KTH ROYAL INSTITUTE
OF TECHNOLOGY

Degree project in computer science and engineering,

First cycle, 15 credits

Accuracy and Robustness of State of the Art Deepfake Detection Models

TOBIAS CARLSSON & OSKAR STRÖMBERG

Stockholm, Sweden 2023

Accuracy and Robustness of State of the Art Deepfake Detection Models

Tobias Carlsson & Oskar Strömberg

Degree Project in Computer Science and Engineering
Date: June 9, 2023
Supervisor: Erik Fransén
Examiner: Pawel Herman
Swedish Title: Precision och Robusthet hos Bästa Tillgängliga Deepfake-detektionsmodeller
School of Electrical Engineering and Computer Science
KTH Royal Institute of Technology

Abstract

With the evolution of artificial intelligence a lot of people have started getting worried about the potential dangers of deepfake images and videos, such as spreading fake videos of influential people. Several solutions to this problem have been proposed with some of the most efficient being convolutional neural networks for face detection in order to differentiate real images from deepfake images generated with a generative adversarial network. One of the currently most prevalent models is the VGGFace which is further analyzed in the report. This project explores how different hyper-parameters affect the effectiveness of existing convolutional neuralnetworks aswell as the robustness in the models. The hyper-parameter that had the biggest effect on accuracy was the amount of conovultion layers in each step of the network. The results showed that while deepfake detection models showed high accuracy on the test set, they are lackluster when it comes to the robustness. The models showed a clear sensitivity for the resolution of test images. This is an issue that can be solved through resizing, this report shows the more concerning issue where the model had a 47 percentage point reduction in accuracy when tested on a different dataset that had fake images generated with a different generative adversarial network. The main takeaways from the project is that current deepfake detection models have to work on generalization in order to effectively classify images.

Sammanfattning

Med utvecklingen av artificiell intelligens har många människor börjat bli oroliga över de potentiella farorna med deepfake-bilder och -videor, exempelvis spridning av falska videor på inflytelserika människor. Flera lösningar på detta problem har föreslagits, varav några av de mest effektiva är konvolutionsneurala nätverk för ansiktsdetektion för att kunna skilja på verkliga bilder och deepfake-bilder som genererats med hjälp av ett generativa motståndarnätverk. En av de främsta nuvarande modellerna kallas VGGFace och analyseras vidare i rapporten. Projektet utforskar hur olika hyperparametrar påverkar effektiviteten hos befintliga konvolutionsneurala nätverk samt undersöker hur robusta modellerna är. Hyperparametern som hade störst effekt på noggrannheten var antalet konvolutionslager i varje steg av nätverket. Resultaten visade att trots deepfake-detektionsmodellernas höga noggrannhet på testdatan så var de bristfälliga gällande robustheten. Modellerna visade tydlig känslighet för upplösningen på testbilder. Detta är ett problem som kan lösas genom att ändra upplösningen på bilderna, men rapporten visar däremot ett mer oroväckande problem där modellen hade en minskning på 47 procentenheter i noggrannhet när den testades på ett annat dataset som hade deepfake-bilder genererade med ett annat generativt motståndarnätverk. De huvudsakliga slutsatserna från projektet är att nuvarande deepfake-detektionsmodeller måste arbeta med generalisering för att kunna klassificera bilder på ett effektivt sätt.

Contents

Chapter 1

Introduction

1.1 Background

In recent years it has become increasingly popular to create deepfake images and videos where it is possible to automatically replace the face of one person with the face of another. Deepfake technology has the potential to greatly improve the entertainment industry by streamlining movie productions as well as allowing foreign movies to be enjoyed in ones native language [19] by matching actors mouth movements to dubbed audio.

It has however also been accompanied by many problems. Celebrity faces have been deepfaked onto pornographic videos and images as well as controversial messages being portrayed by deepfaked political figures. There already exists several effective facial recognition services but state of the art systems based on visual geometry group (VGG) as well as Facenet neural networks are very susceptible to misclassifying deepfake images and videos [13]. Today most deepfake detection tools use deep learning methods with a convolutional neural network (CNN) as a classifier [17].

