



# STN-Net: A Robust GAN-Generated Face Detector

Tanusree Ghosh<sup>(✉)</sup> and Ruchira Naskar

Department of Information Technology, Indian Institute of Engineering Science and Technology, Shibpur 711103, India

2021itp001.tanusree@students.iiests.ac.in, ruchira@it.iiests.ac.in

**Abstract.** Massive advancements in Generative Artificial Intelligence in the recent years, have introduced hyper-realistic fake multimedia content. Where such technologies have become a boon to industries such as entertainment and gaming, malicious uses of the same in disseminating fabricated information eventually have invited serious social perils. Generative Adversarial Network (GAN) generated images, especially non-existent human facial images, lately have widely been used to disseminate propaganda and fake news in Online Social Networks (OSNs), by creating fake OSN profiles. Being visually indistinguishable from authentic images, GAN-generated image detection has become a massive challenge to the forensic community. Even though countermeasure solutions based on various Machine Learning (ML) and Deep Learning (DL) techniques have been proposed recently, most of their performance drops significantly for OSN-compressed images. Also, DL solutions based on Convolutional Neural Networks (CNN) tend to be highly complex and time-consuming for training.

This work proposes a solution to these problems by introducing *STN-Net*, a CNN classifier with an extremely reduced set of parameters, which adopts a carefully crafted minimal image feature set, computed based on *Sine Transformed Noise (STN)*. Despite having a much-reduced feature set compared to other State-of-the-Art (SOTA) CNN-based solutions, our model achieves very high detection accuracy (*average*  $\geq 99\%$ ). It also achieves promising detection performance on post-processed images, which mimic real-world OSN contexts.

## 1 Introduction

Although the origins of image forgery and manipulation can be traced in history as far back as the 1840s [19], contemporary technological advancements have eased forgery creation to a great extent. The invention of Generative Adversarial Networks (GANs) in 2014 [9] is considered one of the milestones in artificial image generation. Eventually, other GAN architectures like PGGAN [13], StyleGAN [15], StyleGAN2 [16], StyleGAN3 [14] etc. have further advanced the capabilities of GANs in generating hyper-realistic and high-quality images. Through easily accessible interfaces<sup>1</sup>, anyone can generate synthetic images in a matter of

<sup>1</sup> <https://thispersondoesnotexist.com/>.

seconds. Whereas such technologies have brought immense progress in various fields like the entertainment and gaming industries, illicit uses of the same have also raised concerns regarding the authenticity and trustworthiness of digital content<sup>2</sup>. Moreover, the prevalence of fake images has become a significant issue due to the ever-increasing presence of Online Social Networks (OSNs) in our daily lives. Illicit users often utilize OSN to spread disinformation and propaganda, potentially harming individuals and society as a whole<sup>3,4</sup>.

While earlier AI-generated face detection was highly dependent on visual inconsistency, like different eye colours in both eyes, asymmetric face shapes, irregular pupil shapes, etc. [5], with the technological advancements of StyleGAN, such artifacts have been largely omitted. A recent study [21] found that regular human observers find AI-generated faces more trustworthy than real faces. Having human accuracy of identifying synthetic faces around  $\approx 50\text{--}60\%$  makes them highly vulnerable to trusting fake content online. Hence, to identify GAN-generated images, automated detectors that rely on apparently ‘hidden’ characteristics of visually indistinguishable hyper-realistic GAN-generated synthetic images are of paramount need in the multimedia forensics community.

A few successful detectors have already been proposed in the literature.

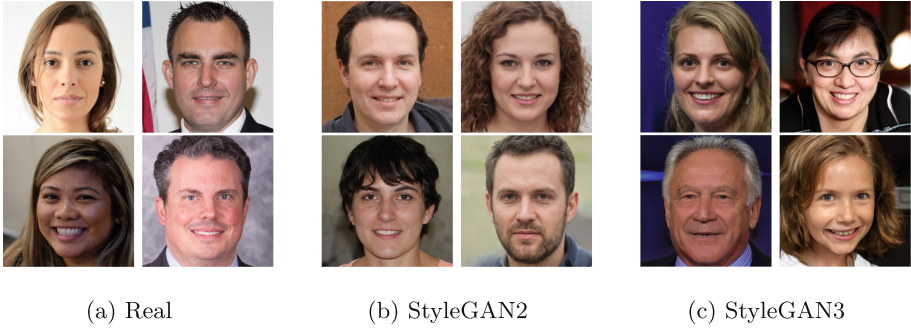
Most of them almost accurately (average detection accuracy as high as 99%) detect synthetic faces in the lab environment, where the training and testing dataset is pre-known [4, 7]. However, deploying these detectors in real-world scenarios remain challenging, as they often face performance degradation for OSN-circulated images. It happens due to the fact that images circulated through OSNs go through various compression algorithms and transformations, which can alter their statistical features and make them more difficult to detect using traditional methods. The specific operations performed by any OSN on images are usually unknown, posing a challenge for designing detectors that can successfully handle various image modifications encountered in real-life situations. A few recent works [22, 23, 31] have addressed this issue and studied the performance of their solutions on post-processed images. They use complex feature sets with deep CNN-based classifiers, which makes them hard to implement in resource-constrained platforms like edge devices. In this work, we formulate the synthetic image detection problem as a binary classification problem between real and fake faces. We propose a solution based on a hand-crafted feature set followed by a well-designed CNN. Our solution achieves a high average accuracy of 99.53% for images from our test set. We evaluate the performance of our solution on post-processed images to understand its real-world usability. Consisting of a minimal feature set compared to SOTA CNN-based solutions, our solution performs well in the context of post-processed images. Specifically, the main contributions of this paper are:

---

<sup>2</sup> <https://www.npr.org/2022/12/15/1143114122/ai-generated-fake-faces-have-become-a-hallmark-of-online-influence-operations>.

<sup>3</sup> <https://edition.cnn.com/2020/02/28/tech/fake-twitter-candidate-2020/index.html>.

<sup>4</sup> <https://edition.cnn.com/2020/02/20/tech/fake-faces-deepfake/index.html>.



