Explaining Low Perception Model Competency with High-Competency Counterfactuals

Sara Pohland
University of California, Berkeley
Berkeley, CA, USA 94720
spohland@berkeley.edu

Claire Tomlin
University of California, Berkeley
Berkeley, CA, USA 94720
tomlin@berkeley.edu

Abstract

There exist many methods to explain how an image classification model generates its decision, but very little work has explored methods to explain why a classifier might lack confidence in its prediction. As there are various reasons the classifier might lose confidence, it would be valuable for this model to not only indicate its level of uncertainty but also explain why it is uncertain. Counterfactual images have been used to visualize changes that could be made to an image to generate a different classification decision. In this work, we explore the use of counterfactuals to offer an explanation for low model competency–a generalized form of predictive uncertainty that measures confidence. Toward this end, we develop five novel methods to generate high-competency counterfactual images, namely Image Gradient Descent (IGD), Feature Gradient Descent (FGD), Autoencoder Reconstruction (Reco), Latent Gradient Descent (LGD), and Latent Nearest Neighbors (LNN). We evaluate these methods across two unique datasets containing images with six known causes for low model competency and find Reco, LGD, and LNN to be the most promising methods for counterfactual generation. We further evaluate how these three methods can be utilized by pre-trained Multimodal Large Language Models (MLLMs) to generate language explanations for low model competency. We find that the inclusion of a counterfactual image in the language model query greatly increases the ability of the model to generate an accurate explanation for the cause of low model competency, thus demonstrating the utility of counterfactual images in explaining low perception model competency [1].

1. Introduction

Convolutional neural networks (CNNs) have shown impressive performance across a range of image classification tasks, but their black-box nature limits their applicability to real-world systems. Without a thorough understanding of these models and their failure modes, one cannot confidently employ such models for critical decision-making tasks. Within the field of explainable artificial intelligence (xAI), there is extensive work on explaining CNN classification decisions to better understand how models generate their output predictions. However, there has been very limited work on explaining model competency to better understand why a model lacks confidence in its prediction.

Previous work has explored the use of saliency mapping methods to offer explanations for model confidence by identifying particular image regions for which the trained classification model is unfamiliar [45]. This is a useful approach when anomalous regions cause the reduction in model competency. However, there are many other non-spatial factors that could lead to a reduction in model confidence, such as changes in image properties like brightness, contrast, or saturation, as well as holistic image corruption like noise or pixelation. We need other methods to offer explanations for low model competency in these cases.

We explore the use of counterfactual images–images associated with high levels of model competency that are similar to the original low-competency image. We develop and compare five approaches for generating counterfactual examples across two distinct datasets with various causes of low model competency. We then evaluate the ability of Multimodal Large Language Models (MLLMs) to generate interpretable explanations for low competency with the aid of these counterfactuals. To our knowledge, this is the first work that uses counterfactual images as an explanatory tool for low model confidence.

2. Background & Related Work

In this work, we offer explanations for why an image classification model lacks confidence in its prediction for a given image. There are many methods to quantify model confidence, but we focus on a particular measure of perception

[1] The code for reproducing our methods and results is available on GitHub: https://github.com/sarapohland/competency-counterfactuals.

model competency, as described in Section 2.1. In Section 2.2, we explore many methods employed to explain the predictions of image classifiers and consider their ability to offer explanations for low model competency. Finally, in Section 2.3, we explore the use of language models to expand on these explanations.

2.1. Quantifying Model Confidence

CNNs for image classification usually output softmax scores, which can be interpreted as the probability that an image belongs to each of the training classes. The maximum softmax probability (MSP) can serve as a measure of the confidence of the vision model for a given image, but this probability tends to be very close to one [17] and is particularly unreliable for data outside of the original training distribution [43]. This has motivated many other approaches to quantify model uncertainty, typically through the use of Bayesian Neural Networks (BNNs) [41, 42], Monte Carlo (MC) dropout [11], or ensembles of models [29]. These methods capture many aspects of uncertainty, but tend not to capture distributional uncertainty resulting from mismatched training and test distributions [47]. This has led to methods that specifically seek to detect inputs that are out-of-distribution (OOD), through classification-based [6, 32, 34, 57], density-based [28, 50, 51, 72], distance-based [19, 30, 58], or reconstruction-based [13, 54, 64] approaches. These methods better address distributional uncertainty, but generally rely on thresholds to generate a binary decision, rather than capturing a holistic measure of uncertainty.

We are interested in *perception model competency*–a generalized form of predictive uncertainty that combines various aspects of uncertainty into a single probabilistic score [49]. To estimate model competency, we employ the PaRCE score [44], which computes the product of the MSP and the probability that the image is in-distribution (ID). To estimate the ID probability, PaRCE uses a function of the reconstruction loss of an autoencoder trained to reconstruct the training images. The scores are calibrated via an ID holdout set such that the PaRCE score directly reflects the prediction accuracy of the perception model.

2.2. Explainable Image Classification

Explainable image classification is a rich field that seeks to offer explanations for why a model makes the decisions that it does [3, 53]. While there are many methods to enhance the understanding of image classifiers, they generally do not deal with *model competency*, and thus cannot offer explanations for how confident a model is in its prediction. We previously explored saliency mapping methods to explain a model's lack of competency, by identifying and displaying key image regions that contribute to the observed low model competency [45]. However, while this work offered useful explanations for images with regional features that were unfamiliar to the perception model, there are many other image properties that do not exist at the regional level but may contribute to low competency. To address this limitation, we explore counterfactual methods.

Rather than seeking to explain why a certain prediction was made, counterfactual methods analyze changes that could be made to the input to obtain a different prediction [62]. Many methods for counterfactual explanations of image classifiers involve pixel-level edits, pinpointing regions for minimal change to achieve the desired class. These approaches frequently use generative models, such as autoencoders, generative adversarial networks (GANs), or diffusion models, to synthesize counterfactual images [23, 24, 33, 37, 55, 56]. There are also a number of optimization-based methods that treat counterfactual generation as a constrained optimization problem in the pixel space [9, 14, 46, 61, 63]. Similarly, one could design an optimization problem in some latent space with the goal of finding minimal perturbations in the latent representation of an image to effect a change in the classification decision [5, 18, 26, 35, 59]. Other methods that perform latent space manipulation focus on leveraging the learned semantic structure for interpretable counterfactual generation [8, 27, 31, 52]. A diverging line of work uses conceptual counterfactuals, emphasizing human-interpretable semantics. These approaches guide concept-level edits, identifying minimal semantic features that need modification to change a classification [1, 2, 12, 15, 16].

While all these methods help explain image classification decisions, none offer explanations for model confidence and all would need to be adapted to varying degrees for this purpose. We explore five novel counterfactual methods that seek to explain why a perception model is not confident for a given image, focusing only on methods that do not require training with low-competency examples.

2.3. Language Models for Anomaly Explanation

In our effort to generate counterfactual explanations for low model competency, we explore the use of MLLMs. Although much work has explored the use of visual language models (VLMs) for OOD and anomaly detection [39], often using CLIP [48] to detect samples that do not belong to any ID class [38, 40] or to distinguish between normal and abnormal samples [25, 70], far less work has considered the use of LLMs to provide explanations for anomaly and OOD detection outcomes [65]. Within the area of LLMs for explanation generation, most work has focused on video anomaly detection (VAD)–the task of identifying unusual or unexpected events in video streams [36, 66–68]–or time series anomaly detection (TSAD)–the task of identifying unusual patterns or behaviors in time-ordered data points [71]. We are interested in the use of LLMs to offer explanations

of low model competency for individual images, which has yet to be explored. Unlike VAD and TSAD, which analyze temporal changes to detect anomalies, our focus is on identifying spatial features that contribute to uncertainty in a single image.

3. Generating Counterfactual Images

In this work, we explore methods to generate high-competency counterfactual images for low-competency samples and offer explanations for low model competency using MLLMs. In this section, we focus on generating counterfactuals.

3.1. Counterfactual Generation Methods

We develop and compare five distinct methods for generating a high-competency counterfactual image that is qualitatively similar to the original low-competency image. Let X be the original image with a competency score, $\hat{\rho}(X)$, which is below some competency threshold. We hope to generate a counterfactual image, X', whose competency score, $\hat{\rho}(X')$, is above this competency threshold.

3.1.1. Image Gradient Descent (IGD)

The goal of the first method is to gradually modify the input image to increase the estimated competency, while maintaining visual similarity. More specifically, we seek to minimize the loss function

$$\mathcal{L}(X') = -\hat{\rho}(X') + \gamma d(X, X'), \tag{1}$$

where $d(\cdot)$ is a distance function and γ is a parameter that trades off between increasing the competency of the counterfactual and maintaining similarity between the counterfactual image and the original. We define distance in terms of the Learned Perceptual Image Patch Similarity (LPIPS) metric, which measures how visually similar the original and counterfactual images are [69].

To obtain a counterfactual image that minimizes Equation 1, we initially set X' to be the original image, X. We then gradually update the image via gradient descent, stopping once the competency of the counterfactual is above the specified threshold or the maximum allowable iterations have been reached.

3.1.2. Feature Gradient Descent (FGD)

In the second method, rather than seeking to maintain visual similarity between the original and counterfactual image, our goal is to increase the estimated competency, while ensuring that the feature vector used for classification does not change substantially. Let $f(\cdot)$ be the feature extractor, which is used to obtain the feature vector, $f(X)$, provided as input to the final softmax layer of the classification model. Our goal now is to minimize

$$\mathcal{L}(X') = -\hat{\rho}(X') + \gamma d(f(X), f(X')). \tag{2}$$

Again, $d(\cdot)$ is a distance function and γ is a tunable parameter. By default, we use the negative cosine similarity to represent the distance between two feature vectors, but other distance metrics can be used as well.

As with IGD, to obtain an image that minimizes Equation 2, we initially set X' to be the original image. We then gradually update the image via gradient descent, stopping once the competency of the counterfactual is above the specified threshold or the maximum allowable iterations have been reached.

3.1.3. Autoencoder Reconstruction (Reco)

Recall from Section 2.1 that we consider a competency estimation method that relies on an autoencoder to reconstruct the input image [44]. Because this reconstruction model outputs images similar to those with which it is familiar, we can treat the reconstructed image as a counterfactual. Let $g(\cdot)$ be the encoder of the reconstruction model and $h(\cdot)$ be the decoder. The counterfactual image is then simply given by $X' = h(g(X))$.

