Detecting Both Machine and Human Created Fake Face Images In the Wild

Shahroz Tariq
The State University of New York, Korea (SUNY-Korea)
Incheon, South Korea
shahroz.tariq@stonybrook.edu

Sangyup Lee
The State University of New York, Korea (SUNY-Korea)
Incheon, South Korea
sangyup.lee@stonybrook.edu

Hoyoung Kim
The State University of New York, Korea (SUNY-Korea)
Incheon, South Korea
hoyoung.kim.1@stonybrook.edu

Youjin Shin
The State University of New York, Korea (SUNY-Korea)
Incheon, South Korea
youjin.shin.1@stonybrook.edu

Simon S. Woo
The State University of New York, Korea (SUNY-Korea)
Incheon, South Korea
simon.woo@sunykorea.ac.kr

ABSTRACT

Due to the significant advancements in image processing and machine learning algorithms, it is much easier to create, edit, and produce high quality images. However, attackers can maliciously use these tools to create legitimate looking but fake images to harm others, bypass image detection algorithms, or fool image recognition classifiers. In this work, we propose neural network based classifiers to detect fake human faces created by both 1) machines and 2) humans. We use ensemble methods to detect GANs-created fake images and employ pre-processing techniques to improve fake face image detection created by humans. Our approaches focus on image contents for classification and do not use meta-data of images. Our preliminary results show that we can effectively detect both GANs-created images, and human-created fake images with 94% and 74.9% AUROC score.

CCS CONCEPTS

• **Security and privacy** → *Social aspects of security and privacy*; • **Computing methodologies** → *Neural networks*;

KEYWORDS

Generative Adversarial Network; Fake Image Detection; Image Forensics

ACM Reference Format:
Shahroz Tariq, Sangyup Lee, Hoyoung Kim, Youjin Shin, and Simon S. Woo. 2018. Detecting Both Machine and Human Created Fake Face Images In the Wild. In *2nd International Workshop on Multimedia Privacy and Security (MPS '18), October 15, 2018, Toronto, ON, Canada.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3267357.3267367

MPS '18, October 15, 2018, Toronto, ON, Canada

ACM ISBN 978-1-4503-5988-7/18/10…$15.00
https://doi.org/10.1145/3267357.3267367

1 INTRODUCTION

The remarkable development of AI and machine learning technologies have assisted in solving the most challenging tasks in the areas of computer vision, natural language processing, image processing, etc. Recently, machine learning algorithms are extensively integrated for photo-editing applications to help create, edit, and synthesize images, and improve image quality. Hence, people without an expert knowledge of photography editing can easily create sophisticated and high quality images. Also, many photo editing tools and apps provide various interesting functionality to attract users such as face swap. For example, face swap apps are widely used to automatically detect faces in photos and swap the face of one person with another person or animal. While face swap is fun and wide-spread in social network or Internet, it can be offensive and someone might not feel comfortable if their faces are swapped or spoofed by someone else for malicious causes. Therefore, abusing these multimedia technologies raise significant social issues and concerns. In particular, one of them is to create fake pornography [6], where anyone can put a victim's face into a naked body to humiliate and intimidate the victim.

In addition, humans can manually create more sophisticated fake or face swap images using high quality photo-editing tools such as Adobe Photoshop. These tools have become much more advanced to create realistic and elaborate fake images, which are difficult to determine the forgery by normal people. The step-by-step instructions and tutorial to create these types of face swaps are easily available in YouTube. Therefore, these technologies can be used for defamation, impersonation, and distortion of facts. Furthermore, these fake information can be quickly and widely disseminated in Internet through social media [6]. Hence, maliciously using these machine learning-enabled multimedia technologies for image forgery can lead into significant problems in not only fake pornography creation, but also hate crimes and frauds. Moreover, recent pioneering advancement

in Generative Adversarial Networks (GANs) has made even easier to create natural, and legitimate looking images [14] from scratch. Although this is a great feat, it comes with major security problems such as using synthetic photos for identification and authentication applications. These images are extremely challenging for normal people to tell whether it is a real human faces or machine generated faces. And GANs can be possibly misused or abused to hurt people similar to DeepFake [28].

