CLASSIFICATION OF REAL AND FAKE FACES

NEURAL NETWORKS AND DEEP LEARNING

Gehan Sherif	Mohamed Saleh	Omar Emad
201901989	201902216	201901607

June 24, 2022

ABSTRACT

Due to the rapid development of both science and technology, especially in the field of artificial intelligence, computers are now capable of creating vivid fake faces using Generative Adversarial Networks (GAN), which could easily deceive human beings, inevitably such technology will be used to create fake news, fake videos, fake pieces of evidence, etc. Therefore, this review aims to discuss the topic of fake face discrimination, the possible techniques used and do a comparative study on the possible architectures of models capable of such feat.

Keywords Deepfake · Fake faces · Classification of real and fake faces · Fake faces detection · Deep learning

1 Problem Definition & Motivation

Technological advances in AI have taken a considerable leap in the current century, giving rise to many applications to aid humankind. However, like any other tool, technology can be a double-edged sword. As a result, applications of harmful impact emerged alongside the good ones. This project focuses on one of the harmful applications called deepfakes, an image or a video that has been altered using AI to misrepresent a person as someone else.

Deepfakes can impact society through impersonation, fake social media accounts, and fake news when spread across social media. They are also used to do atrocious deeds such as creating celebrity pornographic videos, material child sexual abuse, and financial fraud. For example, a deepfake was made of Barack Obama insulting Donald Trump[1]. Another showed Mark Zuckerberg confessing to having total control of people's stolen data. Many more suspicious deepfakes were (and continue to spread) among the public. With the rapid progress in the technology of creating fake faces, it becomes nearly impossible to humans to differentiate between what is real and what is not. However, unlike humans, machines can be aided with AI and deep learning to learn how to differentiate between copies and originals.

The invention of the **Generative Adversarial Network**(GAN) by Ian Goodfellow in 2014, and its subsequent success in generating vividly realistic face images, can be considered as the starting point for deepfake technology [2]. A typical GAN architecture consists of two main sub-modules: a discriminator module and a generator module. The generator's role is to generate fake data from simulated learning of real data and pass it to the discriminator module to see whether it is real or fake. The result is a set of generated fake data that we can fool the discriminator. As the discriminator improves, generated data becomes closer to the realistic ones.

Our main goal is to create a deep learning model that can classify images as real or fake.

2 Literature Review

In a very recent paper by Samy S. Abu-Naser et al., five models were proposed[3]. Four are based on transfer learning, and the fifth is a proposed architecture by the authors. The test accuracy averaged around 96.8% for the models. The real-fake faces dataset was used. It contains the following four classes: fake-easy, fake-mid, fake-hard and real. Fig(1) below shows the architecture of the proposed model.

```
from keras import layers
from keras import models
from keras import backend as K
model = models.Sequential()
model.add(layers.Conv2D(32, (3, 3), activation='relu',
    input_shape=(256, 256, 3)))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(64, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(128, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(256, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(256, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(512, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Flatten())
model.add(layers.Dense(512, activation='relu'))
model.add(layers.Dense(4, activation='softmax'))
```

Figure 1: Model Architecture

The remaining four models used the architectures of **VGG16**, **ResNet50**, **MobileNet** and **InceptionV3** with pre-trained weights and a fully connected Dense layer on top. The output layer uses a **softmax** activation. The optimiser used was **Adam**, with a learning rate of 0.0001. Each model was trained for 150 epochs. Fig(2) summarised the resulting accuracy scores.

Criterion	Proposed Model	VGG16	ResNet50	MobileNet	InceptionV3
Training Accuracy	100%	100%	100%	100%	99.77%
Validation Accuracy	95.21%	93.05%	99.18%	98.14%	99.03%
Training loss	0.000005	0.0002	0.0003	0.0016	0.0075
Validation loss	0.2906	0.2237	0.0265	0.0563	0.0269
Testing Accuracy	95%	93%	99%	98%	99%

Figure 2: Performance Of Models

A different paper by Yuanyuan Li et al. proposes an architecture of 9 convolutional layers and a fully connected layer on top [4]. The network takes as input an image of size $96 \times 96 \times 3$.

