EVALUATION OF CONVOLUTIONAL NEURAL NETWORKS AGAINST DEEPFAKES USING TRANSFER LEARNING

A Paper
Submitted to the Graduate Faculty
of the
North Dakota State University
of Agriculture and Applied Science

By

Siddharth Krishan

In Partial Fulfillment of the Requirements
for the Degree of
MASTER OF SCIENCE

Major Department:
Computer Science

December 2022

Fargo, North Dakota

North Dakota State University
Graduate School

Title

EVALUATION OF CONVOLUTIONAL NEURAL NETWORK AGAINST DEEPFAKES USING TRANSFER LEARNING

By

Siddharth Krishan

The Supervisory Committee certifies that this ***disquisition*** complies with North Dakota State University's regulations and meets the accepted standards for the degree of

MASTER OF SCIENCE

SUPERVISORY COMMITTEE:

Dr. Jeremy Straub
Chair

Dr. Pratap Kotala

Dr. Qifeng Zhang

Approved:

November 17, 2023
Date

Dr. Simone Ludwig
Department Chair

ABSTRACT

The main objective of this paper is to evaluate ResNets, DenseNet, Inception and VGG, against deepfake images, to answer the question: How effectively these Convolutional Neural Network can distinguish between deepfake images and real images.

The dataset was acquired from FaceForensics++ and CelebA datasets for manipulated and unmanipulated images respectively. A custom script using Python and OpenCV was applied to create the final dataset for modelling.

Transfer learning is a technique of applying the learned features by a network to a similar approach. It is employed to save time and resources in training, as it does not require a large dataset to allow the network to learn effectively.

The Convolutional Neural Networks are tested against different deep fakes and the networks are evaluated using metrics like precision, recall, accuracy, loss, and f-1 score. It was observed that all the networks used in the experiment performed exceptionally well, but Inception network was slightly better than the other networks in separating the real and fake images.

ACKNOWLEDGMENTS

I want to present my sincere gratitude to my advisor Jeremy Straub for his motivation and for providing crucial insights and resources, which were very important for this paper. I also want to thank him for believing in me and allowing me to finish my Master of Science degree. I would also like to thank my committee members, Pratap Kotala and Qifeng Zhang, for their invaluable guidance and comments. I am also grateful to CCAST at NDSU for letting me use the resources needed for the training.

I also want to thank my parents, family, and friends for providing me with much-needed support during this journey. I am incredibly obliged to my manager Alissa Bosse at Bobcat, for her constant support and consideration with the flexibility to work. Finally, I want to thank my closest friend Aastha Singh Bhandari, who gave me faint hope when I could not see one.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

The first known attempt at altering an image can be found in one of the iconic portraits of U.S. President Abraham Lincoln, achieved by seamlessly combining two halves from different pictures [1]. Over time, the art of image manipulation and the creation of convincing altered visual representations have significantly improved. In recent times, there has been a leap in image manipulation capabilities, driven by the emergence of "deep fake" technology. This technology enables individuals without specialized training, to generate audio and video content featuring real people saying or doing things they never actually did [2]. A deepfake image created by a synthesis technique that exploits machine learning methods to combine and superimpose a face in an image or video onto target image or video [3]. These machine learning models can also be used to generate fake audio by using small samples of voices of real people. Models using deep learning, which are mostly generative adversarial networks (GANs) [4] and GAN variations [5]–[12] are deployed to create these altered realistic images. They make it very difficult for a person to distinguish them from original images. This will be discussed in detail in a later section.

Advancements in deep learning and the accessibility to cheap computing infrastructure, technical advancements have propelled the creation of deepfakes [13]. Researchers from the Imperial College in London and Samsung's AI research center demonstrated a technology, that can construct a singing or talking portrait by using a single photo and audio file [14]. Similarly, a Chinese smartphone application Zao allows users to swap their face into several clips from movies and T.V. shows using a single image of the person [15]. Another smartphone application FaceApp, let its users alter their facial attributes like hairstyle, gender or age[16]. In the past,

only experts could create these artifacts, but now a free accessible code repository allows any user to exploit this technology [17].

In 2017, an anonymous reddit user posted a video of celebrities with their face morphed onto the bodies of porn actors [18]. This can create a lot of trouble as not only celebrities, but the general public can also be affected by deepfakes. Deepfakes can be used to steal an individual's identity to commit fraud, or to create non-consensual pornographic content, manipulating someone's sexual identity [2]. One example is when a U.K.- based energy company's CEO was scammed out of $243,000 using a deepfake generated voice of his boss who asked to transfer funds in emergency [19], [20]. A short-lived app named 'DeepNude', which can turn an image of a fully clothed woman into an image of woman with no clothes [18], [21]. Revenge porn was used to end a representative's political career, and deepfakes adds a whole new dimension to revenge porn and provide a broader reach of population [17].

Deepfakes can negatively influence the political atmosphere by distorting democratic discourse, manipulating elections, undermining diplomacy, or jeopardizing national security [2]. This technology can be used to post fake videos of political leaders with malign content. A doctored video (not using deepfake technology) was posted online of Nancy Pelosi, the speaker of the US House of Representatives at the time, appearing to disgrace drunkenly through a speech. This video was shared by Donald Trump on Twitter, but it was quickly debunked [22]. Similarly, a deepfake footage of a sketch by Jimmy Fallon was posted for fun in YouTube under the title 'The Presidents'. This issue underscores the pressing concern that, given the capacity of digitally altered graphics to provoke significant disruptions, the advent of deepfake technology presents an even more formidable challenge to the political landscape, owing to its capability to produce exceptionally lifelike and convincing graphic content. Similarly, celebrities are at higher

risk, as their recordings are easily available in the public domain, thus making them easy targets of deepfakes [18], [23]. One example is a deepfake video with likeness to Mark Zuckerberg declaring “whoever controls the data, controls the future”, which was posted on the eve of his Congressional hearing on the matter of data privacy[24]. Hence, deepfakes can also be used to erode trust in institutions [2]. This technology can even be exploited to interfere with military operations by altering satellite images of the Earth [3], [25].

Deepfakes can be deployed for positive applications as well. It can be used in art and cinema to record dialogues or film scenes of actors that have been dead, even to portray a young actor as their old self or vice versa [26]. Another useful application can be found in the paper [27], where GANs are applied for customization of dental crown, which is labor and time demanding task, and requires human expertise.

Any technical advancement can be used for ethical or unethical purposes, and this also applies to deepfakes. As evident from above, presently the purpose of deepfakes for menacing purposes are much more impactful and far reached than their constructive value, thus detecting deepfakes is of importance at present times. So far, several approaches to detect deep fakes have been developed [13], [28]–[33]. Deepfake technology poses a significant threat to the integrity of information on social networking sites, where manipulated images and videos can be employed maliciously to disseminate false information and exert undue influence on a broad scale [34]. Vigilance and proactive measures are imperative for platforms and users alike to detect and mitigate the proliferation of deepfake content and thereby uphold the veracity of online discourse. In response to these challenges, Facebook has amended its policy to encompass the prohibition of deepfake videos featuring individuals uttering fabricated statements [35]. Tech giants like Facebook, Google, Amazon.com and Microsoft are stepping up to fight deep fakes, by

partnering with universities and running a Deepfake Detection Challenge [36]. Microsoft recently unveiled a deepfake detection tool, capable of detecting signs of alteration in the media, which can be undistinguishable to humans [37].

Not only tech giants are fighting against deepfakes, but government and military are also fighting against them. The United States Defense Advanced Research Projects Agency (DARPA) funded a project to develop method to counter deepfakes by detecting them automatically [38]. Recently, in the United States annual defense policy bill for 2020, $5 million authorized Intelligence Advanced Research Projects Activity (IARPA) to start a competition for detecting deepfakes automatically [39].

This paper discusses work that has been done for the development and detection of deepfakes and its improvement. I assess the effectiveness of the state-of-the-art Convolutional Neural Networks (CNNs) developed by others over the years against the deepfakes. These CNNs work near perfectly for ImageNet dataset, that has been a benchmark dataset for computer vision problems. ImageNet is a vast image dataset organized according to WordNet's hierarchy, aiming to provide 1000 images for each of its 100,000+ sets [40]. This word evaluates how well these CNNs perform against deepfakes and whether they maintain their precision as well as they do for the ImageNet dataset.

2. BACKGROUND

2.1. Related Work

The Generative Adversarial Network (GAN) is the core technology used for the generation of deepfakes. In GAN, two models are trained in parallel: one to generate data (G) and another to discriminate between real and generated data (D). G is trained to maximize the likelihood of D making errors, allowing for efficient training without Markov chains or approximate inference networks [4]. In this section, GAN improvements and other methods used to create and improve these realistic computer-synthesized images are explained. The Wasserstein GAN introduced in [9] , improves GANs training phase stability and eliminates problems like mode collapse. The Least Squares Generative Adversarial Networks (LSGANs) introduced in [5] , deals with the vanishing-gradient problem by using the least-squares loss function instead of the sigmoid cross-entropy loss function for the discriminator and minimizing the Pearson x^2 divergence. LSGANs can produce higher-quality images and are more stable than regular GANs. A style-based generator was proposed in [41], as an alternative architecture for GANs, capable of learning and separating high-level attributes. This style-based generator, enhances the generation of stochastic variations in features, enabling scale-specific control in synthesis. This facilitates the manipulation of attributes like color, texture, and style for more diverse and realistic image generation. As a result, the StyleGAN leads to linearity in latent space that helps to control the GAN synthesis.

Laplacian Pyramid Generative Adversarial Networks (LAPGANs) introduced in [11], can generate high-quality natural images using cascades of ConvNets within a Laplacian pyramid framework. A Laplacian pyramid [42] is a linear invertible representation of an image. It contains a set of band-pass images spread out in an octave sequence. The images are generated in

a course-to-fine fashion using GAN to train a separate ConvNet for each pyramid level. BigGANs introduced in [8], train GANs to generate images at higher resolution and observe the instabilities at such scale. To generate high-fidelity images, orthogonal regularization is applied to the generator, which reduces the variance of the generator's input. This aids in providing fine control over the trade-offs between sample fidelity and variety. Also, it was found that the stability in the training of generative models does not rely solely on the generator or discriminator but emerges from their interactions during adversarial training. While strong constraints on the discriminator can promote stability, it often sacrifices performance, leading to the strategy of allowing controlled collapse at later training stages to achieve improved results [8]. A new training methodology to increase fine details in the images is to progressively grow the generator and discriminator is presented in [10]. It starts from a lower resolution and adds new layers as training advances. This increases the level of variation in the generated images and discourages unhealthy competition between generator and discriminator, by starting with simple images and gradually increasing complexity. This gradual learning process maintains stability and balance during training and yield high quality results. Recycle -GAN which was proposed in [6], introduced a novel method for unsupervised video retargeting. This approach translates content from one domain to another while conserving the style native to the domain. It combines spatial and temporal information with the adversarial losses of content translation and style preservation. It also studies spatiotemporal constraints' advantages over spatial constraints. To stabilize the training of the discriminator, a weight normalization technique called Spectral normalization was proposed in [43]. Spectral Normalization works by normalizing the largest singular value of weight matrices in the generator and discriminator networks. This helps control the Lipschitz continuity of these networks, which is essential for stable training. This method

produces images of better quality, as it uses global regularization instead of local regularization on the discriminator. Spectral normalization of GANs generator were used in self-attention GAN (SAGAN) [7] to improve the training dynamics. The SAGAN technology is capable of attention-driven and long-range dependency modeling for tasks like image generation.

