future internet

MDPI

Article

Using Multimodal Foundation Models for Detecting Fake Images on the Internet with Explanations

Vishnu S. Pendyala [1,*] and Ashwin Chintalapati [2]

[1] Department of Applied Data Science, San Jose State University, San Jose, CA 95192, USA
[2] Department of Computer Science, Purdue University, West Lafayette, IN 47907, USA; achintiii@outlook.com
[*] Correspondence: vishnu.pendyala@sjsu.edu

Abstract: Generative AI and multimodal foundation models have fueled a proliferation of fake content on the Internet. This paper investigates if foundation models help detect and thereby contain the spread of fake images. The task of detecting fake images is a formidable challenge owing to its visual nature and intricate analysis. This paper details experiments using four multimodal foundation models, Llava, CLIP, Moondream2, and Gemini 1.5 Flash, to detect fake images. Explainable AI techniques such as Local Interpretable Model-Agnostic Explanations (LIME) and removal-based explanations are used to gain insights into the detection process. The dataset used comprised real images and fake images generated by a generative artificial intelligence tool called MidJourney. Results show that the models can achieve up to a 69% accuracy rate in detecting fake images in an intuitively explainable way, as confirmed by multiple techniques and metrics.

Keywords: misinformation containment; large multimodal models; explainable AI; image processing

check for updates

Citation: Pendyala, V.S.; Chintalapati, A. Using Multimodal Foundation Models for Detecting Fake Images on the Internet with Explanations. *Future Internet* **2024**, *16*, 432. https://doi.org/10.3390/fi16120432

Academic Editor: Gianluigi Ferrari

Received: 17 October 2024
Revised: 15 November 2024
Accepted: 19 November 2024
Published: 21 November 2024

1. Introduction

Even before the arrival of multimodal large language models, the detection of fake images was shown to be difficult [1]. The advent of such models and, consequently, the rapid advancement of image manipulation techniques, often driven by artificial intelligence, has rendered traditional forensic methods increasingly ineffectual. These sophisticated image generation algorithms can create highly realistic and convincing alterations, making it arduous to discern authentic images from fabricated content. The inherent complexity of digital images, characterized by vast amounts of data and intricate patterns, necessitates computationally intensive analysis, often exceeding the capabilities of conventional detection tools. The continuous emergence of novel manipulation methods demands a constant adaptation of detection techniques, rendering the pursuit of a universal solution elusive. The subtle and often imperceptible nature of many image alterations can easily evade human perception, necessitating the development of robust automated detection systems, which this paper intends to address.

Foundation models such as LLaVA [2], Moondream 2 [3], and Gemini 1.5 Flash [4] can generate fake images. The work described in the following sections used the same foundation models to detect fake and authentic images from a given dataset. It compared and contrasted their performance in doing so, both quantitatively and quantitatively using evaluation metrics and explainable AI. The work also used another foundation model called CLIP [5] for the detection. CLIP is a model that learns visual concepts from natural language supervision. Its primary function is to understand the relationship between images and text. While it can be used as a component in image generation systems, it does not generate images. The performance of these four foundation models is explained using Local Interpretable Model-Agnostic Explanations (LIME) [6] and removal-based explanations [7].

1.1. Related Work

For various reasons, misinformation containment is largely an unsolved problem today [8]. While large language models (LLMs) have been used for detecting textual misinformation [9], there is hardly any evidence of using foundation multimodal models for fake image detection. A prior superficial study [10] concluded that GPT-4, Bard, and Bing were unreliable for detecting fake images, but no metrics were provided to support this conclusion. Another study [11] demonstrated that using large language models (LLMs) and that Vision Language Models (VLMs) can significantly improve object detection accuracy. In a different study [12], GPT 3.5, an LLM, was used to extract features to detect out-of-context (OOC) media. The study focused on the automated detection of the misuse of real photographs with conflicting captions. The authors proposed a method that enhances the COSMOS structure, which assesses the coherence between an image and its captions. By employing prompt engineering, the authors developed a robust feature extraction method that captures the correlation between captions.

Explainability techniques, LIME, and SHapley Additive exPlanations (SHAP) have been used to explain the classification of pneumonia from chest X-rays [13] and a rare and aggressive form of childhood eye cancer called retinoblastoma from fundus images [14], as well as the diagnosis of different abnormalities in human kidneys from computer tomography images [15], and more such image processing applications. SHAP has also been used in the context of land cover and land use classification in remote sensing [16]. However, there does not seem to be any evidence of these techniques being used in detecting fake images, which this study attempted.

Existing techniques for fake image detection include deepfake detection. Deepfake detection relies on traditional convolutional neural network-based architectures to make predictions with a focus on classification techniques. These techniques largely focus on maximizing accuracy but provide limited explainability features. Using foundational multimodal models can help provide transparency in detections to understand the strengths and weaknesses of each model. Some of the deepfake detection approaches are discussed below.

1.1.1. Classification Based on Spatial and Temporal Features for Deepfake Detection

Fake image detection approaches typically focus on spatial and temporal features, while foundation models leverage extensive unlabeled data through self-supervised learning, enhancing detection accuracy and reducing demographic bias. Convolutional transfer deepfake detection [17] has been used to learn low-level spatial features and temporal information. Existing CNN architectures are modified to take on multiple Transformer layers, each taking separate temporal and spatial dimensions to optimize performance.

1.1.2. Fake Image Detection Using Foundation Models

Despite this, foundation models can still play an important role in deepfake image detection. Combining traditional deepfake techniques using convolutional neural networks with foundation models can help enhance fake image detection [18]. The research highlights the importance of analyzing biases related to age, gender, and ethnicity when using deepfake detection. Foundation models are designed to reduce bias with their extensive and diverse training data. All four models integrate techniques such as bias audits and detection, post-training mitigation techniques, and constant model evaluation. Studies have shown that even though foundation models such as LLMs are not trained for deepfake detection, they can use their world knowledge to perform reasonably well on the task [19]. Studies have also shown that prompt engineering can improve performance.

1.1.3. Interpretability of Foundation Models in Image-Related Tasks

A novel approach to improve the interpretability of multimodal large language models (MLLMs) by leveraging the image embedding component was proposed by integrating an open-world localization model with an MLLM [20]. The architecture was claimed to significantly enhance interpretability, allowing for the creation of a novel saliency map to explain any output token, the identification of model hallucinations, and the assessment of model biases through semantic adversarial perturbations. In a novel approach, pretrained language models were used to interpret the features learned by image classifiers [21]. By connecting the feature space of image classifiers with language models, the system called TExplain generates textual explanations during inference. These explanations help to identify frequent words and patterns, providing insights into the classifier's behavior, detecting spurious correlations and biases, and enhancing the interpretability and robustness of image classifiers.