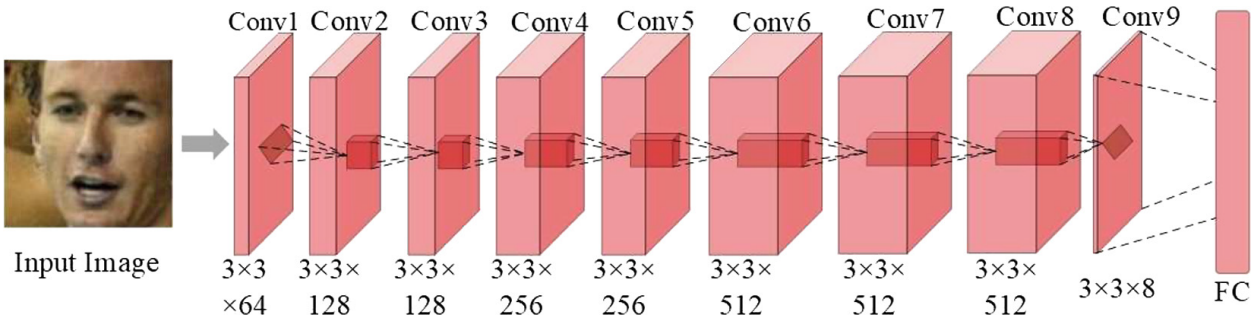


Figure 3: Model Architecture

The first network layer, the processing layer, uses 64 convolution kernels, each of size 3×3, to conduct preliminary convolution on the input. A **LeakyRelu** is then used to get 64 sizes of 96×96 rough feature maps. Then these operations were repeated eight times on the feature maps using multiple convolutions to extrapolate the deep-level semantic information from the given image.

The dataset contains a total of 30,000 images. Part of the data was obtained by web crawling, while the rest came from the LFW faces dataset. The model was trained using Adam as the optimiser with no callbacks for 3000 epochs. The model consequently became over-fit after 15 epochs. The best validation accuracy was reached in the fifth epoch. The test accuracy reached 99.24% on the model, which was re-trained for 15 epochs.

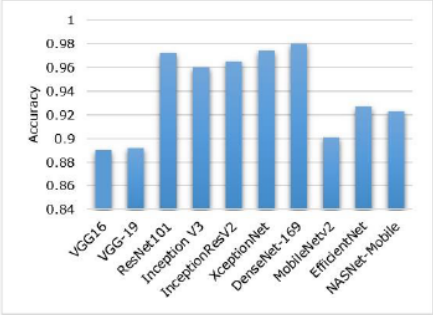
The LFW dataset used here is not balanced. The publishers thoroughly explain the reasons for this on their website. Here is a quote for two of those reasons:

"Many groups are not well represented in LFW. For example, there are very few children, no babies, very few people over the age of 80, and a relatively small proportion of women. In addition, many ethnicities have very minor representation or none at all." [5]

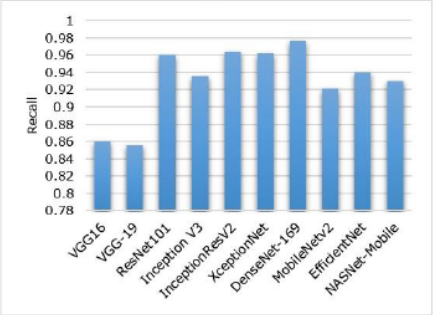
"While theoretically LFW could be used to assess performance for certain subgroups, the database was not designed to have enough data for strong statistical conclusions about subgroups. Simply put, LFW is not large enough to provide evidence that a particular piece of software has been thoroughly tested." [5]

In [6], Momina Masood et al. propose a pre-trained CNN approach to classifying Deepfake videos. They used the DFDC database for training and validation with a random split ratio of 70% training and 30% validation. The DFDC, released by Facebook, was generated using two AI techniques currently undisclosed to the public. It contains about

19,000 original video samples as well as 100,000 deep fakes. Since it is more costly to misclassify fake samples as real than the other way around, the objective of this research was to minimise the percentage of false negatives as much as possible. In the end, the performance of ten pre-trained models was compared. All models were trained for 60 epochs with a learning rate of 0.001. The DenseNet-169 architecture achieved the highest recall value of 97.6%.



(a) Accuracy of Each model



(b) Recall Of Models

Figure 4

In [7], Y. Li et al. present a different approach to detecting deepfakes by exploiting the lack of natural human micro-movements, namely eye-blinking, in deepfake videos. A Spatio-temporal network architecture that uses CNN and RNN layers were utilised to identify the video samples that lack eye-blinking sequences. While the discrimination accuracy of the model is good, this approach is inherently limited by relying too much on a failing of the deepfake generator (lack of blinking) which may be easily overcome by the advent of better, more advanced deepfake generation systems.