Table 1.1.1: The most common types of deepfake videos [14]

NAME	USAGE
Head puppetry	Synthesize entire head and upper shoulder of a person using a source image. The produced result will be a synthesized image where the target behaves the same way as the source image.
Face swapping	Generating a video of target with a replaced synthesized face from the source image with similar facial expressions.
Lip syncing	Manipulation of lips in the target image to make the target appear to be speaking.

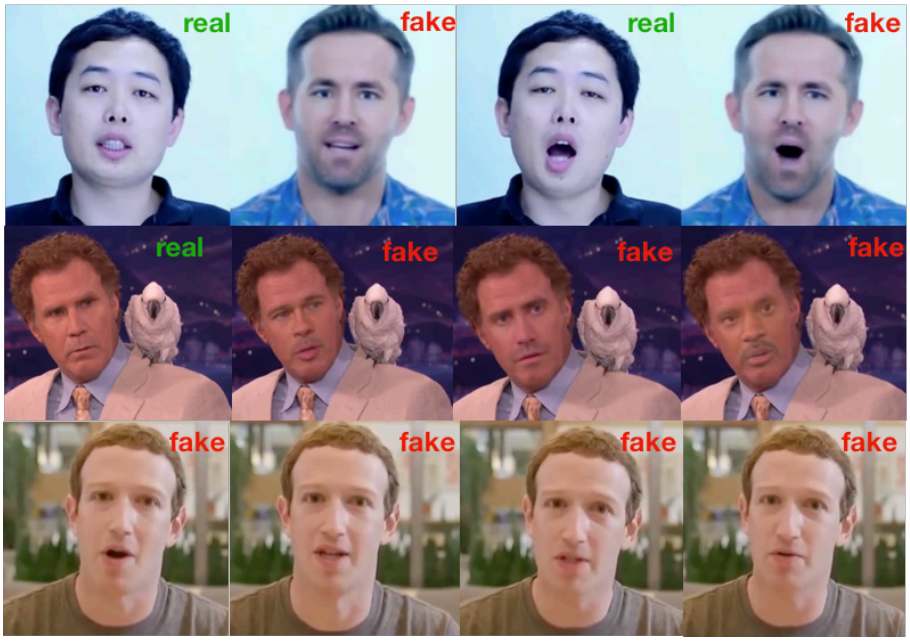

Figure 1.1.1: Examples of head puppetry (row 1), face swapping (row 2) and lip syncing (row 3). Image taken from Deepfake detection: Current challenges and next steps[14]

The biggest problem is the ease in which people can access the ability to create deepfakes. No knowledge of neural networks is necessary. Like Pantserev discusses, there exists desktop applications like FakeApp that are distributed for free on the internet and allow people to change faces in videos for later distribution [15]. It is therefore vital to be able to detect deepfake images and videos which is why technologies that can automatically detect deep fake images in order to alert the public are important to explore.

1.2 Problem

It can be very hard for machine learning models to detect certain deepfake images and with images that contain large amounts of features as a basis for the prediction it can be difficult to determine which features actually contribute to prediction accuracy. John et al. [12] categorizes features into one of three categories: Irrelevant features that never contribute to prediction accuracy, weak relevance features that sometimes contribute to prediction accuracy and strong relevance features that implies that the feature cannot be removed without loss of accuracy. There exists several different approaches to reducing the number of irrelevant features in the data with different efficiencies depending on the source data. Some of the more common generic methods include: feature selection as a heuristic search, filter approaches to feature selection and wrapper approaches to feature selection [3].

Modern technologies for deepfake detection have become quite accurate with many performing at a 90% or higher accuracy rate. But there still are some cases where state of the art CNNs fail to correctly classify the images. As deepfake technology continues to evolve, it is crucial to understand the strengths and weaknesses of detection models

in order to effectively combat the spread of disinformation. We therefore pose the question: What makes modern convolutional neural networks so effective and how robust are they?

1.3 Purpose

This degree project aims to examine the effectiveness and robustness of deepfake image detection models using Convolutional Neural Networks (CNNs). With the proliferation of deepfake technology, there is a growing need for reliable and accurate methods of detecting manipulated images and videos. The purpose of this project is to evaluate the performance of CNN-based models for deepfake detection, comparing their accuracy when modifying the model and against various types of deepfake images.

