Exploring Vision-Language Models for Real-Fake Image Detection with Limited Data

Jared Chan Daniel Yu Sheikh Asif Imran Shouborno
Worcester Polytechnic Institute

Abstract

The detection of fake and real images using deep learning has shown great success on GANs but the ability to perform zero-shot prediction on unseen generative methods is still a challenge. Furthermore, the majority of real-fake image detection methods require mass amounts of training images, ranging in the hundreds of thousands to millions. Our work examines not only how to improve zero-shot prediction capabilities on unseen generators but also doing so with only a few thousand training samples. Specifically, we explore fine-tuning Large Vision-Language Models (VLMs) on limited training data from a subset of the Sentry dataset. Our methods involve experimenting with both traditional VLMs, such as CLIP, in an Active Learning setting and newer VLMs, including LLaVA, which are chatbots that use instructional-tuning to perform few-shot and zero-shot prediction. Additionally, we introduce new methods for improving Active Learning with CLIP, such as loss-weighted uncertainty estimation and weighted backpropagation using uncertainty. We also try various methods for enhancing LLaVA's capabilities for the real-fake detection task by exploiting its ability to explain images. Overall, we find our best method trained on 7.5K images, BW-LMS-T, is able to achieve similar performance compared to CLIP ResNet-50 trained over 481K images. https://github.com/CarrotPeeler/CS545-Real-Fake-Image-Detection.

1. Introduction

In today's digital landscape, distinguishing real images from synthetic ones is increasingly challenging, given the rise of manipulated media. Vision-Language Models (VLMs) offer a promising solution by combining visual and textual understanding. With their ability to comprehend visual and textual data together, VLMs have the potential to greatly improve synthetic image detection, making it easier to combat the spread of misinformation online. However, as the paradigm of deep learning continues to grow, so too does the number of various methods and models for generating fake media. Furthermore, training on images from nearly every possible generative method is infeasible. Thus, our solution aims to discern fake from real images regardless of the generative method and to do so with a small training set.

1.1. Contributions

The major contributions of this paper are as follows:

1. We created a subset of 481K images from the original 3M images included in the Sentry dataset while preserving a large number of StyleGAN3 [13] images and a lower number of diffusion [24] and IF [2] images, which achieves good results with CLIP ResNet-50 on unseen test sets of Sentry, including fake images generated with GAN, diffusion, CogView [5], Midjourney, etc.
2. We experimented with active learning to further reduce the number of training samples needed to achieve a comparable result. With BW-LMS-T, we only needed 7.5k images to train a model that performs similar or better than our baseline trained on 481k images, which is less than 1.6% of our baseline subset or 0.25% of the original Sentry dataset.
3. We explored the capabilities of large language and vision assistants (LLaVA) and trained a LLaVA model with a small amount of data to explain why an image is real or fake in a natural language paragraph. We applied text classification to the generated explanations to assess its strengths and limitations.

2. Literature Review

2.1. Real-Fake Image Detection

The Synthetic Image Detection Challenge ran by the IEEE Video and Image Processing Cup [20] aims to develop robust and automatic tools capable of distinguishing synthetic images from real ones. The challenge is motivated by the recent advancements in AI-based synthetic media generation, which have raised concerns about the trustworthiness of media content and the spread of misinformation online. The challenge has two main objectives:

1. **Robustness**: The detector needs to be able to handle image changes like resizing and compression without losing its ability to spot fake images. These changes can hide important clues and make it harder for the detector to work well.
2. **Generalization**: The detector should work universally well across different sources of synthetic images, including not only Generative Adversarial Networks (GANs) but also more recent diffusion-based models, and be able to generalize to new, unseen architectures.

2.2. Generative Methods

In our dataset, we encounter various generative methods employed to fabricate synthetic images, each with its unique approach and characteristics. Generative Adversarial Networks (GANs) stand as a prominent technique utilized within the dataset. These networks consist of two neural networks, a generator and a discriminator, trained concurrently. The generator crafts fake images, while the discriminator discerns between real and synthetic ones. Through iterative training, GANs yield highly realistic synthetic images, reflecting a key aspect of the dataset.