**Fig. 1.** Example of Real and GAN-generated images from dataset (<https://github.com/NVlabs/fhq-dataset>), (<https://github.com/NVlabs/stylegan2>), (<https://github.com/NVlabs/stylegan3>)

- We introduce a novel feature set *Sine Transformed Noise (STN)* that enhances differentiating features between real and synthetic images. Our feature set, STN is of size  $m \times n \times 1$  for an RGB image of size  $m \times n \times 3$ . STN is the minimal-sized feature map compared to existing feature sets for detecting fake images using any CNN-based network.
- We introduce *STN-Net*, a CNN-based detector utilising STN feature set and augmented STN feature set, which uses very few parameters compared to existing CNN-based detectors while maintaining high detection accuracy. We compare STN-Net with several well-known CNN detectors in the field.
- Proposed STN-Net is tested on various post-processing conditions such as Median filtering, Gaussian Noise addition, Contrast Limited Adaptive Histogram Equalization (CLAHE), Average Blurring, Gamma Correction and Resizing with different parameters, as well as JPEG compression. Our extensive experimental results prove the effectiveness of the proposed network in the presence of such post-processing operations that images undergo in real-world scenarios.

The rest of this paper is organised as follows. Section 2 reviews the relevant related works in the field of GAN image detection. Section 3 presents the proposed STN-Net approach, including the generation of the STN feature set and the architecture of the proposed CNN-based detector. Section 4 presents the experimental setup and evaluation results, comparing STN-Net with other state-of-the-art detectors, while Sect. 5 concludes the paper and provides directions for future research in the field.

## 2 Related Works

While the primary GAN model [9] was able to generate synthetic images that were identifiable with bare eyes, the advanced GAN model StyleGAN and their variants [14–16] generated images have become visually hardly distinguishable

from authentic images. The existing community of Digital Image Forensics started synthetic image detection. GAN image detection is considered a binary classification problem, similar to binary detection of forged images [28]. Having a strong similarity with traditional image forgery detection, most of the earlier solutions consisted of steganalysis-based features like Co-occurrence matrix [20], SRM [17] etc. Later, semantic inconsistency-related features like Corneal specular highlight [18], Landmark locations [32], Irregular pupil shape [11] etc. were explored to identify GAN-faces. Once explored, such artefacts could be used by regular users in their daily lives to some extent to identify fakes. However, with improvements in GAN architecture, such visual inconsistency-based artefacts have been reduced to a great extent, making synthetic images hard to detect.

Another approach to GAN-image detection is purely deep neural network-based. Automatically learned features by DL-based classifiers [4] have been proven to be very successful in detecting synthetic faces. However, a major hurdle for such classifiers is the architectural setting of GAN models. GANs simultaneously use a generator and a discriminator network to learn the data distribution of real datasets and mimic them to generate new data. If a purely DL-based GAN image detector is used as a discriminator inside GAN architecture, eventually, the generator module of GAN will be trained to fool the discriminator, making the DL-based detector useless. As a result, using hand-crafted features in conjunction with deep neural networks (DNN) has gained traction as a prevalent solution method in contemporary contexts [3, 22, 23]. Spatial domain features are primarily used in such solutions. From a different perspective, symmetries in GAN-faces in the frequency domain have also been explored [7, 30].

However, even though progress in GAN image detection is gaining momentum lately, the performance of existing GAN-image detectors in real-world scenarios remains a big challenge. Marra et al. [17] first explored the problem of performance degradation of synthetic image detectors while tested on OSN-like compressed images. Recently, Chen et al. [3] proposed the inclusion of two modules from multiple colour domains, named block attention module and a multi-layer feature aggregation module, into the Xception model to increase robustness against such post-processing degradation. Lately, another problem domain related to GAN-image detection has been explored: The generalisation problem [6]. In real-world scenarios, guessing the exact source model of any GAN image is difficult. Hence, any practical detector should be capable of detecting fake faces even though the training and test datasets mismatch. The work [10] contains a performance comparison between existing solutions.

Recently, the study of anti-forensics in the context of detecting GAN-generated images has gained significance. Carlini et al. [1] explored the anti-forensics aspect of GAN-generated image detectors. They explored five white-box and black-box attack scenarios that severely degraded the performance of GAN-image detection. As our scope for this work is to propose a GAN-image detector that performs well in OSN-context, we consider only common OSN-specific perturbations, as discussed in earlier similar studies [22, 23]. The study

of the detector’s robustness against additional white-box and black-box attacks is reserved for future research endeavours.

We can infer from the high detection performance of existing solutions that detecting GAN faces is not a big hurdle these days while training and testing dataset matches. The bigger problem remains maintaining the detection performance while test images have different statistical characteristics due to unknown manipulation. Hence, we aim to propose a GAN-generated face detection solution robust to image manipulations. In that direction, we design a reliable GAN-face detector combining hand-crafted features with a DNN-based classifier, implementable in real-world scenarios.

### 3 Proposed Methodology

As shown in Fig. 2, our proposed solution consists of two major blocks of operations: *Preprocessing and Feature Extraction* followed by *Deep-Learning-based Classifier*.