Cycle-GAN which was introduced in [44] is an unsupervised image-to image-translation that expands over the base GAN technology by not needing a large dataset of coupled images. It learns the mapping of the distribution from one image such that the distribution on the target image is undifferentiable. It uses an inverse mapping to introduce a cycle consistency loss, thus being named Cycle-GAN. Another image-to-image translation method was investigated in [45] which used conditional adversarial networks. The network introduced in this work can learn the mapping and the loss function used to train the mapping. Thus, their application to different settings with different loss functions is achievable without requiring users to hand-engineer mapping and loss functions. An image-based reenactment system proposed in [46] can replace an individual's inner face with that of a different individual while preserving the original facial expressions. The reenactment pipeline consists of both image retrieval and face transfer. Image retrieval uses temporal clustering of the frames for matching stability of appearance and motion, while for the face transfer, a 2D warping strategy was used to match the coherence of the user's identity.

Glow was introduced in [47] which uses invertible 1x1convolutions in the generative flow of the network. Glow demonstrated a significant improvement in log-likelihood. The technology is capable of efficient realistic-looking synthesis when trained on high resolution faces. A novel approach using space-time architecture to create deep video portraits was presented in [48]. It is based on a rendering-to-video translation network that converts a

sequence of simple computer graphics into photo realistic and coherent video. It transfers head pose, orientation, face expression, eye gaze from a source actor to a target actor. Many methods discussed above use large datasets of a single person to create a personalized head model. In [49], a system with few-shot capability meaning it can create head models with few images of a person was introduced. After a lengthy meta-learning process on large datasets, it can learn neural talking head models of previously unseen people as adversarial training problems. This system uses the person-specific parameters of both generators and discriminator to decrease training time.

2.2. Key Technologies

2.2.1. Convolutional Neural Network

This part aims to explain the parts that make a convolutional neural network and terminologies that will be used in later sections. Convolutional neural networks are commonly referred to as CNNs, and Yann LeCun is credited with the introduction of CNN into computer vision domain with his LeNet-5 network[50]. Multilayer neural networks trained with the back-propagation algorithm constitute the best example of a successful gradient-based learning technique. Given an appropriate network architecture, gradient-based learning algorithms is used to synthesize a complex decision surface that can classify high-dimensional patterns, such as handwritten characters, with minimal preprocessing. Different machine learning models like K-nearest neighbor (KNN), Principal Component Analysis (PCA), Support Vector Machines (SVM) were applied for handwritten character recognition and compared on a standard handwritten digit recognition task were reviewed. Convolutional neural networks, which are specifically designed to deal with the variability of two dimensional (2-D) shapes, are shown to outperform all other techniques in [51]. Real-life document recognition systems are composed of

multiple modules including field extraction, segmentation, recognition, and language modeling. A new learning paradigm, called graph transformer networks (GTN's), allows such multimodule systems to be trained globally using gradient-based methods to minimize an overall performance measure [51]. Two systems for online handwriting recognition are described and experiments demonstrate the advantage of global training, and the flexibility of graph transformer networks. A graph transformer network for reading a bank check is also described in [51]. It uses convolutional neural network character recognizers combined with global training techniques to provide record accuracy on business and personal checks as detailed in [51]. In CNNs, a convolution is performed on the image using filters, which are also called feature detectors or kernels. Different architectures use different sizes and numbers of filters. These filters when convolved with the input image or a feature map from a previous layer, resulting in a new feature map as shown in Figure 1. The size of the feature map is inversely proportional to the size and stride of filter. The depth of the feature map is directly proportional to the number of filters used. The stride is the number of pixels that are skipped over when the filter slides (both vertically and horizontally) over a feature map. An example with a filter size of 3x3 and stride of 2, is shown in Figure 2.

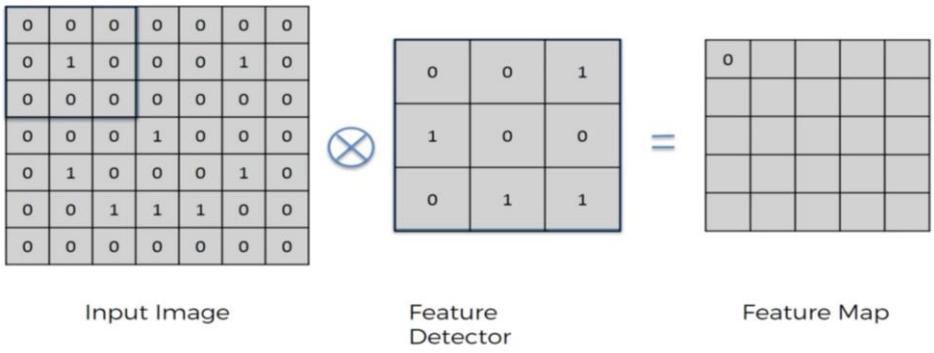

Figure 1: Convolution operation [52]

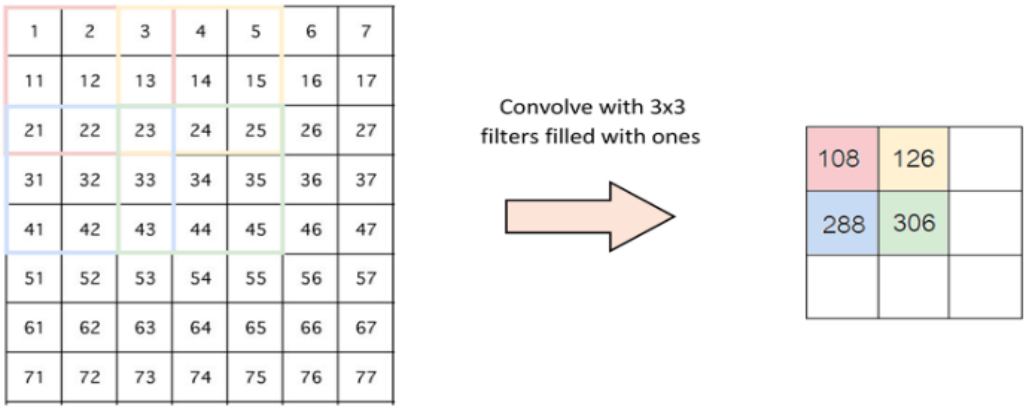

Figure 2: Example of convolutions with stride of 2 pixels [52]

To preserve the spatial information of images, CNNs use a technique called padding to counter the problem of filters not fitting a feature map perfectly [52]. The most common types of padding techniques are valid padding and same padding. The term valid padding means that no padding is used, and filters only use the information from the feature map. Valid padding reduces the dimension of feature maps, as pixels in the image are left out when the filter cannot fill it perfectly, as shown in Figure 3. To preserve the spatial dimensions of feature maps, 'same padding' is used. In same padding, a layer of zeros is added at the edges of the feature map to keep the output dimensions the same as the input dimensions [51], as shown in Figure 4.

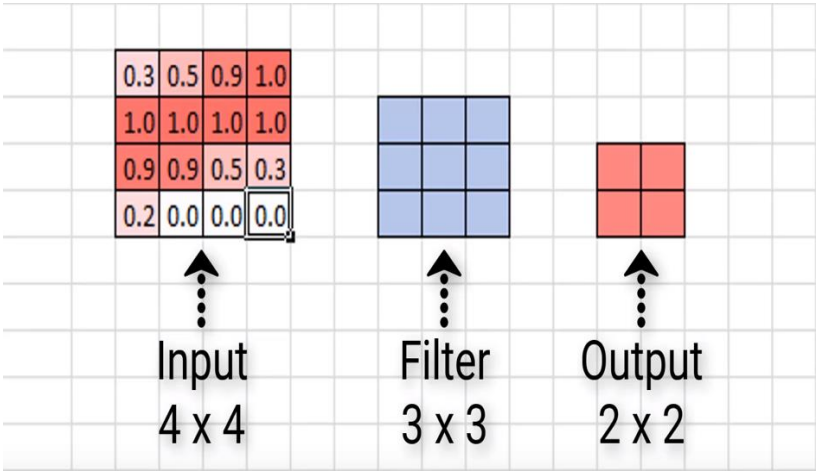

Figure 3: Example of valid padding [53]

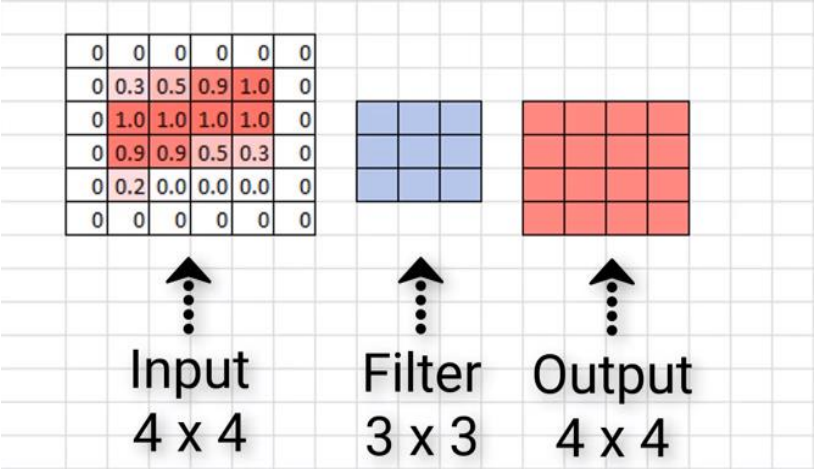

Figure 4: Example of same padding [53]

2.2.2. Pooling Layers

The pooling layers technique is also known as subsampling or down-sampling. Pooling is used to reduce the dimensionality of feature maps while preserving the information within them. This reduces the number of parameters while maintaining spatial invariance and makes training faster [52]. The most common pooling methods are max pooling, average pooling, and sum pooling. In max pooling (shown in Figure 5), the largest value in each stride of the filter is selected. In average pooling, the average of all the values in the stride of the filter is selected. In sum pooling, the sum is used instead of the average [52].

