DETECTING FAKES FACES AND EMOTIONS

Agathe d'Aubenton-Carafa (s243230),
Srijita Sarkar (s242527)

MSc Human-Centered Artificial Intelligence

Upasana Paul (s242577),
Gokul Desu (s242580)

MSc Autonomous Systems

ABSTRACT

The synthesis of fake images has been consistently growing and getting better in the last few years, and while it opens a lot of possibilities such as the creation of large datasets for machine learning, it also raises new problems. One of these is the use of generated images to spread misinformation. As fake news often implicates all kinds of people, it is crucial to be able to differentiate generated images of people from real ones. This is why we trained different models to detect deepfakes, with a success rate of over 90%. We also investigated the ability of new generative models to create images reflecting specific human emotions, as it is a big part of how we perceive an image and react to it. We created a model capable of identifying emotions in real faces and tested its performance on artificially generated facial images, which indicated potential for improvement.

***Index Terms*—** deepfakes, generative models, identify emotions, neural networks

1. INTRODUCTION

In the last few years, synthetic image generation using Generative Adversarial Networks (GANs) has developed considerably, achieving a realism that makes it difficult for humans to tell if the image is real or fake. It has led to problematic uses of AI-generated images such as fake news propagation and truth manipulation. Deepfake images of people violate their privacy, spread misinformation and lead to identity theft and fraud. Thus, in the current era of social media where visuals influence the way information is perceived, it is of utmost importance to know if a piece of digital media is real or not. In this project, we detect deepfakes of people's faces using different deep neural networks. We compare the performances of 5 models - CNN, ResNet50, Encoding, CNN+Encoding, ResNet50+Encoding. We further train two models (both based on the VGG-16 architecture) to recognize emotions from real facial images and test their performances on real and AI-generated images to observe how well a model trained on real data performs on hyper-realistic synthetic ones.

2. BACKGROUND STUDIES

In recent times, there has been a lot of progress in generating and detecting synthetic images. Tariq et al. proposed a neural network-based FakeFaceDetect framework to detect facial deepfakes created by GANs and humans [1]. They achieved about 99% AUROC score using EnsembleShallowNet(V1 & V3) model for GAN images and 74.9% using XceptionNet for human-generated fake images. Hsu et al. worked on generating fake-real image pairs using GANs and then DenseNet to train using pairwise learning to distinguish between real and fake data, achieving 92.9% - 98.9% precision on the different models [2]. Khaireddin and Chen proposed a fine-tuned VGGNet model for facial emotion recognition that optimizes hyperparameters and uses advanced learning rate schedulers, achieving a state-of-the-art accuracy of 73.28% on the FER2013 dataset without utilizing additional training data [3]. Roy et al. developed ResEmoteNet, which combines SE blocks and residual networks to focus on key facial features, improving emotion recognition accuracy to 79.79% on FER2013, 94.76% on RAF-DB, 72.93% on AffectNet-7, and 75.67% on ExpW [4].

3. DATASET DESCRIPTION

We use a dataset of 140K Real and Fake Faces [5] available on Kaggle to detect synthetic facial data. It contains 70K real faces from the Flickr dataset collected by Nvidia and 70K fake faces sampled from the 1 million images generated by StyleGAN. The images have been resized into 64*64 px and the data has been split into train, test, and validation folders having 5000+500+500 real and 5000+500+500 fake images. For emotion detection, we have used the FER-2013 dataset [6], having 28,709 training images, 7178 validation images and 3,589 test images of real faces, also available on Kaggle. These are grayscale images resized to 48*48 px with the face more or less centered and occupying the same amount of space in each photo. Each facial expression belongs to one of seven categories: 0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral. We also created a labeled synthetic face image dataset using the generators Canava and Adobe Firefly to test our Emotion Detection models. We have

10 images belonging to each of the 7 emotions, thus a total of 70 AI-generated test facial images.

4. METHODOLOGY

4.1. Detecting Fake and Real Face Images

4.1.1. Analysis of the Real and Fake Faces Dataset

To start the detection of real faces from fake ones, we had the idea to look at it from a facial recognition point of view. Indeed, facial recognition's goal is to differentiate faces from one another, and as such, is based on facial features. The goal was to see if some features were more common in the synthesized images.