Diffusion-based models emerge as another significant method employed for synthetic image generation. These models adopt iterative diffusion processes, systematically applying noise to an input image to gradually diffuse it and generate the desired output. Notably, they excel in generating realistic images imbued with intricate details and diverse appearances, constituting an integral component of the dataset.

Additionally, Auto Regressive (AR) Models contribute to the dataset's diversity in generative methods. Operating by modeling the conditional probability distribution of each pixel given the preceding pixels in an image, these models generate data pixel by pixel. While they may not attain the same level of realism as GANs, AR Models offer advantages in Interpretability and control over the generation process, adding another layer of complexity to the dataset.

2.3. Vision-Language Models (VLMs)

Vision-language models (VLMs) have been attracting significant attention due to their superior performance in few-shot and zero-shot predictions across various visual downstream tasks [29]. These models combine visual and textual data, enabling them to understand and process information similar to humans. Among these models, we will be focusing on Contrastive Language-Image Pre-training (CLIP) [23] and Large Language and Vision Assistant (LLaVA) [18].

CLIP is an image-text alignment model designed to understand and categorize images by learning visual concepts through natural language descriptions. It excels at connecting images with relevant text, which is useful for image classification and object detection based on textual prompts. However, CLIP is limited to image-text matching and lacks the ability to generate text or comprehend the context of images.

On the other hand, LLaVA extends the functionalities of typical VLMs by adding the ability to engage in conversations, respond to prompts, and interpret feedback. This makes LLaVA not only a tool for image-text alignment, but also a chatbot that can generate textual responses and interact with users. As presented in [17], [16], and [19], LLaVA aims to reproduce GPT-4's vision interpretation capabilities in an open-source manner, which is helpful for researchers.

2.4. Active Learning

Active Learning is a framework for machine learning which lets the models choose the type of data they would like to learn. While accessibility to data is not an issue due to the internet, annotating large amounts of data is costly and infeasible. Thus, active learning proposes to let the model select the quality of the data and the minimum number of samples that are required for annotation and training. In many cases, active learning improves model performance compared to training models on alternative or human-selected data. The typical training cycle for active learning involves a learning model, which selects and generalizes over the data, a method for querying new data, and a pool, which contains all possible data to query. Each active learning iteration, the model queries new data from the pool and appends it to the current training dataset. The learning model then trains over the current version of the training dataset and the process repeats, starting again at the query step [8].

3. Methodology

To address the problem of real-fake image detection using limited data, we explore solutions involving Vision-Language Models (VLMs). Specifically, we try two different approaches: the first trains VLMs in an active learning framework using CLIP and the second utilizes VLM chatbots such as LLaVA. We choose CLIP and LLaVA, as they significantly differ in their capabilities; while CLIP can grasp general associations between textual descriptions and images, it cannot provide in-depth understanding and generative responses for explaining specifics about an image like LLaVA. Prior work [18, 22] shows that instruction-tuning can significantly improve the few-shot and zero-shot performance of LLMs and even VLMs. Therefore, we opt to use LLaVA without active learning as opposed to CLIP.

3.1. Dataset

The Sentry-Image dataset [28] is a resource developed for training models to detect fake or AI generated images. It consists of a variety of generative models including Diffusion, AR, GAN and unknown models. The validation split contains fake images from at least 14 different models, some of which are completely unseen during training, and aims to be a diverse representation of newer generative methods. Since the Sentry dataset contains around 3 million images, we created a subset with 481k training images, which includes 240K real and fake images. For real train data, we use 155K images from CC3M [26], 15K from AFHQv2 [3], and 70K from FFHQ [12]. For fake images, we use 87K StyleGAN3 [14] fake images generated from FFHQ [12], AFHQv2 [3], and MetFaces [14]. To create the validation subset, we pool 187k images from the Sentry validation set, with around 10K images from each dataset. Unseen generators include Cogview2 [4] and Midjourney-V5 [1]. For our experiments, we only train models using this subset.