In order to detect and prevent these malicious effect, diverse detection methodologies can be applied. However, most of prior research is based on analyzing meta-data or characteristics of image compression information, which can easily be cloaked. Also, splicing or copy-move detection techniques are not effective when attackers forge elaborate images using GANs [14]. In addition, there is no existing research to detect GANs created images. Therefore, in this paper, we tackle the problem of detecting both GANs generated human faces and human-created fake images with neural networks using ensemble methods. Specifically, our preliminary research focuses on detecting the following fake face images: 1) GAN created synthetic human faces, and 2) images, where face regions in an image are modified by humans with sophisticated editing tools such as Adobe Photoshop. Our contributions are summarized below:

(1) We develop an ensemble-based neural network classifier to detect GANs created face images with 94-99% of AUROC score.
(2) We develop a neural network classifier to detect sophisticated human-created fake face images with 74.9% AUROC score, leveraging face cropping and noise filtering algorithms.
(3) We propose a fully automated effective end-to-end fake face detection pipeline without any human interventions or any meta-data information.

Our work is the first to detect both GANs and human-created fake face images and our preliminary results show the high accuracy in detecting those fake images. This paper is organized as follows. We discuss related work of fake image detection in Section 2. We explain our approaches in Section 3, and describe our evaluation results in Section 4. Finally, Section 5 offers our conclusions.

2 RELATED WORK

Several researchers [7, 9, 10, 15, 17, 29, 32] have proposed various digital image forensic algorithms and tools to analyze properties of image formats and meta-data, and further deep learning to detect modifications in images. However, detecting forgery and fake components in images remain a challenging problem because attackers are also employing the latest image processing and machine learning techniques to bypass well-known forgery detection techniques. One of the most popular methodologies is analyzing images in frequency domain. When images are compressed using JPEG, the compression history is remained. By analyzing this in the frequency domain, manipulated area on each image can be detected shown by many researchers [19, 20, 29]. However, these methods do not work well against images with sophisticated and smoothed edges, which we consider in this paper.

There is another method, JPEG Ghost [9], which searches for different JPEG quality on the same image and extracts the difference between them. When an image is manipulated, the forged part usually is copied from other images with different JPEG quality. The Normalized pixel distance between an original image and a re-created image with different JPEG quality can clearly show regions with differing quality. This method, however, might not be useful if the forged region comes from the same image quality level.Error level analysis (ELA) [17] is another technique to exploit error rate in JPEG images. Manipulated regions usually have inconsistent error levels from unmodified regions, which can be observed by re-saving the images with different error rate. However, with GANs-created images, ELA still cannot identify the different error level. Hence, this approach is not helpful.

Image classification with neural networks is useful for digital forensics on big data workflows such as health care, security, and anomaly detection [30]. Also, recently these neural network technologies have been used for image forensics. Convolutional Neural Network based image classification and recognition models can be used to discriminate fake images from real ones. VGG16 and VGG19 [26] have improved the large-scale image recognition by increasing the layer depth (23/26 layers) of their CNN-based model. Similarly, these deeper neural network models are more difficult to train. ResNet [11] presents a residual learning framework that makes training of networks much easier. They reformulate the layers as learning residual functions with reference to the layer inputs, instead of learning unreferenced functions. This can benefit on model optimization and improve accuracy from much increased depth. DenseNet [12] is a feed-forward designed network which connects each layer to every other layer. For each layer, the feature-maps of all former layers are used as input. DenseNet requires significantly fewer parameters and computation to achieve state-of-the-art performance. While these image classification models often necessitate significant architecture engineering, NASNet [34] focuses on searching for an architectural building block on a small dataset and transfer the block to a larger dataset. They search for the best convolutional layer on a specific dataset and use that layer to build up a convolutional architecture. Their work contributes to the design of a new search space which enables transferability. Also, Xception [2] has outperformed prior image classifier models on ImageNet dataset by introducing a deep learning classifier with depth-wise separable convolutions.

In addition, there are some investigations for detecting face forgery using deep learning such as FaceForensics [24], which generates refined fake face dataset from utilizing Face2Face [27]. Providing more features to deep learning model [7, 32] have been proposed for image forgery detection. Specifically, noise features extracted from a steganalysis rich model are used as an input to CNN-based model. However, these

approach may not work well with fake images containing smoothed-edge spliced images, since it is difficult to extract the obvious noise features.