1.2. Contributions of This Study

As discovered in the literature review, there is hardly any evidence of using foundation models for fake image detection. This study is unique in using foundation models for fake image detection and explaining the process using explainable AI techniques. In addition to LIME, the work also used removal-based explanation, a technique that has been gaining significance recently. The results show that there is promise in the idea and also demonstrate the need for further improvements in foundation models.

The rest of this article is organized as follows. The next section describes the overall design and the artifacts used for the experiments. The materials detailed include the dataset and foundation models. Since LLMs generate text based on patterns in the input they receive, the way an input "prompt" is formulated can significantly impact the quality and specificity of the output. Prompting strategies refer to the techniques and approaches used to craft effective input prompts that guide the model to produce the most relevant, accurate, or useful responses. Therefore, prompting strategies and experiments with the models are explained next. The following section describes the performance evaluation of the models. Challenges faced, difficulties overcome, and results are discussed next, followed by the application of explainability techniques and the conclusion.

2. Materials and Methods

A broad overview of the approach taken for this work is illustrated in Figure 1. Foundation models such as CLIP were tested on the dataset, and their performance was evaluated using appropriate metrics. The classifications made by the foundation models were then evaluated using various explainable AI techniques such as LIME and removal-based explanations. The dataset and other details are explained below.

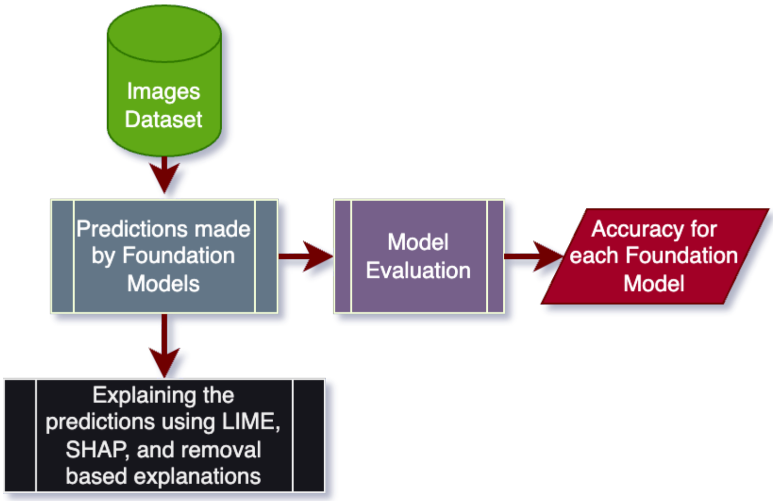
Images Dataset
Predictions made by Foundation Models
Model Evaluation
Accuracy for each Foundation Model
Explaining the predictions using LIME, SHAP, and removal based explanations

Figure 1. Design of the experiments.

2.1. Dataset

The dataset used for the experiments is available in the public domain on Kaggle [22]. It contains the following 11 classes of images: traffic lights, frogs, dogs, cats, airplanes, cars, trucks, ships, birds, horses, and deer. A portion of these images was selected for evaluation (from the test directory), and another portion was selected for n-shot prompting (from the train directory). For the zero-shot prompting, a total of 113 real images and 116 fake images were used as test images.

2.2. Foundation Models

Four different multimodal models, three downloaded from Hugging Face, namely, Llava 1.6, CLIP, and Moondream2, and one other, Gemini 1.5 flash from Google, were used for these experiments. Large Language and Vision Assistant (Llava) [2,23], is an advanced multimodal model that integrates a vision encoder with a language model to enhance visual and language comprehension. The model has demonstrated appreciable visual reasoning capabilities [24]. The second model used was OpenAI's CLIP, a multimodal model specializing in binary image classification. The third model used was Moondream2, also from Hugging Face. Lastly, Gemini 1.5 flash was used, which is Google's fastest multimodal model known for its use with a variety of tasks. It features a long context window, allowing it to process extensive data efficiently, and is optimized for speed with sub-second average first-token latency.

2.3. Overview of Model Architectures

To understand the strengths and weaknesses of each foundation model used in testing and experimentation, the architectures of these four models are observed. This will help make predictions on expected results based on the strengths and weaknesses of these models.

Starting with CLIP, it is a multimodal learning architecture developed by OpenAI. CLIP is an image classifier, meaning that its specialty is in binary image classification. CLIP trains its model on a large-scale dataset containing images and their corresponding text descriptions, which renders it similar in capability to the GPT-2 and GPT-3 models. CLIP uses contrastive learning, a technique largely popular in the field of unsupervised learning. CLIP has several components, one of which is an image encoder, which utilizes architectures and ResNet models to produce high-dimensional vector representations. CLIP also has a text encoder, which allows it to convert textual descriptions into embeddings. CLIP has a vast amount of training data that allow it to make its predictions. CLIP has versatility and is effective for image retrieval, image classification, and matching images with textual descriptions.

Llava 1.6 is an extension of large language models like Llama, which integrates language and vision encoders separately. Unlike CLIP, which has the primary task of zero-shot image classification, Llava's architecture aims for comprehensive multimodal understanding. Llava is classified as an auto-regressive language model based on its transformer architecture. Because of this, Llava can perform tasks that require both textual and visual comprehension, such as visual question answering and image captioning. While CLIP is to be an image classifier based on images and contextual prompts, Llava's goal is to operate as an LLM with additional image processing capabilities. Because of this, one might expect additional n-shot prompting to be successful when using Llava.

Google's Gemini 1.5 Flash Multimodal LLM processes up to 1 million tokens in its long context window and is optimized for low-latency tasks. Similarly to LLava, Gemini's model can act as both a text model and an image model separately. Gemini also uses a dual-encoder structure similar to that of the CLIP image classification model. Unlike Llava or CLIP, Gemini's transformer decoder model architecture means it is designed for lightweight, optimized efficiency. This makes it highly suitable for real-time AI-generated-image detection tasks. However, Gemini 1.5 Flash's explainability features are slightly more limited compared to other foundation models.

Moondream2 is a smaller, multimodal LLM authored by vikhayk on HuggingFace. It uses a transformer-based architecture similar to those of Llava and CLIP. It uses approximately 1.87 billion parameters and training data to generate responses to a given image and text prompt. Given that this is an authored model, it is frequently updated, and potential weaknesses this model could have, such as reduced ability to scale on batched images, are addressed.

