**Challenging Fake Image Detection using GAN Models**

**MRI-GAN: A Generalized Approach to Detect Deep Fakes using Perceptual Image Assessment**

This README provides an overview of the scope of the MRI-GAN project, sample results, and steps required to replicate the work, both from scratch and using pre-trained models. Reproducing the results from scratch involves training all the models. Data processing steps are also outlined below.

The full research paper is available at: <https://arxiv.org/abs/2203.00108>

## TLDR.

## Abstract

DeepFakes are synthetic videos created by replacing a face in an original image with someone else's face. This project focuses on developing deep learning models for classifying DeepFake content. We introduce a novel framework called MRI-GAN, which leverages Generative Adversarial Network (GAN)-based models to detect synthesized videos based on perceptual differences in images. We evaluate our MRI-GAN approach and a plain-frames-based model using the DeepFake Detection Challenge Dataset. Our plain frames-based model achieves 91% test accuracy, while the MRI-GAN framework with Structural Similarity Index Measurement (SSIM) for perceptual differences achieves 74% test accuracy. The MRI-GAN results are preliminary and can potentially be improved further by adjusting loss functions, hyperparameters, or using advanced perceptual similarity metrics.

**MRI-GAN**

**MRI-GAN** generates MRIs of input images. The MRI for a DeepFake image contains artifacts that highlight regions of synthesized pixels, while the MRI of a non-DeepFake image is simply a black image.

<https://d.docs.live.net/f38bd683e309f3ac/Documents/blank.png>

**Steps to Replicate the Work**

**Note:** This is a complex process.

1.**Set Up Development Environment**

* Use conda for Python distribution and related libraries on Ubuntu 20.04 OS.
* Create a new environment using the provided environment.yml file:

bash code:

conda env create -f environment.yml

* Activate the environment.

**2.Download Datasets and Extract**

Download the following datasets:

* [DFDC dataset](https://ai.facebook.com/datasets/dfdc/)
* [Celeb-DF-v2 dataset](https://github.com/yuezunli/celeb-deepfakeforensics)
* [FFHQ dataset](https://github.com/NVlabs/ffhq-dataset)
* [FDF dataset](https://github.com/hukkelas/FDF)

**3.Configure Paths and Parameters**

* Update paths and parameters in the **config.yml** file according to your dataset locations and preferences.

**4.Data Pre-processing**

Execute the following commands in sequence:

* **python data\_preprocess.py --gen\_aug\_plan**
* **python data\_preprocess.py --apply\_aug\_to\_all**
* **python data\_preprocess.py --extract\_landmarks**
* **python data\_preprocess.py --crop\_faces**
* **python data\_preprocess.py --gen\_mri\_dataset**

**5.MRI-GAN Training**

* Configure the **config.yml** file and adjust parameters under **['MRI\_GAN']['model\_params']** as needed.
* Train the MRI-GAN model:

css code:

python train\_MRI\_GAN.py --train\_from\_scratch

* Copy trained MRI-GAN weights:

bash code

cp logs/<date\_time\_stamp>/MRI\_GAN/checkpoint\_best\_G.chkpt assets/weights/MRI\_GAN\_weights.chkpt

* Use the trained MRI-GAN to predict MRIs for the DFDC dataset:

python data\_preprocess.py --gen\_dfdc\_mri

**6.Train and Test the DeepFake Detection Model**

* Generate metadata CSV files using:

css code:

python data\_preprocess..py --gen\_deepfake\_metadata

* For the plain-frames method, configure the **config.yml** file with the following parameters:
* **'train\_transform' : 'complex'**
* **'dataset' : 'plain'**
* Train the model from scratch or resume training if needed:

css code:

python deep\_fake\_detectpy --train\_from\_scratch

* Test the saved model:

css code:

python deep\_fake\_detectpy --test\_saved\_model <path>

* For the MRI-based method, configure the **config.yml** file with the following parameters:
* **'train\_transform' : 'simple'**
* **'dataset' : 'mri'**
* Train the model from scratch or resume training if needed, and test the model similarly.

**7.Other Notes**

* Check the **--help** option of all mentioned scripts for more utility methods, such as resuming training of models if it was stopped prematurely.

**Pre-trained Models**

Download the pre-trained model weights to reproduce the results:

* [MRI-GAN Model with Tau = 0.3 and Generator with the Lowest Loss](https://drive.google.com/uc?id=1qEfI96SYOWCumzPdQlcZJZvtAW_OXUcH)
* DeepFake Detection Models:

1. [Plain-Frames Based Model](https://drive.google.com/uc?id=1_Pxv6ptxqXKtDJNkodkDmMTD_KRo08za)
2. [MRI-Based Model](https://drive.google.com/uc?id=1xKzehNuq1B1th-_-U6OG9v2Q2Odws6VG)

**DeepFake Detection App**

Use the model to test a given video file:

1. Download all pre-trained model weights.
2. Run the command-line app:

css code:

python detect\_deepfake\_app.py --input\_videofile <path to video file> --method <detection method>>

* The **detection method** can be either **plain\_frames** or **MRI**.

**Citation**

If you use this work, please cite it as:

makefile code:

@misc{2203.00108,

Author = {Pratikkumar Prajapati and Chris Pollett},

Title = {MRI-GAN: A Generalized Approach to Detect DeepFakes using Perceptual Image Assessment},

Year = {2022},

Eprint = {arXiv:2203.00108},

}