E. Sabir et al. posit that the deepfake generation algorithms often fail to maintain temporal coherence in the faked videos during synthesis[8].

Yuming Gu et al. introduced a new approach to protecting public figures from deepfake scams using the uniqueness of an individual's face and head movement patterns. The person's subtle facial and head movements can be tracked and discretised into specific action units with differing levels of strength. These units are then passed as features to an SVM classifier to distinguish between real and fake footage samples of the individual[9].

3 Dataset

The dataset we will be using is 140k Real and Fake Faces,, publicly available on Kaggle. As the name suggests, it contains 140K images, 70k of which are real from the Flickr dataset collected by Nvidia. The other 70k are fake faces sampled from the 1 Million FAKE faces (generated using StyleGAN).

(a) Real Faces

(b) Fake Faces

Figure 5

4 Objectives

We have aimed to achieve the following:

(1) Reduce false negatives as it is critical for the model to effectively detect fake faces.

(2) Achieving a relatively high accuracy (over 90%) using transfer learning techniques.

(3) Experiment with the more recent versions of pre-trained models like **ResNet152**.

5 Ethical Considerations

Deepfake technology is a powerful tool which has several applications. It is crucial to think about when its usage becomes morally doubtful and what, if anything, distinguishes the moral wrong that characterises such morally problematic deepfakes. This technology has been used for beneficial purposes. For example, in a video promoting the Malaria Must Die Initiative, it allowed David Beckham to "speak" nine languages in a video promoting the Malaria Must Die Initiative. One of the deepfakes applications is research into software that can artificially reconstruct the voice of persons who cannot speak due to diseases like ALS. So, deepfake should not be banned, but it needs more regulations[10].

Deepfakes can blur the line between real and fake, which raises many serious concerns. We classify deepfakes into four main categories to address them from a regulatory and ethical point of view[11]. The following table summaries the four categories of deepfakes:

	Examples	Advantages	Major Concerns	Unintended Consequences	Legal Responses
Deep Fake Pornography (e.g., revenge porn)	One's face transferred onto porn actor's naked body	—	Invasion into autonomy and sexual privacy; humiliation and abuse	Humiliation; exploitation; physical, mental or financial abuse of individuals or corporations	*Public law* (criminal law, administrative action) *Private law* (torts)
Political Campaigns	Speeches of politicians, news reports, information about socially significant events	Could promote freedom of speech	Damage to reputation, distortion of democratic discourse, hostile governments, impact on election results	Eroding of trust in institutions; deepening social divisions and polarisation; damage to national security and international relations	*Public law* (constitutional law, administrative law, criminal law) *Private law* (defamation, libel, slander, torts; copyright law)
Reduction of Transaction Costs	Translating video records into multiple languages	Facilitation of social interactions, creation of new business models	Ownership of IPRs to the content; privacy	Emergence of new data silos	*Private law*: contract and tort law
Creative and Original Deep Fakes	Nicolas Cage scenes, parody memes	Promotion of creativity and science, free speech	Ownership of IPRs, privacy	Bullying among children	*Private law* (fair use and copyright law, contract law, tort law) *Public law*: constitutional law

Figure 6

Deepfakes detection technologies are required not only because deep fakes are incredibly realistic. From the constitutional and normative law point of view, such detecting technologies are essential. Currently, most multimedia platforms, such as Twitch, Facebook or YouTube, have various legal standards to control the content. Those platforms do the most advanced techniques to detect and remove immoral, illegal or malicious content. Also, several tools are available for platform users to identify and report inappropriate content. Because deep fakes are, in most situations, widely realistic and can difficult be discernible by a naked human eye. Private platform operators and governments have started building and developing deep-fake detection technologies[12].

Deepfakes are morally acceptable depending on the following three main factors[10]:

(1) The deepfaked person's consent.

(2) Explicit announcement of the deepfake nature.

(3) The deepfake should be made without malicious intent.

General Microsoft responsible AI principles: [12]

(1) Fairness: The systems that we develop and deploy should reduce unfairness in our society rather than keep things at the same level or even worsen it.

(2) Reliability Safety: We should ensure that the AI systems we build are consistent with the design ideas and work in a way that is consistent with our values and principles. The systems should have no negative effects on the surrounding environment, and we push the product with quantified and well-understood risks and harms.