More advanced tools such as FotoForensics [18] and MMC image forensic tool [13] utilize a variety of information such as metadata, ELA, and JPEG quality. However, sophisticated attackers can easily detour these tools by hiding or modifying meta-data information. Also, these approaches do not work for images generated by GANs.

Recently, Karras et al. [14] demonstrates that GANs can generate new human faces by growing both the generator and discriminator progressively. The output face images generated by Karras et al. are highly realistic and not trivial for humans to distinguish whether those images are real persons or not. Even though rich applications that this technology can be used in a positive way, this work can create the fake human faces can possibly maliciously used. For example, programmatically creating these realistic fake faces can fool face recognition algorithms, and attackers can create many of these fake images to mislead certain information, generate fake IDs with non-existing persons' faces, etc. and can possibly cause social issues. Currently, no research has performed to detect GANs created synthetic images, which our work focuses on. Deepfakes [28] is another popular tool to create fake multimedia such fake celebrity pornographic videos or revenge porn, where Deepfake pornography surfaced on the Internet in 2017 on Reddit. It is shown that a number of famous actresses' fake porn are created using Deepfake. In addition, Deepfakes can be used to create fake news and malicious hoaxes in politics [4, 23] such as deepfaking Barack Obama [22]. Therefore, it is critical to develop technologies to detect these fake images and stop spreading as quickly as possible.

3 OUR APPROACH

In this section, we explain our approaches to detect 1) GAN-generated fake synthetic face images and 2) sophisticated fake face images created by humans using high quality photo-editing tools.

3.1 Detecting machine (GANs) generated synthetic face images

Currently, traditional digital media forensic tools fail to detect GANs-generated images, because they are progressively generated as a single image. Hence, meta-data or splicing based detection methods are impossible to detect. Hence, our work focuses on distinguishing GANs-generated fake face images from real people' faces, and develop a fundamental enabling classifier technology can detect them with high accuracy. We consider detecting various image sizes from tiny 64×64 to large 1024×1024 pixels images, where we assume attackers can manipulate any size of images. Some examples of real and fake images in different sizes are shown Fig. 1 and 2, where 64×64 image size are extremely challenging to recognize even face region with human eyes.

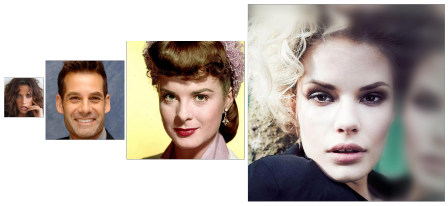

Figure 1: CelebA dataset Images in different sizes from left to right: 64×64, 128×128, 256×256, and 1024×1024 pixels

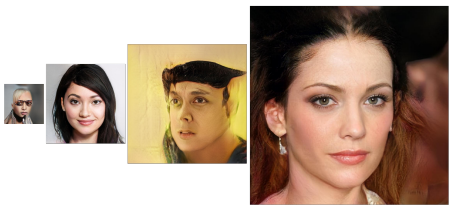

Figure 2: PGGAN dataset Images in different sizes from left to right: 64×64, 128×128, 256×256, and 1024×1024 pixels

In particular, we use the following popular datasets for design, validation, and evaluation:

- **CelebA Dataset:** CelebFaces Attributes Dataset (CelebA) is a large-scale face attributes dataset with more than 200K celebrity images [33].
- **Progressive Growing GANs Dataset (PGGAN)** Consisting of 100K GAN generated fake celebrity images at 1024x1024 resolution using [14].

In addition, we design ensembles of various CNN-based classifiers to detect GAN-created face images. Based on our initial observations and experiments, surprisingly neural networks with great depth such as Xception [2], DenseNet [12, 34] performed poorly on small image sizes such as 64×64. Therefore, we developed our own ensemble model, a shallow convolutional neural network (CNN) architecture. The details of architectures are provided in Table. 1, where we refer to our approach as *Shallow Convolutional Network (ShallowNet)*.