3.1.4. Latent Gradient Descent (LGD)

Improving upon the previous approach, rather than simply using the reconstructed image, we manipulate the latent representation in the reconstruction model to increase the competency of the prediction, while ensuring that the latent vector does not change substantially. Let $z = g(X)$ be the latent representation of the original image and $z' = g(X')$ be the latent representation of the counterfactual image. In this approach, we seek to find the latent vector that minimizes the loss function

$$\mathcal{L}(z') = -\hat{\rho}(h(z')) + \gamma d(z, z'). \tag{3}$$

Once again, $d(\cdot)$ is a distance function and γ is a tunable parameter. By default, we use the negative cosine similarity to represent the distance between two latent vectors, but other distance metrics can be used as well.

To obtain a latent vector that minimizes Equation 3, we initially set z' to be the original latent representation, z. We then gradually update the latent vector via gradient descent, stopping once the competency of the counterfactual is above the specified threshold or once the maximum number of allowable iterations has been reached. We use the decoder of the reconstruction model to generate the counterfactual image from the latent representation: $X' = h(z')$.

3.1.5. Latent Nearest Neighbors (LNN)

Recall from Section 2.1 that competency scores are calibrated via an ID holdout set [44]. In our final method, we first find the latent vector, z_{NN}, from the calibration set that is closest to the latent representation of the image of interest. By default, we use the ℓ_1 norm to find the nearest neighbor, but other distance metrics may be used as well. We then use the reconstruction of this latent vector as the counterfactual image: $X' = h(z_{NN})$.

3.2. Comparison of Counterfactual Images

We compare our counterfactual image generation methods both quantitatively and visually across two datasets and a number of performance metrics.

3.2.1. Datasets

We conduct analysis across two unique datasets. The first dataset is obtained from a simulated lunar environment. The classifier trained on this dataset learns to distinguish between different regions in the environment, such as bumpy terrain, smooth terrain, regions inside a crater, etc. The second dataset contains speed limit signs in Germany [21]. The classifier learns to distinguish between seven common speed limit signs, ranging from 30 to 120 km/hr.

While competency tends to be high for both of these datasets, we identify six key causes of low model competency: spatial, brightness, contrast, saturation, noise, and pixelation [44]. For each dataset, we generate 600 low-competency example images, for which the lack of competency can be attributed to one of these six factors (with 100 images per factor). For the lunar dataset, images with spatial anomalies contain astronauts or human-made structures that were not present in the training set. For the speed limit dataset, spatial anomalies are images of an uncommon speed limit, 20 km/hr, which was not present during training. We generate example images for the other causes of low model competency from high-competency test images by increasing or decreasing the given image property (brightness, contrast, or saturation), adding uniform random noise, or compressing the image to create pixelation. Examples of images with these causes of low model competency are shown in column 1 of Figures 2–7 for the lunar dataset and in Figures 8–13 for the speed dataset in Appendix A.

3.2.2. Evaluation Metrics

Recall from Section 2.2 that in the field of explainable image classification, counterfactual methods analyze changes that could be made to the input to obtain a different prediction through the generation of counterfactual images. There are five desirable properties of counterfactual images [27]: (I) *Validity* The classification model should correctly assign the counterfactual to the desired class. (II) *Proximity* The counterfactual should remain close to the original in terms of some distance function. (III) *Sparsity* A minimal number of features should be changed in generating the counterfactual. (IV) *Realism* The counterfactual should lie close to the data manifold such that it appears realistic. (V) *Speed* The counterfactual should be generated quickly.

We consider the same properties to be desirable for counterfactuals used to explain why model competency is low for a given image. Rather than defining validity in terms of the classifier's prediction, we say that a counterfactual is valid if the competency estimator assigns it a high competency score. We generate a number of metrics to evaluate our counterfactual generation methods in terms of these properties. (1) *Success rate* To measure validity, we compute the percentage of counterfactuals with high model competency. (2) *Perceptual loss* We evaluate proximity using the LPIPS metric for visual similarity [69] described in Section 3.1. (3) *Feature similarity* We evaluate sparsity in terms of the average cosine similarity between the original and valid counterfactual feature vectors used by the classification model. (4) *Latent similarity* We also evaluate sparsity in terms of the average cosine similarity between the original and valid counterfactual latent representations within the autoencoder of the competency estimator. (5) *Fréchet Inception Distance (FID)* We measure realism first in terms of the FID, which is a metric used to assess the quality of images created by a generative model by comparing the distribution of generated images with the distribution of a set of real images [20]. The set of real images we use is the set used to calibrate the competency estimator [44]. (6) *Kernel Inception Distance (KID)* We also assess realism in terms of the KID, which measures the maximum mean discrepancy between features extracted from real and fake images [7]. (7) *Computation time* Finally, we measure speed in terms of the average time required to compute a counterfactual for a single image.

3.2.3. Results & Analysis

We compare the five counterfactual methods discussed in Section 3.1 for both the lunar dataset (Table 1) and the speed dataset (Table 2). For reference, we provide metrics for the original images (Orig) as well. We also visually compare the generated counterfactual images in Figure 1. Several additional example images are visualized in Figures 2–13 in Appendix A.

Comparing the five counterfactual generation methods across both datasets, we first observe that LGD most reliably generates high-competency counterfactual images, achieving a 100% success rate on the speed limit dataset and nearly 100% success for the lunar dataset. LNN also achieves nearly 100% on the lunar dataset but its success rate is closer to 90% for the speed dataset. IGD and FGD tend to perform similarly, generating high-competency counterfactuals for over 95% of the low-competency images in the speed limit dataset but around 80% for the lunar dataset. Finally, Reco is close to 90% successful for lunar but only around 75% for speed, indicating that this method is not the most reliable.

Comparing the proximity of counterfactual images to the original images, we notice that IGD performs the best in terms of perceptual loss, followed by FGD. Reco, LGD, and LNN perform similarly with higher perceptual losses.

We also see similar results for sparsity, for which we consider changes in both the feature vectors and latent representations of the original low-competency images. We

Table 1. Comparison of counterfactual generation methods for the lunar dataset.

Method	Success Rate ↑	Perceptual Loss ↓	Feature Similarity ↑	Latent Similarity ↑	FID ↓	KID ↓	Computation Time ↓
Orig	0.00%	0.00	1.00	1.00	10.81	21.67	0.0002 sec
IGD	80.00%	**0.02**	0.98	0.97	12.39	19.38	1.1911 sec
FGD	81.33%	0.13	**0.99**	0.97	10.97	15.82	3.1559 sec
Reco	88.67%	0.59	0.95	**0.98**	2.63	**2.05**	**0.0053 sec**
LGD	98.33%	0.59	0.95	**0.98**	2.61	2.06	1.0479 sec
LNN	**99.50%**	0.60	0.90	0.92	**2.59**	2.26	0.0069 sec

Table 2. Comparison of counterfactual generation methods for the speed limit dataset.

Method	Success Rate ↑	Perceptual Loss ↓	Feature Similarity ↑	Latent Similarity ↑	FID ↓	KID ↓	Computation Time ↓
Orig	0.00%	0.00	1.00	1.00	29.79	82.23	0.0001 sec
IGD	98.33%	**0.01**	0.54	**0.99**	83.64	315.99	2.8882 sec
FGD	95.83%	0.02	**0.81**	**0.99**	80.11	297.85	5.4005 sec
Reco	74.67%	0.49	0.59	0.88	8.65	8.23	**0.0140 sec**
LGD	**100.00%**	0.47	0.56	0.88	**8.48**	**8.12**	4.2700 sec
LNN	91.33%	0.53	0.41	0.58	9.18	8.72	0.0159 sec

find that FGD tends to produce counterfactual images with the most similar feature vectors, while both IGD and FGD produce counterfactuals with very similar latent representations. We also notice that, overall, more elements of the original feature vectors and latent representations are changed with LNN.

However, we see nearly opposite results in terms of realism. We observe that Reco, LGD, and LNN produce counterfactual images that are much more realistic than the original low-competency images, with little difference between these three methods. In contrast, IGD and FGD produce counterfactuals that are similarly unrealistic to the original images or sometimes even more unrealistic.

The visual comparison of these methods sheds some light on these quantitative results. From rows 1, 3, and 4 of Figure 1, we observe that IGD and FGD sometimes produce counterfactual images with unrealistic artifacts. It is also clear that IGD and FGD often produce counterfactuals that are proximal, but this is not necessarily achieved in a positive way. As is observed in rows 2 and 6 of Figure 1, the differences between the original and counterfactual images are not always clearly observable, which is not beneficial for an explanatory tool.

Finally, comparing the speed of the five methods, we observe that Reco is the fastest on average, but LNN is similarly fast. IGD and LGD are significantly slower than these two methods, and FGD tends to be the slowest. It is also interesting to note that computation time varies significantly with the dataset.

Returning to our visual comparison (Figure 1), we see that Reco, LGD, and LNN often produce counterfactuals that correct the cause of low model competency observed in the original images. In Figure 2, we see objects were removed from examples with spatial anomalies, and in Figure 8, we observe the digit 2 associated with an unfamiliar class was replaced with a digit associated with a seen class. Similarly, Figures 3 and 9 demonstrate that brightness of overexposed images is corrected in the counterfactuals, contrast for high-contrast images is reduced in Figures 4 and 10, and saturation is reduced for overly saturated images in Figures 5 and 11. We also notice that noise was removed from noisy images in Figures 6 and 12, and pixelation was corrected in 7 and 13.

In general, the "best" method depends on which properties of counterfactual image generation are valued most highly. IGD and FGD are probably not particularly useful because they often produce unrealistic images that generally do not address the true cause of low model competency. However, they would be useful if proximity and similarity are a major concern. If speed is a high priority, one might opt for Reco or LNN over LGD. However, if it is most important to reliably produce high-competency counterfactuals, then LGD should be chosen instead. The appropriate method largely depends on how the counterfactual will be used. In the next section, we consider how these counterfactuals might be used by an MLLM to generate language explanations for low model competency.

