BitMind Deepfake Detection Arena

A Comprehensive and Open Benchmark for Detecting AI-Generated Images

Overview

The landscape of open-source computer vision currently grapples with a critical shortage of datasets and evaluation frameworks designed to benchmark systems that distinguish between real and AI-generated images. Previous studies have predominantly targeted content-specific subsets of this problem, such as human faces in images and videos (e.g., DeepfakeBench).

Although these efforts have proven valuable for testing new model architectures under limited conditions, they do not adequately address the broad spectrum of image types encountered in everyday scenarios. This report covers our efforts to fill this gap by developing **Deepfake Detection Arena (DFD-Arena)**, a comprehensive and adaptable benchmark suitable for the diverse and complex nature of in-the-wild images.

Definitions

In this report, we use the term "**synthetic images**" to describe images that are produced by generative AI models, i.e. generative adversarial networks (GAN), vision transformers (VIT), diffusion models, and vision-language models (VLM).

Additionally, we define a synthetic image generated to match the semantic-level characteristics of a "**real image**", or non AI-generated image, as the real image's "**synthetic mirror.**"

Real Image

A real image from FFHQ, a dataset of human faces created by NVIDIA Research as a benchmark intended for GANs.

Using BLIP-2, OPT-6.7b, fine-tuned on COCO VLM + Meta-Llama-3.1-8B-Instruct-bnb-4bit (Unsloth) LLM, we generate the caption:

"A baby lies on a blue blanket in a sunny

Synthetic Mirror

Synthetic image output from our Synthetic Image Generation Pipeline, generated by FLUX.1-dev. using the generation arguments:

```
"guidance_scale": 2,
"num_inference_steps": 100,
"height": 512,
```

setting, surrounded by a blue background, in a portrait view."

"width": 512

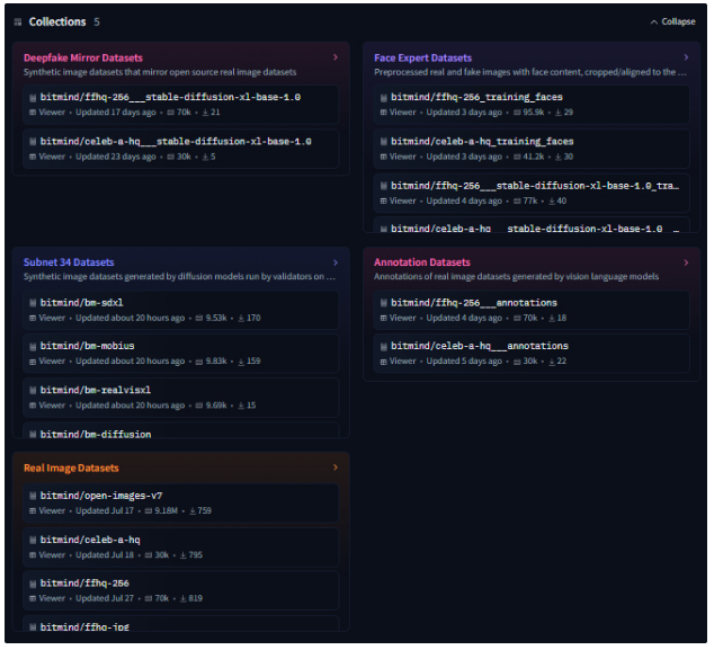
In the subsequent sections, the term "**model**" refers to any algorithm that processes an image to determine its classification, specifically to identify if it is real or synthetic. This includes not only pretrained machine learning architectures, but also heuristic and statistical modeling frameworks.

Furthermore, we define a "**detector**" as an algorithm that either employs a single model or orchestrates multiple models to perform the binary inference. The distinction between "model" and "detector" helps clarify the roles within multi-agent systems and mixture-of-expert frameworks, particularly those utilizing deep learning classifiers.

Data

The BitMind team has curated and generated a large collection of real and synthetic image datasets on HuggingFace with significant topic diversity. These are sourced from a foundation of datasets publicized by leading machine vision research teams–including NVIDIA, Microsoft, and Google–reputable in image segmentation, recognition, and classification spaces.

We have further extended these collections with synthetic mirror datasets generated in-house. These novel datasets are characterized by the semantic balance they share with real counterparts.



BitMind's real and deepfake image datasets are continuing to grow, with new raw and preprocessed general and expert training datasets being continuously added for image diversity.

