The Resilience/Ignorance of Neural Networks in Predicting Distorted Facial Images

In the rapidly evolving field of artificial intelligence, the resilience and adaptability of neural networks in predicting distorted facial images is a topic of significant interest. This project aims to explore the performance of neural networks on the Labeled Faces in the Wild (LFW) people dataset from Sklearn, which comprises 2914 image features and 13233 examples. The goal is to predict the identity of an individual based on an image and to investigate how the prediction accuracy changes when the images are distorted in various ways or generated by a General Adversarial Network (GAN). Can these systems maintain their accuracy when faced with distorted or altered images or GANs? And if they can, what does this mean for the future of facial recognition technology?

Data

This database comprises 13,233 target face images. The database contains images of 5749 different individuals. Of these, 1680 people have two or more images in the database. The remaining 4069 people have just a single image in the database. The images are available as 250 by 250 pixel JPEG images. Most images are in color, although a few are grayscale only. The goal of this dataset originally is to help study the problem of face recognition using previously existing images, that is, images that were not taken for the special purpose of face recognition by machine.

Methods

We used the CNN architecture for our project with 2 convolutional layers followed by ReLU activation functions after each conv laver, and finally, a dense layer. The first convolutional layer has 32 filters with a 5x5 kernel size. The second convolutional layer has 16 filters with a 3x3 kernel size. The dense layer consists of 5749 classes, representing the number of unique individuals in the dataset, and uses a softmax activation function to output probability scores for each class. We conducted two separate training sessions to evaluate the CNN model's performance.

CNN Training 1: This session utilized 4324 images, with 10 images per person for model training for 20 epochs with a 10% validation data. The resulting metrics indicated a training accuracy of 95.27% and a validation accuracy of 29.56%.

CNN Training 2: In this iteration, the CNN model was trained on the entire dataset comprising over 13,233 images. In this training, we chose to overfit the data with no validation dataset, as most of the images (over 4000 images) correspond to 1 person, so it would not make sense to have a validation set.

The GAN architectures include the Generator and Discriminator. The generator has multiple Conv2DTranspose layers with BatchNormalization and LeakyReLU layers following each. The LeakyReLU layer allows for negative gradients. Instead of zeroing out these gradients, they are multiplied by a small alpha value of 0.3. The discriminator model has the two basic Conv2D layers with LeakyReLU and a dropout rate of 0.3. A Fully connected binary classifier network sits at the end which classifies real and fake images.

GAN Training: We trained the initial GAN generator for over 2000 epochs, taking over 6 hours. Subsequently, we fine-tuned the final model using only images of George W. Bush. The dataset contained 530 images of him, and the fine-tuning process was carried out over 1000 epochs, taking about an hour.

Training and Experimental Results

CNN 1: Distortion analysis revealed varying accuracies for different distortion types, with notable performance in noise-distorted images, achieving an accuracy of 100%

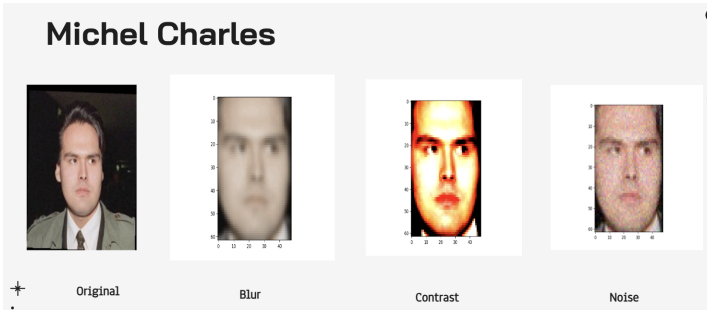
Distortion Type	***Accuracy***
Invert	1.28%

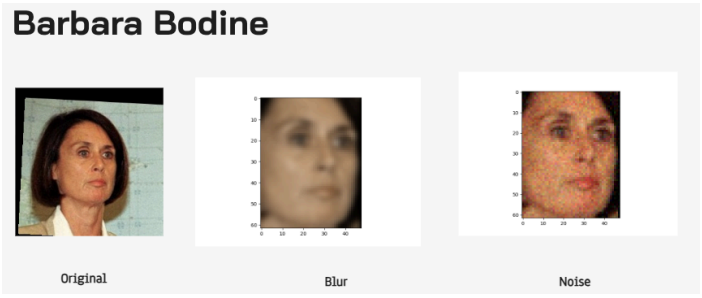
Contrast	1.92%
Blur	82.69%
Noise	100.0%

CNN 2: Distortion analysis showed that the model performed well with certain types of distortions, especially blur and noise, achieving accuracies of 71.7% and 99.18%.