4. Explaining Counterfactual Images

In this section, we consider how to obtain language explanations for low model competency using the counterfactual

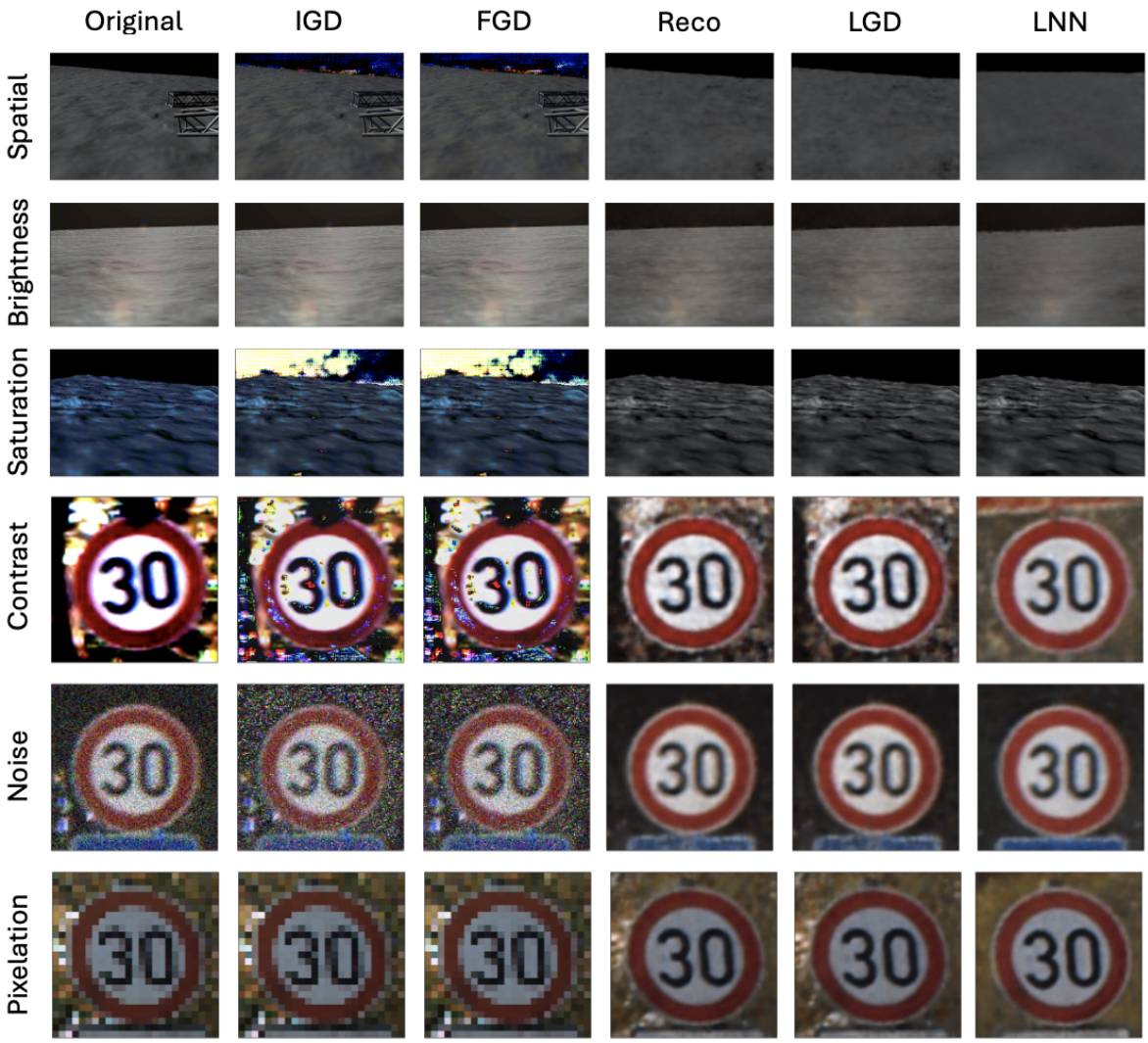


Figure 1. Example counterfactuals generated through different methods (columns) for various causes of low model competency (rows).

images generated in Section 3.

4.1. Counterfactual Explanation Method

We focus on explaining potential causes for low model competency with the help of MLLMs. We consider the explanation provided when the model sees only the original image, as well as the explanation provided upon seeing both the original and counterfactual image. Based on our results from Section 3.2, we use the autoencoder reconstruction (Reco), latent gradient descent (LGD), and latent nearest neighbors (LNN) methods to generate counterfactuals.

4.1.1. LLaMA Model

While a number of MLLMs were considered for the purpose of counterfactual explanation, all explanations are generated using the LLaMA 3.2 model (in the 11B size) [60] because it is a publicly available model that has demonstrated strong performance in Visual Question Answering (VQA) tasks [10]. The LLaMA 3.2 model is a pre-trained and instruction-tuned image reasoning generative model that is optimized for visual recognition, image reasoning, captioning, and answering general questions about an image. This model allows one to set the context in which to interact with the AI model, which typically includes rules, guidelines, or necessary information that help the model respond effectively. It also allows for user prompts, which include the inputs, commands, and questions to the model that could contain an image with text or text only.

4.1.2. Model Prompts

To obtain an explanation from the language model of low model competency for a given image, we first describe the training set, using Prompt B.1 for the lunar dataset and Prompt B.2 for the speed limit dataset. We also give a

description of the competency estimator using Prompt B.3. We then provide instructions about the desired output, using Prompt B.4 if we are not using a counterfactual image and Prompt B.5 otherwise.

4.2. Comparison of Counterfactual Explanations

For each language model explanation, we manually evaluate whether the response correctly describes the true cause of low model competency. We compare the correctness of the explanations that do not use counterfactual images to those aided by the counterfactuals generated by the Reco, LGD, and LNN methods. The accuracies of the explanations across each of the six causes of low model competency, along with the average accuracy, are provided for the lunar dataset in Table 3 and the speed limit dataset in Table 4. Note that we primarily assess the performance of the pre-trained LLaMA model in generating appropriate explanations, but we report the performance for a fine-tuned model as well.

4.2.1. Pre-Trained Model

From our results using the pre-trained LLaMA model (the prominent results displayed in Tables 3 and Table 4), we observe that the explanations generated without the help of counterfactual images were only correct around one-fifth of the time. In contrast, the explanations aided by counterfactual images produced by Reco, LGD, and LNN were correct closer to one-third of the time, indicating that counterfactual images can greatly improve the accuracy of language explanations for low model competency. Examples of this improvement are provided in Figures 14–19 of Appendix D. We did not observe significant differences between the Reco, LGD, and LNN methods.

It should be noted that accuracy varies substantially across the true causes of low model competency. The language model is fairly accurate at identifying noise and pixelation as causes of low competency when a counterfactual image is provided. This may be because noise and pixelation are easily observable features, and image corruption is known to reduce classification performance. The language model can also often identify anomalous objects as a cause for low model competency with the aid of a counterfactual, but the accuracy is much lower than for noise and pixelation. Although correct explanations are often generated for spatial anomalies in the lunar dataset, the language model very rarely notices digits associated with an unknown class in the speed limit dataset. The lower performance may be seen because these spatial anomalies require some high-level understanding of the training set. Finally, the language model is far more accurate in identifying brightness, contrast, and saturation as causes of low model competency when a counterfactual is provided, but accuracy still tends to be low. This poor performance may be observed because brightness, contrast, and saturation are not widely discussed causes of low model competency with which the pre-trained language model would be familiar.

While counterfactual images can greatly increase the ability to generate language explanations that correctly identify the causes of low model competency, accuracy is still not as high as we would hope, especially for particular causes of low model competency. We notice that the language model often hallucinates in its explanations–an issue commonly observed with MLLMs [4]. (See Figure 20 for an example of this.) We also find that the rationale for low model competency is sometimes inverted, especially when using a counterfactual, as in Figure 21.

4.2.2. Fine-Tuned Model

Although it may not always be practical to fine-tune the language model depending on computational constraints and availability of training data, we note that the accuracy of language explanations increases significantly after fine-tuning the model with some image-explanation pairs. (A description of the fine-tuning process is provided in Appendix C.) For both the lunar dataset (Table 3) and the speed limit dataset (Table 4), we notice that the average accuracy of the fine-tuned language explanations is close to 100% across all methods. When fine-tuning is an option, the utility of counterfactual images decreases because the model can learn reasonable explanations using only the original images. A counterfactual image may even become unhelpful for a model that has been fine-tuned well because it introduces additional variance into the data and may serve as a distraction to the fine-tuned language model.

5. Conclusions

In this work, we explore the use of counterfactual images to explain why an image classification model lacks confidence in its prediction. We develop five counterfactual generation methods: image gradient descent (IGD), feature gradient descent (FGD), autoencoder reconstruction (Reco), latent gradient descent (LGD), and latent nearest neighbors (LNN). We evaluate the images generated by these methods in terms of their validity, proximity, sparsity, realism, and speed across two unique datasets with six identified causes of low model competency: spatial, brightness, contrast, saturation, noise, and pixelation. While IGD and FGD generate sparse and proximal solutions, they are slow, unreliable, and tend to generate unrealistic images. Reco, LGD, and LNN tend to generate high-competency counterfactual images that appear more realistic than their original low-competency counterparts and correct for the cause of low competency observed in the original images. The best method among these three depends on the application and the properties of counterfactual images valued most highly.

To further evaluate the utility of counterfactual images as an explanatory tool for low model competency, we develop

Table 3. Accuracy of competency explanations for the lunar dataset across various true causes of low model competency. Results for the pre-trained model are displayed more prominently, while results for the fine-tuned model are provided in parentheses.

Method	Spatial	Brightness	Contrast	Saturation	Noise	Pixelation	Average
None	8% (99%)	1% (90%)	6% (100%)	1% (100%)	6% (100%)	**91%** (100%)	18.83% (98.17%)
Reco	**28%** (95%)	**10%** (83%)	**13%** (96%)	7% (100%)	73% (100%)	77% (100%)	34.67% (95.67%)
LGD	25% (99%)	**10%** (84%)	8% (97%)	**14%** (100%)	73% (100%)	85% (100%)	35.83% (96.67%)
LNN	21% (99%)	7% (81%)	12% (100%)	**14%** (100%)	**82%** (100%)	87% (100%)	**37.17%** (95.83%)

Table 4. Accuracy of competency explanations for the speed limit dataset across various true causes of low competency. Results for the pre-trained model are displayed more prominently, while results for the fine-tuned model are provided in parentheses.

