

Infosys

Responsible AI Office

Infosys Responsible AI Toolkit

Image Explainability

API usage Instructions

Contents

Introduction	1
Image Analysis	1
Explanation of Image with Detected Objects	3

Introduction

Image Explain offers detailed explanations for images generated by Large Language Models (LLMs). It provides an in-depth analysis of the image, highlighting key insights such as the presence of **watermarks**, detection of **potential biases**, and identification of the **visual style** used. In addition, the system evaluates the image using key metrics like **creativity** and **certainty**.

When applicable, Image Explain also performs **object detection**, identifying and labeling objects within the image. It then provides an explanation based on the detected objects. This enhances the interpretability and trustworthiness of AI-generated visuals by combining both image-level and object-level understanding.

Once API swagger page is populated as per instructions given in the github repository Readme file, click on 'try it out' to use required endpoints. Details of endpoints associated with ImageExplainability tenet are outlined below.

Image Analysis

Endpoint – /rai/v1/image-explainability/analyze

Using this API, we can view the explanation of the uploaded image along with additional evaluations. The system also performs the following checks:

- **Watermark Detection:** Determines whether the image was AI-generated by detecting embedded watermarks.
- **Bias Assessment:** Identifies any unfair or skewed representations present in the image.
- **Style Classification:** Identifies the artistic or visual style (e.g., Digital Art) and provides a rationale through style analysis
- **Creativity Score:** Measures the uniqueness and originality of the image generated.
- **Certainty Score:** is a measure of how confident the system is in its analysis of the image

Input:

image: Upload the image you want to analyze. **evaluator:** Use GPT-4o as the

evaluator. **prompt:** Provide the prompt you want to use for analysis or querying the image.

query_flag: By default, query_flag is set to **false**. If you're asking any query related to the image, set query_flag to **true** and provide the query in the prompt field.

image * required
string(\$binary)

generate an image of a nurse 3 (1).PNG

evaluator
string

☒ Send empty value

prompt
string

☒ Send empty value

query_flag
boolean

☒ Send empty value

Response:

```
{
  "image_description": "The image features a person wearing a white lab coat, which is a typical attire for medical professionals. Around the person's neck is a stethoscope, a common tool used by doctors and nurses. The background is a textured, monochromatic surface, and the overall color scheme of the image is grayscale, giving it a classic and formal appearance.",
  "insights": {
    "watermark": "None",
    "bias_type": "No bias",
    "bias_analysis": "No bias identified",
    "style": "Digital Art",
    "style_analysis": "The image is classified as Digital Art due to its clean lines and the use of a monochromatic color palette. The texture in the background and the smooth rendering of the lab coat and stethoscope suggest a digital manipulation or creation process. The grayscale tones and the formal composition contribute to a professional and polished look, typical of digital illustrations used in professional or educational contexts.",
    "query_response": "NA"
  },
  "metrics": {
    "certainty_score": 10,
    "certainty_label": "Highly certain",
    "creativity_score": 60,
    "creativity_label": "Moderately Creative"
  },
  "super_pixels": ""
}
```

Explanation of Image with Detected Objects

Endpoint - /rai/v1/image-explainability/object-detection

Using this API, we can get detailed explanations of images based on detected objects.

Request Payload:

image * required
string(\$binary)

adversarial image with detection.png


evaluator
string

☐ Send empty value

image: Upload an adversarial or non-adversarial image with bounding boxes or object detection annotations. **evaluator:** Use GPT-4o as the evaluator.

Response:

```
{
  "explanation": "The model is able to correctly identify objects like 'person' in the image. However, it falsely detected a 'truck', which is not present in the image. This misidentification could be due to the model's confusion between similar shapes or features, leading to an incorrect classification.",
  "predicted_image": "",
  "time_taken": 11.291
}
```

 [Download](#)