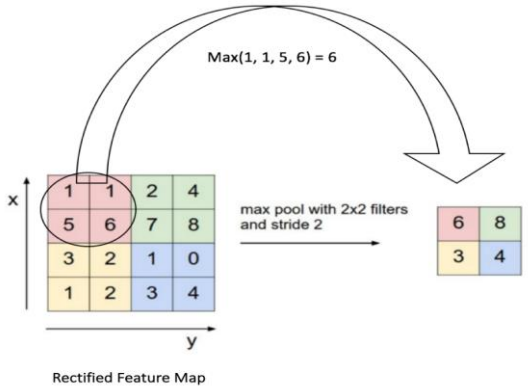

Figure 5: Max pooling [52]

2.2.3. Rectified Linear Unit (ReLU)

Rectified Linear Unit layers adds non-linearity into the ConvNet. There are several functions also to introduce non-linearity such as tanh, sigmoid or leaky ReLU, but ReLU is typically preferred [54]. ReLU helps in reducing the chance of vanishing gradient where the gradient decreases exponentially and the initial layers are not updated effectively, and faster training due to better convergence of the gradient. The ReLU function is defined as $f(x) = max(0, x)$ and is visualized in Figure 6 [52].

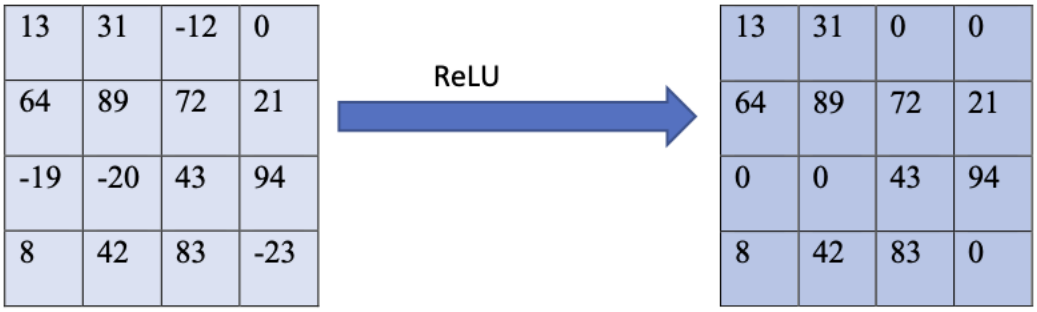

Figure 6: ReLU operation [52]

2.2.4. Fully Connected Layer

Fully connected layers which are also known as hidden layers are the layers traditionally used in multi-layer perceptron. In these layers, every neuron is connected to each neuron in the adjacent layers. These layers are supplied the feature maps after they are flattened [53]. Flattening is the process of transforming the multi-dimensional features maps into single dimensional structure of arrays.

The dropout technique [55] is often used in hidden layers to reduce over-fitting. In the dropout technique, any random neurons are probabilistically disabled from taking part in the learning process [56]. For example, if the probability selected is 0.8, then each time 0.2 or 20% of the neurons in the layer will be deactivated randomly. Due to this, every time the network is

trained it is a different network, like in ensemble learning. Because of this, the output neurons are not dependent on particular neurons, and every neuron is trained equally.

2.2.5. Transfer Learning

Humans can transfer the knowledge learnt on one task over to another related task [57]. Similarly, transfer learning in AI is a technique that uses a neural network trained on one task for another task in a related domain. For example, a neural network could be trained on for image recognition of a type of animal, and it could then be modified to be used for image recognition of a different animal [58].

The transfer of knowledge is done by storing the weights of the network trained on a particular task and using those weights on a different but related task [59]. Related tasks must be similar. For example, transfer learning could not be used to apply a network for a natural language processing problem to a network trained on a computer vision task or vice versa [60]. Freezing and unfreezing layers is a common practice while using transfer learning. Freezing a layer means that the layer will not update its parameters during the training phase and preserves its learning from previous training. Unfreezing a layer conversely, means that the layer is allowed to be modified and learn new features during the training phase [58].

In computer vision tasks, a ConvNet learns to detect edges in earlier layers of complex structures related to specific problem which is learned in the later layers. Transfer learning is used in these cases by freezing early and middle layers and then training the later layers for a new task. This speed up the training process, as there are limited layers to train, and it takes less time for backpropagation [59].

Recent developments in deep learning models, very high accuracies were achieved on several tasks, and these pre-trained models can be used to train on related tasks, making transfer

learning popular [60]. Deep learning requires a lot of time, resources and data to train from scratch. However, using transfer learning the model can be deployed from being trained on different task using limited data. In traditional learning, the model is trained with large amount of data and each model needs to be trained from scratch, which requires increased training time. The difference between traditional learning and transfer learning is shown in Figure 7.

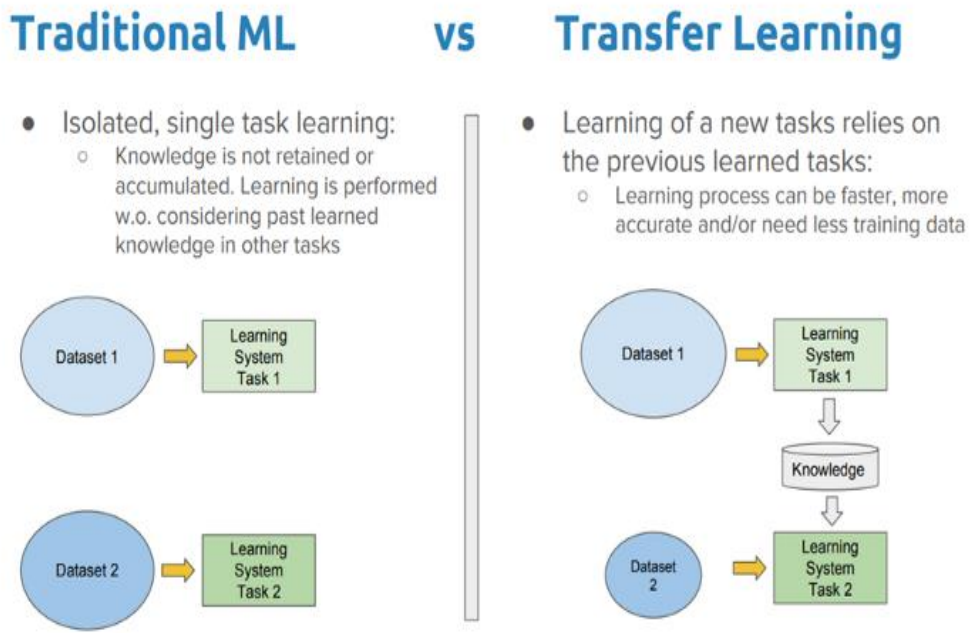

Figure 7: Traditional learning vs Transfer learning [57]

2.3. AlexNet

AlexNet network was proposed by Alex Krizhevsky in [61], and significantly increased the achieved classification accuracy on ImageNet dataset. It consists of five convolution layers and three hidden layers, including the output layer. The architecture for AlexNet is shown in Figure 8.

AlexNet implemented a similar but large architecture (about 60 million parameters) as compared to LeNet introduced by Yann Le Cun in [51]. It has more filters in each layer and five convolution layers as compared to two convolution layers in LeNet. AlexNet introduced some

new approaches to CNNs, which made it perform better on very large datasets. The Rectified Linear Units (ReLU) technique [54] was used for non-linearity instead of the tanh function. This made training of the network considerably faster as shown in Figure 9.

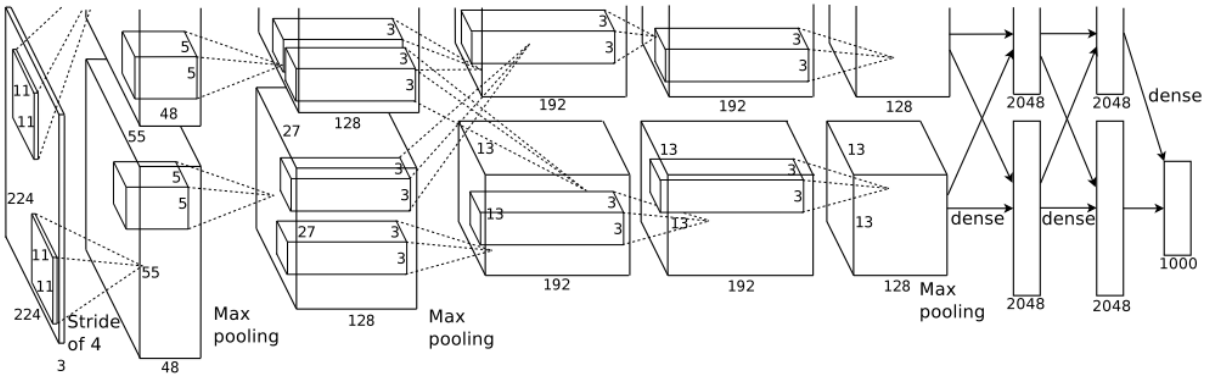

Figure 8: Architecture of AlexNet with delineation of responsibilities between GPU's [61]

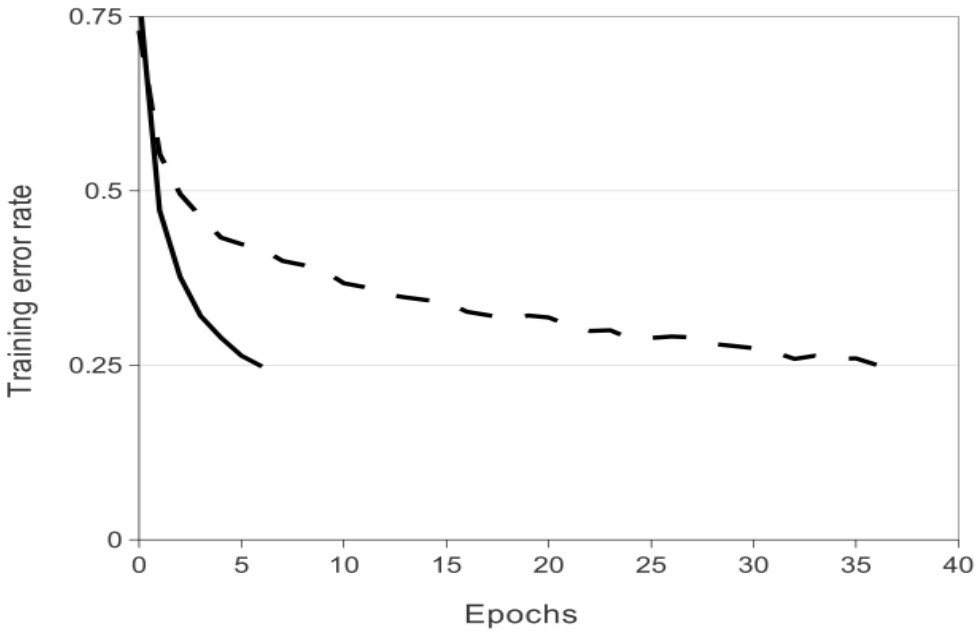

Figure 9: CNN with ReLU vs tanh (dotted line); ReLU reaches 25% error rate 6 times faster [61]

Since the network was considerably larger than LeNet, it was trained using two GPUs in parallel. Each GPU was responsible for half of the neurons and communicated only in specific layers. Some layers took all the feature maps from the previous layer as an input, while other

layers only use feature maps as an input that are available on same GPU [61]. Alex Net also used local response normalization (LRN) and overlapping pooling [61]. In AlexNet, inter-channel LRN was implemented, and normalization was performed across the channels of each pixel coordinates, and across the depth of the channels. Traditionally, pooling layers without overlapping (for a kernel size of *k* x *k*, the stride *(s)* will be greater than or equal to *k)* were used. AlexNet implemented overlapping pooling layers, where the stride *(s)* was less than the size of the kernel *(k).* With each stride, a slice of the previous stride is also preserved, thus conserving information. Additionally, techniques like dropout and data augmentation were introduced to combat problem of overfitting. Data augmentation was performed using principle component analysis (PCA) on RGB values to alter the intensity of the image and to extract random patches from the image and their horizontal reflections. The data augmentation technique led to the increase in the training set by a factor of 2048 [61].