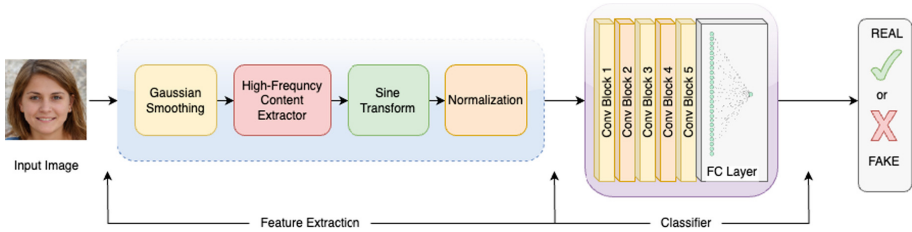


Fig. 2. Proposed Framework

#### 3.1 Preprocessing

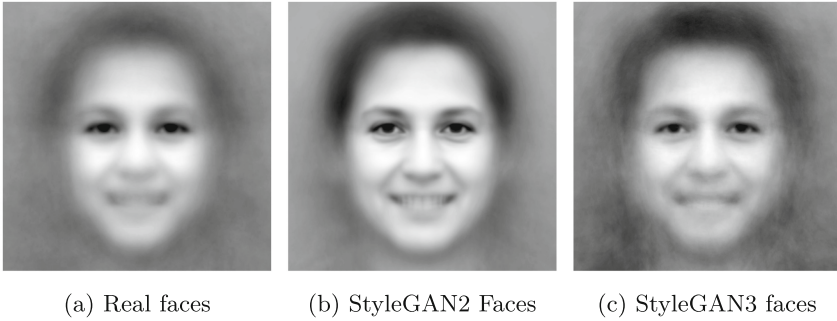
Given an input RGB image  $I(x, y)$ , which represents the intensity of the image at coordinates  $x$  and  $y$  in the spatial domain, we first convert it to grayscale using the Eq. 1. The grayscale value at a pixel location  $(x, y)$  is denoted as  $I_{\text{gray}}(x, y)$ , and it is calculated as a weighted sum of the red, green, and blue channels.

$$I_{\text{gray}}(x, y) = 0.2989 \times R(x, y) + 0.5870 \times G(x, y) + 0.1140 \times B(x, y) \quad (1)$$

where  $R(x, y)$ ,  $G(x, y)$ , and  $B(x, y)$  are the red, green, and blue intensity values of the pixel at coordinates  $(x, y)$ , respectively.

Although a few earlier works [2, 8, 20, 22] have explored strong discriminating features in colour-domain statistics, such methods usually have large feature sizes. We convert a three-channel RGB image to a one-channel grayscale image based on the finding in Fig. 3, which we explain below. Here, we calculate pixel-wise mean values from 400 real images (Fig. 3a), 400 StyleGAN2 generated

images (Fig. 3b) and 400 StyleGAN3 generated images (Fig. 3c). As all synthesised images use the same FFHQ dataset for generating synthetic images, it is trivial that generated photos from both generative architectures look similar to originals, as depicted in Fig. 1. However, as shown in Fig. 3, there are visible differences between the three mean images. GAN-generated images possess much more structure than real ones, especially the mean of StyleGAN2 images, which have a visible common structure. This hints towards the presence of model-specific similar high-frequency components in each type of GAN, which are different from other GAN models and real images. Hence, it is shown that even single-channel grayscale images possess visually discriminating features.



**Fig. 3.** Pixel-wise average of grayscale images

In light of the insights gathered from this discourse, we focus on identifying robust feature representations with high-frequency content in grayscale images generated by GANs. Laplacian of Gaussian (LoG) is a well-known method in image processing for high-frequency image feature extraction in the context of faces [24, 29].

Hereby, we employ Gaussian Blur with kernel size  $3 \times 3$  and 0 standard deviation ( $\sigma$ ) followed by Laplacian kernel to extract high-frequency information. Gaussian kernel, as shown in Eq. 2, is computed using the Gaussian distribution formula. Smoothened image  $I_{smoothed}(x, y)$  is generated by convolving this kernel with  $I_{gray}(x, y)$ , using Eq. 3, where  $i$  and  $j$  are the indices of the kernel's rows and columns, respectively. Here Gaussian blur is used to mitigate the influence of random high-frequency noise on images.

$$kernel(i, j) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{i^2 + j^2}{2\sigma^2}\right) \quad (2)$$

$$I_{smoothed}(x, y) = \sum_{i=-1}^1 \sum_{j=-1}^1 I_{gray}(x + i, y + j) \times kernel(i, j) \quad (3)$$

On smoothened image  $I_{smoothed}(x, y)$ , the Laplacian operator is applied to compute the *Laplacian Response*, which denotes the second order derivative of

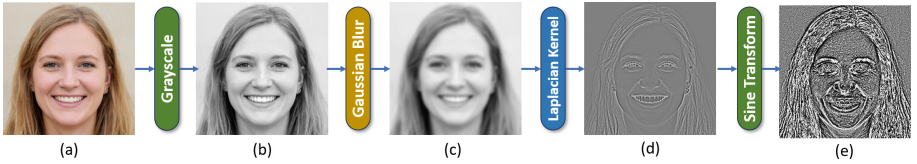
the image intensity, for the spatial coordinates  $x$  and  $y$ , hence obtaining rate of intensity change.

Mathematically, the Laplacian kernel (L) is defined as:

$$L = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

The Laplacian response ( $I_{laplacian}(x, y)$ ) at a pixel location  $(x, y)$  is calculated by convolving the smoothed image  $I_{smoothed}(x, y)$  with the Laplacian kernel using Eq. 4.

$$I_{laplacian}(x, y) = \sum_{i=-1}^1 \sum_{j=-1}^1 I_{smoothed}(x + i, y + j) \times L(i, j) \quad (4)$$



**Fig. 4.** Visualisation of STN-Feature evolution: (a) Input RGB Image; (b) Grayscaled Image; (c) Gaussian Blurred Image; (d) Laplacian Transformed Image; (f) Sine Transformed Feature

### 3.2 Sine-Transformed Noise

Primarily, sine transformations are used in the frequency domain accompanied by Fourier Transform. However, we use sine transformation on direct pixel values obtained by  $I_{laplacian}(x, y)$  using Eq. 5. As shown in Fig. 4, Sine transformation preserves texture and high-frequency information at the pixel level. After extensive experiments with different frequencies, we found the best performance with frequency ten. We have shown a visual representation of feature  $I_{STN}(x, y)$  in Fig. 5 for varying frequencies from 1 to 50.

