12/8/2023

FACEAUTH

Image Classification of Real and Fake Images

Anjali Mudgal

DEEP LEARNING PROJECT

Table of Contents

Introduction

In today's AI-driven world, telling apart real human faces from AI-generated ones, especially with advanced models like DALL-E or Imagen, has become crucial. Our project dives into this challenge by creating an image classification model specifically designed to distinguish between these two types of faces.

We're not only aiming to spot AI-generated images but also identify the specific AI technology behind them. Despite a deep dive into existing research, we couldn't find a ready-made solution to this problem in academic circles. While commercial tools like sensity.ai claim to detect Deep-Fakes, their hidden methods made us curious to explore a fresh approach in our project.

Data

For Real Faces:

Celeb-HQ-2: High-quality celebrity photos sourced from the internet.

Wiki Celeb: Another dataset consisting mainly of celebrity images. Both datasets were chosen for their cleanliness and to ensure no privacy violations occurred during collection.

Resolution: To manage resource consumption, images were settled at 256x256 pixels and cropped to focus only on faces, minimizing extensive preprocessing.

For Fake Faces:

1million Fake Faces: Generated by Nvidia's StyleGan model, comprising a million synthetic faces.

iFakeFaceDB: Comprising 87,000 images generated by StyleGan but subjected to the GANprintR transformation, aimed at removing the GAN fingerprint from the images (details in the referenced paper).

DeepFakeFace: A collection of fake face images generated by various diffusion-based models, including Stable Diffusion Impainting, Insightface toolbox, and Stable Diffusion V1.5. This dataset offers a range of images from GAN and Diffusion based models for comprehensive coverage.

Dataset Split and Balance

Split Method: Employed standard training-test-dev splits.

Set Allocation:

- Training Set: Used for model training.
- Test Set: Utilized for decisions during model training.
- Dev Set: Kept untouched until project presentation for evaluation.

Total Images: Started with 120,000 images for training.

Conditions Ensured:

- Equal numbers of fake and real images in the dataset.
- Balanced distribution within fake images from different models.
- Dev set remained unused until submission day for evaluation.

Class Balance Maintenance: To address the surplus of fake images:

Among fake images, equal distributions from various model sources were ensured: 'inpainting', 'insight', 'text2img', '1m_faces_00', 'iFakeFaceDB'.

Real Image Allocation:

'celebahq256_imgs': 30,000 images, 'wiki': 30,000 images.

This approach aimed to ensure a balanced, representative dataset for robust model training and unbiased evaluation during the project's final stages.

Saving these files in excel, and moved them to train, test and dev folder. Dev test is created to test our models and get final evaluation metrics.

Here's a summary of the processed Excel file used for training:

Processed Excel File Summary:

Columns Included:

image_path: Contains the original path of each image in the dataset.

destination_path: Indicates the final location where the image was moved (train, test, dev).

Folder: which AI model generated that image

split: Specifies the categorization of the image into train, test, or dev.

target_class: Represents the classification of the image as real or fake.

target (numerical): Numeric representation of the target, with '1' for real and '0' for fake.

This Excel file serves as the primary dataset for model training, with each image's path, destination path, split categorization, and corresponding target label (both categorical and numerical) meticulously organized. The 'dev' split remains reserved for final evaluation, ensuring the integrity of the model assessment process.

Modeling

DEIT - Data Efficient Image Transformer

DeiT stands for Distilled data-efficient Image Transformers. This model is an image transformer with a very similar architecture to the Google’s ViT model discussed in the main report with just an addition of a distillation layer in DeiT models. In this model too, images are presented to the model with a fixed-size patch (16x16) linearly embedded that are linearly embedded. From the paper we could understand that the distillation process requires the distillation token.

It learns from a teacher model (CNN) using a distillation token in addition to the usual class ([CLS]) token. The distillation token learns by paying attention to both the class token and patch tokens in the image during training.