Open-Source Image Datasets

ImageNet	A growing dataset that currently contains over 14 million annotated images of real world scenes.
MS-COCO	328,000 images with a wide variety of real world scenes.

CelebA-HQ	30,000 images of celebrity faces with varying poses and backgrounds. CelebA-HQ was introduced by NVIDIA at ICLR.
FFHQ	70,000 high quality 1024x1024 png images of human faces, with varying age, ethnicity, and image backgrounds. FFHQ was made by NVIDIA Research for benchmarking Generative Adversarial Networks (GANs).
Flickr-30k	30,000 images from Flickr, with a variety of real world scenes.
DiffusionDB	14 million synthetic images generated by stable diffusion using prompts and hyperparameters specified by real users.

BitMind Synthetic Mirror Image Datasets

The datasets discussed in this section were created in-house using our open-source Synthetic Image Generation Pipeline which incorporates SoTA models to produce synthetic mirrors while maximizing information density:

1. VLM for image captioning: BLIP-2, OPT-6.7b, fine-tuned on COCO
2. LLM for caption moderation: Meta-Llama-3.1-8B-Instruct-bnb-4bit (Unsloth)
3. Advanced diffusion models including the recent FLUX.1-dev by Black Forest Labs.

Our mirror datasets follow the nomenclature "<real dataset name>__<diffusion model used>". For instance, datasets ending in "Flux" were produced using Flux-1.0-dev, and those with the "SDXL" were created with stable-diffusion-xl-1.0-base.

Real Image Dataset	Diffusion Model	Synthetic Mirrors
CelebA-HQ	SDXL	CelebA-HQ__SDXL
CelebA-HQ	FLUX	CelebA-HQ__Flux
FFHQ	SDXL	FFHQ__SDXL
FFHQ	FLUX	FFHQ__Flux
MS-COCO	SDXL	MS-COCO__SDXL
MS-COCO	FLUX	MS-COCO__Flux

> To efficiently scale operations, the team has also developed open-source tooling for parallelizing dataset generation across a configurable number of GPUs and machines.

These datasets, along with our evaluation framework outlined in the following section, form the basis of an open, reproducible benchmark aimed at assessing a detection algorithm's usefulness in realistic "in-the-wild" settings.

Methodology

Benchmarking Framework: DFD-Arena

1. **Training code** for all models used in SN34 base detectors
2. A DeepfakeDetector class which all SN34 base detectors inherit from. This abstract base class standardizes how detectors initialize and invoke models, and generalizes to Bittensor and

Bittensor-agnostic inference environments, making it significantly easier to:

   a. Modularly select and configure arbitrary detectors in our newly refactored miner.py using a factory pattern
   b. Use our base miner detectors as templates for new detector subclasses
   c. Evaluate arbitrary detection algorithms with our benchmarking framework

3. A **benchmarking class** (Arena) that evaluates all SN34 detectors, and any additional detectors that inherit from our base `DeepfakeDetector(ABC)` class
4. **Benchmarking datasets**
5. An open **leaderboard hosted on Hugging Face**

Our first iteration of detector benchmarking involves three detector designs using two underlying model architectures, each trained on the same subset of our benchmark datasets. We treat the remaining datasets as our holdout set, and use this to evaluate each trained model.

By choosing to not finetune models on train partition of the evaluation datasets, we simulate a realistic lack of a priori knowledge of their distributions.

Model Architectures

NPR (CVPR 2024) is a modified ResNet architecture with transformation in the forward pass function involving a metric called Neighboring Pixel Relationships (NPR). This simple yet powerful extension of the widely-used ResNet is informed by the authors' observation that the upsampling operations common in GANs and diffusers cause nearby pixels in synthetic images to be more correlated than in naturally occurring image data. Using this metric to transform inputs in the forward pass guides the model towards learning features specific to generated images.

UCF (ICCV 2023) is a multi-task disentanglement framework that uncovers the common features across different forgery types. It incorporates a conditional decoder and contrastive regularization loss to improve the disentanglement process and prevent overfitting to irrelevant textures and method-specific forgeries, resulting in a more generalizable detector. The framework uses a modified Xception backbone, which learns method-specific features.

Detectors

All detectors are implemented as subclasses of DeepfakeDetector(ABC), which standardizes basic initialization, population of instance attributes from a .YAML file (including model weights/configs to download from Hugging Face), and invocation for inference.