Specifically, this project will investigate the following research questions:

- How do CNN-based models perform in detecting deepfake images, and what factors have the biggest impact on their accuracy?
- What are the limitations of current CNN-based deepfake detection models, and how robust are they to different types of images?

To address these research questions, this project will review and analyze existing literature on deepfake detection and CNN-based models, and conduct experiments using a dataset of deepfake images. The goal of the project is to contribute to the growing body of knowledge on deepfake detection and provide insights into the strengths and limitations of current detection models.

1.4 Benefits, Ethics and Sustainability

Anyone who wants to utilize CNNs for deepfake image or video detection will benefit from this degree project. Deepfakes could lead to a decrease in trust in institutions through faked videos about police brutality, judge corruption through private discussions or a border guard using racist language[10]. These institutions can benefit from the project by learning how to improve their detection model or by what types of source data to be extra mindful of when implementing deepfake detection models. Andrew Ray also points out the potential problems with political deepfakes where deepfaked videos could be used to influence elections [16]. Therefore all types of elections and its participants could benefit.

Considering this project explores how to improve deepfake detection models it does not tackle a lot of ethical dilemmas. However, there are some ethical issues that could arise with deepfake detection as a whole. A meta-analysis showed that there exists political bias in news media [6]. While bias in favor of one ideology usually gets balanced out by bias in favor of another it still poses the problem of bias in individual sources. Bias seldom affect voters who are already clearly at one side of the political spectrum but centrists tend to vote against whom they hear negative news [2]. Different media could use deepfake detection models to debunk videos that harm their preferred political

view or candidate while avoiding debunking fraudulent media of another candidate or even spreading the deepfaked video.

To properly utilize all potential benefits from deepfake detection models one would have to implement automatic tester for all media posts using one of the available models. This comes with some sustainability issues however as every second roughly 3,400,000 emails are sent[21], 740,000 WhatsApp messages[7], 55,000 facebook posts[20] and 6,000 tweets[9]. These amounts of media posts are clearly not sustainable when it comes to computational limitations, resource allocation as well as energy consumption for running the models at such a capacity.

Chapter 2

Theoretical Background

2.1 Optimizing parameters in a neural network

One of the most important steps in creating an effective neural network is to try and optimize the parameters for the model. As Claesan and De Moor says: " Hyper-parameters are used to configure various aspects of the learning algorithm and can have wildly varying effects on the resulting model and its performance"[5]. One approach to try and find optimal parameters is the trial-and-error grid-search method which has the advantage of being easily parallelized [1]. However, due to dimensionality restraints it is not a sustainable method for four or more hyper-parameters as the amount of functions needed to asses will become too large [18]. This is not workable in most CNN models that utilize more hyper-parameters which is why it is not relevant to this degree project. Hyper-parameter search is an important aspect of figuring out what makes a neural network perform at a high level.

2.2 Adversarial attacks to trick deepfake detection models

While modern deepfake detection models can perform at a high accuracy on datasets with real and fake images and videos there are ways to manipulate data for an adversary in order to trick the detection models. In some cases, a white-box attack can be done where the adversary has full awareness of the architecture of the model and all of its parameters. A proposed method for tricking a deepfake detection model in a white-box attack managed to get a 99.05% attack success rate[11]. However, it is noted that many standard modifications to images like compression are effective in removing adversarial disturbance in an image [8]. Therefore, 99.05% success rate is most likely a bit misleading. In the much more realistic case however an adversary will have to do a black-box attack where they have zero knowledge of the detection model. In this case it's been shown that a Distortion-minimizing Attack can be effective [4]. A Distortion-minimizing Attack is where a loss function is created that is minimized when an image is incorrectly classified. This loss function can then be used to find an optimal perturbation for an image.

2.3 Related Work

- ”Adversarial deepfakes[11]: Evaluating vulnerability of deepfake detectors to adversarial examples”: Research regarding how deepfake detection models can be tricked
- ”RL based hyper-parameters optimization algorithm (ROA) for convolutional neural network”[18]: Research on how to optimize hyper-parameters for a CNN.
- Random search for hyper-parameter optimization”[1]: Description of the problems of finding optimal hyper-parameters

Prior research on adversarial deepfakes found that small modifications on the source data can be useful when trying to trick a detection model. This will be used in the degree project by modifying some of the source images in order to test how robust the CNN is. Previous research has showed the importance of optimal hyper-parameters for generating an efficient model. This influenced the degree projects' decision to test which hyper-parameters have the largest effect on result. Finding which hyper-parameters are most important could allow someone to weight hyper-parameter optimization around the ones with most importance. The amount of previous research regarding robustness and hyper-parameter optimization showed that these are important factors in CNN's which is why our research questions are revolved around these topics.