2.4. Prompting and Experimentation

To meet the respective model nuances, the prompting strategies and responses to the given prompts for each of the models differed slightly. Llava 1.6 was best at interpreting the prompt as a multiple-choice question. An example prompt for the Llava 1.6 model is as follows:

```
Is the image real or fake?
(A) Real
(B) Fake
```

The model outputs responses as an answer to the multiple-choice question, either A/B or (A) Real/(B) Fake.

For OpenAI's CLIP model, since it is a binary classification model, the prompts are passed in as labels. The first label provided is "You are an image forensics expert. This is a real image with no technical interference". The second label provided is "You are an image forensics expert. This is a fake image with technical interference". From there, the softmax function is used to get the models to output the respective probabilities for each label. The label with the higher probability determines if the model classifies the image as real or fake.

The Moondream2 model had a slightly modified prompt. It had difficulty answering the prompt as a multiple-choice question. The prompt that worked best with this model is as follows: "Determine if the image is real or fake and answer clearly". This prompted the model to respond in a concise statement whether it believed the image provided was real or fake. The response from this model is of the form "This image is real/fake", followed by a brief description of the image. The following are examples of real responses that the model output:

1. The image is real, as it shows a horse standing in a field of tall grass.
2. The image is a fake or computer-generated (CGI) image of a city street with a traffic light.
3. The image is real, as it shows a large ship sailing across the ocean.
4. The image is a fake, as it is a 3D rendering of a cat sitting on a blue background.

String parsing was then used to determine the prediction the LLM made and classify the prediction as real or fake.

Google Gemini answers the multiple-choice question similarly to the Llava model. However, at first, its responses were a bit indecisive, and on some occasions, it would not come to a clear decision. Thus, we had to use contextual priming to obtain an updated prompt: "You are an image forensics expert. Analyze the image and state concisely whether it is (A)real or (B)fake? Choice:("

The following are examples of responses generated:

1. Choice: **(A)real** This image appears to be a real photograph. There are no obvious signs of manipulation or digital alteration.
2. Choice: **(B)fake** The image appears to be generated using AI. The frog's texture and lighting are too perfect and unrealistic, and the water's reflections are artificial. The overall image lacks the natural imperfections and noise found in real-world photographs.

2.5. Evaluation

Experiments were run by taking a random sample of 25 images from the test images and having the model predict whether each of those 25 images was real. This was carried out 20 separate times, and with each iteration, the accuracy score, precision score, F1 score, and the Matthews correlation coefficient were obtained, all from the sklearn.metrics library.

In accumulating these 20 iterations, the mean value of each metric was taken to generate an average accuracy, precision, F1, and MCC value.

2.6. Challenges and Difficulties

Throughout this process, there were some challenges faced. Firstly, the majority of the models were computationally intense, which made for a slow process in terms of gathering the results and running the experiments. Next, selecting different models that would be suited for the task at hand was an unexpected challenge. One image classifier and three multimodal image/text models were used, and a host of other potential candidate models considered were not successful because they were unable to process images. Some of these models had limited capability as image captioning models without much scope for using them for the problem being addressed. Each model also had different prompting techniques required to achieve the task at hand, and different response methods.

3. Results and Discussion

The results in Table 1 show that the models had varying levels of success with both zero-shot and n-shot prompting.

Table 1. Performance metrics for various models.

Model	Prompting Technique	Accuracy	Precision	F1 Score	MCC
Llava	0-Shot	0.62	0.73	0.583	0.347
	1-Shot	0.65	0.77	0.609	0.38
	2-Shot	0.65	0.76	0.613	0.39
	3-Shot	0.66	0.77	0.613	0.391
	4-Shot	0.696	0.794	0.657	0.424
CLIP	0-Shot	0.48	0.446	0.407	−0.07
	1-Shot	0.452	0.460	0.567	−0.122
	2 Shot	0.488	0.502	0.609	−0.020
	3-Shot	0.488	0.505	0.558	−0.057
	4-Shot	0.500	0.451	0.505	0.024
Moondream2	0-Shot	0.632	0.784	0.579	0.375
	1-Shot	0.650	0.594	0.732	0.435
	2-Shot	0.583	0.529	0.660	0.248
	3-Shot	0.583	0.578	0.705	0.188
	4-Shot	0.667	0.639	0.760	0.300
Gemini 1.5 Flash	0 Shot	0.55	0.6804	0.5148	0.2075
	1-Shot	0.6375	0.6990	0.5839	0.3181
	2-Shot	0.6	0.5741	0.5512	0.1095
	3-Shot	0.575	0.7749	0.5344	0.3253
	4-Shot	0.6875	0.6966	0.6325	0.4013

The Llava model had a relatively steady and moderate increase in all metrics as the model gained more context due to relative prompts. It had the highest accuracy, precision, and Matthews correlation coefficient with four-shot prompting. Each prompt seemed to train the model better and add clarity to the model. The few-shot learning strategy helped it steadily improve.

The CLIP model for image classification performed the worst of all the models. This could be because CLIP is not a text generator, rather it is an image classifier. Image classification models are typically specialized to categorize given images into predefined classes. Although CLIP allows the customization of labels, it is still based largely on pretraining. Because we are not attempting to classify images into predetermined classes, this could explain why CLIP was not quite as successful.

Both Moondream2 and Gemini 1.5 Flash performed similarly in terms of their overall results. The results from one-shot prompting were better than that of zero-shot prompting;

however, the two- and three-shot prompting results were worse as the metrics dipped once again. Four-shot prompting seemed to be enough for both models to determine consistent patterns across the dataset and generate better results.

3.1. LIME Explainability

To understand the decision-making process of each of the models as more prompts and context were added, LIME explainability was used first. LIME generates several perturbed samples of images and makes predictions on those perturbed samples. From there, LIME assigns weights to those perturbed samples based on their proximity to the original instance, and a linear regression model is run on the perturbed samples and predicted outcomes. From the results of this model we can identify which features both positively and negatively impact model confidence in its predictions. Figures 2–4 illustrate this idea.

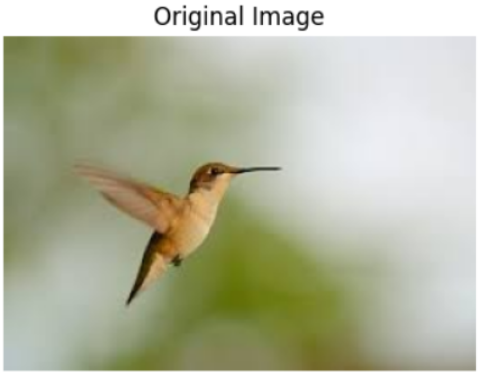

Figure 2. A picture of a bird.