The training was done by the pretrained model downloaded from huggingface. I used BCEWithLogitsLoss() as the loss function and stochastic gradient descent as the optimizer.

Training

As mentioned in the Data section of the report, we collected fake images dataset from various sources that were iFakeFaceDB, deepFakeFace and diffusion-based model generated images. We trained our model with the combination of all the fake images i.e, training set had images from iFakeFaceDB set, deepFakeFace set and diffusion models generated images. Other than this, we also trained our model on each of these specific datasets. The thought process behind this was to know how biased a model is getting to one particular dataset if the model is given exclusively those pictures or how well can the model generalize with just a single source of images.

```
PRETRAINED_MODEL ="facebook/deit-base-distilled-patch16-384"
```

```
def model_definition():
    model = AutoModelForImageClassification.from_pretrained(PRETRAINED_MODEL,
                                                            num_labels=1,
                                                            ignore_mismatched_sizes=True)

    model = model.to(device)
    optimizer = torch.optim.SGD(model.parameters(), lr=LR, momentum=MOMENTUM)
    criterion = nn.BCEWithLogitsLoss()

    scheduler = ReduceLROnPlateau(optimizer, mode='min', factor=0.5, patience=LR_PATIENCE, verbose=True)

    print(model, file=open(f'summary_{MODEL_NAME}.txt', 'w'))

    return model, optimizer, criterion, scheduler
```

Figure 1. Model Definition

Results

Model	Accuracy	Precision	Recall	AUROC	F1Score
deit	0.83607	0.82924	0.87737	0.83243	0.85262
deit_iFakeFaceDB	0.66425	0.61683	1	0.63467	0.76301

The DeiT model gave decent metrics on our dev set. As we can see that accuracy is around 83%. The only problem we had while training this model was that the training time for this model was high as compared to other models we trained. The only downside to this model was that the training time was very high which we got to know when we trained the model.

t-SNE Plot

t-SNE stands for t-Distributed Stochastic Neighbor Embedding. It is a non-linear dimensionality reduction technique. It aims to find a low-dimensional representation of the data (usually 2D or 3D) while preserving local structure and relationships between data points.

In most of the research paper that we have seen especially 'Towards Universal Fake Image Detectors that Generalize Across Generative Models' Two main points were made by this paper which offered us some insights and the way we can visualize the latent vector approach:

When they observed the tSNE plots for a GAN based images trained classifier the cluster for real images was more open and the cluster for GAN generated images was more compressed. Also the images that were trained on diffusion generated images they were falling in the same cluster as the real images. Which gave insight into this might be because the model is learning the fingerprint of fake images but not the real images. function can easily distinguish FGAN from the other three, but the learned real class does not seem to have any property (a space) of its own but is rather used by function to form a sink class, which hosts anything that is not FGAN.