2.4 Convolutional Neural Networks

A Convolutional neural network (CNN) is an artificial neural network architecture that is best suited for image processing tasks, such as feature extraction in images and image classification. CNNs uses a mathematical operation called convolution, which in mathematical terms is a function that is a product of two functions which expresses how one function is applied on the other. The kernel size in a CNN determines the area of input data considered at each step of the convolution operation, allowing the network to capture different levels of detail and features. A kernel size of 3 will in each step consider a 3x3 grid of pixels from the source image.

The CNN consists of an input layer, an output layer and hidden layers. Each hidden layer is comprised of several layers that perform operations on the input. These layers are: convolutional layers, pooling layers and fully-connected layer. The convolutional layer performs the convolution operation and produces a number of filter maps, which captures the features produced by the convolutional filters. Pooling layers are used reduce the dimension of the feature maps, which reduces the complexity of the convolutional layers and increasing the computational performance of the network. After a series of convolutional and pooling layers the feature maps are flattened into a one-dimensional vector and passes through the fully-connected layer to make a prediction.

The number of convolutional layers and pooling layers as well as the number of filters in each convolutional layer are some of the parameters which are used to optimise the models performance. Other important parameters can be: dropout rate for dropout

regularization, batch normalization for normalizing the inputs to each layer, the stride of the convolution and padding of the input.

2.5 Face recognition using CNNs

The first successful use of CNNs for facial recognition was the AlexNet in 2012, developed for the ImageNet challenge. The AlexNet showed that a deep convolutional layer is suitable for face detection in images. Later developments would show that a deep convolutional network is even better at face detecction when using a small kernel size for the convolution, the smallest size possible being a 3x3 grid in order to be able to identify left, right, top, bottom and center pieces of a grid. This would become the VGGFace model which has achieved accuracy of up to 99 percent on some data sets. This model has been shown to perform equally well on the task of deepfake-detection. However, there is a limited amount of data sets available of labeled deepfake images and the VGGFace model has been proven to vary a lot depending on the data set used for validation. It is therefore uncertain if deepfake-detection models are generalized well enough to accurately predict deepfake images from any given data set.

2.6 VGGFace model

The VGGFace model is a deep convolutional neural network specifically designed for facial recognition tasks, to investigate the performance of automated face recognition systems. The VGGFace model is based on the VGG-16 architecture, which consists of 16 weight layers, including 13 convolutional layers and 3 fully connected layers, followed by a softmax layer for classification. The model is trained on the VGGFace dataset, a large-scale face dataset containing over 2.6 million images of more than 2,600 distinct individuals. These images vary in terms of pose, expression, and lighting conditions, ensuring a robust and diverse training set.