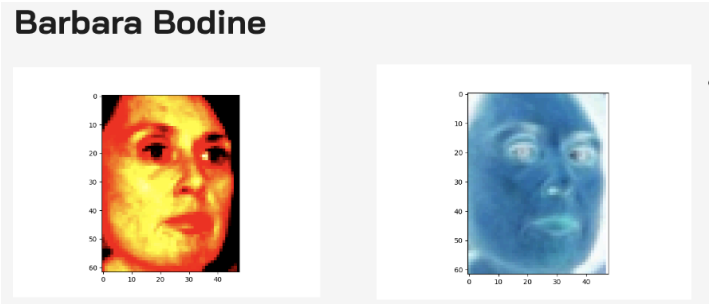
Distortion Type	***Accuracy***
Invert	0.87%
Contrast	20.96%
Blur	71.7%
Noise	99.18%

CNN algorithm predicted these images correctly

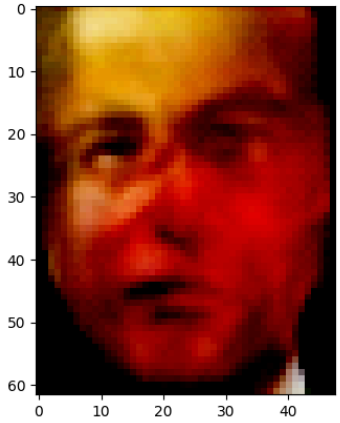




CNN algorithm predicted these images incorrectly



When it came to predicting fake GAN images, the CNNs demonstrated remarkable resilience. Both CNNs, when tested with the base GAN, had a prediction rate of 0.0% (0 out of 1000). Even after fine-tuning, the prediction rate only slightly increased for **CNN 1** to 0.004% (4 out of 1000), while the other maintained a prediction rate of 0.0%.



The image displayed above is a representation of George W. Bush, generated by our fine-tuned GAN, which was accurately predicted by **CNN 1**.

Interpretation

In the process of blurring and adding noise to the images, the resilience of the CNN models was evident. The model trained with at least 10 images per person performed slightly better than the other. When predicting images with altered contrast, the overfitted model surpassed the other. This can be attributed to the overfitting potentially enabling the model to familiarize itself with all the images, thereby increasing its prediction accuracy. However, both models struggled with inverted images, which is understandable given the significant difference from the original images.

When presented with the generated images by the GAN, the CNN model proved to not be resilient. One of the CNN models managed to correctly predict a mere 4 out of the 1000 generated images, while the other failed to make any correct predictions. Interestingly, this was

the desired outcome, as a number of the generated images looked nothing like G.W. Bush and were of poor quality.

Conclusion/Future Work

The findings from our experiment highlight the resilience of CNNs in recognizing facial images under various distortions, with particularly strong performance against noise and blur distortions. However, both models struggled with high-contrast and inverted images, indicating potential area for further enhancement.

Given additional time, we would spend more time on CNNs and GANs architecture. We would like to explore different model architectures for both CNNs and GANs. While we managed to experiment with CNNs, doing this for GANs proved to be more challenging due to the extensive training time they require, making it time-consuming to experiment with alternative architectures. Also, GANs require a substantial amount of data - often more than 100,000 images - for effective training. Therefore, working with a dataset larger than our current one would be crucial.

Command Line Arguments

Options:
'-c', '--CNNModel' – cnn1 or cnn2, defaults to cnn1,
'-t', '--train' – True or False, defaults to False,
'-e', '--Epochs' – takes any integer, defaults to 500.

Python3 test.py (options)
This command line uses the epochs and train options

Python3 cnn.py (options)
This command line uses the train and CNNModel options. Epochs preset to 10 for overfitting the cnn2 model and 20 for cnn1 model with best model saved based on validation accuracy.

Python3 GAN.py (options)
This command line uses the epochs and train options

PS: We were unable to push these models to our github repo. Here is a link to a google drive containing our models.
https://drive.google.com/drive/folders/1g0RzWBCJeLRx9oEcyslpnvZ5b-3BJwqp?usp=sharing

References

Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller.
Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments.
University of Massachusetts, Amherst, Technical Report 07-49, October, 2007.