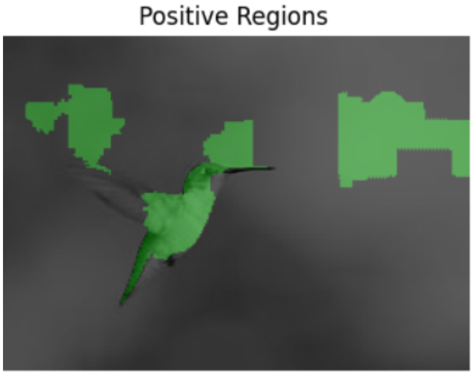

Figure 3. Positive regions of confidence.

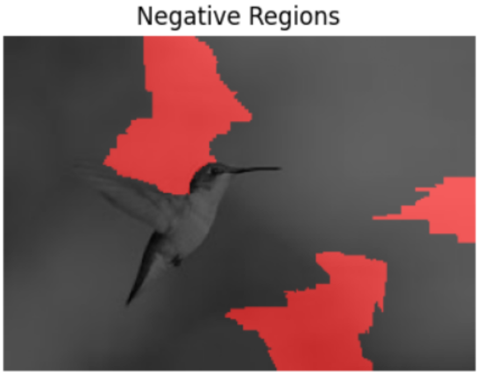

Figure 4. Negative regions of confidence.

In the example image in Figure 2, the bird appears to be in mid-flight. The Llava model predicts that this image is real, which is indeed the correct prediction, and that there is no fabrication. Running the LIME explainability technique on the model for this image reveals the regions of positive and negative confidence for the model when making its prediction. The positive regions are shown in Figure 3 and the negative regions in Figure 4. The results shown are for the Llava model.

The results from LIME are aligned with human intuition. The "positive regions" are the regions that are heavily strengthening the model's confidence that the image is real (which was its prediction). The results appear to make sense, as the regions are indeed important in determining that the bird in the photo seems lifelike and does not appear to have been manipulated. The negative regions of confidence mainly appear to be in the background. The background of this image is quite blurry and can seem to raise doubts as to whether or not this image has been fabricated.

The LIME explainability results differed for all models in regard to how different prompting techniques influenced the different confidence regions. Starting with Llava, the regions of confidence did not differ despite the increased context provided by the model. This meant that for each image, the exact same positive regions of confidence and negative regions of confidence were always generated. This means that the inherent features that the model relies on to make predictions do not change. However, the model can calibrate its understanding of the features better and avoid overfitting with more provided examples.

With regard to the CLIP image classification model, LIME reveals that the regions of confidence changed when prompts were added. Figure 5 shows the results for a real image of a dog with zero-shot prompting, while Figure 6 with one-shot prompting.

The regions of confidence differ as a contextual prompt is provided. The positive regions shift from a large portion of the background to a more prominent region of the image, the dog's face. The negative regions also shift from a large portion of the background to the grass in front of and near the subject. CLIP's focus shifts to new areas of the image when a contextual prompt is added. However, the regions of confidence stay the same when additional prompts are added.

The Moondream2 model highlights explainability regions similarly to the Llava model. With additional contextual prompting, the same regions of positive and negative confidence are found. However, unlike the Llava model, Moondream2 does not show continued improvement as more contextual prompts are added. Contextual prompting has been proven to be less effective when using Moondream2.

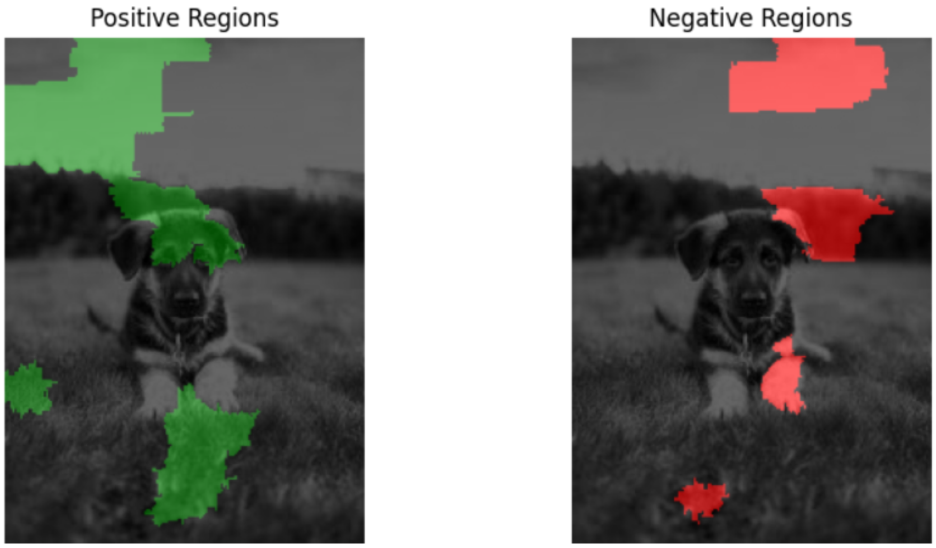

Figure 5. Positive and negative regions generated by CLIP with zero-shot prompting.

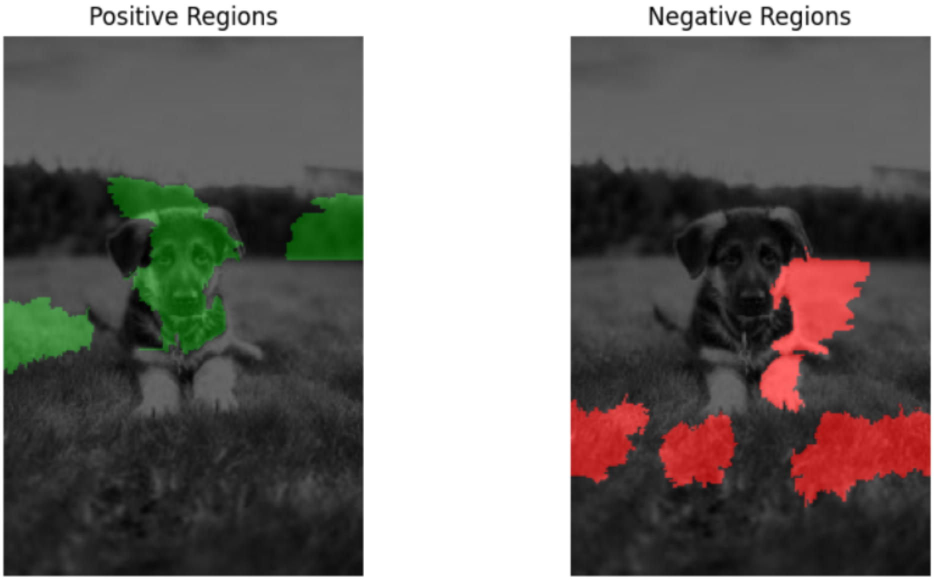

Figure 6. Positive and negative regions generated by CLIP with one-shot prompting.