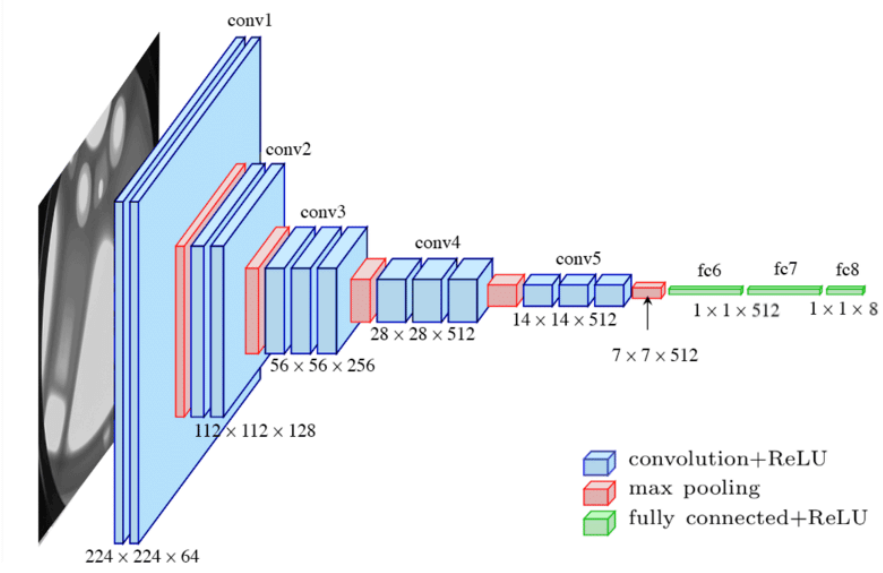

Figure 2.6.1: VGGFace Architecture

Chapter 3

Method

The project used two datasets in order to test the efficiency of the model. The main dataset used was Kaggle's 140k real and fake faces which consists of 70k real and 70k fake faces that are divided into train, validate and test sets. The dataset is generated using StyleGAN. The other dataset was used to test robustness in the model and the fake images was created manually by humans. We chose this dataset because the perturbations will most likely be very different than those existing in the GAN created images. This second dataset however only consists of 1000 real and 1000 fake faces. Below are description of the exact tests that were conducted.

3.1 The baseline model

The baseline model follows the same structure as VGGFace, except with only one convolution layer between each max pooling. This is to reduce the computational complexity of the tests and to limit the hyperparameters to test. Furthermore by limiting the baseline model, the accuracy will be relatively lower than the state-of-the art VGGFace model, making it easier to see what hyperparameters increasse the baseline models accuracy closer to that of VGGFace.

3.2 Testing hyperparameters

One part of the work that was done was to test which hyper-parameters has the biggest effect on the results in order to answer the question ”what makes modern CNNs so effective?”. This was done by modifying a baseline CNN model and seeing how it performed in comparison to the unmodified version in the following ways:

- The kernel size in the model was changed. The baseline was a 3x3 grid for the convolution. Both 2x2 and 5x5 grids were tested.
- The amount of convolutional layers between each pooling layer was modified. The modifications changed the baseline model to be more similar to the existing vggface model. This was done by adding a second convolution layer after each pooling step.

- The source images were cut in half to effectively only train and test on half the input data to see if it could reduce overfitting and work better in general application.

The three mentioned hyper-parameters were chosen since there are not many other easily changeable parameters in consideration of the projects limitations. It is also these hyper-parameters that have the biggest effect on accuracy according to previous research.

3.3 Testing robustness

In order to answer the second part of the research question ”How robust are current CNN models” we decided to modify the test data. This was done to see how the model would perform in a real world scenario where data is unpredictable and there may be adversaries who try to exploit and deceive the model. The following tests were conducted:

- The resolution of test data was reduced to half the resolution of the source data.
- The source data was reduced to 128px but resized to the trained resolution of 224px before prediction was done
- The model was tested on a completely seperate dataset in order to better determine the models ability to generalize. It was tested on the kaggle Real and Fake Face Detection dataset.

3.4 Limitations

There were limitations to the possible experiments that were capable of being performed, because of computation time. This limited for example how many layers that could be added to the baseline to increase performance and how much data could be used to test the VGGFace model.

Chapter 4

Result

4.1 Performance of hyperparameters

Model	Precision (Fake)	Precision (Real)	Precision weighted average
Baseline model	0.84	0.98	0.91
Doubled conv layers	0.96	0.93	0.94
Deepest 3 layers	0.80	0.99	0.90
Shallowest 3 layers	0.8	0.99	0.99

Table 4.1.1: Hyperparameter fine-tuning performance

The hyperparameters that were tested were the kernel size and number of convolution layers. From testing it turns out that the optimal kernel size was 3, because both increasing the kernel size to 4 and decreasing it to 2 led to a decrease in precision. Testing of the number convolution layers showed that doubling the number of convolution layers between each pooling greatly increased the precision, but only doubling the deepest or the topmost layers did not have the same great effect. Sadly, the computation of further increasing the number convolutional layers was to great and could not be performed.

4.2 Robustness in the model

As mentioned in 3.3 the robustness of the model was tested by first testing the model on images in lower resolution, specifically 128px instead of the 224px that the model was trained on. The model had a 75% accuracy on fake images and a 50% accuracy on real images which added up to a weighted average of 63%. In comparison the baseline model performed at 84% accuracy on fake images, 98% on real which added up to a weighted average of 91%.