(3) Privacy & Security: We need to make sure that the user's data we use does not leak or disclosed. We also need to know the source of the data used to train our models and how it was collected.

(4) Inclusiveness: Making sure that we are intentionally inclusive and diverse with the approaches we take towards AI.

(5) Transparency: it has mainly two sides; it implies that the individuals who design AI systems should be honest about how and why they are utilising AI, as well as the limitations of their systems. Transparency also implies that people should be able to understand the behaviour of AI systems.

(6) Accountability: we are accountable for how our technology impacts the world. Also, part of our accountability is to help our customers and partners be accountable by putting some guidelines and principles for using our products.

6 Exploring Data & Pre-processing

6.1 Exploring Data

Plotting 9 images from real and fake images:

Figure 7

We wanted to try this trick proposed in [13] Ranjan, R et al., where the models were not trained on all three colour channels. However, instead of one colour channel, we started by splitting some of the images to see if we could spot a deepfake with our naked eyes, then trained the best model again on the red channel only it gave worse results than the model trained on all 3 colour channels.

Plotting each channel of RGB separately for 6 images of real and fake faces:

Figure 8

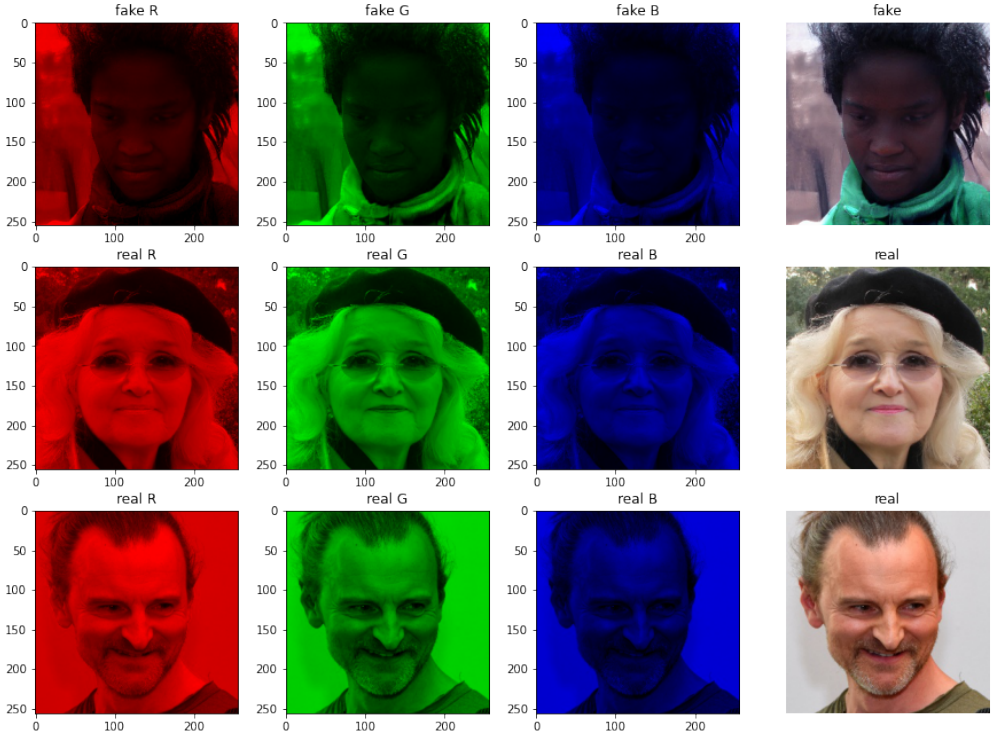


Figure 9

6.2 Pre-processing

To simulate the effect of noise on the fake faces sent using various media platforms, we added heavy image augmentations to the training set to avoid overfitting. We have used ImageDataGenerator from Keras to add image augmentations and chose a batch size to be 256 and dimensions of image 256 $\times$256.

The preprocessing function was changed to fit the pre-trained model being used.
The data is augmented using the following transformations:

(1) Width shift with range 0.1.

(2) Height shift with range 0.1.

(3) Random rotations with degree range 40.

(4) Horizontal flipping.