In ShallowNet, we use L2 kernel regularizer of 0.0001 in each Conv2D layer. We also use various batch normalization, dropout values, and Max Pooling methods as shown in Table 1. ShallowNet is able to detect the subtle difference between real and fake images even on tiny images such as 64×64, which are impossible for human eyes to distinguish, and provides very high accuracy. We developed three different versions of ShallowNet with different layer settings. ShallowNetV1 has lower performance on smaller size images. Hence, in V2 and V3, we developed more shallower architectures, where V2 and V3 are quite similar in depth. But the major difference is the introduction of Max pooling layer in V3, which yields better performance on small size images. Another advantage of our approach is that, due to the shallow layers, the training time is significantly reduced.

The end-to-end classification pipeline is shown in 3. For training, the images in PGGAN and CelebA dataset are given

ShallowNetV1	ShallowNetV2	ShallowNetV3
Conv-BN-ReLU-DO-Conv-BN-ReLU-DO-Conv-BN-ReLU-MP-DO	Conv-ReLU-DO-Conv-ReLU-DO-Conv-ReLU-DO	Conv-ReLU-DO-Conv-ReLU-DO-Conv-ReLU-MP-DO
Conv-BN-ReLU-DO-Conv-BN-ReLU-DO-Conv-BN-ReLU-MP-DO		
Conv-BN-ReLU-DO-Conv-BN-ReLU-DO-Conv-BN-ReLU-MP-DO	Conv-ReLU-DO-Conv-ReLU-DO-Conv-ReLU-DO	Conv-ReLU-DO-Conv-ReLU-DO-Conv-ReLU-MP-DO
Conv-BN-ReLU-DO-Conv-BN-ReLU-DO-Conv-BN-ReLU-MP-DO		
Conv-BN-ReLU-DO-Conv-BN-ReLU-DO-Conv-BN-ReLU-MP-DO	Conv-ReLU-DO-Conv-ReLU-DO	Conv-ReLU-DO-Conv-ReLU-DO
Conv-BN-ReLU-DO-Conv-BN-ReLU-DO		
F-D-ReLU-BN-DO-D-S	F-D-ReLU-BN-DO-D-S	F-D-ReLU-BN-DO-D-S

Table 1: ShallowNet architectures for GANs generated image detection. Each row represents a block in the architecture. L2 kernel regularizer of 0.0001 is used in each Conv2D layer. (Note: Conv=Conv2D, BN=Batch Normalization, DO=Dropout, MP=MaxPooling, F=Flatten,D=Dense & S=Sigmoid)

'FAKE' and 'REAL' labels respectively, and then they are passed through deep neural networks to create a classification model. Our classification model is the ensemble of three different shallow models. We report our detection accuracy in result section.

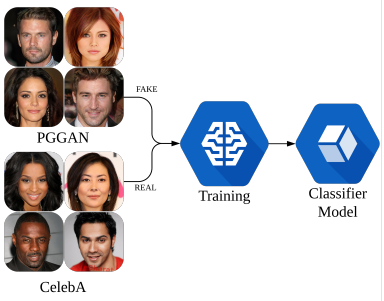

Figure 3: GANs Fake Image Detection, where PGGAN (FAKE) is the GANs generated fake images and CelebA (REAL)is used for normal real human faces

3.2 Fake face image creation by humans

Although it is better to automate the fake face image creation process, our another goal is to detect fake images created by humans. In particular, we aim to detect very sophisticated fake images created by humans. Unfortunately, there are not many available human-created fake face image datasets. Hence, several members of our lab manually created the fake face images using Adobe Photoshop CS6. The initial input images are collected from Google and Bing image search including human faces with diverse viewing angles, where we used noncommercial reuse with modifications search setting for searching. We consider men and women of different ages and races. In addition, we include more challenging samples with strong make-ups, glasses, sunglasses, and hats. While many images consists of one person's face, we use several images with multiple people to create several fake faces, as shown in Figs. 5.(f), and 6.

Also, as shown in Fig. 4, we divided the quality of creating images into three levels based on different editing complexity: 1) (Lv.1-image) crop and paste, 2) (Lv.2-image) crop, paste and smooth edges, and 3) (Lv.3-image) crop, paste, smooth edges, and adjust the level of color and light. The Lv.1-image is easy to recognize by normal people, and Lv.3-image is more difficult as shown in Fig. 4. Moreover, six different ranges of fake region modifications are performed as shown in Fig. 5: (a) one part of a face (e.g. eye, nose, and mouth), (b) two or more parts of a face (e.g. eyes, and nose and mouth), (c) half of a face, (d) a whole face, (e) extra addition (e.g. sunglasses, and hair), and (f) several face modification in an image.