The following detectors have been implemented:

- NPRDetector(DeepfakeDetector) which initializes and invokes pretrained NPR model inference
- UCFDetector(DeepfakeDetector) which initializes and invokes pretrained UCF model inference
- CAMODetector(DeepfakeDetector) which implements Content-Aware Model Orchestration (CAMO). It initializes and invokes different pretrained models using a hard mixture-of-experts framework that routes classified image content to the most suitable models for inference.
    - CAMO is BitMind's current base miner detector and enables specialized models to excel without needing to generalize outside their niche. Our most performant version features two UCF backbones—a face expert and a generalist. This iteration of CAMO has significantly outperformed previous detectors.

```
# camo.yaml Configuration File
object_detection: False
content_type:
```

```yaml
  general:
    detector: 'UCF'
    detector_config: 'ucf.yaml'  # Default model for 'general'
  face:
    detector: 'UCF'
    detector_config: 'ucf_face.yaml'  # Default model for 'face'
```

Default YAML configuration file for CAMO instructing it to route image content between a face expert and a generalist UCF detector with different configurations (weights and preprocessing).

```python
@DETECTOR_REGISTRY.register_module(module_name='CAMO')
class CAMODetector(DeepfakeDetector):
    """
    This DeepfakeDetector subclass implements Content-Aware Model Orchestration
    (CAMO). A GatingMechanism, a Gate subclass, routes image content to
    DeepfakeDetectors that initialize models pretrained on the content type.
    ...
    """

    def __init__(self, model_name: str = 'CAMO', config: str = 'camo.yaml', cuda: bool =
        """
        Initialize the CAMODetector with dynamic model selection based on config.
        """
        self.detectors = {}
        super().__init__(model_name, config, cuda)
        self.gate = GATE_REGISTRY["GATING_MECHANISM"](object_detection=self.object_detect

    def load_model(self):
        """
        Load detectors dynamically based on the provided configuration and registry.
        """
        for content_type, detector_info in self.content_type.items():
            model_name = detector_info['model']
            detector_config = detector_info['detector_config']

            if model_name in DETECTOR_REGISTRY:
                self.detectors[content_type] = DETECTOR_REGISTRY[model_name](
                    model_name=f'{model_name}_{content_type.capitalize()}',
                    config=detector_config,
                    cuda=self.device
                )
            else:
                raise ValueError(f"Detector {model_name} not found in the registry for {c

    def __call__(
            self, image
    ) -> float:
        try:
            # Determine image content type.
            content_type, content_data = self.gate(image)
            pred = self.detectors[content_type](content_data)
        except Exception as e:
            print(f"Error performing inference: {e}")
        return pred
```

CAMODetector(DeepfakeDetector) uses a GatingMechanism(Gate) instance to route image inference invocation to a detector based on classified image content.

Training Set

In order to perform a fair evaluation, all detectors benchmarked on DFD-Arena must use the same data to train any underlying models. The training dataset we use for all 3 models reported in our results consists of two real datasets and two synthetic datasets: FFHQ and MSCOCO, and their stable diffusion mirrors (FFHQ__SDXL, MSCOCO__SDXL) .

The datasets were chosen for the following reasons:

1. Using synthetic data whose semantic distribution matches that of the real datasets helps avoid learning a trivial topic model
    a. Consider an extreme example, wherein all of the synthetic training data are images of cars, and all of the real data are images of trees - training on this dataset would risk the model only learning to differentiate between trees and cars rather than real and synthetic.
2. Including a faces-only dataset like FFHQ allows us to test our hypotheses around content aware routing — i.e., what performance benefits can CAMO provide by routing images to specialist models for specific image types?
3. Including a dataset with highly diverse content (MSCOCO) provides a realistic topic distribution to represent what may be encountered in the wild.

Holdout Set

We accordingly used the remaining datasets in our benchmark collection as our holdout. Given that several of these datasets contain millions of images, we randomly sample and aggregate 1,000 images from each dataset in order to reduce the overall runtime of the benchmarking process.

- ImageNet
- DiffusionDB
- CelebA-HQ
- CelebA-HQ__SDXL
- CelebA-HQ__Flux
- Flickr30k
- Flickr30k__SDXL
- MS-COCO__Flux

We choose to not finetune on training partitions of the evaluation datasets in order to make the benchmark more representative of a detector's performance in real-world scenarios wherein finetuning is not an option.