To do so, we used the facial-recognition Python library. This module usually works by taking an image and encoding it in a 128 features vector using a trained neural network. This feature vector is then compared using the Euclidean distance to vectors from images of known people. If the distance between two vectors is lower than a threshold, then it is recognized as the same person, otherwise it is seen as two different people.

As we were not interested in recognizing a person, we stopped at the encoding part and decided to compare encodings from real and fake images using the Euclidean norm of the vectors.

4.1.2. Model Architectures

In order to differentiate the synthesized face images from real ones, we decided to start really simply by trying a convolutional neural network (CNN). The convolutional part is composed of three convolutional layers with ReLu activation, and dropout at 0.5 and max pooling after the first two. We also applied batch normalization after each convolutional layer. The fully connected part is made of three linear layers, with ReLu activation after the first two. As we use of BCEWithLogitsLoss as our loss, we do not apply a sigmoid function after the last linear layer.

Since it seems like there was a difference in the encoding of the real and fake faces (see Figure 3), we made a second model using only the encodings of the faces as an input and going through the exact same fully connected part as the precedent model.

The third model we built is a double-input one, taking both the images and the encodings as input. The images first go through the same convolutional part as the first model, and we concatenated them to the encoding before going through the fully connected part (again the exact same as before).

Since the image classification was the part that worked worse than the other one, we tried another image classification method, which is to fine-tune a pre-trained Resnet50 model by training just the last layer (and changing the output size to have a binary classification).

Finally, we tried again a double input model but this time using a pre-trained Resnet50 model removing the last layer to analyze images and concatenate the output to the encodings to go through the same fully connected model as before.

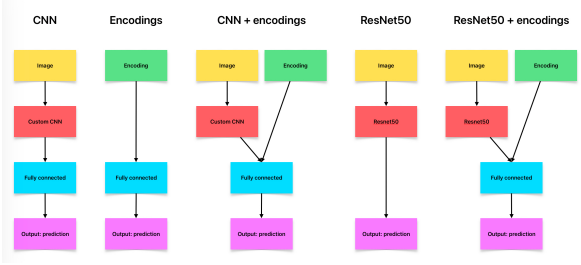


Fig. 1. Comparison of the different models for fake image detection

To facilitate comparison between models (see Figure 1, the loss, optimizer and training parameters are the same for each one (see Table 1).

Loss	BCEWithLogitsLoss
Optimizer	Adam
Batch size	32
Number of epochs	20
Learning rate	0.001

Table 1. Loss, optimizer and model parameters

4.2. Emotion Detection

4.2.1. Data Preprocessing for FER-2013 Dataset

We have first normalized the images by rescaling their pixel values so that they lie in the range [0,1]. To increase the size and diversity of the dataset, we have performed data augmentation to the training set by applying random transformations such as +-20 degrees rotation, up to 20% horizontal and vertical shift and horizontal flipping. The categorical class labels have been one-hot encoded since it is a multi-class classification problem. To improve generalization, for each epoch, the training data is shuffled.

4.2.2. Model Architectures

We have trained two neural network-based architectures on real images of people's faces present in the FER-2013 dataset. We have tested our models on both real test images present in the dataset and the labeled synthetic dataset generated by us, containing faces showing the 7 different emotions. We have compared the performances of the models on real as well as fake data to study how well a model trained on real faces generalizes to capture nuanced emotions on hyper-realistic synthetic faces.

We designed Model 1 (see Figure 2) by combining a pre-trained VGG16 base model with additional custom layers to fine-tune the model to our dataset. An initial Conv2D layer was used to convert grayscale images to the three-channel RGB format, as required by VGG16. We used the feature extraction layers of VGG16 (13 convolutional layers grouped into 5 blocks, each followed by a max-pooling layer), pre-trained on the ImageNet Dataset [7]. We excluded the top fully connected layers of VGG16 which are fine-tuned to the ImageNet Dataset. We flattened the output of the VGG16 and add a custom Dense layer of 512 units with ReLu activation, followed by a Dropout layer with a rate of 0.5 to reduce overfitting. The output layer is a fully connected layer with the softmax activation function to output class probabilities for each emotion.