Method	Spatial	Brightness	Contrast	Saturation	Noise	Pixelation	Average
None	**2%** (99%)	4% (100%)	0% (100%)	0% (100%)	10% (100%)	**98%** (100%)	19.00% (99.83%)
Reco	1% (100%)	12% (99%)	**3%** (98%)	**14%** (100%)	**74%** (100%)	81% (100%)	**30.83%** (99.50%)
LGD	0% (100%)	**21%** (100%)	1% (100%)	12% (100%)	64% (100%)	81% (100%)	29.83% (100.00%)
LNN	0% (100%)	12% (100%)	1% (99%)	11% (100%)	70% (100%)	81% (100%)	29.17% (99.83%)

a pipeline to generate language explanations using a pre-trained MLLM with the aid of high-competency counterfactual images. We find that, while explanations generated without the help of counterfactual images were only correct around one-fifth of the time, the explanations aided by counterfactual images produced by Reco, LGD, and LNN were correct closer to one-third of the time. This indicates that counterfactual images can greatly improve the accuracy of language explanations for low model competency. We also find that the accuracy of explanations increases to nearly 100% after fine-tuning the MLLM with a few thousand image-explanation pairs.

Although counterfactual images appear useful for explaining the reason why an image classifier lacks confidence in its prediction, much work could be done to improve the utility of these counterfactuals. Most immediately, one could more carefully select optimization parameters for the gradient descent-based methods and improve the stopping criterion. In addition, one could consider other distance metrics in the loss functions. It would also be interesting to combine these methods–in a single objective or by utilizing multiple counterfactual images. One might also design a metalearner to dynamically select the most appropriate counterfactual for an image, rather than relying on a fixed generation method.

There is also much work to be done in generating language explanations from the provided counterfactual images. First, one could evaluate other pre-trained MLLMs, beyond LLaMA. One might also explore the design of VLMs specifically for the purpose of low model competency explanation and analyze the generalizability of such methods to new datasets. It would also be beneficial to explore methods to reduce language model hallucinations–potentially through prompt engineering techniques or post-processing filters.

To more fully understand the utility of counterfactual images and language explanations, as well as how to improve them, it would be valuable to perform user studies. Future work should analyze how useful counterfactual images are to human users, allowing the user to play the role of the MLLM and evaluating how often they determine the correct cause of low competency with and without the aid of a counterfactual. It would also be interesting to receive feedback from users about the perceived utility of these counterfactuals. Similarly, users could evaluate how accurate and useful the language explanations are to them. While expanding on the analysis of counterfactual methods and explanations, it would also be useful to conduct evaluations with more diverse and complex datasets.

Finally, there remains the question of what should be done with these explanations. It would be interesting to explore the use of counterfactual images and their language explanations as a corrective tool to improve model predictions. For example, if a model is not confident because image brightness is high, perhaps the system adjusts brightness before making a prediction. We may also use these explanations to train better models. One might use knowledge of low model competency causes for data augmentation.

References

[1] Abubakar Abid, Mert Yuksekgonul, and James Zou. Meaningfully debugging model mistakes using conceptual counterfactual explanations. In *International Conference on Machine Learning (ICML)*, 2022. 2

[2] Arjun Akula, Shuai Wang, and Song-Chun Zhu. Cocox: Generating conceptual and counterfactual explanations via fault-lines. In *Conference on Artificial Intelligence*, 2020. 2

[3] Sajid Ali, Tamer Abuhmed, Shaker El-Sappagh, and et al. Explainable Artificial Intelligence (XAI): What we know and what is left to attain Trustworthy Artificial Intelligence. *Information Fusion*, 99(C), 2023. 2

[4] Zechen Bai, Pichao Wang, Tianjun Xiao, and et al. Hallucination of multimodal large language models: A survey, 2024. 7

[5] Rachana Balasubramanian, Samuel Sharpe, Brian Barr, and et al. Latent-cf: A simple baseline for reverse counterfactual explanations, 2021. 2

[6] Abhijit Bendale and Terrance E. Boult. Towards open set deep networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2

[7] Mikołaj Bińkowski, Dougal J. Sutherland, Michael Arbel, and et al. Demystifying MMD GANs. In *International Conference on Learning Representations (ICLR)*, 2018. 4

[8] Saloni Dash, Vineeth N Balasubramanian, and Amit Sharma. Evaluating and mitigating bias in image classifiers: A causal perspective using counterfactuals. In *Winter Conference on Applications of Computer Vision (WACV)*, 2022. 2

[9] Amit Dhurandhar, Pin-Yu Chen, Ronny Luss, and et al. Explanations based on the missing: Towards contrastive explanations with pertinent negatives. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018. 2

[10] Xixi Ga, Wenjie Liu, Tongyu Zhu, and et al. Evaluating robustness and diversity in visual question answering using multimodal large language models, 2024. 6

[11] Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In *Proceedings of Machine Learning Research (PMLR)*, 2016. 2

[12] Asma Ghandeharioun, Been Kim, Chun-Liang Li, and et al. Dissect: Disentangled simultaneous explanations via concept traversals, 2022. 2

[13] Dong Gong, Lingqiao Liu, Vuong Le, and et al. Memorizing Normality to Detect Anomaly: Memory-augmented Deep Autoencoder for Unsupervised Anomaly Detection. In *International Conference on Computer Vision (ICCV)*, 2019. 2

[14] Yash Goyal, Ziyan Wu, Jan Ernst, and et al. Counterfactual visual explanations. In *International Conference on Machine Learning (ICML)*, 2019. 2

[15] Yash Goyal, Amir Feder, Uri Shalit, and et al. Explaining classifiers with causal concept effect (cace), 2020. 2

[16] Sadaf Gulshad and Arnold Smeulders. Counterfactual attribute-based visual explanations for classification. *International Journal of Multimedia Information Retrieval*, 2021. 2

[17] Chuan Guo, Geoff Pleiss, Yu Sun, and et al. On calibration of modern neural networks. In *Proceedings of Machine Learning Research (PMLR)*, 2017. 2

[18] Victor Guyomard, Françoise Fessant, Tassadit Bouadi, and et al. Post-hoc counterfactual generation with supervised autoencoder. In *European Conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD)*, 2022. 2

[19] Dan Hendrycks, Steven Basart, Mantas Mazeika, and et al. Scaling out-of-distribution detection for real-world settings. In *Proceedings of Machine Learning Research (PMLR)*, 2022. 2

[20] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, and et al. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *International Conference on Neural Information Processing Systems (NIPS)*, 2017. 4

[21] Sebastian Houben, Johannes Stallkamp, Jan Salmen, and et al. Detection of traffic signs in real-world images: The German Traffic Sign Detection Benchmark. In *International Joint Conference on Neural Networks (IJCNN)*, 2013. 4

[22] Edward J. Hu, Yelong Shen, Phillip Wallis, and et al. Lora: Low-rank adaptation of large language models, 2021. 18

[23] Paul Jacob, Éloi Zablocki, Hédi Ben-Younes, and et al. Steex: Steering counterfactual explanations with semantics. In *European Conference on Computer Vision (ECCV)*, 2022. 2

[24] Guillaume Jeanneret, Loïc Simon, and Frédéric Jurie. Diffusion models for counterfactual explanations. In *Asian Conference on Computer Vision (ACCV)*, 2023. 2

[25] Jongheon Jeong, Yang Zou, Taewan Kim, and et al. Winclip: Zero-/few-shot anomaly classification and segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2

[26] Shalmali Joshi, Oluwasanmi Koyejo, Warut Vijitbenjaronk, and et al. Towards realistic individual recourse and actionable explanations in black-box decision making systems, 2019. 2

[27] Saeed Khorram and Li Fuxin. Cycle-consistent counterfactuals by latent transformations. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2, 4

[28] Diederik P. Kingma and Prafulla Dhariwal. Glow: Generative Flow with Invertible 1x1 Convolutions. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018. 2

[29] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems (NIPS)*, 2017. 2

[30] Kimin Lee, Kibok Lee, Honglak Lee, and et al. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018. 2

[31] Yan Li, Shasha Liu, Chunwei Wu, and et al. Dcfg: Discovering directional counterfactual generation for chest x-rays. In *International Conference on Bioinformatics and Biomedicine (BIBM)*, 2021. 2

[32] Shiyu Liang, Yixuan Li, and R. Srikant. Enhancing The Reliability of Out-of-distribution Image Detection in Neural Net-

works. In *International Conference on Learning Representations (ICLR)*, 2018. 2

[33] Shusen Liu, Bhavya Kailkhura, Donald Loveland, and et al. Generative counterfactual introspection for explainable deep learning. In *Global Conference on Signal and Information Processing (GlobalSIP)*, 2019. 2

[34] Weitang Liu, Xiaoyun Wang, John D. Owens, and et al. Energy-based out-of-distribution detection. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 2

[35] Arnaud Van Looveren and Janis Klaise. Interpretable counterfactual explanations guided by prototypes. In *European Conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD)*, 2021. 2

[36] Hui Lv and Qianru Sun. Video anomaly detection and explanation via large language models, 2024. 2

[37] Silvan Mertes, Tobias Huber, Katharina Weitz, and et al. Ganterfactual–counterfactual explanations for medical non-experts using generative adversarial learning. In *Frontiers in Artificial Intelligence*, 2022. 2

[38] Yifei Ming, Ziyang Cai, Jiuxiang Gu, and et al. Delving into out-of-distribution detection with vision-language representations. In *International Conference on Neural Information Processing Systems (NIPS)*, 2024. 2

[39] Atsuyuki Miyai, Jingkang Yang, Jingyang Zhang, and et al. Generalized out-of-distribution detection and beyond in vision language model era: A survey, 2024. 2

[40] Atsuyuki Miyai, Qing Yu, Go Irie, and et al. Locoop: few-shot out-of-distribution detection via prompt learning. In *International Conference on Neural Information Processing Systems (NIPS)*, 2024. 2

[41] Radford M. Neal. Bayesian learning via stochastic dynamics. In *Advances in Neural Information Processing Systems (NIPS)*, 1992. 2

[42] Radford M. Neal. *Bayesian Learning for Neural Networks*. Springer, 1996. 2

[43] Yaniv Ovadia, Emily Fertig, Jie Ren, and et al. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 2

[44] Sara Pohland and Claire Tomlin. Parce: Probabilistic and reconstruction-based competency estimation for cnn-based image classification, 2024. 2, 3, 4

[45] Sara Pohland and Claire Tomlin. Understanding the dependence of perception model competency on regions in an image. In *Explainable Artificial Intelligence (xAI)*, 2024. 1, 2