The python code used for pre-processing:

```
BATCH_SIZE = 256
HEIGHT = 256
WIDTH = 256

train_datagen =  ImageDataGenerator(
      preprocessing_function = tf.keras.applications.efficientnet.preprocess_input
    ,
      width_shift_range=0.1,
      height_shift_range=0.1,
      rotation_range=40,
      horizontal_flip=True,
    )

train_generator = train_datagen.flow_from_directory(TRAIN_DIR,
                                                    shuffle = True,
                                                    seed = 7,
                                                    target_size=(HEIGHT, WIDTH),
                                                    batch_size=BATCH_SIZE,
                                                    color_mode='rgb',
                                                    class_mode='categorical',
                                                    subset='training')
```

7 Hyper-Parameters

Due to the limited computational resources, we used 10% of the data and trained all the models with the following hyper-parameters

```
EPOCHS = 10
base_learning_rate = 0.0009
Optimizer = tf.keras.optimizers.Adam(learning_rate = base_learning_rate)
EarlyStopping = keras.callbacks.EarlyStopping(monitor="val_loss", min_delta=0,
    patience=3)

#Classification Head Creation
def MakeModel(Transfer_Model):
  for layer in Transfer_Model.layers:
    layer.trainable = False

  x = Flatten()(Transfer_Model.layers[-1].output)
  x = Dense(758, activation='relu')(x)
  x = Dropout(0.5)(x)
  x = Dense(512, activation='relu')(x)
  x = Dropout(0.3)(x)
  x = Dense(256, activation='relu')(x)
  output = Dense(2, activation='softmax')(x)

# return new model
  return Model(inputs = Transfer_Model.inputs, outputs = output)
```

The makeModel function takes the base class of a transfer learning model, freezes all of its layers and then adds the classification head. The purpose of the function is to automate the model-making process as we tried around 30 different transfer learning models.

8 Choosing Transfer Learning Model

Transfer learning models transfer gained knowledge while solving a problem to a different but related problem. However, we have chosen a big dataset; we used the pre-trained models to save massive training efforts by leveraging previous knowledge.

Initialising the network with pre-trained weights results in better performance than random weights.

The features are transferred poorly when the dataset is not similar to that used in the transfer learning model.

In [14], They estimated the accuracy of each transfer learning model on the ImageNet-1k validation set for an image classification task.

The below figure shows a comparison between different transfer learning models where the ball size represents the model complexity, the x-axis represents the accuracy, and the y-axis represents the computational complexity. [14]

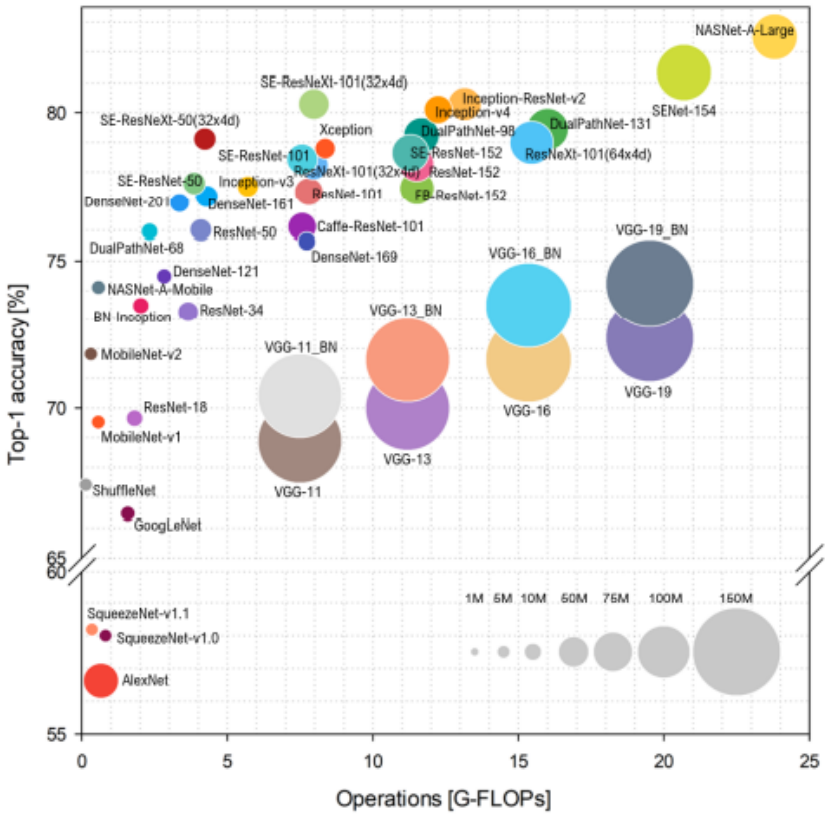


Figure 10

we were not sure how to pick promising models so we did somewhat of a comparative study between 29 pre-trained models trained on 10% of the data.