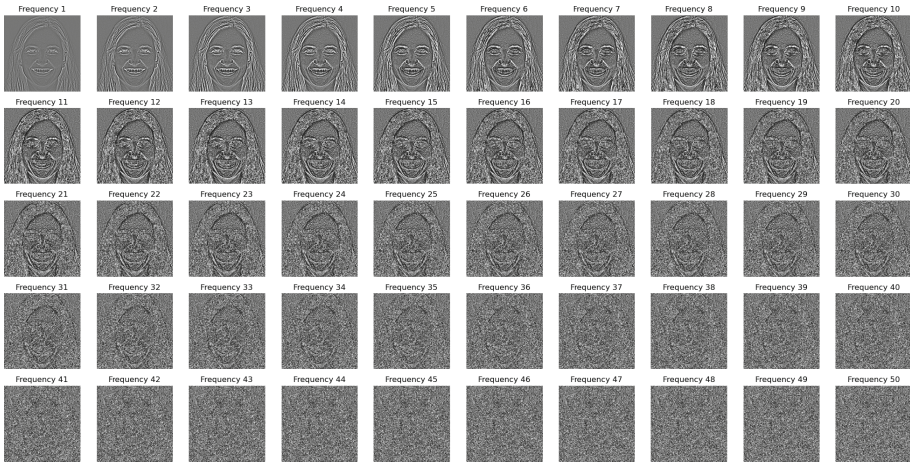
$$I_{STN}(x, y) = \sin \left( 2\pi \times frequency \times \frac{I_{laplacian}(x, y)}{255.0} \right) \quad (5)$$

We utilize the normalized form of the  $I_{STN}(x, y)$  as feature, as shown in Eq. 6.

$$I_{STN_{Normalized}}(x, y) = \frac{I_{STN}(x, y) - \min(I_{STN}(x, y))}{\max(I_{STN}(x, y)) - \min(I_{STN}(x, y))} \times 255.0 \quad (6)$$

Given an RGB image, the evolution of feature  $I_{STN}(x, y)$  is pictorially shown in Fig. 4.

It is evident from the visualisation that for frequency 1, the feature set visually depicts merely high-frequency edge information. As the frequency increases to 10, the features get much more accentuated.



**Fig. 5.** Visualisation of STN-feature for various frequency values

### 3.3 Classifier

Convolutional Neural Networks (CNN) are conventionally used for image classification for their efficient capability of extracting features hierarchically. As shown in Fig. 6, our proposed CNN-based classifier consists of five convolution blocks. Each block consists of a *Convolutional Layer* with  $3 \times 3$  sized kernel, followed by a *Batch Normalization* layer and *Max Pooling Layer* with window size  $2 \times 2$ . The batch Normalization layer is used to increase stability in training, whereas the Max Pooling Layer is used to downsample the feature size after each convolution for computational efficiency.

The number of filters in the convolutional layer varies in each block. Block 1 has 8 filters, Block 2 has 16 filters, Block 3 has 32 filters, Block 4 has 64 filters, and Block 5 has 128 filters. It is because, with deeper levels, the network needs to capture more complex and abstract features, which require a larger number of filters. In each block, ‘ReLU’ is used as an activation function to induce non-linearity. The last convolution block is followed by a flattening layer, which reshapes the high-dimensional feature maps into a one-dimensional vector, then connected to a fully connected (dense) layer with 64 neurons. The final dense layer consists of a single neuron, representing the output layer of the classifier with the sigmoid activation function.



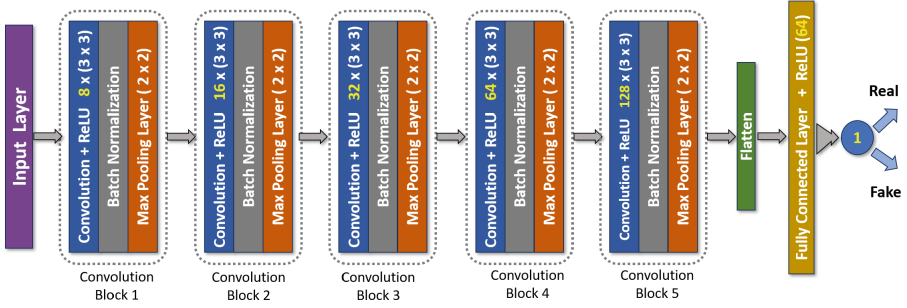


Fig. 6. STN-Net Architecture

## 4 Experimental Results and Analysis

We study the detection performance of our solutions on two versions of StyleGAN-generated images. StyleGAN2-generated images are widely available throughout the OSN and easily accessible to ordinary people, whereas StyleGAN3 is the improved version of StyleGAN2, with exceptionally good hyperphotorealism. Detection of both StyleGAN model-generated images is essential in identifying manipulated or fake images on online social networks in current time.

### 4.1 Solution Models

We experiment with our proposed feature set and CNN in three different settings:

- In the Baseline configuration, our proposed CNN model is utilized for binary classification without preprocessing. In this case, the input provided to our model consists of only grayscale image  $I_{gray}(x, y)$ . The performance of this model serves as an indicator of the effectiveness and quality of our CNN framework. It is further mentioned as ‘*Only CNN*’ setting.
- In the second case, we examine the influence of various preprocessing stages: Gaussian Blurring (GB), Laplacian Transformation (LT) and Sine Transformation (ST) separately. In all cases, the classifier is fixed. It is further mentioned as ‘*Feature + CNN*’ setting.
  - GB + ST: Sine Transformation is directly applied on Gaussian Blur operated grayscale image.
  - LT + ST: The input grayscale image undergoes two transformations: the Laplacian Transformation and the Sine Transformation.
  - GB + LT: The grayscale image undergoes Gaussian blur and subsequent Laplacian transformation without any sine transformation applied.
  - GB + LT + ST: Input image in preprocessed with all above-discussed operations. Along with CNN, this model is STN-Net (Single Layer).

- In the last case, input to CNN consists of two layers of GB + LT + ST feature set stacked together. While all other preprocessing parameters are kept the same, the kernel for the Gaussian Blur of the second layer is set to 5. It is further mentioned as ‘*Augmented Feature + CNN*’ setting. We call this model STN-Net (Dual Layer).