[46] Rafael Poyiadzi, Kacper Sokol, Raul Santos-Rodriguez, and et al. Face: Feasible and actionable counterfactual explanations. In *AAAI/ACM Conference on AI, Ethics, and Society (AIES)*, 2020. 2

[47] Joaquin Quiñonero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil Lawrence, editors. *Dataset shift in machine learning*. The MIT Press, Cambridge, MA, 2008. 2

[48] Alec Radford, Jong Wook Kim, Chris Hallacy, and et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, 2021. 2

[49] Vickram Rajendran and William LeVine. Accurate layerwise interpretable competence estimation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 2

[50] Jie Ren, Peter J. Liu, Emily Fertig, and et al. Likelihood ratios for out-of-distribution detection. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 2

[51] Danilo Jimenez Rezende and Shakir Mohamed. Variational Inference with Normalizing Flows. In *Proceedings of Machine Learning Research (PMLR)*, 2015. 2

[52] Pau Rodríguez, Massimo Caccia, Alexandre Lacoste, and et al. Beyond trivial counterfactual explanations with diverse valuable explanations. In *International Conference on Computer Vision (ICCV)*, 2021. 2

[53] Tilman Räuker, Anson Ho, Stephen Casper, and et al. Toward Transparent AI: A Survey on Interpreting the Inner Structures of Deep Neural Networks, 2023. 2

[54] Mohammad Sabokrou, Mohammad Khalooei, Mahmood Fathy, and et al. Adversarially learned one-class classifier for novelty detection. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[55] Pouya Samangouei, Ardavan Saeedi, Liam Nakagawa, and et al. Explaingan: Model explanation via decision boundary crossing transformations. In *European Conference on Computer Vision (ECCV)*, 2018. 2

[56] Sumedha Singla, Brian Pollack, Junxiang Chen, and et al. Explanation by progressive exaggeration. In *International Conference on Learning Representations (ICLR)*, 2020. 2

[57] Yiyou Sun and Yixuan Li. Dice: Leveraging sparsification for out-of-distribution detection. In *European Conference on Computer Vision (ECCV)*, 2022. 2

[58] Yiyou Sun, Yifei Ming, Xiaojin Zhu, and et al. Out-of-Distribution Detection with Deep Nearest Neighbors. In *Proceedings of Machine Learning Research (PMLR)*, 2022. 2

[59] Jayaraman J. Thiagarajan, Vivek Narayanaswamy, Deepta Rajan, and et al. Designing counterfactual generators using deep model inversion. In *International Conference on Neural Information Processing Systems (NIPS)*, 2024. 2

[60] Hugo Touvron, Thibaut Lavril, Gautier Izacard, and et al. Llama: Open and efficient foundation language models, 2023. 6

[61] Simon Vandenhende, Dhruv Mahajan, Filip Radenovic, and et al. Making heads or tails: Towards semantically consistent visual counterfactuals. In *European Conference on Computer Vision (ECCV)*, 2022. 2

[62] Sahil Verma, Varich Boonsanong, Minh Hoang, and et al. Counterfactual explanations and algorithmic recourses for machine learning: A review, 2022. 2

[63] Tom Vermeire, Dieter Brughmans, Sofie Goethals, and et al. Explainable image classification with evidence counterfactual. *Pattern Analysis and Applications (PAA)*, 25, 2022. 2

[64] Yan Xia, Xudong Cao, Fang Wen, and et al. Learning discriminative reconstructions for unsupervised outlier removal. In *International Conference on Computer Vision (ICCV)*, 2015. 2

[65] Ruiyao Xu and Kaize Ding. Large language models for anomaly and out-of-distribution detection: A survey, 2024. 2

[66] Yuchen Yang, Kwonjoon Lee, Behzad Dariush, and et al. Follow the rules: Reasoning for video anomaly detection with large language models. In *European Conference on Computer Vision (ECCV)*, 2025. 2

[67] Luca Zanella, Willi Menapace, Massimiliano Mancini, and et al. Harnessing large language models for training-free video anomaly detection. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[68] Huaxin Zhang, Xiaohao Xu, Xiang Wang, and et al. Holmes-vad: Towards unbiased and explainable video anomaly detection via multi-modal llm, 2024. 2

[69] Richard Zhang, Phillip Isola, Alexei A. Efros, and et al. The unreasonable effectiveness of deep features as a perceptual metric. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3, 4

[70] Qihang Zhou, Guansong Pang, Yu Tian, and et al. Anomalyclip: Object-agnostic prompt learning for zero-shot anomaly detection, 2024. 2

[71] Jiaxin Zhuang, Leon Yan, Zhenwei Zhang, and et al. See it, think it, sorted: Large multimodal models are few-shot time series anomaly analyzers, 2024. 2

[72] Bo Zong, Qi Song, Martin Renqiang Min, and et al. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *International Conference on Learning Representations (ICLR)*, 2018. 2

A. Comparison of Counterfactual Images

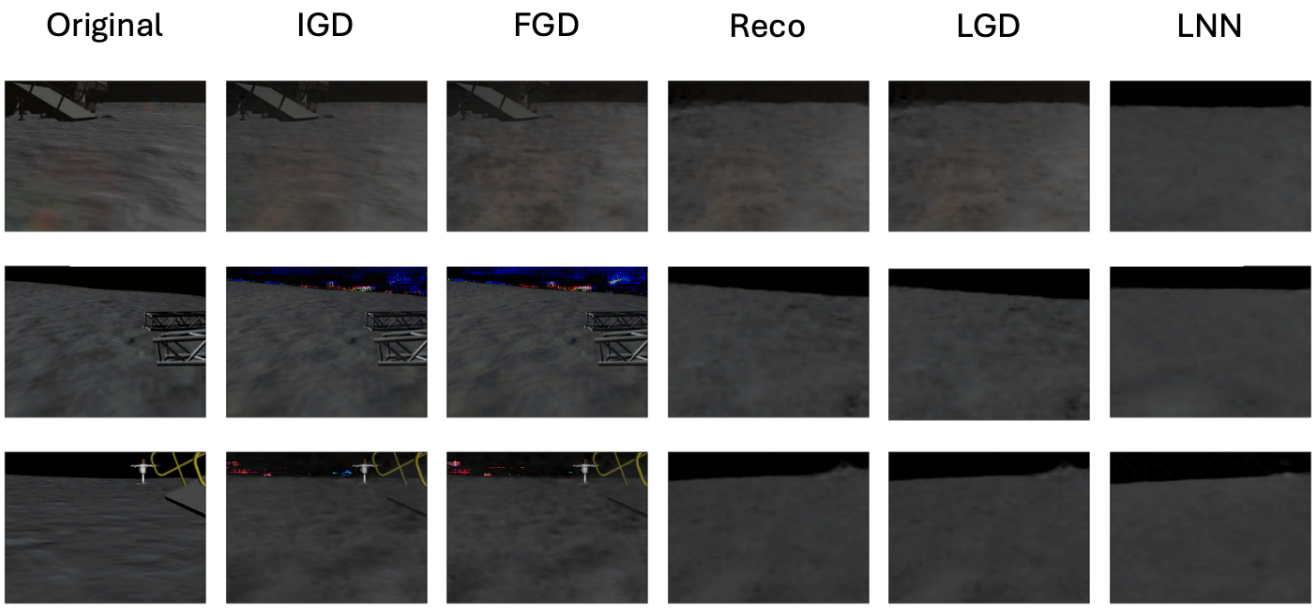


Figure 2. Counterfactual images generated using Image Gradient Descent (IGD, Column 2), Feature Gradient Descent (FGD, Column 3), Autoencoder Reconstruction (Reco, Column 4), Latent Gradient Descent (LGD, Column 5), or Latent Nearest Neighbors (LNN, Column 6) for three low-competency examples in the *lunar* dataset with *spatial* anomalies (Original, Column 1).

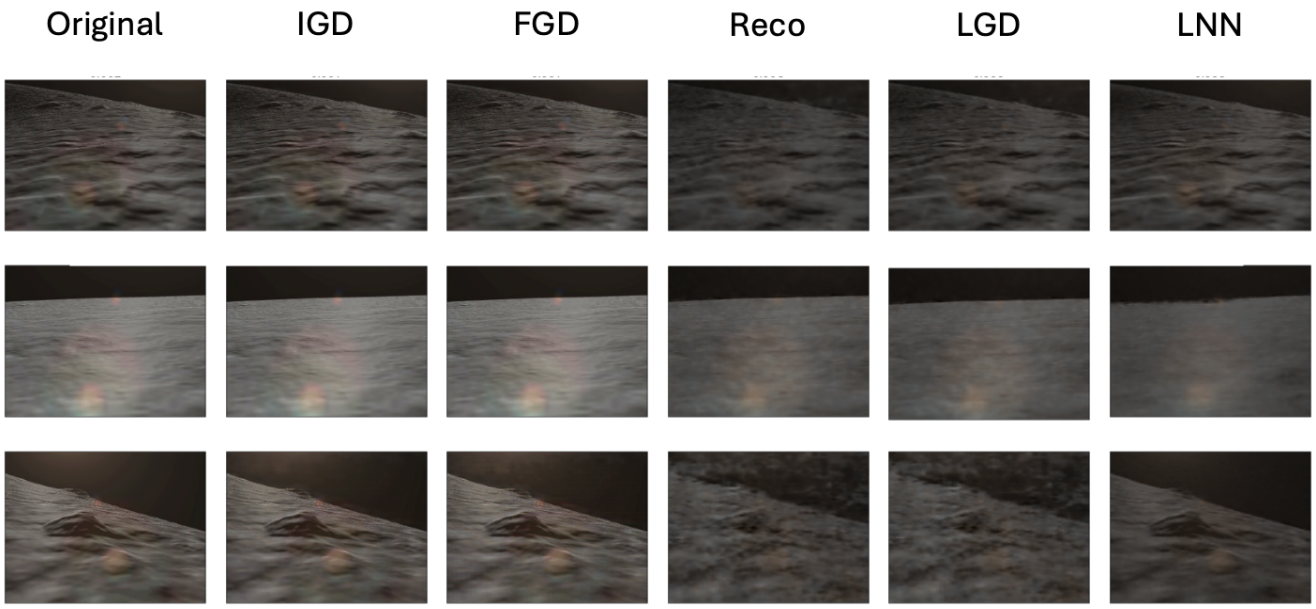


Figure 3. Counterfactual images generated using Image Gradient Descent (IGD, Column 2), Feature Gradient Descent (FGD, Column 3), Autoencoder Reconstruction (Reco, Column 4), Latent Gradient Descent (LGD, Column 5), or Latent Nearest Neighbors (LNN, Column 6) for three low-competency examples in the *lunar* dataset with modified *brightness* (Original, Column 1).