9 Proposed Models

9.1 Sample From The Dataset

The used portion of the dataset is split as the following:

(1) The training set of 100k images.

(2) The test set of 20k images.

(3) The validation set of 20k images.

The tables below represent the transfer leaning models used and accuracy and recall for each one.

Model	InceptionResNetV2	InceptionV3	MobileNet	MobileNetV2	VGG16	VGG19	Xception
Accuracy	78.04%	76.7%	82.7%	81.25%	78.95%	78.55%	77.3%
Recall	73.6%	77.4%	88.5%	88.7%	84.5%	85.7%	80.1%

EfficientNets	B0	B1	B2	B3	B4	B5	B6	B7
Accuracy	79.25%	80.65%	80.4%	82.70%	80.0%	77.39%	78.35%	78.45%
Recall	90.8%	76.8%	74.9%	88.7%	75.9%	69.8%	77.7%	72.4%

ResNets	50	101	152	50V2	101V2	152V2
Accuracy	83.2%	80.34%	84.85%	80.35%	80.1%	80.75%
Recall	95.5%	90%	94.4%	89.7%	85.3%	86.6%

EfficientNetV2s	B0	B1	B2	B3
Accuracy	80.1%	83.6%	81.5%	81.9%
Recall	95.4%	94.4%	94.1%	92.6%

DenseNets	121	169	201
Accuracy	82.3%	79.9%	86.1%
Recall	91.9%	96.8%	90.2%

DenseNet201	RGB	R
Accuracy	86.1%	77.75%
Recall	90.2%	80.7%

9.2 The Whole Dataset

The dataset is split as the following:

(1) The training set of 100k images.

(2) The test set of 20k images.

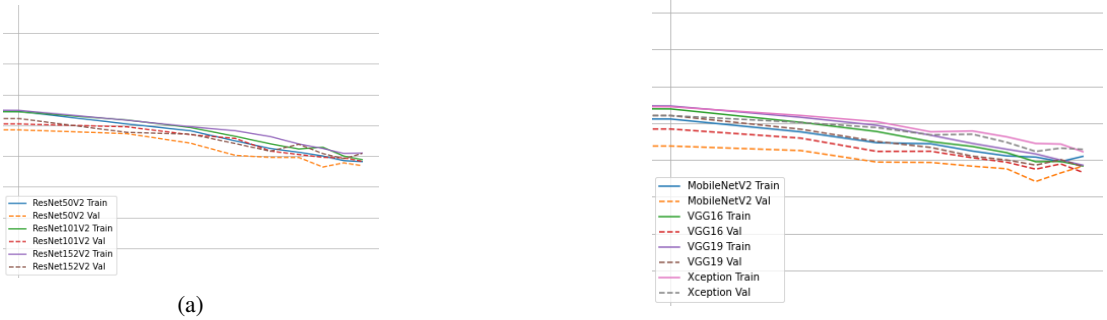
(3) The validation set of 20k images.

The following table shows the used transfer learning model and accuracy of each model trained by all the images in the dataset:

Model	ResNet152	DenseNet201	MobileNet	EfficientNetB3
Accuracy	90.2%	71.9%	74.2%	87.7%

10 Discussion

Compared with models trained on the same dataset, there is a VGG16 model with 98.8% accuracy, InceptionResNetV2 with 99.96% accuracy and DenseNet121 with 0.95% accuracy which was trained on grayscale images. we can notice that our models fail in comparison to these three models, which is expected as the models were not trained long enough on all of the data; even the learning curves show room for improvement in terms of lower validation loss.



(a) (b)

Figure 11

Since The Recall score is of great interest in our problem because it is very consequential to classify a deepfake as real, the most promising models in terms of recall were (trained on 10% of the dataset)

Model	DenseNet169	ResNet50	EfficientNetV2B0	EfficientNetV2B1	EfficientNetV2B2
Recall	96.8%	95.5%	95.4%	94.4%	94.1%

Gragnaniello, Diego, et al stated in a critical analysis of the state of the art review is that the overall accuracy is always above 90% regardless of the type architecture used and further improvements achieve about 97% if training is done with StyleGAN2 images[15].