Test Data	Precision (Fake)	Precision (Real)	Precision weighted average
128px Images	0.75	0.50	0.63
224px Images	0.88	0.85	0.86
Second Dataset	0.33	0.53	0.43

Table 4.2.1: Robustness performance

When the model was tested on 128px images but resized the image to the original 224px it had an accuracy of 88% on fake images and 85% on real images which added up to a weighted average of 86%. Finally the model was tested on the separate dataset mentioned in 3.3. On this dataset the model had an accuracy of 33% on fake images and a 53% accuracy on real images which gave the weighted average of 43%. The model performed just above guessing for real images and actively worse than guessing for the fake.

Compared to the baseline model and testdata we can see that the biggest decrease in performance came from testing on a different dataset with a decrease of 48 percentage points. Change in resolution of the images did not have a big effect on the accuracies of the model as long as the model itself resized the images before execution.

Chapter 5

Conclusions

5.1 Discussion

5.1.1 Robustness

While the results of the model seemed promising at first the project clearly showed some problems when it comes to the robustness and generalization of the model. We saw a pretty big loss of accuracy when the model classified the same dataset on 128px resolution. This is somewhat expected as the model most likely learned how to recognize patterns that appear on 224px and any other resolution would change how these patterns appear to the model. We did some quick tests on different resolutions aswell such as the source resolution of 256px and saw similar accuracies there. The second test of robustness somewhat proved that the model is able to adapt to new resolutions reasonably well as long as the image is resized to the trained 224px resolution before classifying. The small 3 percentage point reduction in accuracy that we saw in comparison to the baseline model could be due to some smaller features in the images being lost when the resolution is cut in half. The most impactful result about the robustness was how poorly the model performed on another dataset under the same constraints as it had in the training dataset. This shows that the model is overfitted and does not do well on a variety of images. One likely explaination as to why the performance dropped drastically is the fact that the fake images were generated using different methods for the different datasets and the model learned to recognize patterns in one type of deepfakes but totally fails on another.

5.1.2 Hyperparameters

The results showed that increasing the kernel size did not improve the accuracy, which is reasonable because the smaller the kernel size the more local features can be captured. Since deepfake images share the most common features with a real image it is therefore most likely that it is small local features that can distinguish deepfakes from real images, which this result shows. Although the accuracy did neither improve when decreasing the kernel size to 2, this is because a kernel size of 3 is the smallest possible grid to be able to capture center, top, bottom, left and right positions of a grid, which shows that the positional information of the pixels is also important for classifying

images. In conclusion the kernel size of 3 is the smallest and most optimal size for discovering small local features which are vital in detecting deep-fake images.

Increasing the convolution layers lead to an overall increase in performance, but only when all layers were doubled. It would have been interesting to test if this were to be true when further increasing the number of convolution layers and testing if there were a plateau when having 3 convolution layer between each max pooling and if this is the reason that the VGGFace model does not have 3 convolution layers between the first max pooling as seen in figure 3.4.1 and instead have 2 convolution layers in conv1 and conv2. Sadly this would require more computational power than was available, but the results of double convolution layers shows that the number of convolution layers is a key parameter in the performance of VGGFace.

5.2 Future Work

Considering how poorly the model performed on a dataset the model was not trained on, one area of important future work would be how to better generalize detection models to work better on different types of images. Perhaps a dataset with fake images created from different generative models could be created and trained on or maybe it can be solved through better pre-processing of data before classification is done. With the lack of labeled deepfake datasets to train on future work could also consist of creating more labeled data constructed from a variety of deepfake models.

5.3 Final Words

While current deepfake detection models look very promising there are still some problems when it comes to generalization and robustness. Overall the models are effective but seem very niched and the problem of efficient implementation must also be solved in order to properly utilize the tools.

Bibliography

[1] Bergstra, James and Bengio, Yoshua. “Random search for hyper-parameter optimization.” In: *Journal of machine learning research* 13.2 (2012).

[2] Bernhardt, Dan, Krasa, Stefan, and Polborn, Mattias. “Political polarization and the electoral effects of media bias”. In: *Journal of Public Economics* 92.5-6 (2008), pp. 1092–1104.