Figure 4: Three quality levels of creating images: Lv.1 is cropped and pasted, Lv.2 is cropped, pasted and smoothed edges, and Lv.3 is cropped, pasted, smoothed edges, and adjusted color and light levels.

3.3 Detecting human-created fake face images

In our approach, we do not use meta-data as they can be forged as well. And we only distinguish fake and genuine images with only RGB channel information. We divide our detection method into two stages, where the first stage is to perform pre-processing to crop and filter face regions.

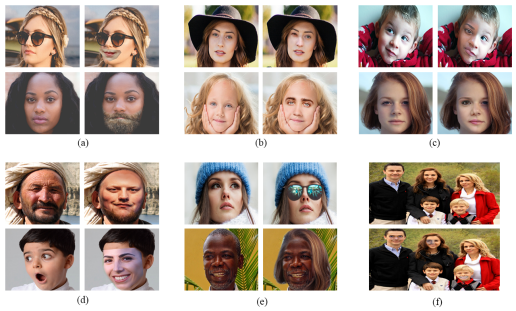

Figure 5: Range of fake parts, where left is original image and the right is fake. (a) one part of a face, (b) two or more parts of a face, (c) half of a face, (d)the whole face, (e) extra addition (e.g. sunglasses, and hair), and (f) several faces in an image.

Once we obtain the cropped and aligned faces, we train our classifier model to distinguish fake images created by humans from unmodified real images.

Cropping face regions: To distinguish human-created fake faces from real faces, face region must be inspected first. We used the following state-of-art face detection algorithms: SeetaFace engine [25], MTCNN [8, 31], YOLO [21] and Dlib [16]. In the presence of multiple faces, as shown in Fig. 6, missed detection and false positive can occurs. Based our initial experiments, we found that MTCNN performed the best with the low false positives, a missed detection.

Noise Filtering Heuristic: We further perform a noise filtering algorithm to reduce false positives of detected face regions. We assume that typically faces are located in the middle of the image, where it would be the largest part. Therefore, our algorithm compares all detected face regions with the largest detected face area and ignores the face regions smaller than 1/3 of that largest region. All the face regions after the noise-filtering are used as an input (green boxes) as shown in Fig. 6, while faces in red boxes are ignored. For every cropped face input, the classifier calculates the probability of being fake and takes the maximum probability among all inputs to finally determine whether the image is fake or not.

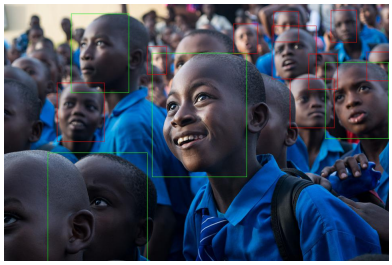

Figure 6: An example of MTCNN face detection and noise filtering, where the faces in red rectangles are ignored by the filtering algorithm, and the faces in green are used as an input to the classifier.

Classifier Training: We present our end-to-end fake image detection pipeline in Fig.7. We trained the following CNN-based models on our dataset to detect fake face images: VGG16 [26], VGG19 [26], ResNet [11], DenseNet [12], NASNet [34] XceptionNet [2], and ShallowNetV1. For training, we provided the cropped face images into each model separately and created classifiers for them. Every classifier is assigned with the same task of calculating the probability of an image being fake. We also used different image sizes from 32×32, 256×256 to examine how the image resolution will impact the accuracy. We present our detection performance in next section.

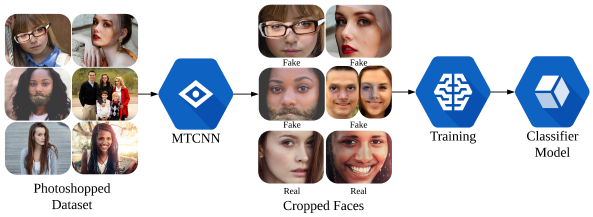

Figure 7: The end-to-end pipeline to distinguish fake human-created images from real images.

4 EVALUATION

We performed extensive experimentation on both scenarios. The details of experiment along with their results are given in the following sections.