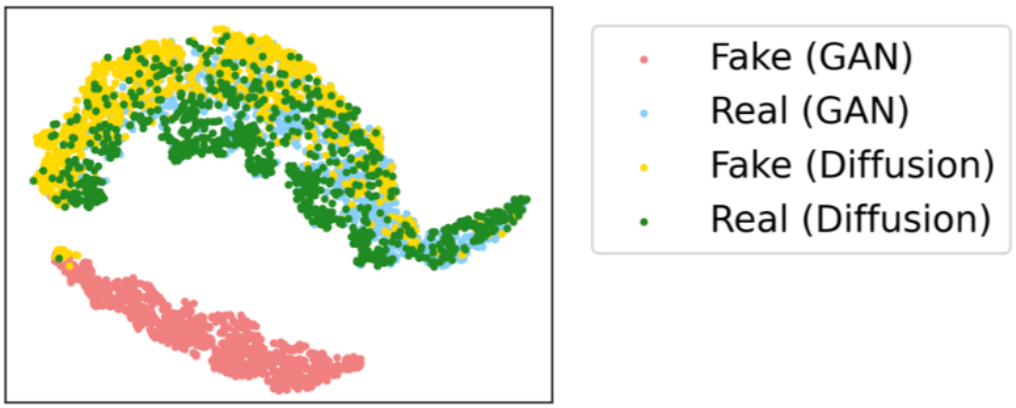

Figure 2. tSNE visualization of Real and Fake images

Performing frequency spectrum analysis of the images from GAN and diffusion. From the average frequency spectra of each domain in this paper we can see that see a distinct and repeated pattern in StarGAN and CycleGAN. However, this pattern is missing in real and diffusion-based images. So, while the diffusion network might have his property of their own, they are different from the one shared by GAN.

Since point-to-point pixel classification won't work, So, any classification decision of an image should be made after it has been mapped into some feature space. This feature space should capture low level details of the image. Hence, we would be using the embeddings from the transformer.

To investigate the clusters of latent images, we made the tSNE plots for the embedded images. We removed the last layer of our trained transformer, then passed 50 set of images from each class to it. The results are shown in the below figures.

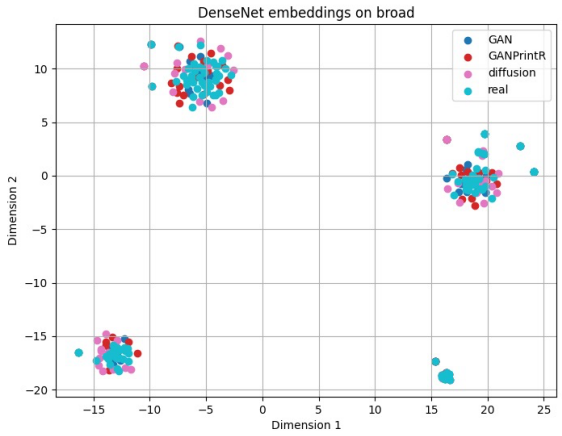

Figure 3. tSNE of Densenet trained on Complete Dataset

For DenseNet, since it is convolutional based, we had to perform an average pooling after removing the last layer and the dimension size was 1058. The results were not what we expected in this case since the real class points are overlapping everywhere. This might be because of loss of information while pooling or while reducing the 1058-dimensional space to 2 dimensions. This might have been the reason the results in conclusion are not in line with the plot we have.

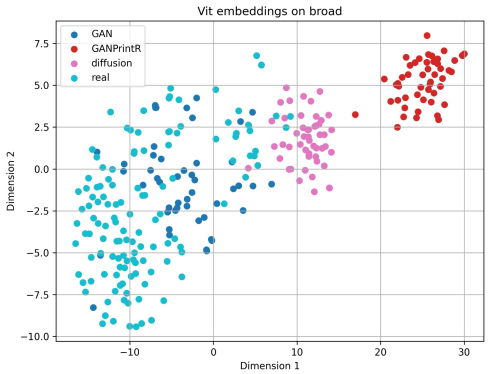

Figure 4. tSNE of Google's ViT on complete dataset

ViT network trained on combination of images performed the best, As seen in figure it can form the clusters for diffusion and GANprintR. Some of the real images are still merging with the GAN image clusters which was slightly unexpected.