2.4. VGG

The VGG architecture was presented from Visual Geometry Group, University of Oxford and further increased the depth of the network to 19 layers in [62] and improved over AlexNet [61] by using a small convolution filters of size 3x3 with a stride fixed at 1 pixel. This is the smallest size that preserves the notion of center, left/right, and up/down. They also utilized 1x1 convolution filter in one of their configurations and same padding of 1 pixel was used to maintain a uniform spatial resolution. Each layer followed stacks of similar convolutional layers. This is followed by max pooling layer using a 2 x 2 window, with a stride of 2.

After each max-pooling step, the total number of filters (with the width of the convolutional layers) increased by a factor of 2, starting from 64 in the first stack to 512 in the last 2 stacks of convolutional layers. VGG did not use local response normalization, as it did not

improve performance of their network. All configurations follow the same generic design with a different depth of convolutional layers and the same depth of fully connected (FC) layers. The first two FC layers had 4096 units using the ReLU activation function. The last FC layer had 1000 units using the SoftMax activation function. This configuration is shown in Figure 10. For a single test scale, it achieved a top-1 error of 25.5% and a top-5 error of 8.0%. For multi test scales, the top-1 error was 24.8% and top-5 error was 7.5%.

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Figure 10: Different VGG configurations [62]

2.5. Inception or GoogLeNet

Inception architecture demonstrated a new state of the art performance for classification and detection at ImageNet Large Scale Visual Recognition Challenge 2014 (ILSVRC14) [63]. The core idea behind this architecture was to use filters of different sizes that function on a

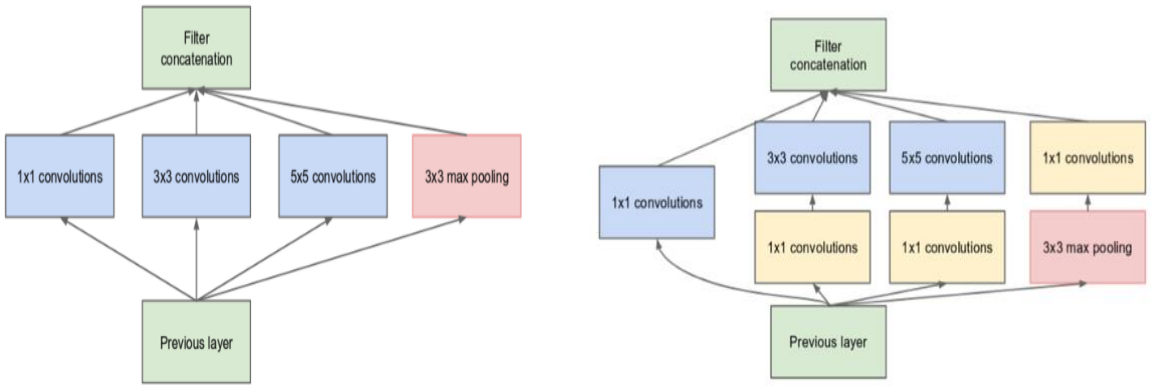

Figure 11: Inception module, naive version(left) and with dimensionality reduction(right) [63]

similar level, making the network wider as shown in the Figure 11. An additional 3x3 max pooling layer using ‘same’ padding was added at the same level of those convolutional features.

Deep convolutional neural networks are computationally expensive and this network is both deeper and wider; however, it still keeps the computational cost constant. To achieve this, 1x1 convolutions were used before 3x3 and 5x5 convolutions to limit the number of channels as depicted in Figure 12. Using a 1x1 size convolution layer, the computable parameters are reduced to about one-tenth of the parameters that would be present without 1x1 size convolutions. Using a dimension reduction module, the architecture was popularly known as GoogLeNet (named to pay homage to LeNet by Yann LeCun). GoogLeNet has nine inception modules that are stacked linearly. It uses a global average pooling of size 7x7 at the end of the last inception module and before the fully connected layer. The fully connected layer utilizes the SoftMax activation function.

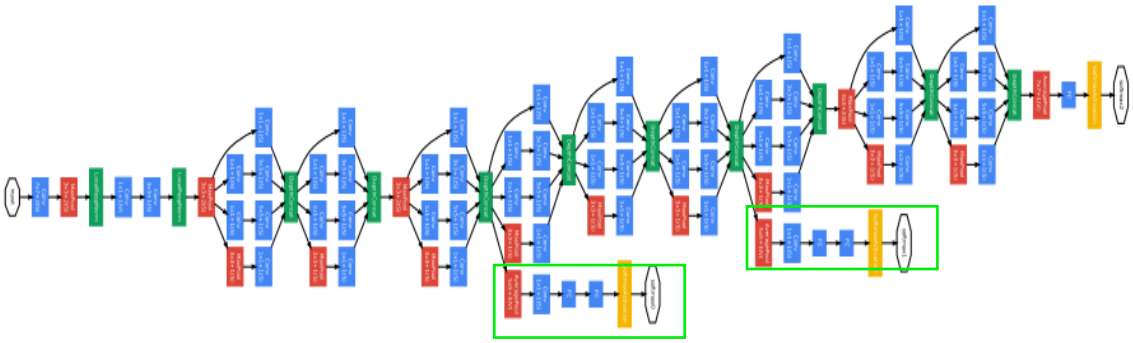

Figure 12: Complete GoogLeNet [63]

The blocks contained inside the green rectangle boxes as seen in Figure 12 are auxiliary classifiers, which are introduced to avoid the vanishing gradient problem. These auxiliary classifiers add global average pooling and followed by a 1x1 convolutions filters for dimensionality reduction. Then a fully connected layer with 1024 units (70% dropout rate) and a ReLU activation function is used. The last fully connected layer in this auxiliary classifier is the same as the one used at the end of the last inception module.

Further improvements were made to the Inception/GoogLeNet architecture which were termed as Inception v2 and Inception v3. In these updates, the 5x5 convolutions were broken into two stages of 3x3 to remove the representational bottleneck [64]. Also, the NxN filters were implemented as a combination of a 1xN and a Nx1 convolutions. Additionally, batch normalization [65] was used for auxiliary classifiers, and label smoothing was used to prevent overfitting [64].

2.6. Residual Networks (ResNets)

In the early stages of deep learning, it was believed that the network will perform better as its depth is increased, since it can explore a greater number of parameters. But it was observed that after a certain depth the performance starts to decrease. This decrease in performance occurs because of the vanishing gradient and degradation problems.

The vanishing gradient problem happens when a network is large and the gradients from the latter part of the network become very small and do not affect the weights at the initial layers, thus making the extra depth of the network useless. The degradation problem is when by increasing the depth of a network, the performance actually decreases rather than increasing or is similar to a shallower network.

The vanishing gradient and the degradation problems were resolved by introducing residual block as shown in Figure 13. These residual blocks used 'identity mapping' or 'skip connections'. Skip connections helped with the vanishing gradient problem, as they allowed the gradient to travel back to the initial layers through these connections. Using an identity layer aids with the degradation problem, as this layer learns the identity mapping in addition to the direct mapping. The skip connections do not add any extra parameters, thus it does not increase computational cost.

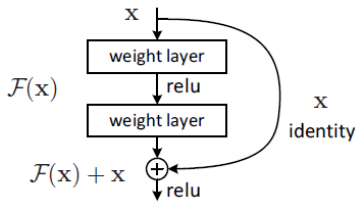

Figure 13: A single residual block with identity mapping [66]

The ResNet-34 network presented in [66], is depicted in the top part of Figure 14. The notable thing here is the dotted skip connections, which represents scaling down of the dimensions by increasing the strides of convolutions. It also uses average pooling at the end of the last convolution layer. Using these residual blocks, networks with more than 100 layers have been formed. ResNet-101 and ResNet-152 are networks with 101 and 152 layers respectively.

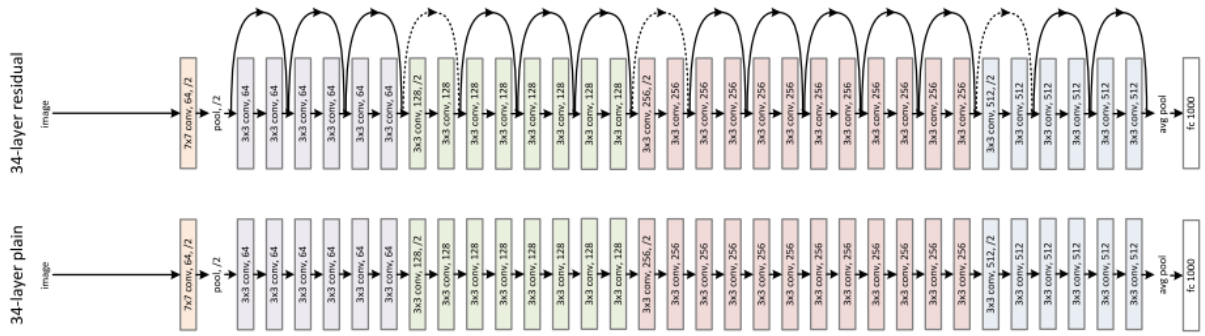

Figure 14: TOP**:** 34-layer network with residual blocks. BOTTOM**:** 34-layer plain network [66]

2.7. DenseNet

Densely Connected Convolutional Networks introduced in [67], also known as DenseNet, involve every layer receiveing features from all previous layers, and passing on their feature maps to all of their successor layers. As a result, this network gains the advantage of feature reuse and requires fewer parameters to learn, the network does not need to re-learn previously absorbed features, because of this the information flow between layers was improved and also prevent the vanishing gradient problem from occurring.