3.2. Deep Bayesian Active Learning

For active learning, we use the Deep Bayesian Active Learning framework [7], which uses Bayesian uncertainty estimation to query potential samples for training. We use CLIP ResNet-50 modified from recent work [21] as the learning model. In accordance with [21], we keep the CLIP ResNet-50 backbone frozen and only add a single linear layer to the end of the model, which will be trained for binary classification. The aforementioned work demonstrates that the feature space of a large pretrained vision-language model not trained for real-fake detection is unbiased towards any one binary class.

We randomly sample a balanced number of images from among all possible 481K training images as the initial training data. Following training over the initial data, the model enters the active learning cycle, where it queries new data, adds it to the current training data, and trains over both the new and old samples. This process repeats until the model converges over the validation data or the entire pool of 481K images is exhausted. As Bayesian uncertainty estimation is costly for querying over all 481K samples, we use an approximation in the form of Monte Carlo Dropout sampling. To sample, the learning model predicts over all 481K image samples several times, and each time, the model performs dropout. This repetitive process yields several sets of prediction probabilities over the data, thus approximating the uncertainty distribution over the data. A subset of the 481K samples with highest uncertainty are then queried and added to the current training datapool.

To statistically measure the uncertainty of the prediction probabilities obtained, methods such as Max Entropy [25], BALD [9], Variation Ratios [6], and Mean Standard Deviation [11, 15] are used. We also develop our own versions of these methods which aim to improve the quality of queried samples. Our versions include modifications to the original methods, such as factoring loss into uncertainty estimation, weighing backpropagation via uncertainty, and balancing class distribution for querying.

Since uncertainty is based on prediction probabilities, the correctness of these probabilities is ignored. For example, the model may be very certain about an image belonging to one class over the other; in this scenario, uncertainty estimation methods will ignore this sample for selection. However, if the model is actually incorrect about this prediction and is simply overconfident, this sample should be selected for further training. Thus, incorporating loss into uncertainty can improve the quality and types of samples selected for learning. For any of the uncertainty methods mentioned before, we can combine loss and uncertainty via the following formulation. First, we compute the normalized uncertainty scores for each queried image, $\omega \in \mathbb{R}^n$, where n is the number of queried images.

$$\omega = \frac{\epsilon - \min(\min(\epsilon), 0.1)}{\max(\epsilon) - \min(\min(\epsilon), 0.1)} + 1 \tag{1}$$

This formulation uses Min-Max Normalization to change the uncertainty scores to range between 1 and 2 instead of 0 to 1. This prevents images with low uncertainty scores of 0 from being multiplied by large loss values of 100. Therefore, incorrect but high certainty images are no longer ignored. Furthermore, if the lowest uncertainty score is still relatively high and not close to 0, those scores are still preserved as long as they are above 0.1. The following formulation is for $\alpha \in \mathbb{R}^n$, which is the vector of final scores computed by multiplying the loss values, $\lambda \in \mathbb{R}^n$, for each queried image by their normalized uncertainty scores, ω. Note that the loss values are taken at prediction time and are never changed.

$$\alpha = \lambda \cdot \omega \tag{2}$$

Our second modification involves using uncertainty scores as weights during back propagation such that the learning model is penalized more so for incorrectly predicting newer and more uncertain images during training. The formulation for the weights is as follows:

$$\beta = \lambda \cdot \omega^3 \tag{3}$$

β is a vector of the new weighted loss values for each queried sample. Here ω is cubed to scale the normalized uncertainty scores to values which are appropriate for the weights. Note, the loss values, λ, are constantly updated after each epoch of the learning model's training and are not fixed. Additionally, to ensure newer queried samples are given more importance than samples already trained on, we exponentially decay the weights at a fixed rate, which is applied after each epoch.