## 4.2 Dataset

- For StyleGAN2 face detection, we select 20,000 random real-face images from the FFHQ dataset [15] and 20,000 random synthetic face images from the StyleGAN2 dataset [16]. We divide these 40,000 images: 28,000 for training, 8,000 for validation and 4,000 for testing. We consider image size  $256 \times 256$  for all experiments.
- For StyleGAN3 face detection, we select 1,592 StyleGAN3 face images generated from the FFHQ dataset, provided by official dataset [14], and we collect the same number of images from the FFHQ dataset as real face images. We further divide these images: 2,228 for training, 638 for validation and 318 for testing. We use  $256 \times 256$  sized images in all experiments.

## 4.3 Settings

We use the ‘Adam’ optimizer and ‘Binary Cross-Entropy’ loss function for all experiments. We train our classifier for 40 epochs for each case of StyleGAN2 and 100 epochs for StyleGAN3 detector. We use a Learning Rate (LR) scheduler for efficient convergence. For the first ten epochs, LR is set to  $1 \times 10^{-3}$ ; for the subsequent ten epochs, LR is multiplied by  $1 \times 10^{-1}$ ; while for the last 20 epochs, LR is multiplied by  $1 \times 10^{-3}$ .

## 4.4 Performance Evaluation

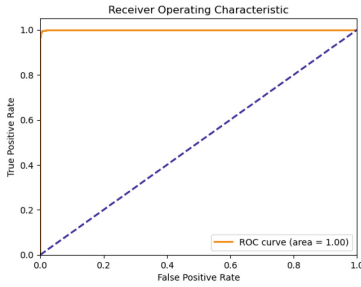
For detection performance evaluation, we consider five metrics: Accuracy, Area Under the ROC Curve (AUC) Score, Precision, Recall and F-1 Score. *Accuracy*, the most common metric for classification tasks, is the ratio of correctly predicted instances to the total number of instances in the test dataset. *Precision* is the ratio of correctly predicted positive instances, called true positives to the total number of predicted positive instances (sum of true positives and false positives). *Recall*, also known as Sensitivity or True Positive Rate, is the ratio of correctly predicted positive instances (true positives) to the total number of actual positive instances (sum of true positives and false negatives). The ROC curve plots the True Positive Rate (Recall) against the False Positive Rate. AUC measures the area under this curve, providing a single value that represents the overall discriminative power of a model. An AUC of 1 indicates perfect separation between the classes, while an AUC of 0.5 indicates random guessing. In the context of our problem of detecting real and synthetic images, as discussed earlier, we utilize a balanced dataset for training and testing our models. Hence,

accuracy is considered a suitable metric. However, as our main focus is not to misclassify any fake images as real, ‘Recall’ is chosen as a metric. Higher precision signifies fewer ‘fake’ predictions that are ‘real’.

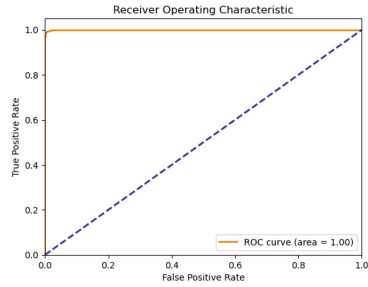
As shown in Table 1, our baseline model, with only grayscale image as input, performs quite well in discriminating StyleGAN2 images from Real images with an average accuracy greater than 91%, while performing exceptionally well in

**Table 1.** Detection Performances

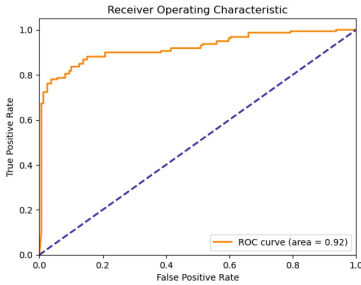
Detector type	GB	LT	ST	StyleGAN2					StyleGAN3				
				Accuracy	AUC	Precision	Recall	F1-Score	Accuracy	AUC	Precision	Recall	F1-Score
Baseline (Gray)	-	-	-	91.79	97.51	94.04	89.49	91.47	80.90	88.70	75.64	<b>89.96</b>	81.77
	Y	-	Y	98.19	99.80	98.11	98.31	98.14	80.80	90.90	77.78	86.21	81.31
1-layer (Gray)	-	Y	Y	89.89	95.86	92.23	85.34	88.33	80.77	86.66	80.25	81.63	80.54
	Y	Y	-	98.39	99.88	98.97	97.76	98.26	81.38	84.53	77.15	89.23	82.37
	Y	Y	Y	99.38	99.96	99.45	99.08	99.24	<b>84.21</b>	92.34	<b>82.14</b>	87.35	<b>84.28</b>
2-layer (Augmented)	Y	Y	Y	<b>99.53</b>	<b>99.97</b>	<b>99.55</b>	<b>99.28</b>	<b>99.43</b>	83.49	<b>93.17</b>	80.18	88.71	83.85



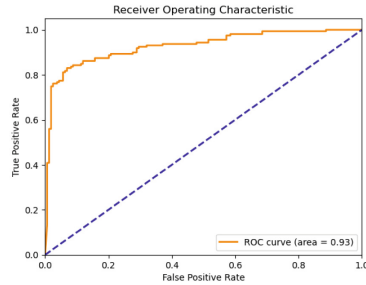
(a) STN-Net(SL)



(b) STN-Net(DL)



(c) STN-Net(SL)



(d) STN-Net(DL)

**Fig. 7.** ROC curve for Single Layer and Dual Layer STN-Net: (a) and (b) for StyleGAN2 dataset, (c) and (d) for StyleGAN3 dataset

terms of Precision and Recall for StyleGAN3 images. As discussed earlier, this result proves the strength of our designed CNN.

It is very evident from Table 1 that models with Gaussian Blur transformation bring significant performance enhancement to CNN model. While Sine transformation is constant in the first two cases of ‘Feature + CNN’ setting, only LT transformed models face sharp performance degradation, compared with only GB models for both StyleGAN2 and StyleGAN3. We may take the insight that LT alone enhances high-frequency components, many of which may be unnecessary, while GB smoothens them, making CNN learn more generalised features. GB + LT performs slightly better than GB + ST in most of the metrics for StyleGAN2 images. For StyleGAN3 we perform similar tests. Even though GB + LT performed well together compared to their single transformation performance, GB + LT + ST outperformed them, proving the effectiveness of the proposed Sine Transformation on synthetic image detection. For StyleGAN3 image detection, this feature pipeline achieves the best performance with an average accuracy of 84.21%.