An example can be seen in Figure 15, where the growth rate (k) is set to 4. This means that each layer contributes four feature maps to the subsequent layers. It uses the concatenation of filters from each layer to other layers. Hence, in a dense block network with k growth rate and N number of layers, each layer will have k_0+k*N number of feature maps present, where k_0 is the number of channels in the input image. Another benefit of DenseNet is memory and computational efficiency as at each layer, the number of features maps to be learned remains the same, but each layer can learn collectively using previous features maps as well.

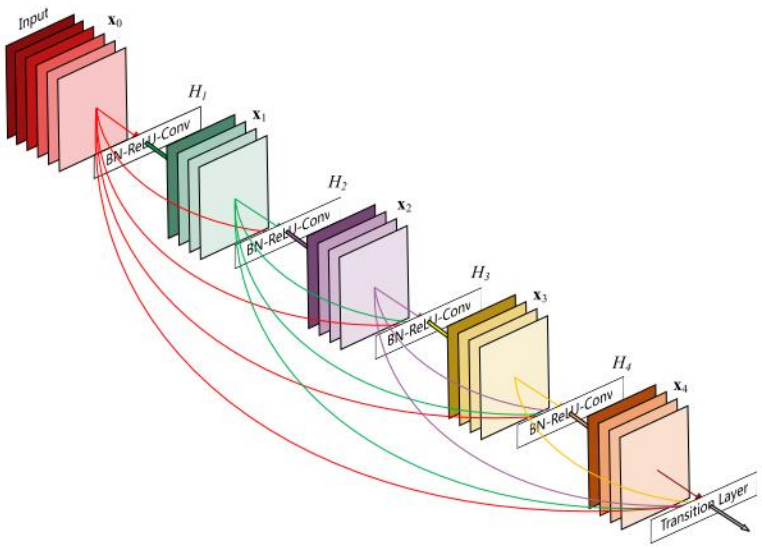

Figure 15: A 5-layer dense net with growth rate of k=4 [67]

Since down-sampling layers to change the size of feature maps is important, and concatenation is not possible for feature maps of different sizes, dense blocks with different sizes of feature-maps were used. In order to decrease the size of the feature maps and connect dense blocks, DenseNet uses transition layers which are comprised of a 1x1 convolution layer followed by a 2x2 average pooling layer. The full deep DenseNet is shown in Figure 16.

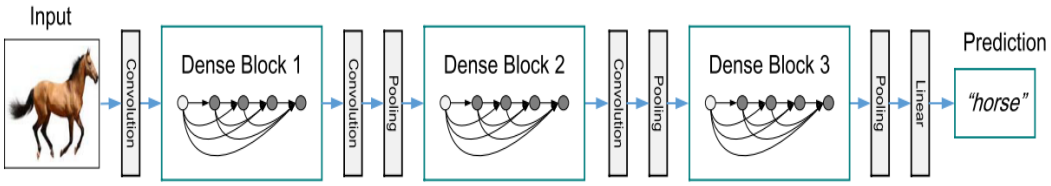

Figure 16: A deep DenseNet with three dense blocks [67]

Different configurations of DenseNet are shown in Figure 17. A DenseNet-121 architecture uses a growth rate (k) of 32 (each layer in the dense block adds 32 new feature-maps).

Layers	Output Size	DenseNet-121($k = 32$)	DenseNet-169($k = 32$)	DenseNet-201($k = 32$)	DenseNet-161($k = 48$)
Convolution	112 × 112	7 × 7 conv, stride 2			
Pooling	56 × 56	3 × 3 max pool, stride 2			
Dense Block (1)	56 × 56	[1 × 1 conv; 3 × 3 conv] × 6	[1 × 1 conv; 3 × 3 conv] × 6	[1 × 1 conv; 3 × 3 conv] × 6	[1 × 1 conv; 3 × 3 conv] × 6
Transition Layer (1)	56 × 56	1 × 1 conv			
	28 × 28	2 × 2 average pool, stride 2			
Dense Block (2)	28 × 28	[1 × 1 conv; 3 × 3 conv] × 12	[1 × 1 conv; 3 × 3 conv] × 12	[1 × 1 conv; 3 × 3 conv] × 12	[1 × 1 conv; 3 × 3 conv] × 12
Transition Layer (2)	28 × 28	1 × 1 conv			
	14 × 14	2 × 2 average pool, stride 2			
Dense Block (3)	14 × 14	[1 × 1 conv; 3 × 3 conv] × 24	[1 × 1 conv; 3 × 3 conv] × 32	[1 × 1 conv; 3 × 3 conv] × 48	[1 × 1 conv; 3 × 3 conv] × 36
Transition Layer (3)	14 × 14	1 × 1 conv			
	7 × 7	2 × 2 average pool, stride 2			
Dense Block (4)	7 × 7	[1 × 1 conv; 3 × 3 conv] × 16	[1 × 1 conv; 3 × 3 conv] × 32	[1 × 1 conv; 3 × 3 conv] × 32	[1 × 1 conv; 3 × 3 conv] × 24
Classification Layer	1 × 1	7 × 7 global average pool			
		1000D fully-connected, softmax			

Figure 17: DenseNet architectures for ImageNet [67].

3. EXPERIMENTAL SETUP

3.1. Dataset

The Face Forensics++ dataset, consisting of 1000 original video sequences manipulated with four automated face manipulation methods (Deepfakes, Face2Face, Face Swap, and Neural Textures). The dataset used for this work contains a total of 977 YouTube videos and all videos contain a trackable frontal face without obstructions. The dataset has videos in three different compression rates: c0/raw (compression rate is 0), uncompressed; c23 (high quality with a compression rate of 23); and c40 (low quality with a compression rate of 40). This work has used the c23 videos, as this quality level provides a balance between memory size and video quality (to provide information in the image for better feature identification). This work test the deep fakes manipulation methods using the ConvNets.

The experimental process is as follows. First, the images for the deep fake category are extracted from the videos of FaceForensics++ dataset using OpenCV and Python on all four types of manipulation methods. Every two seconds, one frame is extracted and the Haar-Cascades classifier (based on Viola and Jones face detection technique introduces in [68]) is applied to detect the faces in the captured frame. The area with the detected face area is cropped and saved as an image. The extracted images are curated to remove the false positive images (images that do not contain faces and still captured) and approximately 10,000 images are produced. A sample of these images are shown in Figure 18. These images are then randomly split into two scoops using a 80:20 ratio (i.e., 8000 images for training and 2000 images for testing). For the experiment, only the face part of the image is used. This allows the CNN to only learn the features of the face and better distinguish between deepfakes and unmanipulated images.

Second, the images for unmanipulated (non-fake) category used for this work were sourced from the CelebFaces Attributes Dataset (CelebA). CelebA is a large-scale face attributes dataset with more than 200,000 celebrity images as shown in Figure 19. This experiment uses the aligned and cropped subset of the Celeb-A dataset, which is closer to has been extracted for use as the manipulated (fake) image dataset. For this dataset also, 8000 images were randomly selected for training and 2000 images were randomly selected for test purposes. The unmanipulated images were downloaded from a separate dataset because it allows the model to learn from different sets of images, not just the original faces to which the deepfake algorithm was applied. This demonstrates the potential efficacy of this technique for other uses. It also improves the learning capability of the CNNs by increasing their robustness and preventing overfitting in the model.

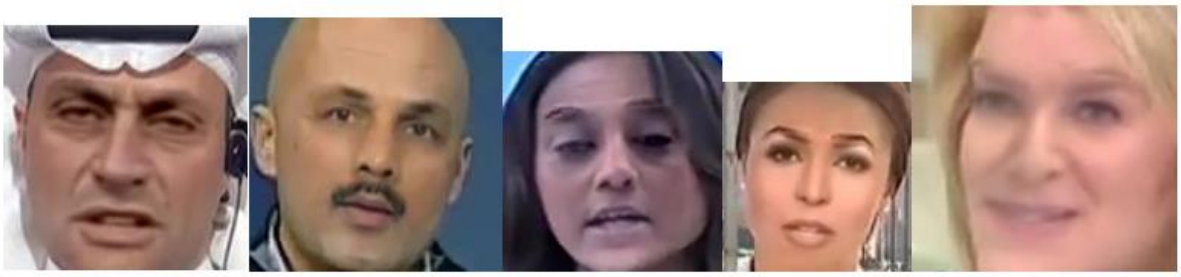

Figure 18: Example of DeepFakes Images [69]

Figure 19: Real Images from the dataset Celeb-A [70]

3.2. Approach

Both sets of matrices lies with values for each pixel ranging from 0 to 255 for grayscale images, and from 0 to 255 for each channel (red, green, and blue). The R-G-B channels are stacked over each other, and their values are combined, giving the pixel color. The values of three channels for each pixel are specified as (R, G, B). A (0,0,0) for a pixel is the color white, while (255,255,255) is the color black. Different combination of values produces different colors.

The pre-trained ConvNets models (inception, ResNet. DenseNet, and VGG) used in this experiment are state-of-the-art architecture, that has been shown to produce highest level of accuracy with ImageNet data (standard for comparing ConvNets performance) as described previously. The motivation of this experiment is to assess the ability of these networks to distinguish between real and fake images. Specifically, it seeks to determine if they can distinguish the features of unmanipulated and manipulated faces. The model is run as a binary classifier with possible outputs of fake (as 0) or real images (as 1) using different ConvNets to evaluate their performance.

The training of the models used for this work has been expedited using transfer learning. These models were downloaded from the Keras[71] library including their respective pre-trained weights on the ImageNet dataset. An image size of 224 x 224 was selected as the input to the model. Since the model was trained using the custom dataset described previously, only the convolution layers of the model are kept. The fully connected layer of the pre-trained model is not used. This creates the base model to which other parts of the model will be added.

The earlier layers of the base model were frozen and the later layers were kept unfrozen to allow the model to learn from the features of the new dataset. A pooling layer was added on

top of the base model. In this case, a global average pooling layer was applied. This layer takes the average of the output feature maps provided by the base model and converts it into a 1-D array. This 1-D array is then supplied to the next fully connected layer which, in this case, is the prediction/final layer. Since the model was developed for binary classification, the prediction layer consists of one neuron with sigmoid activation. The model uses binary cross-entropy to calculate its loss function. "Adam or adaptive moment estimation"[72] was used as the optimizer for updating weights.

The ImageDataGenerator function from the Keras[71] library was used to augment the data by rotating, zooming, and flipping the images vertically and horizontally, improving the model's performance in recognizing the images. The training and test set were divided into batches of 16. The model updated its weights after it has processed each batch. The number of batches was equal to the total number of samples in the dataset divided by the batch size.

$$\#\ \text{of Batches} = \frac{\text{Total \# of Samples}}{\text{Batch Size}} \quad \text{(Eq. 1)}$$

The final model was trained with the epoch value set as ten. The accuracy metric was used to evaluate the model performance during training phase. The accuracy of the model is the ratio of number of correct predictions to the total number of predictions.

$$Accuracy = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}} \quad \text{(Eq. 2)}$$

VGG, Inception v3, ResNets, and DenseNet were employed as base models to run on the data and to evaluate their performance metrics using a confusion matrix, as accuracy alone is not an absolute measurement of a model's performance.