Model 2 (see Figure 2) was available on Kaggle [8] which showed high accuracy on the FER-2013 dataset, and we used this to compare with the performance of our model (Model 1). It also relies on the feature extraction layers of the VGG16 architecture with an initial Conv2D layer and custom layers such as Dropout(0.5) to reduce overfitting, Flatten layer to convert into a 1D vector and pass it to three consecutive Batch Normalization and Dense layers with 32 units and ReLu activation. The output layer is a dense layer with softmax. However, while training the model, we noticed that the accuracy metric used by the author was Binary Accuracy and we corrected it to Categorical accuracy, which should be used to evaluate one-hot encoded targets in multi-class classification.

Fig. 2. The two model architectures for emotion detection

5. RESULTS

5.1. Detecting Fake and Real Face Images

5.1.1. Analysis of the Real and Fake Faces Dataset

We computed the encoding for each image of the train set, which is 5000 real images and 5000 generated ones (side note: the ability to have those encodings on the synthesized images proves that faces are detected on it, and as such that the quality of the generated data is good). We then displayed the statistical repartition of the norm of the encodings by class.

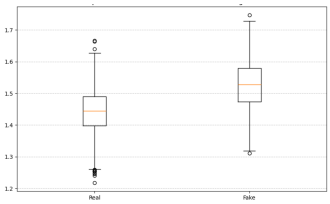


Fig. 3. Statistical repartition of the norm of the encodings based on their class

We can see that the norm of the encodings of the fake images is usually higher than that of the real ones. However, there is still an overlap between the two, so we decided not to classify them based on a threshold on the norm. Instead, since there seem to be some differences between the encodings of the real and generated images, we choose to use the encodings as input for our models.

5.1.2. Model Performances

Table 2 presents the accuracy scores obtained by each of the models presented in Figure 1.

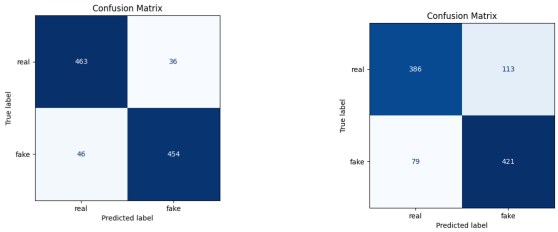
Model	Training accuracy	Test accuracy
CNN	98.52%	80.78%
Encodings	98.97%	91.78%
CNN + Encodings	99.21%	87.99%
ResNet50	98.58%	86.29%
ResNet50 + Encodings	99.56%	91.69%

Table 2. Training and test accuracy for each model

We can see that the model using only the encodings as an input performs better than the other and that adding the image as an input can even decrease the performance of the model, depending on how the image is analyzed. Our hypothesis to explain the better performance of the models using encodings is that those vectors are made to accentuate differences between faces, which may help to reduce overfitting as we have a more diverse train set.

We noticed another thing by looking at the confusion matrix (see Figure 4), that is the encoding model classify better the real images while the CNN model classify better the fake

images. We hypothesize that this difference could be due to the nature of the encodings, that are made to be used on real images of human faces, so they are likely more consistent on those images, while the CNN can learn to detect odd things in the generated ones that may not even be on the face of the person.



(a) Encoding model (b) CNN model

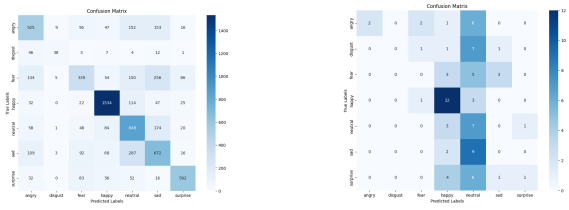
Fig. 4. Confusion matrix for two different models

5.2. Emotion Detection

Table 3 shows the performances of Models 1 and 2 on the FER-2013 dataset (real faces) and the synthetic dataset created by us. Model 1 achieves a training accuracy of 65.25% and a test accuracy of 63.36% on real data. Its performance dropped considerably when testing on the AI-generated face images (27.85%). The performance of Model 2 on training data dropped drastically after correcting the accuracy metric from Binary (originally used by the authors in [8]) to Categorical (from around 85% to 27.85%). It however shows slightly better results when tested on fake images (29.27%) compared to real ones (22.60%).