Following the earlier result proving the effectiveness of GB transformation, in ‘Augmented Feature + CNN’ case, we use two layers of GB + LT + ST feature as discussed earlier. It achieves the best performance for all metrics with 99.53% average accuracy for StyleGAN2 case. For StyleGAN3, two layers of GB + LT + ST feature achieve best AUC score.

We show ROC curve for STN-Net for both StyleGAN2 and StyleGAN3 datasets in Fig. 7.

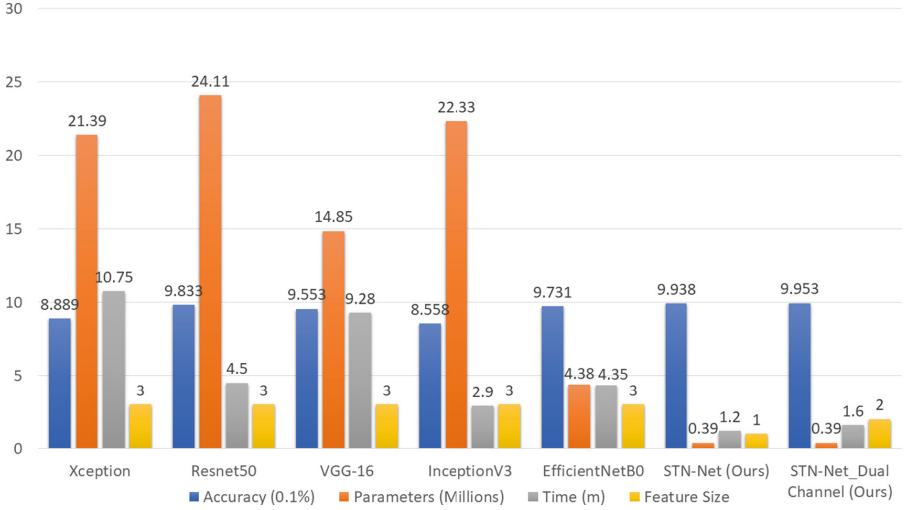
#### 4.5 Performance Comparison with Other Solutions

Table 2 compares detection performance for StyleGAN2 images with other State-of-the-art (SOTA) solutions. Our best model performs better than most SOTA models, only slightly less than the solution proposed by Qiao et al. [23]. However, our model attains such performance with the minimum size feature set of 1 grayscale channel compared with other solutions. While their solution [23] utilizes colour domain information with ten channels through a CNN, other works [20, 22] utilize co-occurrence and cross co-occurrence metrics with three channels and six channels features respectively. Chen et al. [3] fuse information from multiple colour domains with a six channels-sized feature set. Frank et al. [7] use frequency domain artefacts.

**Table 2.** Performance comparison with SOTA

Metric	Ours (Best Model)	Qiao [23]	Nowroozi [22]	Nataraj [20]	Chen [3]	Frank [7]
Accuracy (%)	99.53	99.80	99.33	96.11	97.70	98.58

As shown in Fig. 8, we compare the performance of our two models, STN-Net (Single-Layer) and STN-Net (Dual-Layer) with other available transfer-learning-based models: XceptionNet [4], ResNet50 [12], VGG-16 [25], InceptionV3 [26]



**Fig. 8.** Comparison with other CNNs

and EfficientNetB0 [27] for StyleGAN2 image detection. In all these cases, we import pre-trained models from the Keras library. We use weights from ‘imagenet’. For all these models, on top of base pre-trained models, we add a *Global Average Pooling Layer* followed by a Fully Connected Layer with 256 neurons. The final layer of the model is added, consisting of a single neuron, which is activated using the sigmoid activation function. Raw RGB image is provided as input in every case. All models are compiled using the Adam optimizer and the binary cross-entropy loss function.

In Fig. 8, Accuracy is multiplied by 0.1 for better visibility in the chart. Parameters are shown in millions, and the Average time to train per epoch is shown in minutes. Among transfer learning models, ResNet50 performs best with the detection accuracy of 98.33%, having the largest number of parameters, 24.11 million and three channels of feature space. Our model having  $\approx 1.61\%$  parameters of ResNet50, with grayscale images, achieves 99.53% detection accuracy.

As shown in Table 3, we further compare the computational complexity of the above-mentioned transfer learning-based models with our proposed solutions regarding Floating point operations (FLOPs) and Average Latency. We have calculated the Average Latency as the average inference time for 500 test samples in all cases. Further, for both our solutions, we have calculated latency with the preprocessing steps included, i.e., the calculation of the STN feature. It is evident from both Fig. 8 and Table 3 that our solutions obtain excellent performance despite having the lowest number of model parameters, training time, FLOPs and output latency.

**Table 3.** Computational Complexity Comparison with other CNN models

Model	FLOPs (Billion)	Latency (Only DL model) [Millisecond]	Latency (With Preprocessing) [Millisecond]
STN-Net (Single Layer)	<b>0.06</b>	<b>31.30</b>	32.87
STN-Net (Dual Layer)	0.07	33.36	34.46
Xception	5.95	35.11	–
VGG-16	20.05	36.48	–
InceptionV3	3.85	36.95	–
EfficientNetB0	0.51	60.88	–
Resnet50	5.04	87.74	–