A confusion matrix is used to calculate the TP (True Positive), FP (False Positive), TN (True Negative), and FN (False Negative). A confusion matrix is an estimation tool providing a summary in a tabular form of correct and incorrect predictions given by the model[73], as shown in Figure 20. An example to show the use of confusion matrix in a cancer detection model is also shown in Figure 20.

n=165	**Predicted: NO**	**Predicted: YES**	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

Predicted Class	Actual Class
No Cancer	No Cancer
Has Cancer	Has Cancer
No Cancer	No Cancer
No Cancer	No Cancer
No Cancer	Has Cancer
Has Cancer	Has Cancer
No Cancer	No Cancer
Has Cancer	No Cancer
No Cancer	No Cancer
No Cancer	No Cancer
Has Cancer	No Cancer
No Cancer	No Cancer

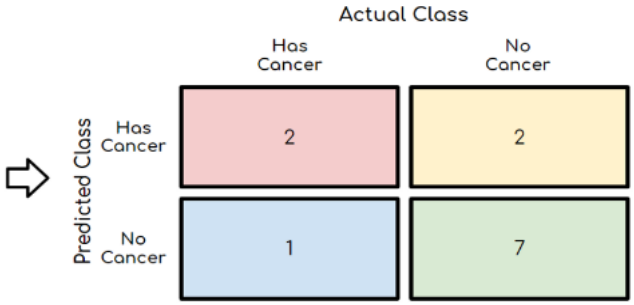

Figure 20: TOP: A confusion matrix [74], BOTTOM: An example of a confusion matrix for Cancer detection [75]

The recall provides us with the performance metric is defined as the actual positives that were identified correctly[74]. False positive rate (FPR) is the percentage of true negatives incorrectly predicted as positives[76]. Precision identifies the number of positive identifications that were correct[76]. The formulas for recall, precision and FPR are as follows:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

(Eq. 3)

$$\text{FPR} = \frac{\text{FP}}{\text{TN} + \text{FP}}$$

(Eq. 4)

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

(Eq. 5)

An F1 score is also calculated for each model. The F1 score can range between 0 and 1, where 1 is a perfect score (signifying the best model performance)[75]. F1 can be calculated using the formulae:

$$\text{F1 score} = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}}$$

(Eq. 6)

4. RESULTS

This chapter presents the results of the experiment performed on the deepfake images using the ConvNet models ResNets, DenseNet, Inception, and VGG16 as classification models using transfer learning and trained weights from the ImageNet dataset. As previously discussed, all these models were chosen as these ConvNets have consistently achieved high performance in image classification tasks. The dataset used in the research is balanced, i.e., both classes have an equal number of images. Details on the performance metrics like accuracy, validation accuracy, precision, recall, and f-1 score are also captured in this chapter for all these models. Likewise, as stated in the previous chapter, these metrics are used to evaluate a model as a classifier, giving us insights into the ability of the model to distinguish and precisely classify the images. For each metric, the higher the score, the better the model performs in classifying the images.

4.1. Deepfakes

On the Deepfakes data, InceptionV3 gets an accuracy of 99.96% on the training set and 99.92% on the validation set. The DenseNet achieves an accuracy of 99.91% on the training set, while it achieves an accuracy of 99.97% on the validation set. ResNets achieves an accuracy of 99.93% on the training set and 99.80% on the validation set. Compared to other models, the VGG achieves the lowest accuracy on both training and validation set, achieving 99.80% and 99.44%, respectively.

The InceptionV3 performs the best on the training set, while DenseNet achieves the best accuracy on the validation set. All the models perform exceptionally well on the deepfake dataset and attain high scores of 0.99 for precision, recall, and f-1 score. Therefore, all the models effectively separated and classified the real and fake images apart on the Deepfake type of image manipulation.

4.2. FaceShifter

The Inception ConvNet model again scores the highest accuracy of 99.84% on the training set and an accuracy of 99.92% on the validation set. ResNet achieves 99.79% accuracy on the training set but outperforms other models on the validation set with an accuracy of 99.95%. In comparison, VGG achieves the lowest accuracy of 99.45% on the training set, and validation set accuracy of 99.75%, while DenseNet achieves the lowest accuracy of 99.62% against the validation set of FaceShifter manipulation.

For the FaceShifter dataset, all the models perform exceptionally well and attain high scores of 0.99 for the precision, recall, and f-1 score. Hence, all the models are also effective in separating and classifying the real and fake images apart on this type of image manipulation.

4.3. Face2Face

Inception again leads the other models in accuracy on the training set, while VGG16 has the lowest accuracy on the training set. In the case of the validation set, ResNets performs the best in the FaceShifter dataset, while DenseNet achieves the least accuracy.

The accuracy of Inception on the training set and validation set is 99.91% and 99.56%, respectively. DenseNet achieves an accuracy of 99.80% on the training set and 98.41% on the validation set. ResNets achieves 99.79% on training set accuracy, which is just below DenseNet by 0.01% but outclasses other models with a validation set accuracy of 99.85%. VGG reaches an accuracy of 99.35% and 99.70% on the training and validation set, respectively.

Again, the models can perform exceptionally well on the dataset. They attain high scores of 0.99 on precision, recall, and f-1 score. Thus, the models can separate and classify the image manipulated using the Face2Face method.

4.4. FaceSwap

ResNets achieves an accuracy of 99.90% on the training set, and on the validation set, it attains an accuracy of 99.94%. Inception also performs almost equally well on the training set, with an accuracy of 99.89%, but only 99.75% on the validation set. DenseNet has an accuracy of 99.81% on the training set but performs equally well as ResNets on the validation set with an accuracy of 99.94%. VGG reaches an accuracy of 98.83% on the training set and 99.80% accuracy on the validation set.

ResNets, Inception, and DenseNet all score the precision, recall, and f-1 score of 0.99 each. Consequently, they can easily and effectively distinguish real and fake images. VGG16 obtain a precision of 0.92 and recall of 0.96 on the FaceSwap data, thus having an f-1 score of 0.94. The f-1 score of 0.94 is still very high for VGG, which means that it can separate the real and fake images with high confidence, but compared to other models in this research, it is slightly less.

4.5. Comparison of the Models

Table 1: Comparison of ConvNets against different type of image manipulation techniques

Type of Fake	CNN	Training Accuracy	Validation Accuracy	Precision	Recall	F-1 Score
Deepfakes	Inception	99.96%	99.92%	0.99	0.99	0.99
	ResNet	99.93%	99.80%	0.99	0.99	0.99
	DenseNet	99.91%	99.97%	0.99	0.99	0.99
	VGG16	99.80%	99.44%	0.99	0.99	0.99
Faceshifter	Inception	99.84%	99.92%	0.99	0.99	0.99
	ResNet	99.80%	99.62%	0.99	0.99	0.99
	DenseNet	99.79%	99.95%	0.99	0.99	0.99
	VGG16	99.45%	99.75%	0.99	0.99	0.99
Face2Face	Inception	99.91%	99.56%	0.99	0.99	0.99
	ResNet	99.78%	98.41%	0.99	0.99	0.99
	DenseNet	99.69%	99.85%	0.99	0.99	0.99
	VGG16	99.35%	99.70%	0.99	0.99	0.99
FaceSwap	Inception	99.89%	99.75%	0.99	0.99	0.99
	ResNet	99.90%	99.94%	0.99	0.99	0.99
	DenseNet	99.81%	99.94%	0.99	0.99	0.99
	VGG16	98.83%	99.80%	0.92	0.96	0.94

As shown in Table 1, all the models perform exceptionally well in classifying different kinds of fake images from authentic ones. As evident from the table above, all models have similar training set, and validation set accuracy; it shows that the models do not exhibit overfitting to the data, i.e., they can recognize the unobserved images and the images from the training set.

Inception outclasses other models in training set accuracy on all kinds of image manipulation techniques, except for FaceSwap, in which ResNets performs marginally better than Inception. VGG has the lowest accuracy on the training set in all types of fake images. Overall, ResNets and Inception perform best against all four types of manipulation used in this research, as they provide consistent and balanced performance on both the training and validation set. DenseNet performs better than VGG16 on the training and validation set but is not as consistent as ResNets and Inception.