Another observation made when looking at the regions generated with Moondream2 shown in Figures 7 and 8 is that the regions of positive and negative confidence take up a much greater portion of the overall image as opposed to with other models. These larger regions suggest that the model might be leveraging broader features rather than focusing on the fine-grained details of the image. In addition, Moondream might outperform other models in tasks where understanding the overall scene is crucial. This also means that the model could use improvement on tasks involving fine detail analysis of an image.

With the Gemini 1.5 Flash model, each additional contextual prompt changes the regions of high and low confidence. The results obtained from analyzing a real image of a cat are shown in Figures 9–13.

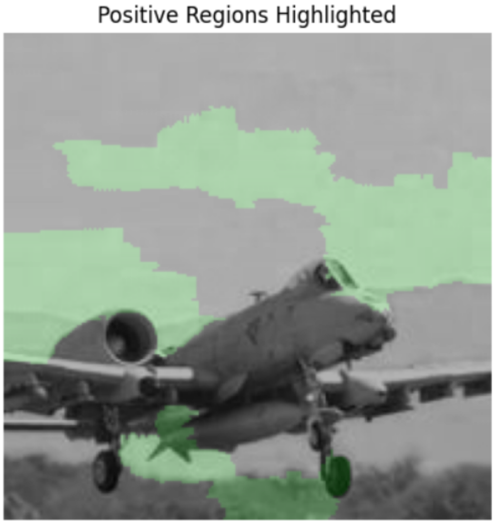

Figure 7. Positive regions generated by Moondream2 model.

These results show that Gemini is the only model where positive and negative regions change for every single prompting technique. The results obtained from the previous section showed that the biggest difference in results occurred when going from zero-shot prompting to one-shot prompting. This somewhat tracks when looking at the results generated by LIME. The positive regions shift more from the carpet to the cat's face and body, which is the subject of the image. The negative regions, which were largely centered on the cat's features, shifted toward the carpet and surrounding area. In general, the extra prompting did not help Gemini that much. However, it was taking the prompt into context, and the regions of confidence changed more often than with any other model.

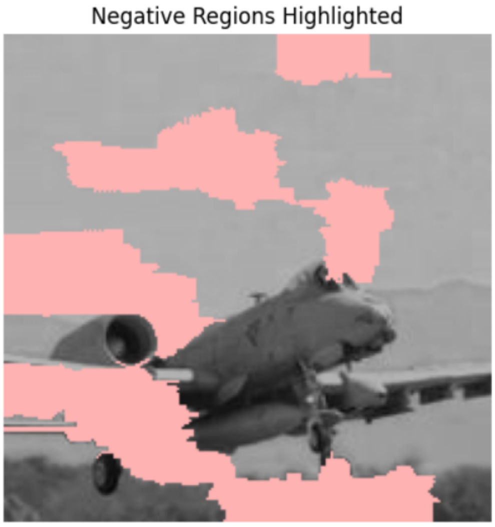

Figure 8. Negative regions generated by Moondream2 model.

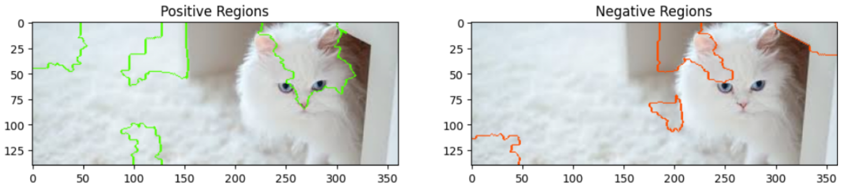

Figure 9. Positive and negative regions generated by Gemini for zero-shot prompting.

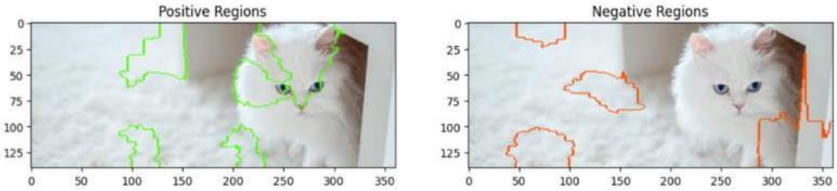

Figure 10. Positive and negative regions generated by Gemini for one-shot prompting.

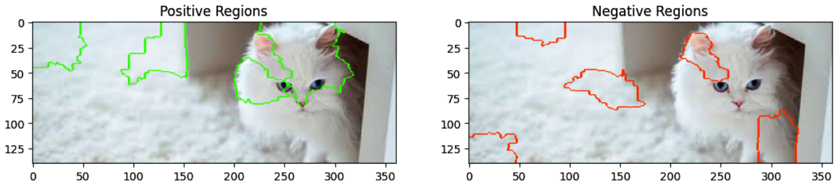

Figure 11. Positive and negative regions generated by Gemini for two-shot prompting.

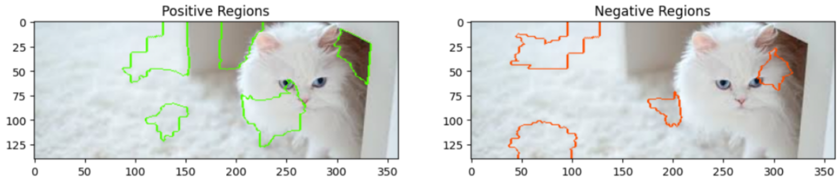

Figure 12. Positive and negative regions generated by Gemini for three-shot prompting.

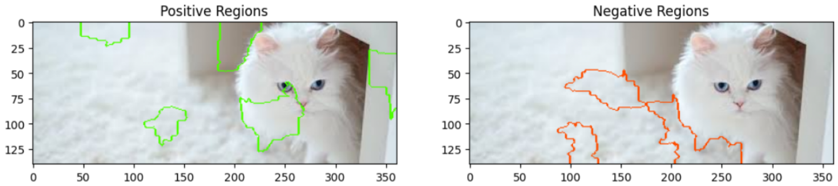

Figure 13. Positive and negative regions generated by Gemini for four-shot prompting.