3.3. Natural Language Explanation Generation with Large Language and Vision Assistant

LLaVA, as introduced in [17], [16], and [19], utilizes visual instructions with large language assistants to reproduce the multimodal capabilities of GPT-4 in an open-source project. LLaVA is capable of analyzing objects, features, and anomalies of images out of the box. However, upon experimentation, we noticed that an untrained LLaVA-1.6 model containing 7 billion parameters is unable to identify fake images out of the box. It considers most of the images to be real. However, it's possible to explain why an image is fake in a natural language due to its distinguishable features. Therefore, we decided to fine-tune LLaVA-1.6-7b with a synthetically generated dataset containing explanations of why a given image is real or fake. Providing expert knowledge such as the capabilities of existing GAN, diffusion, and autoregressive models and telling the default LLaVA what was used to generate a fake image helps it prepare explanations thanks to our guided prompts. We then train an unguided explainer with the explanations so that the model can answer why an image is real or fake for an unseen image. Besides, to see the merit of explanations, which is akin to chain-of-thought prompting [27] followed by fine-tuning, we also attempt direct classification with LLaVA, where the chatbot directly answers whether the image is real or fake. We noticed that explanation with text classification outperforms direct classification with LLaVA, which is consistent with the reasoning capabilities of large language models that aren't trained with a large amount of data for a complex dounstream task. It's notable that we limited the subset to only 5500 images for this part, which is less than 1.2% of our cherry-picked baseline CLIP ResNet-50 training subset. The explanations used to trained the model and the explanations generated on unseen test images can be found at the WPI SharePoint link included in our Github repository.

Training a large model requires a substantial amount of time and resources. We utilized 48GB of GPU memory to fine-tune our LLaVA model with a parameter-efficient fine-tuning method called LoRA [10]. LoRA delegates fine-tuning to low-rank adapters obtained by low-rank decomposition to approximate the parameters that need to be changed in the original model. It has seen substantial use in fine-tuning large language models as it significantly reduces resource and time requirements. Despite the benefits provided by LoRA, we still needed 48GB GPU to run our fine-tuning task, which is understandable given the scale of the model.

4. Experiments and Results

4.1. Active Learning

To study the effect of our uncertainty estimation methods against prior work, we compare them by accuracy against the Sentry subset test data. All active learning trials are completed with the same hyperparameters to ensure results are fair for comparison. All results below are conducted with the same training hyperparameters and use image augmentation techniques such as Gaussian blur, JPEG compression, random horizontal flip, and random crop according to prior work [21].

Table 1. Active Learning Hyperparameters

Hyperparameter	**Value**
Momentum Beta	0.9
Image input size (pixels)	224
Optimizer	AdamW
Learning Rate	1×10^{-4}
Size of Initial Training Data (images)	10K
Query Size (images)	2.5K
Active Learning Iterations	20
Training Epochs for Initial Data	30
Training Epochs for Queried Data	30
Monte Carlo Dropout Iterations	10
Weighted Loss Decay Rate (BW-LMS)	0.99

Table 2. BW-LMS-T Hyperparameters

Hyperparameter	**Value**
Momentum Beta	0.9
Image input size (pixels)	224
Optimizer	AdamW
Learning Rate	1×10^{-4}
Size of Initial Training Data (images)	2K
Query Size (images)	500
Active Learning Iterations	20
Training Epochs for Initial Data	60
Training Epochs for Queried Data	5
Monte Carlo Dropout Iterations	10
Weighted Loss Decay Rate (BW-LMS)	0.75

All methods involve training CLIP ResNet-50 with active learning using the hyperparameters in Table 1. After all active learning iterations, the train dataset finalizes with 60K train samples. Table 3 shows the accuracy scores for each method. The cc3m and celeba-hq datasets contain real images while the other datasets are fake.

In Table 3, we find that by using Mean Standard Deviation with loss, weighted backpropagation via uncertainty, and class balanced querying together (BW-LMS), we obtain the best results across all 14 test sets. Compared to uncertainty estimation methods from past work, such as Mean Standard Deviation, Variation Ratios, and BALD, our BW-LMS method outperforms them in 7 out of the 14 test datasets. Additionally, BW-LMS has the best performance across all StyleGAN3 test sets, except for MetFaces, where