4.1 GANs-Generated Image Detection

To discriminate GANs-generated synthetic images from real celebrity images, we trained different deep learning models using Keras python deep learning library [3] and evaluated their performance with test dataset. We used 200K images for training (20% for validation) and 18K for testing so that the distribution of both classes is 50:50. The experiments are categorized by the resolution of images that were used for training: 64×64, 128×128, 256×256, and 1024×1024 as shown in Table 2. We used different ensembles of these methods to provide more diversity to our discriminators. And each model takes an image as an input and outputs the probability of an input image being GANs-generated image or not. We used the Area Under the Receiver Operating Characteristic curve (AUROC) [5] to measure performance, as shown in Table 2 The AUROC is used to measure the practicality of a model and a greater area indicates a more practical model, which are significantly above random chance.

As expected, the performance of ShallowNet outperformed other neural network models as shown in Table 2. We achieved the best performance using an ensemble model of ShallowNetV1 and ShallowNetV3 as shown in Table 2 with 93.99% to 99.99% accuracy. Our results clearly indicate that it is quite difficult to detect the difference between real and GANs-generated images at lower resolutions with the state-of- the-art deep neural network architectures such as XceptionNet as shown in Table 2. Deeper network such as XceptionNet and

NASNet are quite huge for low resolution image. Hence, we found a gradual decrease in accuracy as the resolution of image becomes lower, while all ShallowNet models outperform all other approaches in 1024×1024 image size. In particular, the ensemble improves the overall performance across all image sizes. Hence, except tiny 64×64 image size, our approaches perform very well for detecting GAN-generated images.

	AUROC (%)			
Method	**64×64**	**128×128**	**256×256**	**1024×1024**
VGG19	56.69	55.13	57.13	60.13
XceptionNet	79.32	79.03	82.03	85.03
NASNet	83.55	90.55	92.55	96.55
ShallowNetV1	84.94	98.12	99.82	99.99
ShallowNetV2	79.82	99.98	99.99	99.99
ShallowNetV3	90.85	99.99	99.99	99.99
Ensemble ShallowNet (V1 & V3)	**93.99**	**99.99**	**99.99**	**99.99**

Table 2: AUROC scores of different deep learning models and their comparison with ShallowNet

4.2 Human-Created Image Detection

We evaluated the following DNN models to detect human-created fake images: XceptionNet, VGG19, ResNet, NASNet, and ShallowNet (SNv1). In addition, we also tried linear classifier, XGBOOST[1], which provides the gradient boosting framework with simple CNN model to extract the features. In addition, we evaluated different cropped and noise filtered input face images from 32×32 to 256×256 pixels to examine the effect of different input image sizes vs. performance. For testing, we tested 85 test images (52 fake and 33 real images) and measured the AUROC scores.

Different CNN-based models: To compare the performance among different models, we fixed the input test image size to 128×128 pixels, which is the most common input image sizes that most classifiers can accept from MTCNN outputs. With the fixed input image size, we can determine the best performing model. The AUROC score results are provided in Fig.8, where VGG19 was 51% slightly better than random guess, ResNet50 was 49.1%, which was the lowest and worse than random guess. XceptionNet outperformed other competing techniques with 74.9% AUROC score. The AUROC score for SNv1, CNN+XGBOOST, and NASNet was 66.3%, 52.4%, and 59% respectively. Except XceptionNet, all other approaches did not perform well. We believe XceptionNet's depth-wise separable convolutions modules better enabled the improved detection of facial changes.

Impacts of different input image sizes: We focus on capturing differing performance with varying cropped input image sizes with the best performing XceptionNet model. We hypothesize that different image sizes would yield different detection performance similar to GANs case. Therefore, we train the classifier models for distinguishing fake and real

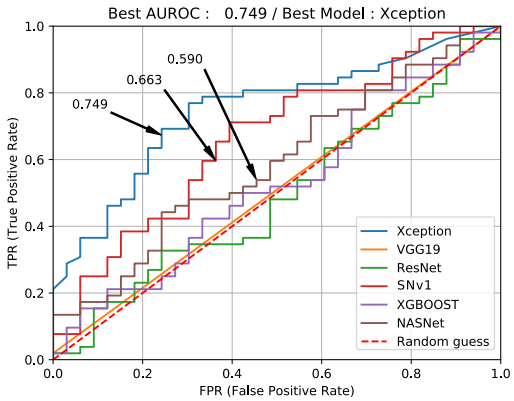

Figure 8: Human-created fake image detection with different CNN-based models with 128x128 image size