There was a limitation obtained from using LIME. A small portion of images when run with the LIME explainer generated very little positive and very little negative regions. This could be because of the following:

1. The model is overfitting to specific features irrelevant during training;
2. The model might have learned misleading correlations;
3. Approximating the model's behavior could have limitations.

Further understanding these issues requires continued research and testing.

3.2. Jaccard Index

To further quantify the results we interpreted using LIME, the Jaccard similarity coefficient can be used. The Jaccard Index, also known as the Jaccard similarity coefficient, is a statistical measure used to quantify the similarity and diversity between two sets. In the context of image analysis and model interpretability, particularly with LIME, the Jaccard Index can be employed to evaluate the overlap between regions identified as positively or negatively contributing to a model's decision.

Mathematically, the Jaccard Index is defined as

$$\text{Jaccard Index} = \frac{|A \cap B|}{|A \cup B|}$$

where $|A \cap B|$ represents the size of the intersection of sets A and B (i.e., the common elements between the two sets), and $|A \cup B|$ denotes the size of the union of sets A and B (i.e., all unique elements present in either set).

3.3. Jaccard Index for Evaluating Overlap Between Positive and Negative Regions

The Jaccard Index, also known as the Jaccard similarity coefficient, is a statistical measure used to quantify the similarity and diversity between two sets. In the context of image analysis and model interpretability, particularly with LIME, the Jaccard Index can be employed to evaluate the overlap between regions identified as positively or negatively contributing to a model's decision.

Significance of the Jaccard Index

In this study, the Jaccard Index is utilized to compare the overlap between the positive regions (areas of the image that positively influence the model's prediction) and the negative regions (areas that negatively influence the model's prediction) highlighted by LIME:

- Jaccard Index close to 1: A high Jaccard Index indicates significant overlap between positive and negative regions. This suggests that the model may rely on the same areas of the image for both positive and negative contributions, potentially indicating regions of ambiguity or mixed relevance.
- Jaccard Index close to 0: A low Jaccard Index suggests minimal overlap, indicating that the model clearly distinguishes between different parts of the image for positive and negative contributions. This separation can reflect a well-defined decision-making process where distinct regions contribute either positively or negatively, but not both.
- Moderate Jaccard Index: A moderate value suggests that while there is some overlap, there are also distinct areas that are uniquely positive or negative. This could reflect a

balanced interpretation of the model, where certain features play dual roles depending on the context.

The Jaccard Index thus provides a quantitative measure to assess the extent to which the model's focus areas overlap for positive and negative influences, offering insights into the model's interpretability and decision-making behavior. The results from taking each model's average Jaccard index over a sampling of images are shown in Table 2.

Table 2. Jaccard scores for different models.

Model	**Jaccard Score**
Llava	0.0
CLIP	0.0
Moondream2	0.758
Gemini	0.0

The Moondream2 model is the only model with a Jaccard score that is not 0. This confirms our suspicions that this model relies on the same regions for both positive and negative contributions, which could imply that it is not great at very fine, detailed tasks. The other three models have a Jaccard score of 0, signifying no overlap, which can mean a more precise decision-making method.

3.4. Removal-Based Explanations

Another approach to understanding model explainability is the removal-based explanation approach. This approach involves systematically occluding varying regions of an image and observing how these changes reflect in the model's predictions. This approach can help identify the most and least influential regions of an image that contribute to the model's decision making. The steps for generating removal-based explanations are as follows:

1. Define the prediction function.
2. Generate the baseline predictions for the image using the original prediction function.
3. Define a function to occlude regions in an image and create a list of regions to occlude. Change the region size and step size of the occlusion as needed.
4. For each region, measure the change in probabilities from the baseline prediction to the new prediction of the occluded image.
5. Output the most and least influential regions of the image.

The exact formula for calculating the most and least influential regions of the image is simply using the change in the Euclidean distance between the array of baseline probabilities and the array of new probabilities. Euclidean distance allows us to consider the difference across all dimensions, or elements in the array. In addition, it allows the comparison of probability vectors in their entirety. The Euclidean distance can be used to effectively compare the relative importance of each feature and is an efficient computation. The format of the probability logits for the Llava 1.6 model is as follows when rounded to three significant figures:

$$\begin{bmatrix} 5.628 \times 10^{-9} & 6.580 \times 10^{-9} & 5.843 \times 10^{-7} & 8.407 \times 10^{-7} & 8.424 \times 10^{-7} & 8.423 \times 10^{-7} \\ 3.244 \times 10^{-9} & 2.710 \times 10^{-9} & 8.083 \times 10^{-6} & 1.673 \times 10^{-5} & 1.667 \times 10^{-5} & 1.676 \times 10^{-5} \\ 7.354 \times 10^{-9} & 7.297 \times 10^{-9} & 3.910 \times 10^{-5} & 5.414 \times 10^{-5} & 5.399 \times 10^{-5} & 5.411 \times 10^{-5} \\ 3.539 \times 10^{-12} & 3.148 \times 10^{-12} & 6.416 \times 10^{-4} & 3.512 \times 10^{-8} & 3.483 \times 10^{-8} & 3.512 \times 10^{-8} \\ 7.207 \times 10^{-11} & 1.090 \times 10^{-10} & 4.333 \times 10^{-4} & 1.199 \times 10^{-6} & 1.188 \times 10^{-6} & 1.196 \times 10^{-6} \\ 6.364 \times 10^{-11} & 5.660 \times 10^{-11} & 2.718 \times 10^{-5} & 4.118 \times 10^{-7} & 4.065 \times 10^{-7} & 4.085 \times 10^{-7} \end{bmatrix}$$

The resulting array of all Euclidean distances generated by the repeated occlusion of portions of the image is shown in Table 3.

Table 3. Top 5 most and least influential regions.

Region (Coordinates)	Influence Value
Most Influential Regions	
(90, 135, 60, 30)	11.738492
(60, 120, 60, 30)	11.481618
(30, 120, 60, 30)	11.475345
(60, 135, 60, 30)	11.300630
(90, 120, 60, 30)	11.210486
Least Influential Regions	
(270, 165, 60, 30)	9.290663
(240, 165, 60, 30)	9.257188
(270, 120, 60, 30)	9.470703
(270, 60, 60, 30)	9.491793
(270, 135, 60, 30)	9.506325

Since CLIP is a pure image classification model, the probability logits outputted are much simpler to understand. It is the probability that the image satisfies the first prompt followed by the probability that the image follows the second prompt. The format is as follows:

$$\text{Baseline Probabilities: } [0.43507442,\ 0.5649257]$$

Since the Euclidean distance calculations are carried out on an array of two numbers instead of several probability logits like with Llava, the calculations are much simpler and more efficient. The results are shown in Table 4.