Table 3. Active Learning Method Comparison

Dataset	Method				
	MS	BW-MS	BW-LMS	VR	BD
cc3m	95.05	93.90	94.04	94.81	**95.15**
celeba-hq	68.16	68.47	68.21	**71.77**	66.24
cogview2-22k	**83.74**	83.23	82.60	80.10	83.07
if-ddim-50-15k	95.79	**96.18**	96.09	95.45	95.44
if-ddpm-50-15k	96.17	95.99	**96.27**	95.85	96.01
if-dpmsolver++-10-15K	95.77	**96.17**	96.10	94.97	95.77
if-dpmsolver++-25-15K	96.19	96.19	**96.76**	95.81	95.87
midjourneyv5-5k	78.75	**83.10**	79.56	76.34	80.12
sdv15r-dpmsolver-25-15k	99.03	99.17	**99.48**	98.67	99.01
sdv2-dpmsolver-25-10K	97.56	98.15	**99.03**	97.83	97.27
stylegan3-r-afhqv2-512x512	74.07	71.09	**79.25**	75.15	71.98
stylegan3-t-afhqv2-512x512	71.88	69.76	**77.48**	73.74	69.31
stylegan3-t-ffhqu-1024x1024	60.73	62.09	**67.23**	64.87	59.82
stylegan3-t-metfaces-1024x1024	81.17	80.88	82.73	72.37	**86.38**

MS = Mean Standard Deviation
LMS = Mean Standard Deviation with Loss
VR = Variation Ratios
BD = BALD
BW = Uses class balanced querying and weighted back propagation

Table 4. Final Method Comparison

Dataset	Method			
	Baseline	BW-LMS-T	LLaVA-E	LLaVA-D
cc3m	**95.38**	94.66	42.8	90.0
celeba-hq	9.18	71.68	87.6	**100.0**
cogview2-22k	81.14	84.18	**90.4**	40.0
if-ddim-50-15k	95.95	**96.31**	82.0	26.0
if-ddpm-50-15k	**97.59**	97.02	74.4	18.0
if-dpmsolver++-10-15K	**96.54**	96.18	77.6	34.0
if-dpmsolver++-25-15K	**97.94**	97.68	88.4	28.0
midjourneyv5-5k	**93.17**	72.69	88.4	72.0
sdv15r-dpmsolver-25-15k	99.39	**99.67**	54.4	16.0
sdv2-dpmsolver-25-10K	99.26	**99.49**	75.2	12.0
stylegan3-r-afhqv2-512x512	97.27	**99.37**	2.4	0.0
stylegan3-t-afhqv2-512x512	96.88	**99.37**	1.2	0.0
stylegan3-t-ffhqu-1024x1024	94.97	89.34	27.2	4.0
stylegan3-t-metfaces-1024x1024	96.88	87.11	**99.6**	99.0
Train Size (Images)	481K	7.5K	5.5K	3K

Baseline = CLIP ResNet-50 without active learning
BW-LMS-T = Tuned Version of BW-LMS
LLaVA-E = LLaVA with Explainer + Text Classifier
LLaVA-D = LLaVA with Direct Classifier

it places second. Note that these results are based on hyperparameters used for comparison, and they do not reflect the best performance obtainable.

4.2. Comparing VLM Methods

Using our best performing method, BW-LMS, we tune its active learning hyperparameters to further improve its results on the Sentry test set. The tuned hyperparameters are shown in Table 2. Most notably, we decrease the query size to 500 such that the model only learns the top 250 samples with highest uncertainty. This reduces the amount of samples with low uncertainty that are mixed into the group of queried images. We also decrease the training epochs per query to prevent overfitting and increase the initial training epochs to 60.

We also perform a final comparison in Table 4 of our best active learning and LLaVA method against the baseline results. The baseline method trains CLIP ResNet-50 on all 481K train samples without active learning for 1 epoch, whereas our best active learning method, BW-LMS-T, is only trained on 7.5K images and LLaVA-E is trained on 5.5K. The baseline method nets very high accuracy scores across all datasets. However, the baseline performs terribly on celeba-hq, with 9.18% accuracy. BW-LMS-T slightly outperforms the baseline on several datasets but exceptionally beats the baseline on celeba-hq by 62.5%. LLaVA-

E, which involves an explainer and text classifier for binary classification, performs the best on cogview2-22k and stylegan3-t-metfaces-1024x1024 with the highest accuracy out of all methods for those test sets. However, LLaVA-E performs poorly on all other stylegan3 test sets in addition to cc3m and sdv15r-dpmsolver-25-15k. Additionally, our other LLaVA method, LLaVA-D, obtains perfect accuracy over the celeba-hq test set.