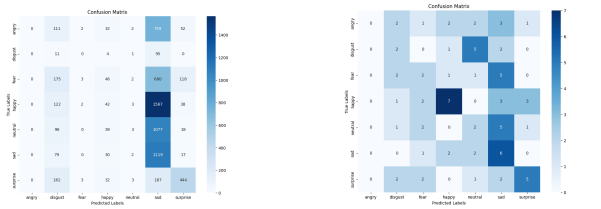
Model	Training Accuracy	Testing Accuracy on Real Images	Testing Accuracy on Fake Images
Model 1	65.25%	63.36%	26.83%
Model 2	27.85%	22.60%	29.27%

Table 3. Training and testing accuracies on real and fake face images for each model.



(a) Testing on Real Images (b) Testing on Fake Images

Fig. 5. Confusion matrices for Model 1



(a) Testing on Real Images (b) Testing on Fake Images

Fig. 6. Confusion matrices for Model 2

Figure 5 shows the confusion matrices of testing Model 1 on real and fake data respectively. The confusion matrix for real images shows a clear diagonal dominance, thus establishing that the model can distinguish between the different emotion types. However, this is not the case for testing on fake images. Figure 6 shows the confusion matrices for Model 2 and the lack of diagonal dominance shows that the model struggles to differentiate between the classes for both real and fake data. The model seems to be biased towards majority classes such as 'Happy' and 'Neutral' for real faces.

6. CONCLUSION

We have demonstrated a difference between the images of faces generated by StyleGAN and real face images and exploited it to create a model that is able to differentiate the two with an accuracy of almost 92%. The poor performance of the VGG16-based emotion detection models on synthetic images shows that a model which performs fairly well on real images fails to generalize on synthetic data. This could be due to the inability of synthetic images to capture true emotions and nuances in facial features, unlike actual photographs, even though they are hyper-realistic.

7. FURTHER WORK

In order to improve and extend our work, we could add data from other generative models to our training dataset to differenciate fake images from real ones. Indeed our model is trained only on the StyleGAN generated ones, but what we realistically need is a model that performs well on all kind of generated images. We could try to increase our model performance for emotion detection to better capture differences in facial features and emotions in synthetic images. One idea is to also train the model on real as well as a robust dataset of synthetic face images (that is adequately representative of different genders, ages and ethnicity). We could better handle class imbalances in the FER-2013 dataset through more data augmentation to prevent bias towards the majority classes.

8. REPOSITORY

The code is included in the Github repository: `https://github.com/Lukog05/Deep-Learning-Project/tree/main` [9].

9. REFERENCES

[1] S. Tariq, S. Lee, H. Kim, Y. Shin, and S. S. Woo, "GAN is a Friend or Foe? A Framework to Detect Various Fake Face Images," in *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing (SAC '19)*, New York, NY, USA, 2019, pp. 1296–1303, Association for Computing Machinery.

[2] Chih-Chung Hsu, Yi-Xiu Zhuang, and Chia-Yen Lee, "Deep fake image detection based on pairwise learning," *Applied Sciences*, vol. 10, no. 1, pp. 370, 2020.

[3] Y. Khaireddin and Z. Chen, "Facial emotion recognition: State of the art performance on fer2013," *arXiv preprint arXiv:2105.03588v1*, 2021.

[4] A. K. Roy, H. K. Kathania, A. Sharma, A. Dey, and M. S. A. Ansari, "Resemotenet: Bridging accuracy and loss reduction in facial emotion recognition," *arXiv preprint arXiv:2409.10545v2*, 2024.

[5] Xhlulu, "140k real and fake faces," Kaggle, 2020.

[6] Mandar Sambare, "Fer-2013 dataset," Kaggle, 2021.

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. IEEE, 2009, pp. 248–255.

[8] Vishrut Grover, "Emotion recognition with vgg16," `https://www.kaggle.com/code/vishrutgrover/emotion-recognition-with-vgg16`, 2021, Accessed: [05.12.2024].

[9] Lukog05, "Deep learning project," `https://github.com/Lukog05/Deep-Learning-Project/tree/main`, 2024.