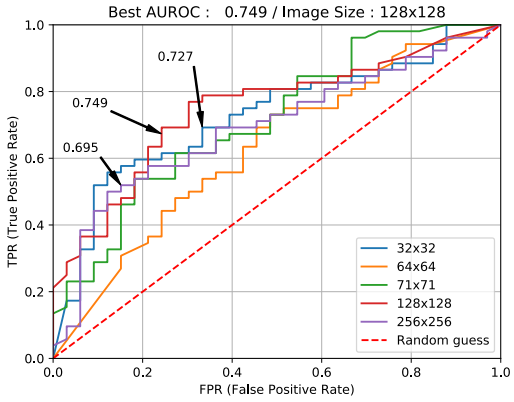

Figure 9: Human-created fake image detection withing varying image sizes using XceptionNet

images with the following four different cropped input image sizes: 32×32, 64×64, 71×71, and 128×128. The AUROC-curve result is shown in Fig. 9. The 128×128 images perform the best with 74.9% AUROC score, while 32×32, 64×64, and 71×71 achieved 72.6%, 63%, and 71.4% AUROC score, respectively. We also trained XceptionNet with a rather smaller dataset of 256×256 size image to examine, if a larger image size can yield better accuracy. The 256×256 achieved 69.5% AUROC score which is quite close to 74.9% of 128x128, which showed that with more training examples the higher resolution images can yield better result on our method.

5 CONCLUSION

We developed the neural network based classifiers to detect both GANs and human-created fake face images without resorting any meta-data information. Our preliminary results show the promising results in detecting GANs generated images with high accuracy with our proposed ShallowNet. While more research is needed for detecting human-created

fake images due to various complexity in fake image generation, we believe more training data can help improve performance. For future work, we plan to further enhance our face detection and noise-filtering algorithms and produce more human-created fake face images. Also, we will train our models with different levels of photoshop which might potentially strengthen our results.

ACKNOWLEDGEMENT

This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the ICT Consilience Creative program(IITP-2017- R0346-16-1007) supervised by the IITP (Institute for Information & Communications Technology Promotion), and by NRF of Korea by the MSIT (NRF-2017R1C1B5076474).