Each model's performance metrics on the holdout datasets provide us with the following insights:

1. Evaluating on additional stable diffusion datasets (DiffusionDB, our SDXL mirrors) shows how well different models generalize to images produced by the same diffusion models used to generate its training data.
2. Evaluating on our Flux mirrors helps us understand how well different models generalize to data produced by diffusion models whose outputs it was not trained on.
3. SDXL synthetic mirrors test models' ability to generalize to new data (with a different semantic distribution) from the same model that produced the synthetic half of its training data.
4. Real image datasets like ImageNet and Flickr30k test models' ability to generalize to real-world images with a wide variety of content and image characteristics.

Hypotheses

1. Although UCF and NPR performance have not been directly compared in computer vision literature prior to our standardized DFD-Arena benchmark, we believe UCF will yield better overall performance metrics than NPR across different synthetic image sources, due to its

overall performance metrics than NPR across different synthetic image sources, due to its unique focus on learning to disentangle forgery methods. A finetuned face expert UCF model also ranks #1 on DeepfakeBench, a limited benchmark that only evaluates models trained on cropped and aligned face images.

2. Since our initial CAMO configuration orchestrates content-based image routing between a UCF expert and a UCF generalist under the hood, CAMO should be at least as performant as the UCF generalist.
3. More broadly, breaking down detection tasks into specialized sub-problems will significantly boost accuracy.
4. Given 1-3, CAMO should outperform all other models in our initial benchmark results.

Results

The green scores below indicate the value is the top performance on the given metric/column.

Average Performance Metrics

Detector	Accuracy	Precision	Recall	F-1	MCC
NPR	0.7169	0.9193	0.5996	0.7258	0.5044
UCF	0.7229	0.9436	0.592	0.7275	0.5285
CAMO	0.7555	0.9442	0.647	0.7679	0.5707

Dataset-Specific Accuracy

Detector	CelebA-HQ	Flickr30k	ImageNet	CelebA-HQ-SDXL	CelebA-HQ-Flux	Flickr30k-SDXL	Diffusion DB	MS-COCO-Flux
NPR	0.987	0.916	0.834	0.876	0.386	0.846	0.302	0.588
UCF	0.995	0.981	0.847	0.85	0.484	0.794	0.256	0.576
CAMO	0.999	0.979	0.831	0.961	0.682	0.722	0.28	0.59

Discussion

Our analysis has provided valuable insights into the performance of different models in deepfake detection:

1. CAMO Model Effectiveness
    a. Performance Metrics - CAMO not only achieved the highest accuracy scores for CelebA-HQ, Diffusion DB, and CelebA-HQ-SDXL, but also performed the best overall.
    b. Key Takeaway - The routing strategy using a combination of a generalist and a face-specific expert, orchestrated by a dlib face detector, proves to be highly effective. This approach validates our hypothesis that breaking down detection tasks into specialized sub-problems can significantly enhance accuracy.
2. UCF Generalization Limitations
    a. Performance Comparison - UCF exhibited poor performance when generalizing to Flux-generated images compared to NPR.
    b. Future Directions - The limited diversity in its training data (only two image sources) may be restricting UCF's performance. Integrating more varied image sources could potentially allow it to better demonstrate its feature disentanglement capability and overall recall.

3. NPR Generalization Capabilities
    a. General Performance - When compared to UCF, NPR showed superior generalization across CelebA-HQ-Flux and Flickr30k-SDXL. As discussed in (2a), this relationship may change if both models are trained on synthetic data from additional sources.
    b. Implications for Integration - The robust performance suggests that NPR's focus on artifacts common to various generative models may present a a further opportunity to explore the integration of NPR's methodologies into broader detection frameworks like CAMO, potentially as an expert model or through leveraging its transformation techniques within other architectures.

Conclusion

Our work lays the foundation for the first open-source benchmark for synthetic image detection. We are committed to continually expanding upon this initial version by incorporating additional open-source data and systematically generating our own.

The early CAMO implementation's relative simplicity and extensibility, coupled with its impressive performance on our benchmark, are encouraging. Building off of this work, we will continue contributing additional detection model integrations, drawing from both open-source work and our own research.

In the context of our Bittensor subnet, the Deepfake Detection Arena repository will soon replace our `base_miner` directory, serving as the central hub for training and evaluating detection systems. To add an extra layer of open competition and transparency, we will launch a Hugging Face leaderboard to accompany the Arena.

We are excited to continue developing the most comprehensive benchmark for detecting AI-generated images, with the ultimate goal of creating the most effective synthetic image detection framework.