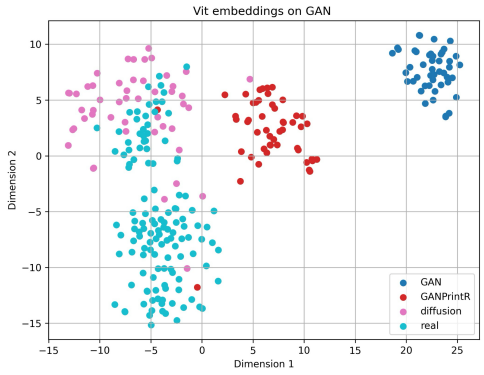

Figure 5. tSNE of Google's ViT on GAN generated images

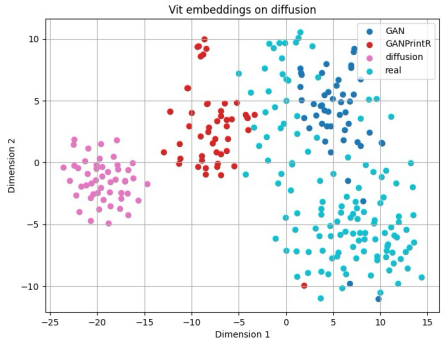

Figure 6. tSNE of Google's ViT on diffusion models generated images

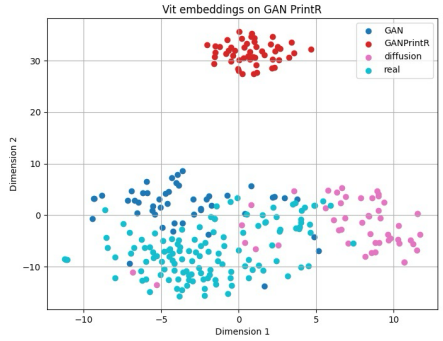

Figure 7. tSNE of Google's ViT on GAN PrintR images

From figure we can see that the GAN images are able to be in a completely different cluster, but the model also understands GANprintR and diffusion. From Figure:7 we can see that the GANprintrR images are able to be in a completely different cluster, but the distinction of other 2 with real images might still be difficult. Which is why we have the worst results on this one.

So, our best model is when the dataset with multiple AI generate images is used and then the latent representation of images from transformer is used, enabling the model to learn some representations of real.

Summary

The FaceAuth project aimed to develop a robust image classification model capable of differentiating between real human faces and AI-generated faces. Data sourcing involved curated datasets for real faces (Celeb-HQ-2, Wiki Celeb) and synthetic faces from various AI models (1million Fake Faces, iFakeFaceDB, DeepFakeFace). A meticulous dataset split and balance were maintained, ensuring equal representation of real and fake images across training, test, and dev sets. The project's primary dataset, meticulously organized in an Excel file, became the cornerstone for model training.

The team experimented with the Data Efficient Image Transformer (DEIT) model, a variant of ViT (Vision Transformer), employing it for training and evaluation. Through DEIT, the team observed promising results, achieving an accuracy of around 83% on the dev set. Additionally, t-SNE analysis was conducted to visualize the latent spaces of the trained models, offering insights into the clustering behavior of real and AI-generated images.

Conclusion

The project highlighted the challenges and advancements in detecting AI-generated faces, showcasing the potential of DEIT in discerning between real and synthetic images. The t-SNE analysis provided nuanced insights into the model's learning patterns, offering perspectives on the clustering behavior of different image types. Notably, using a dataset comprising multiple AI-generated images enabled the model to better discern representations of real faces.

References

1. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., & Jégou, H. (2021, January 15). *Training data-efficient image transformers & distillation through attention*. arXiv.org. https://arxiv.org/abs/2012.12877
2. Tsang, S.-H. (2022, August 14). *Review - DEIT: Data Efficient Image Transformer*. Medium. https://sh-tsang.medium.com/review-deit-data-efficient-image-transformer-b5b6ee5357d0#:~:text=DeiT%20has%20the%20same%20architecture,is%20a%20way%20to%20train
3. Neves, J. C., Tolosana, R., Vera-Rodriguez, R., Lopes, V., Proença, H., & Fierrez, J. (2020, July 1). *GANPRINTR: Improved fakes and evaluation of the state of the art in face manipulation detection*. arXiv.org. https://arxiv.org/abs/1911.05351
4. *Papers with code - ifakefacedb dataset*. Dataset | Papers With Code. (n.d.). https://paperswithcode.com/dataset/ifakefacedb
5. OpenRL. (n.d.). *OpenRL/deepfakeface · datasets at hugging face*. OpenRL/DeepFakeFace · Datasets at Hugging Face. https://huggingface.co/datasets/OpenRL/DeepFakeFace