5. Conclusion and Future Work

VLMs such as CLIP and LLaVA are promising methods in the task of real versus fake image detection. The use of more complex models like CLIP:ViT-L/14 may further enhance these capabilities. Furthermore, employing active learning strategies can boost model generalization across unseen synthetic image generators given only a few thousand training samples, thereby improving the robustness and effectiveness of these systems. Once fully trained, LLaVA not only differentiates between real and fake images but also explains the reasoning in natural language, adding a layer of transparency and user interaction. In the future, we plan to expand the training data set as a promising approach to refine these models into expert real-fake explainers. In addition, GradCAM could be used to verify whether our CLIP ResNet models look at the same features mentioned by the LLaVA explainer. We hope that with these advancements, we can help tackle misinformation and raise the integrity of digital media.

References

[1] Midjourney. https://www.midjourney.com. Accessed: 2024-05-01. 3

[2] DeepFloyd Lab at StabilityAI. DeepFloyd IF: a novel state-of-the-art open-source text-to-image model with a high degree of photorealism and language understanding. https://www.deepfloyd.ai/deepfloyd-if, 2023. Retrieved on 2023-11-08. 1

[3] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 3

[4] Ming Ding, Zhuoyi Yang, Wenyi Hong, Wendi Zheng, Chang Zhou, Da Yin, Junyang Lin, Xu Zou, Zhou Shao, Hongxia Yang, and Jie Tang. Cogview: Mastering text-to-image generation via transformers. In *Advances in Neural Information Processing Systems*, pages 19822–19835. Curran Associates, Inc., 2021. 3

[5] Ming Ding, Zhuoyi Yang, Wenyi Hong, Wendi Zheng, Chang Zhou, Da Yin, Junyang Lin, Xu Zou, Zhou Shao, Hongxia Yang, and Jie Tang. Cogview: Mastering text-to-image generation via transformers. *arXiv preprint arXiv:2105.13290*, 2021. 1

[6] L.C. Freeman. *Elementary Applied Statistics: For Students in Behavioral Science*. Wiley, 1965. 3

[7] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep bayesian active learning with image data, 2017. 3

[8] Jacob Gildenblat. Overview of active learning for deep learning. https://jacobgil.github.io/deeplearning/activelearning#active-learning--higher-accuracy-with-less-data-annotation, 2020. Accessed: 2024-05-01. 2

[9] Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. Bayesian active learning for classification and preference learning, 2011. 3

[10] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 4

[11] Michael Kampffmeyer, Arnt-Børre Salberg, and Robert Jenssen. Semantic segmentation of small objects and modeling of uncertainty in urban remote sensing images using deep convolutional neural networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 680–688, 2016. 3

[12] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4396–4405, 2019. 3

[13] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. In *Proc. NeurIPS*, 2021. 1

[14] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. In *Advances in Neural Information Processing Systems*, pages 852–863. Curran Associates, Inc., 2021. 3

[15] Alex Kendall, Vijay Badrinarayanan, and Roberto Cipolla. Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding, 2016. 3

[16] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning, 2023. 2, 4

[17] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023. 2, 4

[18] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023. 2

[19] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, 2024. 2, 4

[20] University Federico II of Naples and NVIDIA. Synthetic image detection challenge. 1

[21] Utkarsh Ojha, Yuheng Li, and Yong Jae Lee. Towards universal fake image detectors that generalize across generative models, 2024. 3, 4

[22] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022. 2

[23] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. 2

[24] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, 2022. 1

[25] Claude Elwood Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423, 1948. 3

[26] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of ACL*, 2018. 3

[27] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022. 4

[28] Lei Bai Jingjing Qu Chengyue Wu Xihui Liu Wanli Ouyang Zeyu Lu, Di Huang. Seeing is not always believing: Benchmarking human and model perception of ai-generated images. In *Advances in Neural Information Processing Systems*, 2023. 2

[29] Jingyi Zhang, Jiaxing Huang, Sheng Jin, and Shijian Lu. Vision-language models for vision tasks: A survey, 2024. 2