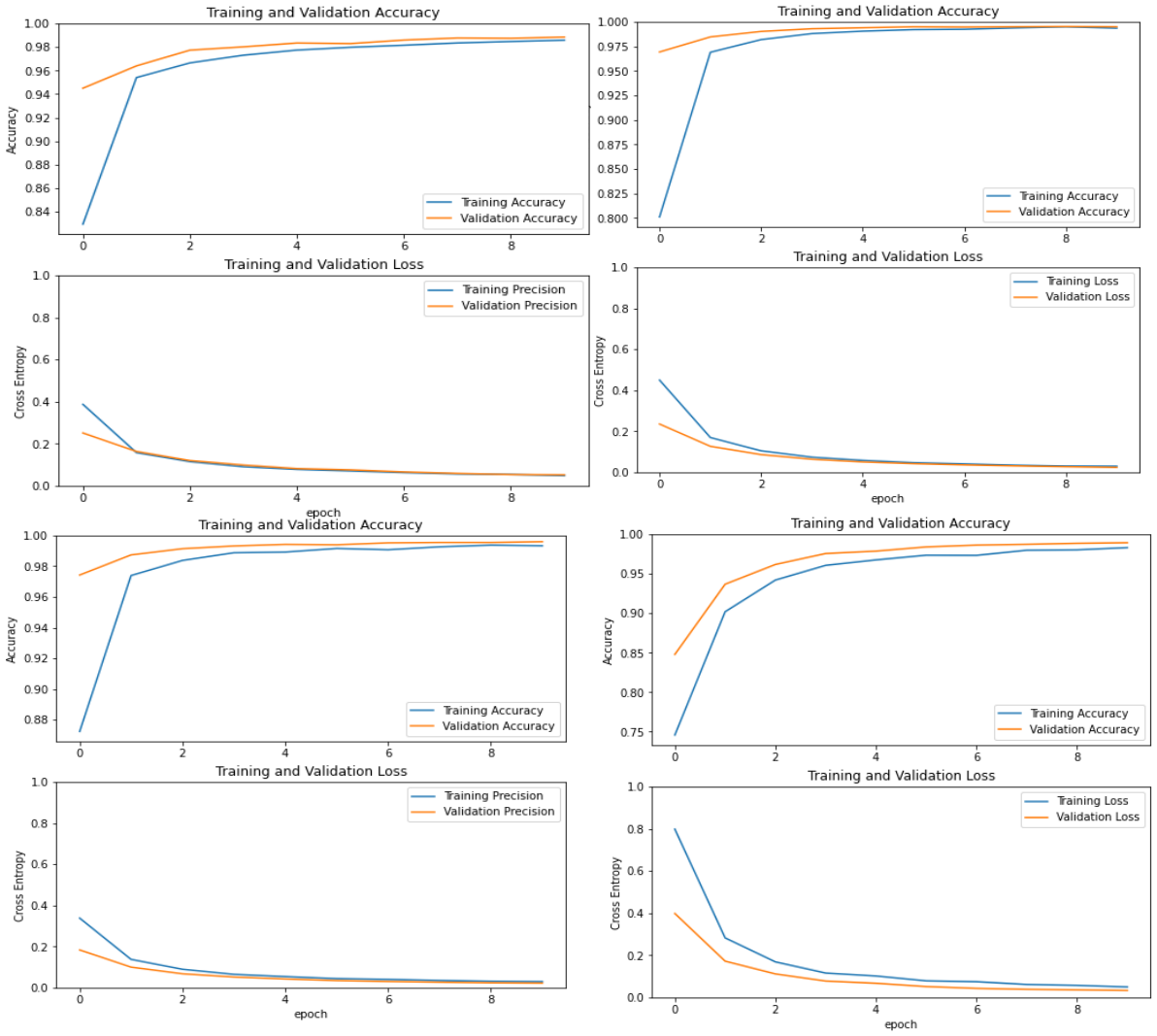

Figure 21: Accuracy and Loss of different models against deepfakes. DenseNet (top left), ResNets (top right), Inception (bottom left), and VGG16(bottom right)

Figure 21 shows the accuracy and loss of different models against deepfake images. ResNets and DenseNets are able to get to the top accuracy quickly on the validation set than Inception and VGG models. However, ResNets and Inception get better accuracy than DenseNet, and ResNets reach higher accuracy than Inception. VGG has the lowest initial accuracy on the validation set compared to other models and reaches its top accuracy gradually. VGG accuracy is like DenseNet. Every model has identical accuracy on both training set and validation set, hence depicting the absence of dropout issue.

Compared to the performance on the ImageNet dataset of these models, VGG16 has the lowest accuracy of 90.10%[71]. ResNet152v2 has the highest accuracy of 94.20%[71], followed by InceptionV3, with an accuracy of 93.70%[71] a DenseNet, with an accuracy of 93.2%[71]. However, on our dataset, all the models have achieved high accuracy and precision, recall, and f-1 score.

5. DISCUSSION AND CONCLUSION

Every development in science and technology can be used for ethical or unethical tasks. Deepfake is no different; it can be used in the cinema and music industries for various ethical tasks, like creating the voice of a dead artist or generating new music. The other side of deepfake is much more notorious and causes severe implications to society, like creating a fake video of a leader saying something nefarious. With the growth in technology, the accessibility to the software used to create deepfakes has increased, and any simple person can use it to create deepfakes. Social media companies are more prone to the unethical use of deepfakes as this content is mainly circulated through these networks. So, identifying and preventing the circulation of fake content is extremely important.

This research aimed to test the performance of the top performing ConvNet and check their effectiveness in classifying a fake image using transfer learning. Based on the metrics obtained after running the models, it can be concluded that these pre-trained models can effectively identify fake and real images. This study also indicates that these models can effectively understand the difference between manipulated and unmanipulated facial features.

The real and fake image datasets were taken from different sources to increase robustness, and only the face part from the image was used for training and testing the models. It allowed the model to learn the features of the face effectively. While limited data for the fake images makes the model's training challenging, the transfer-learning approach proved effective, and pre-trained models on ImageNet weights resulted in effective learning of the models. The models can correctly predict the image and have exceptionally good precision, recall, and f-1 scores.

The deepfake images included in the dataset come from a very raw input, i.e., performing the deepfake manipulation at a lower level which exhibits the manipulation overlay of features of one person to another, which is easy to recognize. The approach was effective in learning the features of fake images that exhibit the exploitation of visual artifacts. Overall, more research is needed to understand deepfake technology better and expand its scope, as increasing deep-learning updates and new development techniques for more realistic-looking deep fake images can be more challenging to identify. The approach taken in this research has shown to be effective with a limited amount of data and can be further applied to higher-fidelity images with more data. Also, rather than using transfer learning on high-fidelity fake images, the existing ConvNet architectures can be trained from scratch, which is more time-consuming and expensive. The approach used in this study shows that the existing networks can be used to identify the

The future work and progression of this study can be to test these models on much more real-looking fake images created using Generative Adversarial Networks and to create a custom pipeline that extracts visual artifacts of the faces to exploit the visual discrepancies produced by the fake images.

REFERENCES

[1] L. Moran, “Iconic Abraham Lincoln portrait revealed to be TWO pictures stitched together | Daily Mail Online,” dailymail.co.uk. Accessed: Oct. 09, 2020. [Online]. Available: https://www.dailymail.co.uk/news/article-2107109/Iconic-Abraham-Lincoln-portrait-revealed-TWO-pictures-stitched-together.html

[2] B. Chesney and D. Citron, “Deep fakes: A looming challenge for privacy, democracy, and national security,” *Calif Law Rev*, vol. 107, no. 6, pp. 1753–1820, 2019, doi: 10.15779/Z38RV0D15J.

[3] T. T. Nguyen, C. M. Nguyen, D. T. Nguyen, D. T. Nguyen, and S. Nahavandi, “Deep Learning for Deepfakes Creation and Detection: A Survey,” Sep. 2019, [Online]. Available: http://arxiv.org/abs/1909.11573

[4] I. J. Goodfellow *et al.*, “Generative adversarial nets,” in *Advances in Neural Information Processing Systems*, 2014.

[5] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley, “Least Squares Generative Adversarial Networks,” *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2017-Octob, pp. 2813–2821, 2017, doi: 10.1109/ICCV.2017.304.

[6] A. Bansal, S. Ma, D. Ramanan, and Y. Sheikh, “Recycle-GAN: Unsupervised Video Retargeting,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Aug. 2018, pp. 122–138. doi: 10.1007/978-3-030-01228-1_8.

[7] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, “Self-attention generative adversarial networks,” *36th International Conference on Machine Learning, ICML 2019*, vol. 2019-June, pp. 12744–12753, 2019.

[8] A. Brock, J. Donahue, and K. Simonyan, “Large scale GaN training for high fidelity natural image synthesis,” *7th International Conference on Learning Representations, ICLR 2019*, pp. 1–35, 2019.

[9] M. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein generative adversarial networks,” *34th International Conference on Machine Learning, ICML 2017*, vol. 1, pp. 298–321, 2017.

[10] T. Karras, T. Aila, S. Laine, and J. Lehtinen, “Progressive growing of GANs for improved quality, stability, and variation,” *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*, pp. 1–26, 2018.

[11] E. Denton, S. Chintala, A. Szlam, and R. Fergus, “Deep generative image models using a laplacian pyramid of adversarial networks,” *Adv Neural Inf Process Syst*, vol. 2015-Janua, pp. 1486–1494, Jun. 2015, [Online]. Available: http://arxiv.org/abs/1506.05751

[12] A. Makhzani and B. Frey, “PixelGAN autoencoders,” in *Advances in Neural Information Processing Systems*, 2017.

[13] D. Güera, E. J. Delp, D. Guera, and E. J. Delp, “Deepfake Video Detection Using Recurrent Neural Networks,” 2019. doi: 10.1109/AVSS.2018.8639163.

[14] J. Vincent, “New deepfake tech turns a single photo and audio file into a singing video portrait - The Verge,” The Verge. Accessed: Oct. 09, 2020. [Online]. Available: https://www.theverge.com/2019/6/20/18692671/deepfake-technology-singing-talking-video-portrait-from-a-single-image-imperial-college-samsung

[15] J. Porter, “Another convincing deepfake app goes viral prompting immediate privacy backlash - The Verge,” The Verge. Accessed: Oct. 09, 2020. [Online]. Available: https://www.theverge.com/2019/9/2/20844338/zao-deepfake-app-movie-tv-show-face-replace-privacy-policy-concerns

[16] “FaceApp - Free Neural Face Transformation Filters.” Accessed: Oct. 09, 2020. [Online]. Available: https://www.faceapp.com/

[17] C. Wang, “Deepfakes, Revenge Porn, And The Impact On Women,” Forbes. Accessed: Oct. 09, 2020. [Online]. Available: https://www.forbes.com/sites/chenxiwang/2019/11/01/deepfakes-revenge-porn-and-the-impact-on-women/#4878e2181f53

[18] B. Marr, “The Best (And Scariest) Examples Of AI-Enabled Deepfakes,” Forbes. [Online]. Available: https://www.forbes.com/sites/bernardmarr/2019/07/22/the-best-and-scariest-examples-of-ai-enabled-deepfakes/

[19] J. Damiani, “A Voice Deepfake Was Used To Scam A CEO Out Of $243,000,” Forbes. Accessed: Sep. 24, 2020. [Online]. Available: https://www.forbes.com/sites/jessedamiani/2019/09/03/a-voice-deepfake-was-used-to-scam-a-ceo-out-of-243000/

[20] C. Stupp, “Fraudsters Used AI to Mimic CEO’s Voice in Unusual Cybercrime Case - WSJ,” The Wall Street Journal. Accessed: Oct. 09, 2020. [Online]. Available: https://www.wsj.com/articles/fraudsters-use-ai-to-mimic-ceos-voice-in-unusual-cybercrime-case-11567157402

[21] S. Samuel, “AI deepfake app DeepNude transformed photos of women into nudes,” Vox. Accessed: Sep. 24, 2020. [Online]. Available: https://www.vox.com/2019/6/27/18761639/ai-deepfake-deepnude-app-nude-women-porn

[22] S. Parkin, “The rise of the deepfake and the threat to democracy | Technology | The Guardian,” The Guardian. Accessed: Oct. 09, 2020. [Online]. Available: https://www.theguardian.com/technology/ng-interactive/2019/jun/22/the-rise-of-the-deepfake-and-the-threat-to-democracy

[23] Bloomberg, “How ‘Deep Fakes’ Became Easy — And Why That’s So Scary,” Fortune. Accessed: Sep. 24, 2020. [Online]. Available: https://fortune.com/2018/09/11/deep-%20fakes-obama-video/

[24] B. Warner, “Deepfake Zuckerberg Video Goes Viral on Eve of House A.I. Hearing | Fortune,” Fortune. Accessed: Oct. 09, 2020. [Online]. Available: https://fortune.com/2019/06/12/deepfake-mark-zuckerberg/

[25] T. Fish, “Deep fakes: AI-manipulated media will be ‘WEAPONISED’ to trick military,” Express.co.uk. [Online]. Available: https://www.express.co.uk/news/science/1109783/deep-fakes-ai-artificial-intelligence-photos-video-weaponised-china

[26] T. Bradshaw, “Deepfakes: Hollywood’s quest to create the perfect digital human | Financial Times,” Financial Times. Accessed: Oct. 10, 2020. [Online]. Available: https://www.ft.com/content/9df280dc-e9dd-11e9-a240-3b065ef5fc55

[27] J.-J. Hwang, S. Azernikov, A. A. Efros, and S. X. Yu, “Learning Beyond Human Expertise with Generative Models for Dental Restorations,” pp. 1–18, 2018, [Online]. Available: http://arxiv.org/abs/1804.00064

[28] F. Matern, C. Riess, and M. Stamminger, “Exploiting visual artifacts to expose deepfakes and face manipulations,” *Proceedings - 2019 IEEE Winter Conference on Applications of Computer Vision Workshops, WACVW 2019*, pp. 83–92, 2019, doi: 10.1109/WACVW.2019.00020.