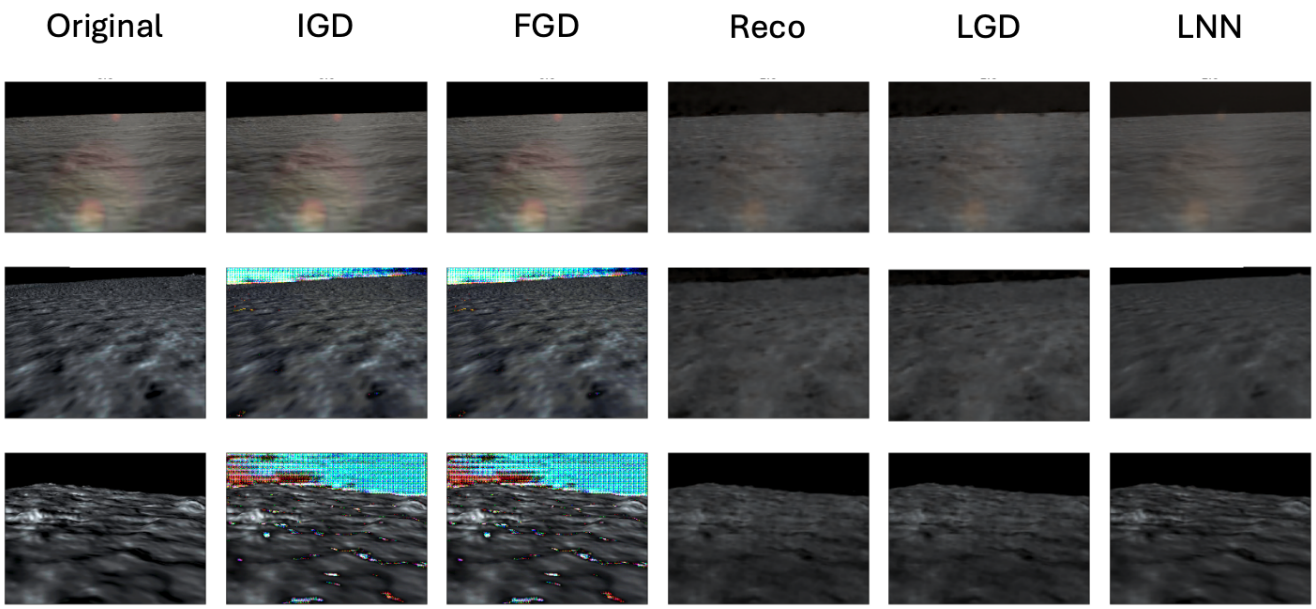


Figure 4. Counterfactual images generated using Image Gradient Descent (IGD, Column 2), Feature Gradient Descent (FGD, Column 3), Autoencoder Reconstruction (Reco, Column 4), Latent Gradient Descent (LGD, Column 5), or Latent Nearest Neighbors (LNN, Column 6) for three low-competency examples in the *lunar* dataset with modified *contrast* (Original, Column 1).

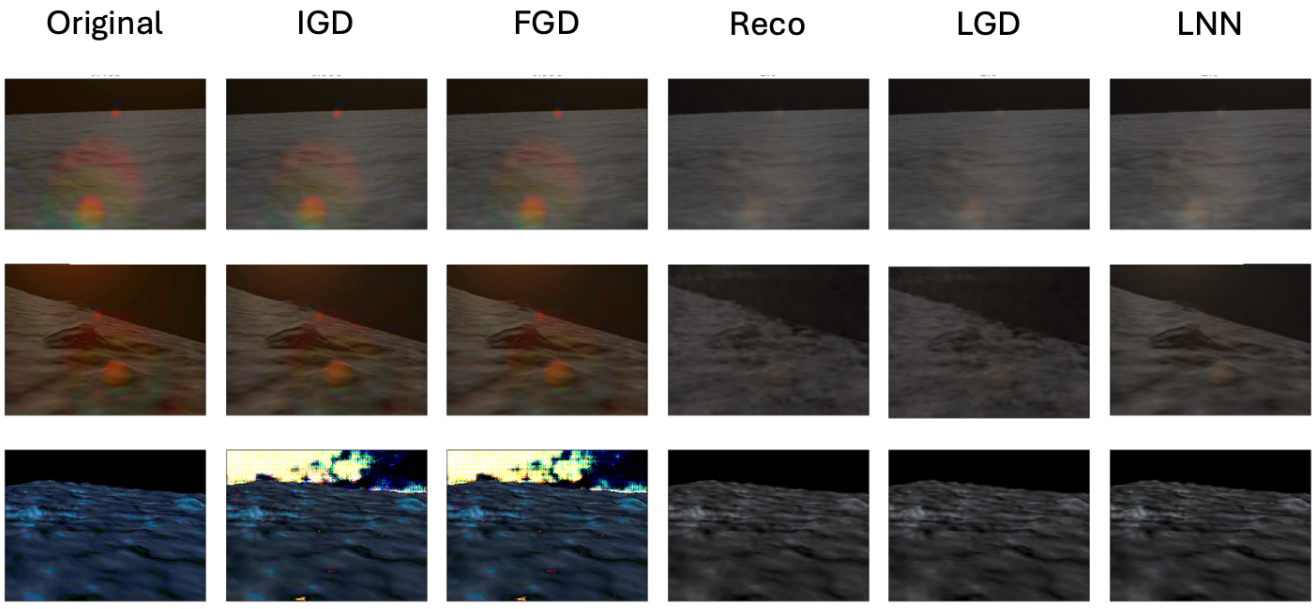


Figure 5. Counterfactual images generated using Image Gradient Descent (IGD, Column 2), Feature Gradient Descent (FGD, Column 3), Autoencoder Reconstruction (Reco, Column 4), Latent Gradient Descent (LGD, Column 5), or Latent Nearest Neighbors (LNN, Column 6) for three low-competency examples in the *lunar* dataset with modified *saturation* (Original, Column 1).

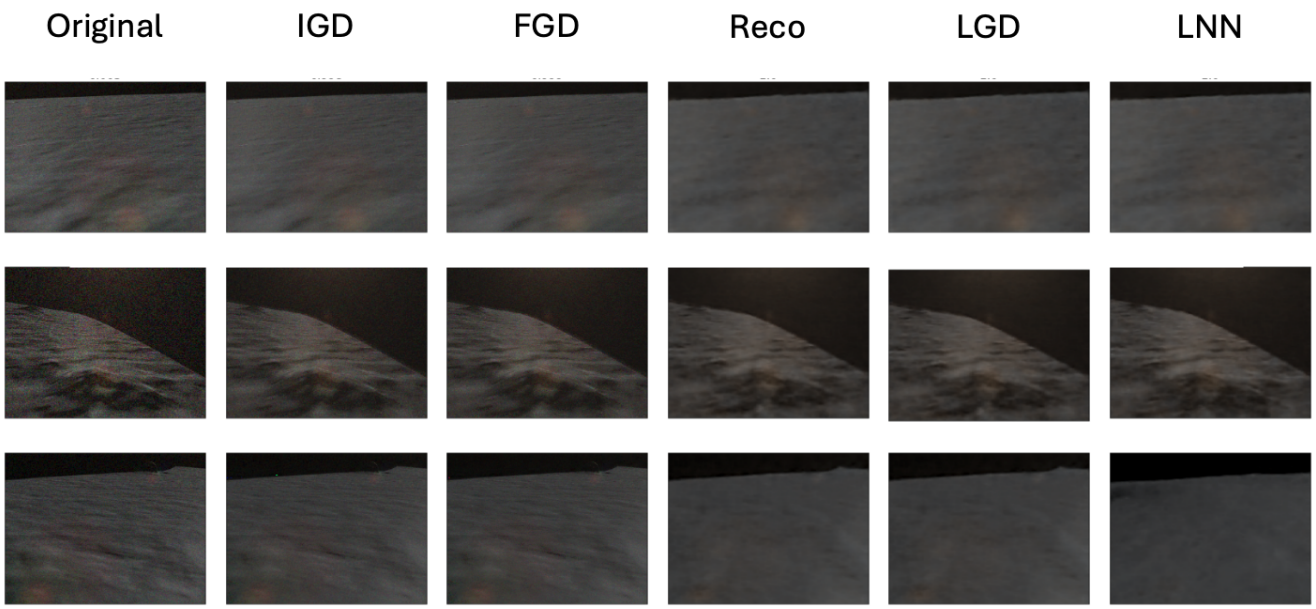


Figure 6. Counterfactual images generated using Image Gradient Descent (IGD, Column 2), Feature Gradient Descent (FGD, Column 3), Autoencoder Reconstruction (Reco, Column 4), Latent Gradient Descent (LGD, Column 5), or Latent Nearest Neighbors (LNN, Column 6) for three low-competency examples in the *lunar* dataset with additive *noise* (Original, Column 1).