11 Conclusion

DenseNet201 shows quite promising performance on only 10% of the data and 10 training epochs. Further experimentation on architecture of classification head, learning rates and prepossessing of data needs to be done. If we were to choose pre-trained models to train on the entire dataset we would go with ResNets,DenseNets and EffecientNetV2s as they achieved an average accuracy of 82.8%, 82.77% and 81.775% respectively. DenseNet169 did achieve the highest recall score of 96.8% which is very close to that achieved by the DenseNet169 in [6], Momina Masood et al.

12 Load Distribution

We have worked together on searching for topic, choosing dataset, setting objectives, and writing literature review and the whole code.

The load distribution for writing the report is as the following:

Gehan Sherif	Mohamed Saleh	Omar Emad
Ethical Considerations	Problem Definition & Motivation	Dataset
Choosing Transfer Learning Model	Hyper-Parameters	Proposed Models
Exploring Data & Pre-processing	Conclusion	Discussion

13 References

[1] S. Agarwal, H. Farid, Y. Gu, M. He, K. Nagano, and H. Li, "Protecting World Leaders Against Deep Fakes," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshop

[2] Goodfellow, I., Pougetabadie, J., Mirza, M., et al.: Generative Adversarial Nets. In: Neural Information Processing Systems, pp. 2672–2680 (2014)

[3] Salman, Fatima Maher & Abu-Naser, Samy S. (2022). Classification of Real and Fake Human Faces Using Deep Learning. _International Journal of Academic Engineering Research (IJAER)_ 6 (3):1-14.

[4] Li, Y., Meng, J., Luo, Y., Huang, X., Qi, G., Zhu, Z. (2021). Deep Convolutional Neural Network for Real and Fake Face Discrimination. In: Jia, Y., Zhang, W., Fu, Y. (eds) Proceedings of 2020 Chinese Intelligent Systems Conference. CISC 2020. Lecture Notes in Electrical Engineering, vol 705. Springer, Singapore. https://doi.org/10.1007/978-981-15-8450-3-62.

[5] "LFW Face Database : Main", Vis-cs.umass.edu, 2022. [Online]. Available: http://vis-www.cs.umass.edu/lfw/index.html.

[6] M. Masood, M. Nawaz, A. Javed, T. Nazir, A. Mehmood and R. Mahum, "Classification of Deepfake Videos Using Pre-trained Convolutional Neural Networks," 2021 International Conference on Digital Futures and Transformative Technologies (ICoDT2), 2021, pp. 1-6, doi: 10.1109/ICoDT252288.2021.9441519.

[7] Y. Li, M.-C. Chang, and S. Lyu, "In ictu oculi: Exposing ai generated fake face videos by detecting eye blinking," arXiv preprint arXiv:1806.02877, 2018.

[8] E. Sabir, J. Cheng, A. Jaiswal, W. AbdAlmageed, I. Masi, and P. Natarajan, "Recurrent Convolutional Strategies for Face Manipulation Detection in Videos," Interfaces (GUI), vol. 3, p. 1, 2019.

[9] B. Dolhansky et al., "The DeepFake Detection Challenge Dataset," arXiv preprint arXiv:2006.07397, 2020.

ps, 2019, pp. 38-45.

[10] A. de Ruiter, “The distinct wrong of Deepfakes Philosophy & Technology,” SpringerLink, 10-Jun-2021. [Online]. Available: https://link.springer.com/article/10.1007/s13347-021-00459-2.

[11] Meskys, E., Kalpokiene, J., Jurcys, P. and Liaudanskas, A., 2022. Regulating Deep Fakes: Legal and Ethical Considerations. [online]Papers.ssrn.com.
Available at: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3497144.

[12] Our approach to responsible AI at Microsoft. [Online]. Available: https://www.microsoft.com/en-us/ai/our-approach?activetab=pivot1:primaryr5.

[13] Ranjan, R et al, Comparative assessment of CNN architectures for classification of breast FNAC images

[14] Bianco, S., Cadène, R., Celona. N. and Paolo, 2018. Benchmark Analysis of Representative Deep Neural Network Architectures. Available: https://www.researchgate.net/publication/328509150_Benchmark_Analysis_of_Representative_Deep_Neural_Network_Architectures/citation/download

[15] Gragnaniello, Diego, et al. "Are GAN generated images easy to detect? A critical analysis of the state-of-the-art." 2021 IEEE International Conference on Multimedia and Expo (ICME). IEEE, 2021.