[29] C. C. Hsu, Y. X. Zhuang, and C. Y. Lee, “Deep fake image detection based on pairwise learning,” *Applied Sciences (Switzerland)*, vol. 10, no. 1, 2020, doi: 10.3390/app10010370.

[30] O. de Lima, S. Franklin, S. Basu, B. Karwoski, and A. George, “Deepfake Detection using Spatiotemporal Convolutional Networks,” 2020, [Online]. Available: http://arxiv.org/abs/2006.14749

[31] Z. Liu, P. Luo, X. Wang, and X. Tang, “Deep learning face attributes in the wild,” *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2015 Inter, pp. 3730–3738, 2015, doi: 10.1109/ICCV.2015.425.

[32] P. Korshunov and S. Marcel, “Vulnerability assessment and detection of Deepfake videos,” *2019 International Conference on Biometrics, ICB 2019*, 2019, doi: 10.1109/ICB45273.2019.8987375.

[33] S. H. Yoo, S. K. Oh, and W. Pedrycz, “Optimized face recognition algorithm using radial basis function neural networks and its practical applications,” *Neural Networks*, vol. 69, pp. 111–125, 2015, doi: 10.1016/j.neunet.2015.05.001.

[34] J. Villasenor, “Deepfakes, social media, and the 2020 election,” Brookings. Accessed: Oct. 10, 2020. [Online]. Available: https://www.brookings.edu/blog/techtank/2019/06/03/deepfakes-social-media-and-the-2020-election/

[35] S. Ghaffary, “Facebook’s new deepfake ban won’t apply to a fake viral video of Nancy Pelosi - Vox,” Vox. Accessed: Oct. 10, 2020. [Online]. Available: https://www.vox.com/recode/2020/1/7/21055024/facebook-ban-deepfake-video

[36] B. Morris, “Tech Companies Step Up Fight Against ‘Deepfakes’ - WSJ,” The Wall Street Journal. Accessed: Oct. 10, 2020. [Online]. Available: https://www.wsj.com/articles/tech-companies-step-up-fight-against-deepfakes-11574427345

[37] L. Kelion, “Deepfake detection tool unveiled by Microsoft - BBC News,” BBC. Accessed: Oct. 10, 2020. [Online]. Available: https://www.bbc.com/news/technology-53984114

[38] W. Knight, “The US military is funding an effort to catch deepfakes and other AI trickery | MIT Technology Review,” MIT Technology Review. Accessed: Oct. 10, 2020. [Online]. Available: https://www.technologyreview.com/2018/05/23/142770/the-us-military-is-funding-an-effort-to-catch-deepfakes-and-other-ai-trickery/

[39] N. Strout, “A new $5M competition to help the Pentagon detect deepfakes,” C4ISRNET. Accessed: Oct. 10, 2020. [Online]. Available: https://www.c4isrnet.com/intel-geoint/2019/12/27/a-new-5m-competition-to-help-the-pentagon-detect-deepfakes/

[40] “ImageNet.” Accessed: Oct. 30, 2023. [Online]. Available: https://www.image-net.org/

[41] T. Karras, S. Laine, and T. Aila, “A Style-Based Generator Architecture for Generative Adversarial Networks,” *IEEE Trans Pattern Anal Mach Intell*, vol. 43, no. 12, pp. 4217–4228, 2018, doi: 10.1109/TPAMI.2020.2970919.

[42] P. J. Burt and E. H. Adelson, “The Laplacian Pyramid as a Compact Image Code,” *IEEE Transactions on Communications*, 1983, doi: 10.1109/TCOM.1983.1095851.

[43] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, “Spectral normalization for generative adversarial networks,” *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*, 2018.

[44] J. Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017. doi: 10.1109/ICCV.2017.244.

[45] P. Isola, J. Y. Zhu, T. Zhou, and A. A. Efros, “Image-to-image translation with conditional adversarial networks,” in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017. doi: 10.1109/CVPR.2017.632.

[46] P. Garrido *et al.*, “Automatic face reenactment,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2019-June, no. 4, pp. 4217–4224, Aug. 2014, doi: 10.1109/CVPR.2014.537.

[47] D. P. Kingma and P. Dhariwal, “Glow: Generative flow with invertible 1×1 convolutions,” in *Advances in Neural Information Processing Systems*, Jul. 2018, pp. 10215–10224. [Online]. Available: http://arxiv.org/abs/1807.03039

[48] H. Kim *et al.*, “Deep video portraits,” *ACM Trans Graph*, vol. 37, no. 4, 2018, doi: 10.1145/3197517.3201283.

[49] E. Zakharov, A. Shysheya, E. Burkov, and V. Lempitsky, “Few-shot adversarial learning of realistic neural talking head models,” in *Proceedings of the IEEE International Conference on Computer Vision*, May 2019, pp. 9458–9467. doi: 10.1109/ICCV.2019.00955.

[50] Y. LeCun, K. Kavukcuoglu, and C. Farabet, “Convolutional networks and applications in vision,” in *ISCAS 2010 - 2010 IEEE International Symposium on Circuits and Systems: Nano-Bio Circuit Fabrics and Systems*, 2010. doi: 10.1109/ISCAS.2010.5537907.

[51] Y. LeCun, L. L. Bottou, Y. Bengio, and P. P. Haffner, “Gradient-based learning applied to document recognition,” *proc. OF THE IEEE*, 1998, doi: 10.1109/5.726791.

[52] R. Prabhu, “Understanding of Convolutional Neural Network (CNN) — Deep Learning | by Prabhu | Medium,” Medium.com. Accessed: Oct. 20, 2020. [Online]. Available: https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148

[53] “Zero Padding in Convolutional Neural Networks explained - deeplizard,” Deeplizard. Accessed: Oct. 20, 2020. [Online]. Available: https://deeplizard.com/learn/video/qSTv_m-KFk0

[54] V. Nair and G. E. Hinton, “Rectified linear units improve Restricted Boltzmann machines,” in *ICML 2010 - Proceedings, 27th International Conference on Machine Learning*, 2010.

[55] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” *Journal of Machine Learning Research*, 2014.

[56] J. Brownlee, “A Gentle Introduction to Dropout for Regularizing Deep Neural Networks,” Machine Learning Mastery. Accessed: Oct. 20, 2020. [Online]. Available: https://machinelearningmastery.com/dropout-for-regularizing-deep-neural-networks/

[57] D. Sarkar, “A Comprehensive Hands-on Guide to Transfer Learning with Real-World Applications in Deep Learning | by Dipanjan (DJ) Sarkar | Towards Data Science,” TowardsDataScience. Accessed: Oct. 27, 2020. [Online]. Available: https://towardsdatascience.com/a-comprehensive-hands-on-guide-to-transfer-learning-with-real-world-applications-in-deep-learning-212bf3b2f27a

[58] S. Martin, “What Is Transfer Learning? | NVIDIA Blog,” NVIDIA. Accessed: Oct. 27, 2020. [Online]. Available: https://blogs.nvidia.com/blog/2019/02/07/what-is-transfer-learning/

[59] N. Donges, “What Is Transfer Learning? A Simple Guide | Built In,” builtin. Accessed: Oct. 27, 2020. [Online]. Available: https://builtin.com/data-science/transfer-learning

[60] S. Ruder, “Transfer Learning - Machine Learning’s Next Frontier,” ruder.io. Accessed: Oct. 27, 2020. [Online]. Available: https://ruder.io/transfer-learning/

[61] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” *Commun ACM*, vol. 60, no. 6, pp. 84–90, 2017, doi: 10.1145/3065386.

[62] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, pp. 1–14, 2015.

[63] C. Szegedy *et al.*, “Going deeper with convolutions,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 07-12-June, pp. 1–9, 2015, doi: 10.1109/CVPR.2015.7298594.

[64] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the Inception Architecture for Computer Vision,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-Decem, pp. 2818–2826, 2016, doi: 10.1109/CVPR.2016.308.

[65] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *32nd International Conference on Machine Learning, ICML 2015*, 2015.

[66] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-Decem, pp. 770–778, 2016, doi: 10.1109/CVPR.2016.90.

[67] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, Institute of Electrical and Electronics Engineers Inc., Nov. 2017, pp. 2261–2269. doi: 10.1109/CVPR.2017.243.

[68] P. Viola and M. Jones, “Rapid Object Detection using a Boosted Cascade of Simple Features,” 2001.

[69] A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Niessner, "FaceForensics++: Learning to Detect Manipulated Facial Images," *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2019-October, pp. 1–11, Jan. 2019, doi: 10.1109/ICCV.2019.00009.

[70] "CelebA Dataset." Accessed: Nov. 06, 2023. [Online]. Available: https://mmlab.ie.cuhk.edu.hk/projects/CelebA.html

[71] "Keras Applications," Keras.io. Accessed: Oct. 24, 2020. [Online]. Available: https://keras.io/api/applications/

[72] D. P. Kingma and J. L. Ba, "Adam: A Method for Stochastic Optimization," *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, Dec. 2014, doi: 10.48550/arxiv.1412.6980.

[73] S. Narkhade, "Understanding AUC - ROC Curve. In Machine Learning, performance… | by Sarang Narkhede | Towards Data Science," Towards Data Science. Accessed: Oct. 22, 2020. [Online]. Available: https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5

[74] "Simple guide to confusion matrix terminology," Data School. Accessed: Oct. 22, 2020. [Online]. Available: https://www.dataschool.io/simple-guide-to-confusion-matrix-terminology/

[75] I. Chelliah, "Confusion Matrix - Clearly Explained," https://towardsdatascience.com/confusion-matrix-clearly-explained-fee63614dc7.

[76] "What is a confusion matrix?," Edpresso. Accessed: Oct. 22, 2020. [Online]. Available: https://www.educative.io/edpresso/what-is-a-confusion-matrix