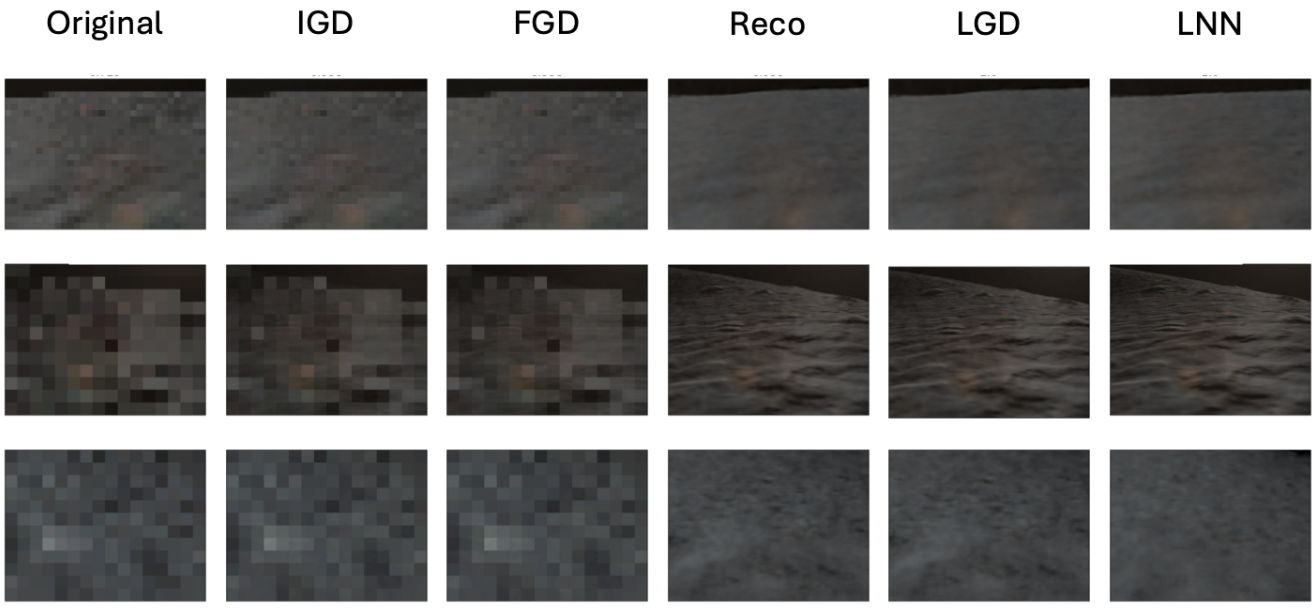


Figure 7. Counterfactual images generated using Image Gradient Descent (IGD, Column 2), Feature Gradient Descent (FGD, Column 3), Autoencoder Reconstruction (Reco, Column 4), Latent Gradient Descent (LGD, Column 5), or Latent Nearest Neighbors (LNN, Column 6) for three low-competency examples in the *lunar* dataset with *pixelation* (Original, Column 1).

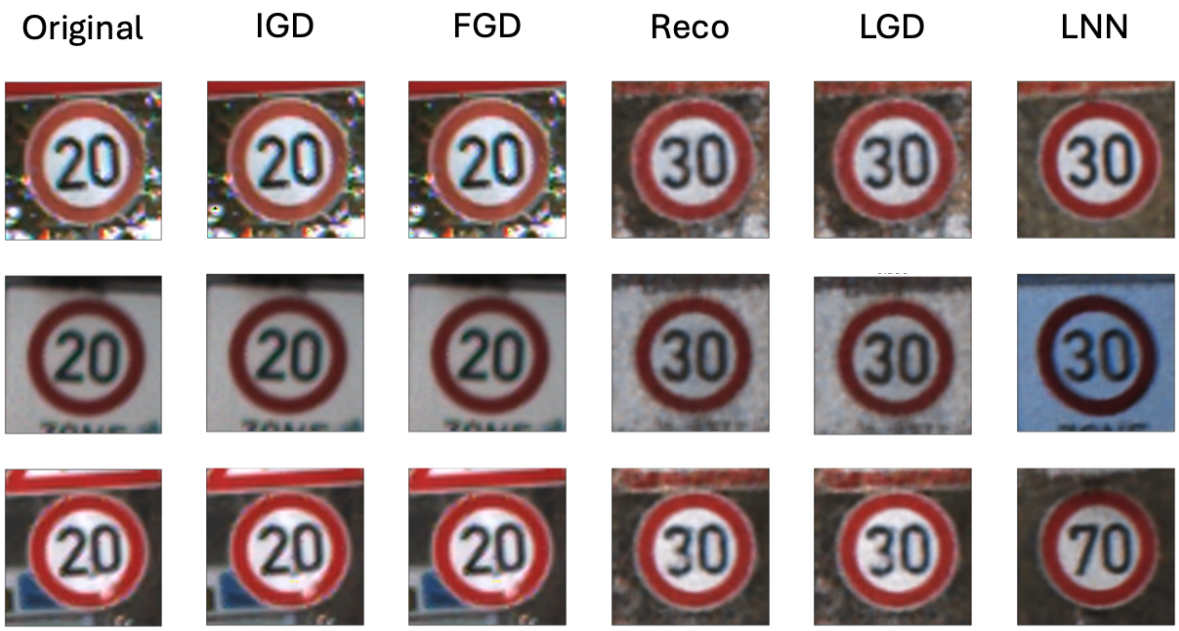


Figure 8. Counterfactual images generated using Image Gradient Descent (IGD, Column 2), Feature Gradient Descent (FGD, Column 3), Autoencoder Reconstruction (Reco, Column 4), Latent Gradient Descent (LGD, Column 5), or Latent Nearest Neighbors (LNN, Column 6) for three low-competency examples in the *speed limit* dataset with *spatial* anomalies (Original, Column 1).

Figure 9. Counterfactual images generated using Image Gradient Descent (IGD, Column 2), Feature Gradient Descent (FGD, Column 3), Autoencoder Reconstruction (Reco, Column 4), Latent Gradient Descent (LGD, Column 5), or Latent Nearest Neighbors (LNN, Column 6) for three low-competency examples in the *speed limit* dataset with modified *brightness* (Original, Column 1).

Figure 10. Counterfactual images generated using Image Gradient Descent (IGD, Column 2), Feature Gradient Descent (FGD, Column 3), Autoencoder Reconstruction (Reco, Column 4), Latent Gradient Descent (LGD, Column 5), or Latent Nearest Neighbors (LNN, Column 6) for three low-competency examples in the *speed limit* dataset with modified *contrast* (Original, Column 1).

Figure 11. Counterfactual images generated using Image Gradient Descent (IGD, Column 2), Feature Gradient Descent (FGD, Column 3), Autoencoder Reconstruction (Reco, Column 4), Latent Gradient Descent (LGD, Column 5), or Latent Nearest Neighbors (LNN, Column 6) for three low-competency examples in the *speed limit* dataset with modified *saturation* (Original, Column 1).

Figure 12. Counterfactual images generated using Image Gradient Descent (IGD, Column 2), Feature Gradient Descent (FGD, Column 3), Autoencoder Reconstruction (Reco, Column 4), Latent Gradient Descent (LGD, Column 5), or Latent Nearest Neighbors (LNN, Column 6) for three low-competency examples in the *speed limit* dataset with additive *noise* (Original, Column 1).

Figure 13. Counterfactual images generated using Image Gradient Descent (IGD, Column 2), Feature Gradient Descent (FGD, Column 3), Autoencoder Reconstruction (Reco, Column 4), Latent Gradient Descent (LGD, Column 5), or Latent Nearest Neighbors (LNN, Column 6) for three low-competency examples in the *speed limit* dataset with *pixelation* (Original, Column 1).

B. Language Model Prompts

Prompt B.1: Description of Lunar Training Set

I trained a CNN for image classification from a set of images obtained from a simulated lunar environment. The classifier learns to distinguish between different regions in this environment, such as regions with smooth terrain, regions with bumpy terrain, regions at the edge of a crater, regions inside a crater, and regions near a hill.

Prompt B.2: Description of Speed Limit Training Set

I trained a CNN for image classification from a dataset containing speed limit signs. The classifier learns to distinguish between seven (7) different speed limits: 30, 50, 60, 70, 80, 100, and 120 km/hr.

Prompt B.3: Description of Competency Estimator

In addition to the classification model, I trained a reconstruction-based competency estimator that estimates the probability that the classifier's prediction is accurate for a given image.

Prompt B.4: Instructions without Counterfactual Image

Here is an image for which the classifier is not confident. In a single sentence, explain what properties of the image itself might lead to the observed reduction in model confidence.

Prompt B.5: Instructions using Counterfactual Image

Here are two images side-by-side. The first (on the left) is the original image, for which my classifier is not confident. The second image (on the right) is a similar image, for which my model is more confident. In a single sentence, explain what properties of the original image might have led to the observed reduction in model confidence.

C. Fine-Tuning Language Model

To fine-tune the LLaMA model to generate model competency explanations for a given dataset, we first collect 3000 additional low-competency images that have not been used for training or evaluation–500 from each of the six low competency categories. For each dataset, we gather 500 previously unseen images with spatial anomalies and generate 500 images with each of the image modifications from previously unused high-competency images. We automatically assign each of these new low-competency images a sample explanation based on their known cause of low model competency. For example, for an image with increased saturation, we would assign the explanation: "The original image is over-saturated."

We perform fine-tuning on a single NVIDIA GeForce RTX 4090, which has 24GB of GDDR6X RAM. To significantly reduce the size of the pre-trained model, we load the model in 4-bit quantization. To further reduce the computational effort required for fine-tuning, we use LoRA (Low-Rank Adaptation)–a technique for efficiently fine-tuning large pre-trained models by introducing learnable low-rank matrices into specific parts of the model [22]. Rather than updating the weights of the entire model, we only modify the vision layers, language layers, attention modules, and MLP modules during fine-tuning. We use a supervised fine-tuning (SFT) trainer with the Adam optimizer. We fine-tune for 10 epochs, using a training batch size of 4 and 4 gradient accumulation steps.

D. Language Model Explanations

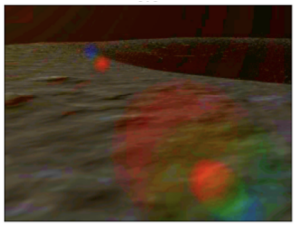
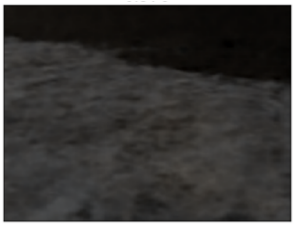
Original Image	Counterfactual Image	Original Explanation	Counterfactual Explanation
		*The image appears to be a **pixelated** and **blurry** representation of a lunar surface, with **indistinct features** and a **lack of clear boundaries** between different regions. This level of **pixelation** and **blurriness** may be due to the **low resolution** of the image or the presence of **noise**, which can make it challenging for the model to accurately identify and classify the different features in the image. As a result, the model may be uncertain about its predictions, leading to a reduction in confidence.*	*The model's reduced confidence in the first image can be attributed to the presence of a **rainbow-colored reflection**, which may have introduced **color inconsistencies** or **over-saturation** that deviate from the expected **color palette** of the simulated lunar environment. This discrepancy could have caused the model to struggle with accurately classifying the image, resulting in lower confidence.*

Figure 14. (i) Low-competency image with increased *saturation*. (ii) Counterfactual generated by Reco. (iii) Incorrect LLaMA explanation generated from original image alone. (iv) Correct LLaMA explanation generated with aid of counterfactual.