**Table 4.** Detection performance on post-processed images

Operations	Parameters	Baseline Model	STN-Net	Dual layer STN-Net	CSC-Net [23]	Co-Net [20]	CC-Net [22]
Median Filter	$3 \times 3$	91.62	96.92	<u>98.78</u>	<b>99.35</b>	81.48	85.13
	$5 \times 5$	<u>91.12</u>	63.77	78.92	<b>93.80</b>	75.98	83.65
Gaussian Noise	1.0	91.82	<u>99.21</u>	<b>99.38</b>	94.43	76.35	93.68
	2.0	91.84	<u>99.28</u>	<b>99.33</b>	74.25	76.73	96.80
CLAHE	$3 \times 3$	85.54	<u>97.10</u>	<b>98.99</b>	94.70	51.43	50.32
Average Blurring	$3 \times 3$	91.37	86.11	<u>94.84</u>	<b>97.30</b>	93.68	86.90
	$5 \times 5$	<b>90.40</b>	50.12	53.77	82.68	<u>88.23</u>	76.63
Gamma Correction	0.8	90.10	<b>99.28</b>	<u>99.26</u>	95.08	82.28	83.15
	0.9	91.69	<u>99.23</u>	<b>99.45</b>	98.00	87.23	90.98
	1.2	89.56	<b>99.40</b>	<u>99.36</u>	96.90	87.20	85.53
Resizing	0.5	<u>91.42</u>	72.40	86.31	79.80	57.93	<b>92.47</b>
Average	–	90.58	87.53	<b>91.67</b>	<u>91.48</u>	78.04	84.11

#### 4.6 Performance in the Context of OSN

As previously discussed, the exact operations of OSN platforms on images are unknown but must be investigated further by the research community to develop any solution for fake image detection in practical cases. Hence, to check the robustness of our solutions, we apply common post-processing operations like Median filtering, Gaussian Noise addition, Contrast Limited Adaptive Histogram Equalization (CLAHE), Average Blurring, Gamma Correction and Resizing with different parameters on StyleGAN2 dataset as shown in Table 4. The best performance of each operation is marked in bold, and the second-best performance is underlined. We examine performances in terms of detection accuracy for our three models: Baseline (Only CNN), STN-Net (Single Layer) and STN-Net (Dual Layer). Our model STN-Net (Dual Layer) achieves the best average accuracy of

91.67%, closely followed by CSC-Net [23]. Both versions of STN-Net perform exceptionally well on Gaussian noise-added images. While for Gaussian noise with a standard deviation of 2.0, Single layer STN has a performance drop of 0.10%, Dual-Layer STN has 0.20% performance drops. Interestingly, our baseline model has a much smaller performance drop than others. Detailed results are shown in Table 4.

#### 4.7 Performance in the Context of JPEG Compression

As discussed earlier, even though mostly GAN-generated images are by default in PNG format, they are commonly converted into JPEG format while uploaded or downloaded to or from OSNs. Unlike lossless PNG images, JPEG images use lossy compression, which enables discarding of some image data to reduce file size. This can lead to a loss of image quality and a change in the statistical properties of images.

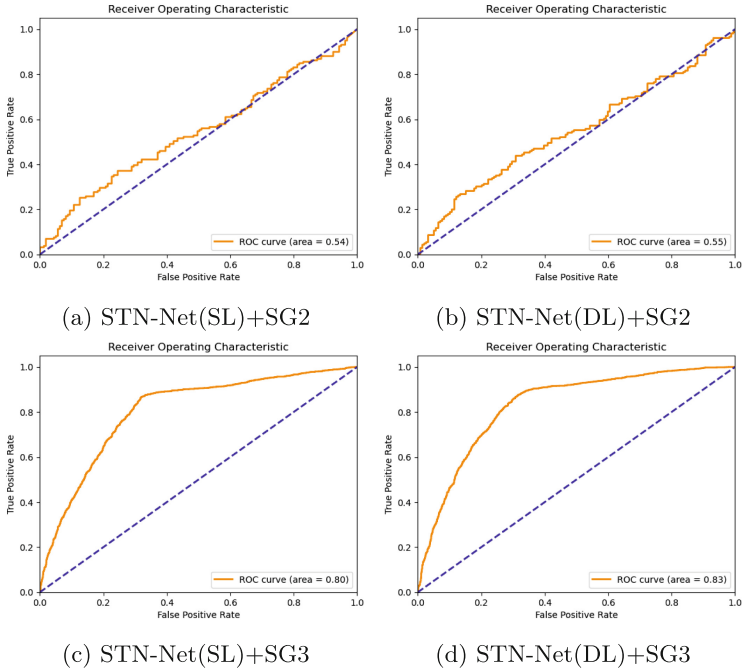
**Table 5.** Performance on JPEG Compression

Quality Factor	Base Model	STN-Net	Dual-Layer STN-Net	CSC-Net [23]	Cross Co-Net [22]	Co-Net [20]
90	91.38	99.33	<b>99.33</b>	97.53	94.50	95.58
80	91.37	98.71	<b>99.08</b>	97.44	88.66	94.93
70	91.29	97.99	<b>98.69</b>	97.23	83.50	94.03
60	91.34	96.90	<b>98.26</b>	96.83	94.00	94.65
50	91.22	96.88	<b>97.89</b>	96.51	80.05	96.66

CNN models learn abstract information from provided training data. Most CNN-based GAN-image detectors face performance drop issues when training data is from PNG images and testing data from JPEG images. We check the performance of our models on JPEG quality factors: 90, 80, 70, 60 and 50 and compare their performance with SOTA solutions. As shown in Table 5, the Dual layer variant of our proposed STN model performs best for all mentioned JPEG compression levels.

#### 4.8 Generalization Performance

We further explore the generalization capability of our proposed solutions: STN-Net with both single-layer and dual-layer variations. We show results using ROC curve, as shown in Fig. 9. Firstly, we test StyleGAN3-generated faces on models trained on the StyleGAN2 dataset (Fig. 9a, Fig. 9b). Next, we test StyleGAN2-generated faces on models trained on StyleGAN3 dataset (Fig. 9c, Fig. 9d). As shown in Fig. 9, while the training set is from StyleGAN3, in both single-layer and dual-layer versions, detection performance for StyleGAN2 images is satisfactory



**Fig. 9.** Generalization Performance: (a) and (b) trained on StyleGAN2 while tested on StyleGAN3, (c) and (d) trained on StyleGAN3 while tested on StyleGAN2

with AUC score  $\geq 80\%$ . While the training set is from StyleGAN2, in both single-layer and dual-layer versions, the detection performance for StyleGAN3 images is worse than the previous case. However, it still performs better than random guesses.