[3] Blum, Avrim L and Langley, Pat. “Selection of relevant features and examples in machine learning”. In: *Artificial intelligence* 97.1-2 (1997), pp. 245–271.

[4] Carlini, Nicholas and Farid, Hany. “Evading deepfake-image detectors with white-and black-box attacks”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*. 2020, pp. 658–659.

[5] Claesen, Marc and De Moor, Bart. “Hyperparameter search in machine learning”. In: *arXiv preprint arXiv:1502.02127* (2015).

[6] D’Alessio, Dave and Allen, Mike. “Media bias in presidential elections: A meta-analysis”. In: *Journal of communication* 50.4 (2000), pp. 133–156.

[7] Diamandis, Peter. *64 Billion Messages in 24 Hours: Key Takeaways From WhatsApp’s Massively Disruptive Statistics*. 2014. URL: `https://www.huffpost.com/entry/64-billion-messages-in-24_b_5160021` (visited on 03/29/2023).

[8] Dziugaite, Gintare Karolina, Ghahramani, Zoubin, and Roy, Daniel M. “A study of the effect of jpg compression on adversarial images”. In: *arXiv preprint arXiv:1608.00853* (2016).

[9] Engineering. *New Tweets per second record, and how!* 2013. URL: `https://blog.twitter.com/engineering/en_us/a/2013/new-tweets-per-second-record-and-how` (visited on 03/29/2023).

[10] Helmus, Todd C. *Artificial Intelligence, Deepfakes, and Disinformation: A Primer*. Tech. rep. RAND CORP SANTA MONICA CA, 2022.

[11] Hussain, Shehzeen, Neekhara, Paarth, Jere, Malhar, Koushanfar, Farinaz, and McAuley, Julian. “Adversarial deepfakes: Evaluating vulnerability of deepfake detectors to adversarial examples”. In: (2021), pp. 3348–3357.

[12] John, George H, Kohavi, Ron, and Pfleger, Karl. “Irrelevant features and the subset selection problem”. In: *Machine learning proceedings 1994*. Elsevier, 1994, pp. 121–129.

[13] Korshunov, Pavel and Marcel, Sebastien. “Deepfakes: a new threat to face recognition”. In: *Assessment and detection* (2018).

[14] Lyu, Siwei. “Deepfake detection: Current challenges and next steps”. In: *2020 IEEE international conference on multimedia & expo workshops (ICMEW)*. IEEE. 2020, pp. 1–6.

[15] Pantserev, Konstantin A. “The malicious use of AI-based deepfake technology as the new threat to psychological security and political stability”. In: *Cyber defence in the age of AI, smart societies and augmented humanity* (2020), pp. 37–55.

[16] Ray, Andrew. “Disinformation, deepfakes and democracies: The need for legislative reform”. In: *TheUNIVERSITY OF NEW SOUTH WALES LAW JOURNAL* 44.3 (2021), pp. 983–1013.

[17] Suganthi, ST, Ayoobkhan, Mohamed Uvaze Ahamed, Bacanin, Nebojsa, Venkatachalam, K, Štěpán, Hubálovskỳ, Pavel, Trojovskỳ, et al. “Deep learning model for deep fake face recognition and detection”. In: *PeerJ Computer Science* 8 (2022), e881.

[18] Talaat, Fatma M and Gamel, Samah A. “RL based hyper-parameters optimization algorithm (ROA) for convolutional neural network”. In: *Journal of Ambient Intelligence and Humanized Computing* (2022), pp. 1–11.

[19] Usukhbayar, Binderiya, and Homer, Sean. “Deepfake Videos: The Future of Entertainment”. In: (2020).

[20] Zephoria. *The Top 10 Valuable Facebook Statistics – Q2 2021*. 2021. URL: `https://zephoria.com/top-15-valuable-facebook-statistics/` (visited on 03/29/2023).

[21] Zettasphere. *Mind boggling stats for 1 second of internet activity*. 2019. URL: `https://www.zettasphere.com/mind-boggling-stats-for-1-second-of-internet-activity/` (visited on 03/29/2023).

Appendix - Contents

TRITA-EECS-EX-2023:285