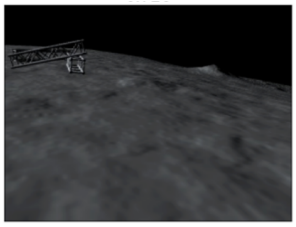
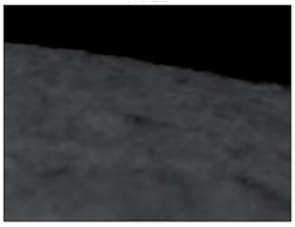
Original Image	Counterfactual Image	Original Explanation	Counterfactual Explanation
		*The image appears to be a grayscale representation of a simulated lunar environment, featuring a **blurry** and **low-resolution** depiction of the terrain. The model's lack of confidence may be attributed to the image's **poor quality**, which could lead to difficulties in accurately identifying the features and patterns necessary for classification. The model may struggle to distinguish between the different regions of the terrain due to the **low resolution** and **lack of clear boundaries**, resulting in a reduced confidence in its predictions.*	*The model's reduced confidence in the original image may be attributed to the presence of a **metal object** in the top left corner, which introduces an element of unpredictability and challenges the model's ability to accurately classify the image. This **object** may have caused the model to hesitate or struggle with making a definitive prediction, resulting in lower confidence.*

Figure 15. (i) Low-competency image with *spatial* anomaly (ladder at top left). (ii) Counterfactual generated by LGD. (iii) Incorrect LLaMA explanation generated from original image alone. (iv) Correct LLaMA explanation generated with aid of counterfactual.

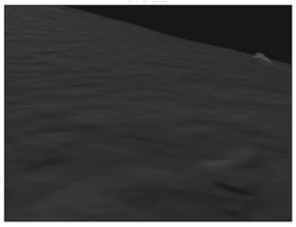
Original Image	Counterfactual Image	Original Explanation	Counterfactual Explanation
		*The image appears to be a grayscale representation of a lunar surface, featuring a dark background with a lighter-colored foreground. The foreground exhibits a **mix of smooth and bumpy textures**, with some areas displaying a more uniform tone and others showing a greater degree of variation. This **complexity in the image's texture and tone** may contribute to the model's reduced confidence, as it may struggle to accurately classify the image due to the ambiguity in the visual features presented.*	*The model's lower confidence in the first image can be attributed to the presence of **noise** or **low-contrast features**, which may have resulted in a less robust representation of the image. This, in turn, could have led to a decrease in the model's ability to accurately classify the image.*

Figure 16. (i) Low-competency image with reduced *contrast*. (ii) Counterfactual generated by LNN. (iii) Incorrect LLaMA explanation generated from original image alone. (iv) Correct LLaMA explanation generated with aid of counterfactual.

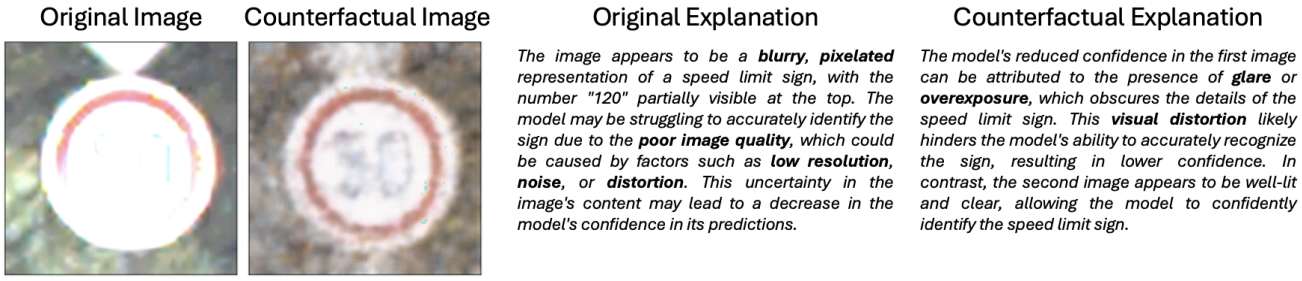


Figure 17. (i) Low-competency image with increased *brightness*. (ii) Counterfactual generated by Reco. (iii) Incorrect LLaMA explanation generated from original image alone. (iv) Correct LLaMA explanation generated with aid of counterfactual.

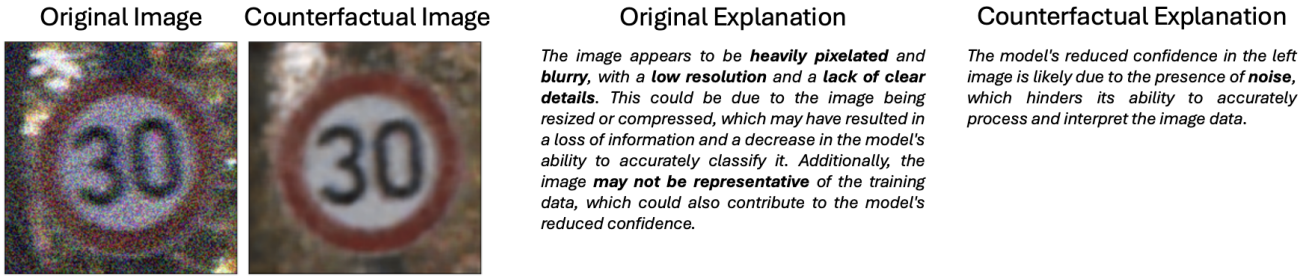


Figure 18. (i) Low-competency image with additive *noise*. (ii) Counterfactual generated by LGD. (iii) Incorrect LLaMA explanation generated from original image alone. (iv) Correct LLaMA explanation generated with aid of counterfactual.

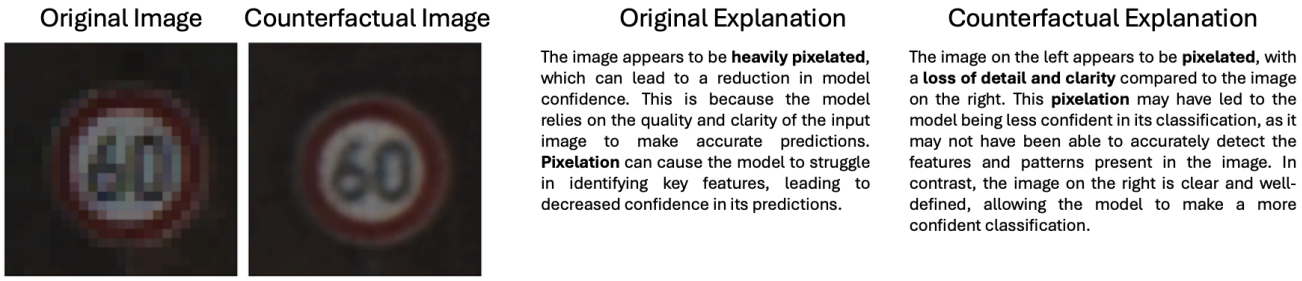


Figure 19. (i) Low-competency image with *pixelation*. (ii) Counterfactual generated by LNN. (iii) Correct LLaMA explanation generated from original image alone. (iv) Correct LLaMA explanation generated with aid of counterfactual.

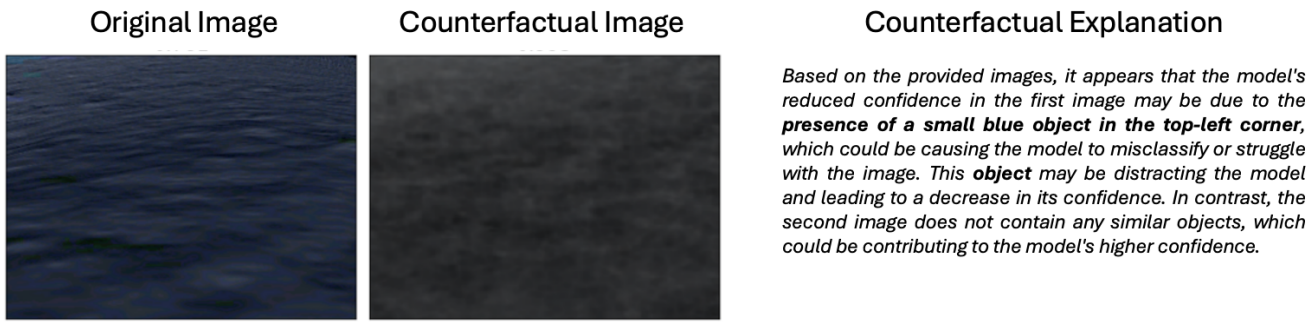


Figure 20. (i) Low-competency image with increased saturation. (ii) Counterfactual generated by LGD. (iii) Incorrect LLaMA explanation that contains a *hallucination*.

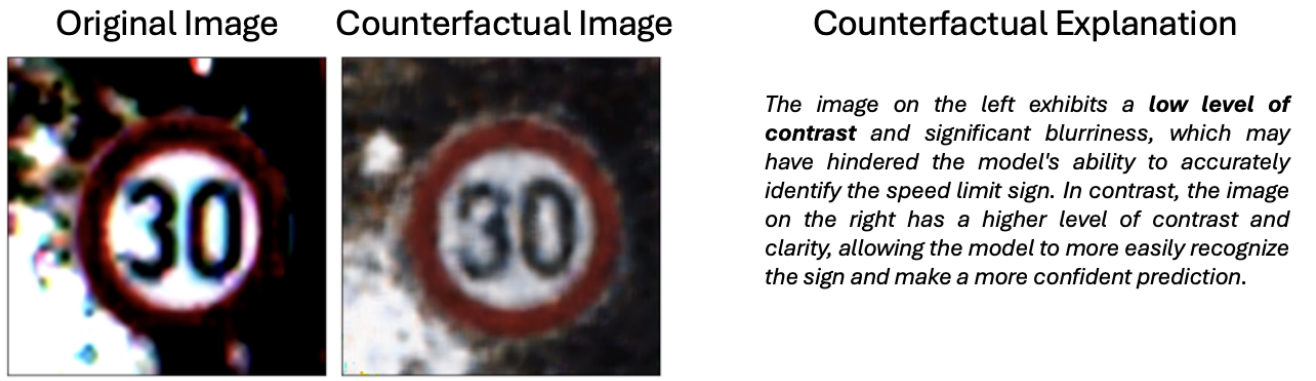


Figure 21. (i) Low-competency image with increased contrast. (ii) Counterfactual generated by Reco. (iii) LLaMA explanation that *inverts* the reason for low competency.