REFERENCES

[1] Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. ACM, 785–794.
[2] François Chollet. 2017. Xception: Deep learning with depthwise separable convolutions. *arXiv preprint arXiv:1610.02357* (2017).
[3] François Chollet et al. 2015. Keras. https://keras.io.
[4] Jon Christian. 2018. Experts fear face swapping tech could start an international showdown. Retrieved July 5, 2018 from https://theoutline.com/post/3179/deepfake-videos-are-freaking-experts-out
[5] Hyoungseok Chu. 2017. AUROC. Retrieved Jun 25, 2018 from https://github.com/hyoungseokchu/AUROC
[6] Samantha Cole. 2018. We Are Truly Fucked: Everyone Is Making AI-Generated Fake Porn Now. *Vice* (25 Jan 2018). https://motherboard.vice.com/en_us/article/bjye8a/reddit-fake-porn-app-daisy-ridley
[7] Davide Cozzolino, Giovanni Poggi, and Luisa Verdoliva. 2017. Recasting residual-based local descriptors as convolutional neural networks: an application to image forgery detection. In *Proceedings of the 5th ACM Workshop on Information Hiding and Multimedia Security*. ACM, 159–164.
[8] Iván de Paz Centeno. 2018. MTCNN. Retrieved Jun 25, 2018 from https://github.com/ipazc/mtcnn
[9] Hany Farid. 2009. Exposing digital forgeries from JPEG ghosts. *IEEE Trans. Information Forensics and Security* 4, 1 (2009), 154–160.
[10] Mohammad Farukh Hashmi, Aaditya R Hambarde, and Avinash G Keskar. 2013. Copy move forgery detection using DWT and SIFT features. In *Intelligent Systems Design and Applications (ISDA), 2013 13th International Conference on*. IEEE, 188–193.
[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.
[12] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. 2017. Densely Connected Convolutional Networks. In *CVPR '17*, Vol. 1. 3.
[13] Multimedia Computing Lab in Kaist. 2015. MMC Image Forensic Tool. Retrieved July 22, 2018 from http://forensic.kaist.ac.kr/
[14] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. 2017. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196* (2017).
[15] A. Kashyap, B. Suresh, M. Agrawal, H. Gupta, and S. D. Joshi. 2015. Detection of splicing forgery using wavelet decomposition. In *International Conference on Computing, Communication Automation*. 843–848. https://doi.org/10.1109/CCAA.2015.7148492
[16] Davis E King. 2009. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research* 10 (July 2009), 1755–1758.
[17] Neal Krawetz. 2007. A Picture's Worth: Digital Image Analysis and Forensics. *Hacker Factor Solutions* (2007), 16–20.
[18] Neal Krawetz. 2012. FotoForensics. Retrieved July 5, 2018 from http://fotoforensics.com/
[19] Zhouchen Lin, Junfeng He, Xiaoou Tang, and Chi-Keung Tang. 2009. Fast, automatic and fine-grained tampered JPEG image detection via DCT coefficient analysis. *Pattern Recognition* 42, 11 (2009), 2492–2501. https://doi.org/10.1016/j.patcog.2009.03.019
[20] S. Murali, Govindraj B. Chittapur, H. S. Prabhakara, and Basavaraj S. Anami. 2013. Comparision and analysis of photo image forgery detection techniques. *arXiv preprint arXiv:1302.3119* (2013).
[21] Joseph Redmon and Ali Farhadi. 2018. YOLOv3: An Incremental Improvement. *arXiv preprint arXiv:1804.02767* (2018).
[22] Aja Romano. 2018. Jordan Peele's simulated Obama PSA is a double-edged warning against fake news. Retrieved August 15, 2018 from https://www.vox.com/2018/4/18/17252410/jordan-peele-obama-deepfake-buzzfeed
[23] Kevin Roose. 2018. Here Come the Fake Videos, Too. Retrieved July 5, 2018 from https://www.nytimes.com/2018/03/04/technology/fake-videos-deepfakes.html
[24] Andreas Rössler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. 2018. FaceForensics: A Large-scale Video Dataset for Forgery Detection in Human Faces. *arXiv preprint arXiv:1803.09179* (2018).
[25] seetaface. 2016. SeetaFaceEngine. Retrieved Jun 25, 2018 from https://github.com/seetaface/SeetaFaceEngine
[26] Karen Simonyan and Andrew Zisserman. 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv preprint arXiv:1409.1556* (2014), 1–14.
[27] Justus Thies, Michael Zollhofer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. 2016. Face2face: Real-time face capture and reenactment of rgb videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2387–2395.
[28] Wikipedia. 2018. Deepfake. Retrieved July 5, 2018 from https://en.wikipedia.org/wiki/Deepfake
[29] Jianquan Yang, Guopu Zhu, Jiwu Huang, and Xi Zhao. 2015. Estimating JPEG compression history of bitmaps based on factor histogram. *Digital Signal Processing* 41 (2015), 90–97. https://doi.org/10.1016/j.dsp.2015.03.014
[30] Saber Zerdoumi, Aznul Qalid Md Sabri, Amirrudin Kamsin, Ibrahim Abaker Targio Hashem, Abdullah Gani, Saqib Hakak, Mohammed Ali Al-Garadi, and Victor Chang. 2017. Image pattern recognition in big data: taxonomy and open challenges: survey. *Multimedia Tools and Applications* (2017), 1–31.
[31] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. 2016. Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks. *arXiv prerpint arXiv:1604.02878* (2016).
[32] Peng Zhou, Xintong Han, Vlad I. Morariu, and Larry S. Davis. 2018. Learning Rich Features for Image Manipulation Detection. *arXiv preprint arXiv:1805.04953* (2018).
[33] Xiaogang Wang Ziwei Liu, Ping Luo and Xiaoou Tang. 2015. Deep Learning Face Attributes in the Wild. In *Proceedings of International Conference on Computer Vision (ICCV '15)*.
[34] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V. Le. 2017. Learning Transferable Architectures for Scalable Image Recognition. *arXiv preprint arXiv:1707.07012* (2017).