Table 4. Top 5 most and least influential regions (CLIP model).

Region (Coordinates)	Euclidean Distance
Most Influential Regions	
(180, 138, 60, 30)	0.34646508
(180, 135, 60, 30)	0.3147826
(180, 15, 60, 30)	0.2995943
(150, 15, 60, 30)	0.29548472
(150, 105, 60, 30)	0.28423652
Least Influential Regions	
(270, 138, 60, 30)	0.005460461
(270, 135, 60, 30)	0.005460461
(270, 120, 60, 30)	0.005460461
(270, 105, 60, 30)	0.005460461
(270, 90, 60, 30)	0.005460461

This explainability technique does have limitations. Firstly, the technique systematically occludes parts of the image based on region size and step size. This means that these metrics are crucial in generating the regions that are occluded, and the optimization of these metrics is key. Next, because of this, the most and least influential regions remain consistent, which means that we cannot use removal-based explanations to identify the effectiveness of zero- vs. n-shot prompting. The most/least influential regions will remain the same, similar to what can be realized with LIME.

The result of using this explainability technique on a sample image is illustrated in Figures 14 and 15.

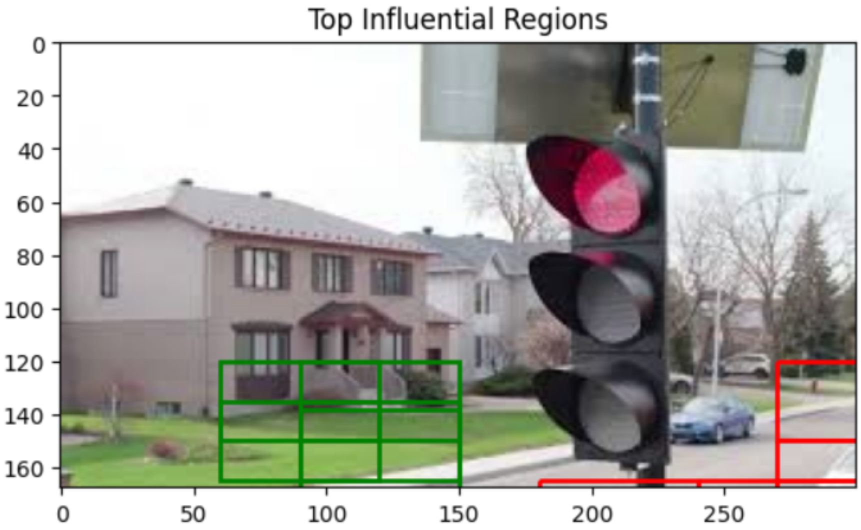

Figure 14. Most/least influential regions generated by Llava.

The comparison reveals interesting results. The Llava model's most influential regions comprised more of the house and grass in the background. CLIP's most influential regions comprised near and around the traffic light. However, both model's least influential regions were around the bottom right-hand corner of the image, which makes sense. In considering the road at the bottom right of the image is not the main focus of the image, it makes sense that these regions would be the least influential for either model to make a prediction.

One aspect to note is that this explainability technique requires direct access to logit probabilities generated by the model. Gemini 1.5 Flash does not provide access to those logits, and neither does Moondream2, so the explainability technique is not used for these models.

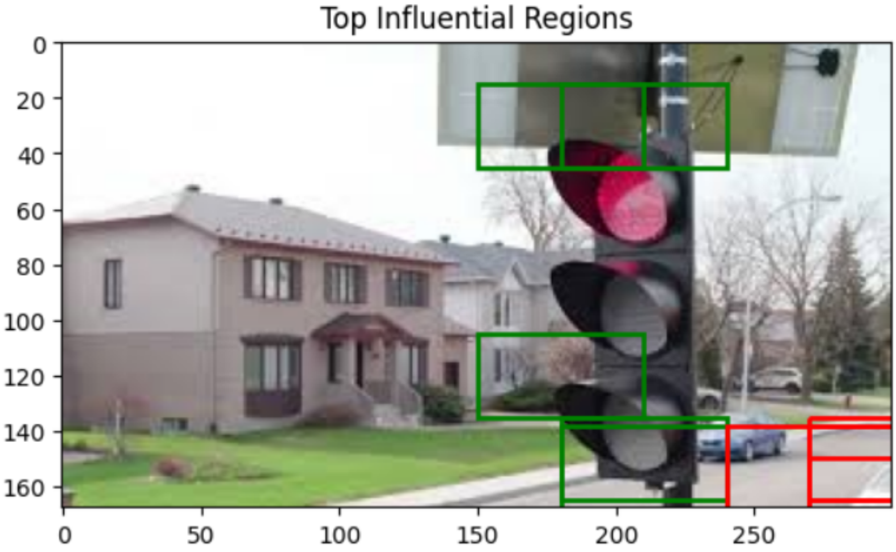

Figure 15. Most/least influential regions generated by CLIP.

4. Conclusions

As image generation and manipulation become more prevalent, the implications for misinformation, security, and trust in visual media grow more profound. This work proved that foundation models work reasonably well for fake image detection. Accuracy levels rose as high as 69.6 percent and the models Llava, CLIP, and Gemini each had MCC scores consistently in the range of 0.3 and 0.4. This indicates a positive correlation between the model's predictions and the desired outcomes. Multiple explainability techniques confirm that the functionality of the models aligns with human intuition. The highlighted regions examined heavily by the LIME and integrated gradient explainability techniques are prominent regions in the images. The multiple foundation models used for this work show promise that they can be used for fake image detection. It was observed that the models get better with training using few-shot learning. Specifically, Llava and Gemini showed considerable improvements as new contextual prompts were added, and few-shot prompting improved the accuracy and F1 score of each model. A future direction can be to fine-tune the models with additional training data and investigate their performance. Since the dataset of images represented 11 classes, there are other images we can pull to test these models further. Another research direction is to combine image analysis with the analysis of related information such as text and audio. Combining textual misinformation as well as fake images could be an intriguing tactic for analyzing multimodal models further. Yet, another research approach is to use interdisciplinary knowledge such as from digital forensics to enhance the analysis.

Author Contributions: Conceptualization, V.S.P.; methodology, V.S.P.; software, A.C. and V.S.P.; validation, A.C.; formal analysis, V.S.P. and A.C.; investigation, A.C. and V.S.P.; resources, A.C.; data curation, A.C.; writing—original draft preparation, V.S.P. and A.C.; writing—review and editing, V.S.P. and A.C.; visualization, A.C.; supervision, V.S.P.; project administration, V.S.P.; funding acquisition, V.S.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: No new data were created or analyzed in this study.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Zheng, L.; Zhang, Y.; Thing, V.L. A survey on image tampering and its detection in real-world photos. *J. Vis. Commun. Image Represent.* **2019**, *58*, 380–399. [CrossRef]
2. Liu, H.; Li, C.; Wu, Q.; Lee, Y.J. Visual instruction tuning. In Proceedings of the 17th International Conference on Neural Information Processing Systems (NIPS'04), Vancouver, BC, Canada, 13–18 December 2004.
3. Kumar, V. Moondream 2. 2024. Available online: https://huggingface.co/vikhyatk/moondream2 (accessed on 29 October 2024).
4. Google. Gemini Flash 1.5. Available online: https://ai.google.dev/gemini-api/docs/models/gemini#gemini-1.5-flash (accessed on 10 August 2024).
5. Radford, A.; Kim, J.W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. Learning transferable visual models from natural language supervision. In Proceedings of the International Conference on Machine Learning, Virtual, 18–24 July 2021; pp. 8748–8763.
6. Ribeiro, M.T.; Singh, S.; Guestrin, C. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. *arXiv* **2016**, arXiv:1602.04938.
7. Covert, I.; Lundberg, S.; Lee, S.I. Explaining by removing: A unified framework for model explanation. *J. Mach. Learn. Res.* **2021**, *22*, 1–90.
8. Pendyala, V.S. Misinformation Containment Using NLP and Machine Learning: Why the Problem Is Still Unsolved. In *Deep Learning Research Applications for Natural Language Processing*; IGI Global: Hershey, PA, USA, 2023; pp. 41–56.
9. Pendyala, V.S.; Hall, C.E. Explaining Misinformation Detection Using Large Language Models. *Electronics* **2024**, *13*, 1673. [CrossRef]
10. Al-Janabi, O.M.; Alyasiri, O.M.; Jebur, E.A. GPT-4 versus Bard and Bing: LLMs for Fake Image Detection. In Proceedings of the 2023 3rd International Conference on Intelligent Cybernetics Technology & Applications (ICICyTA), Denpasar, Bali, Indonesia, 13–15 December 2023; pp. 249–254.
11. Fan, Y.; Nie, J.; Sun, X.; Jiang, X. Exploring Foundation Models in Detecting Concerning Daily Functioning in Psychotherapeutic Context Based on Images from Smart Home Devices. In Proceedings of the 2024 IEEE International Workshop on Foundation Models for Cyber-Physical Systems & Internet of Things (FMSys), Hong Kong, China, 13–15 May 2024; pp. 44–49. [CrossRef]
12. Wu, G.; Wu, W.; Liu, X.; Xu, K.; Wan, T.; Wang, W. Cheap-fake Detection with LLM using Prompt Engineering. In Proceedings of the 2023 IEEE International Conference on Multimedia and Expo Workshops (ICMEW), Brisbane, Australia, 10–14 July 2023; pp. 105–109.

13. Hosen, M.H.; Saha, A.; Uddin, A.; Ashraf, K.; Nawar, S. Enhancing Pneumonia Detection: CNN Interpretability with LIME and SHAP. In Proceedings of the 2024 6th International Conference on Electrical Engineering and Information & Communication Technology (ICEEICT), Dhaka, Bangladesh, 2–4 May 2024; pp. 794–799. [CrossRef]
14. Aldughayfiq, B.; Ashfaq, F.; Jhanjhi, N.; Humayun, M. Explainable AI for retinoblastoma diagnosis: Interpreting deep learning models with LIME and SHAP. *Diagnostics* **2023**, *13*, 1932. [CrossRef] [PubMed]
15. Bhandari, M.; Yogarajah, P.; Kavitha, M.S.; Condell, J. Exploring the capabilities of a lightweight CNN model in accurately identifying renal abnormalities: Cysts, stones, and tumors, using LIME and SHAP. *Appl. Sci.* **2023**, *13*, 3125. [CrossRef]
16. Temenos, A.; Temenos, N.; Kaselimi, M.; Doulamis, A.; Doulamis, N. Interpretable Deep Learning Framework for Land Use and Land Cover Classification in Remote Sensing Using SHAP. *IEEE Geosci. Remote Sens. Lett.* **2023**, *20*, 8500105. [CrossRef]
17. Sun, W.; Ma, Y.; Zhang, H.; Wang, R. ConTrans-Detect: A Multi-Scale Convolution-Transformer Network for DeepFake Video Detection. In Proceedings of the 2023 29th International Conference on Mechatronics and Machine Vision in Practice (M2VIP), Queenstown, New Zealand, 21–24 November 2023; pp. 1–6. [CrossRef]
18. Cunha, Y.M.B.G.; Gomes, B.R.; Boaro, J.M.C.; Moraes, D.d.S.; Busson, A.J.G.; Duarte, J.C.; Colcher, S. Learning Self-distilled Features for Facial Deepfake Detection Using Visual Foundation Models: General Results and Demographic Analysis. *J. Interact. Syst.* **2024**, *15*, 682–694. [CrossRef]
19. Jia, S.; Lyu, R.; Zhao, K.; Chen, Y.; Yan, Z.; Ju, Y.; Hu, C.; Li, X.; Wu, B.; Lyu, S. Can chatgpt detect deepfakes? a study of using multimodal large language models for media forensics. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 17–18 June 2024; pp. 4324–4333.
20. Giulivi, L.; Boracchi, G. Explaining Multi-modal Large Language Models by Analyzing their Vision Perception. *arXiv* **2024**, arXiv:2405.14612.
21. Asgari, S.; Khani, A.; Khasahmadi, A.H.; Sanghi, A.; Willis, K.D.; Amiri, A.M. texplain: Post-hoc Textual Explanation of Image Classifiers with Pre-trained Language Models. In Proceedings of the ICLR 2024 Workshop on Reliable and Responsible Foundation Models, Vienna, Austria, 10 May 2024.
22. Alaa, M. MidJourney ImageNet: Real vs. Synth. 2024. Available online: https://www.kaggle.com/datasets/mariammarioma/midjourney-imagenet-real-vs-synth (accessed on 18 November 2024).
23. Liu, H.; Li, C.; Li, Y.; Lee, Y.J. Improved baselines with visual instruction tuning. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 16–22 June 2024; pp. 26296–26306.
24. Rajesh, K.; Raman, M.; Karim, M.A.; Chawla, P. Bridging the Gap: Exploring the Capabilities of Bridge-Architectures for Complex Visual Reasoning Tasks. *arXiv* **2023**, arXiv:2307.16395.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.