We may infer that StyleGAN3-trained models learn more generalised features than StyleGAN2-trained models. StyleGAN3 is an improved version of StyleGAN2. Hence, it is possible that models that learn statistical features of StyleGAN3 images naturally cover many of StyleGAN2 features.

## 5 Concluding Remarks

In this work, we propose a solution to identify authentic and GAN-generated face images in the context of OSNs. Hence, we test our proposed detector’s performance against standard perturbation in OSN. However, we have not included the study of other sophisticated black-box and white-box attacks [1] like adaptive attacks in this work. We wish to include such studies in future work.

In this work, we introduce a feature *Sine Transformed Noise (STN)* that is highly capable of discriminating between real and GAN images. Accompanied by a well-designed deep neural network, STN is capable of performing at



par with SOTA solutions in ideal scenarios and achieves prominent performance for post-processed and compressed images. Compared with other SOTA solutions, STN-Net uses lightweight CNN with fewer parameters, lesser computational complexity and high inference time. All these advantages make STN-Net very usable in real-world scenarios.

## References

1. Carlini, N., Farid, H.: Evading deepfake-image detectors with white-and black-box attacks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pp. 658–659 (2020)
2. Chen, B., Ju, X., Xiao, B., Ding, W., Zheng, Y., de Albuquerque, V.H.C.: Locally GAN-generated face detection based on an improved Xception. *Inf. Sci.* **572**, 16–28 (2021)
3. Chen, B., Liu, X., Zheng, Y., Zhao, G., Shi, Y.Q.: A robust GAN-generated face detection method based on dual-color spaces and an improved Xception. *IEEE Trans. Circ. Syst. Video Technol.* **32**(6), 3527–3538 (2021)
4. Chollet, F.: Xception: deep learning with depthwise separable convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1251–1258 (2017)
5. Ciftci, U.A., Demir, I., Yin, L.: FakeCatcher: detection of synthetic portrait videos using biological signals. *IEEE Trans. Pattern Anal. Mach. Intell.* (2020). <https://doi.org/10.1109/TPAMI.2020.3009287>
6. Cozzolino, D., Gagnaniello, D., Poggi, G., Verdoliva, L.: Towards universal GAN image detection. In: 2021 International Conference on Visual Communications and Image Processing (VCIP), pp. 1–5. IEEE (2021)
7. Frank, J., Eisenhofer, T., Schönherr, L., Fischer, A., Kolossa, D., Holz, T.: Leveraging frequency analysis for deep fake image recognition. In: International Conference on Machine Learning, pp. 3247–3258. PMLR (2020)
8. Fu, Y., Sun, T., Jiang, X., Xu, K., He, P.: Robust GAN-face detection based on dual-channel CNN network. In: 2019 12th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), pp. 1–5. IEEE (2019)
9. Goodfellow, I., et al.: Generative adversarial nets. In: Advances in Neural Information Processing Systems, vol. 27 (2014)
10. Gagnaniello, D., Cozzolino, D., Marra, F., Poggi, G., Verdoliva, L.: Are GAN generated images easy to detect? A critical analysis of the state-of-the-art. In: 2021 IEEE International Conference on Multimedia and Expo (ICME), pp. 1–6. IEEE (2021)
11. Guo, H., Hu, S., Wang, X., Chang, M.C., Lyu, S.: Eyes tell all: irregular pupil shapes reveal GAN-generated faces. In: ICASSP 2022–2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2904–2908. IEEE (2022)
12. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
13. Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of GANs for improved quality, stability, and variation. arXiv preprint [arXiv:1710.10196](https://arxiv.org/abs/1710.10196) (2017)

14. Karras, T., et al.: Alias-free generative adversarial networks. In: *Advances in Neural Information Processing Systems*, vol. 34, pp. 852–863 (2021)
15. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4401–4410 (2019)
16. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and improving the image quality of StyleGAN. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8110–8119 (2020)
17. Marra, F., Gragnaniello, D., Cozzolino, D., Verdoliva, L.: Detection of GAN-generated fake images over social networks. In: *2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, pp. 384–389. IEEE (2018)
18. Matern, F., Riess, C., Stamminger, M.: Exploiting visual artifacts to expose deepfakes and face manipulations. In: *2019 IEEE Winter Applications of Computer Vision Workshops (WACVW)*, pp. 83–92. IEEE (2019)
19. Mishra, M., Adhikary, F.: Digital image tamper detection techniques-a comprehensive study. arXiv preprint [arXiv:1306.6737](https://arxiv.org/abs/1306.6737) (2013)
20. Nataraj, L., et al.: Detecting GAN generated fake images using co-occurrence matrices. arXiv preprint [arXiv:1903.06836](https://arxiv.org/abs/1903.06836) (2019)
21. Nightingale, S., Agarwal, S., Härkönen, E., Lehtinen, J., Farid, H.: Synthetic faces: how perceptually convincing are they? *J. Vis.* **21**(9), 2015–2015 (2021)
22. Nowroozi, E., Mekdad, Y.: Detecting high-quality GAN-generated face images using neural networks. In: *Big Data Analytics and Intelligent Systems for Cyber Threat Intelligence*, pp. 235–252 (2023)
23. Qiao, T., et al.: CSC-Net: cross-color spatial co-occurrence matrix network for detecting synthesized fake images. *IEEE Trans. Cogn. Dev. Syst.* (2023). <https://doi.org/10.1109/TCDS.2023.3274450>
24. Sharif, M., Mohsin, S., Javed, M.Y., Ali, M.A.: Single image face recognition using Laplacian of Gaussian and discrete cosine transforms. *Int. Arab J. Inf. Technol.* **9**(6), 562–570 (2012)
25. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint [arXiv:1409.1556](https://arxiv.org/abs/1409.1556) (2014)
26. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2818–2826 (2016)
27. Tan, M., Le, Q.: EfficientNet: rethinking model scaling for convolutional neural networks. In: *International Conference on Machine Learning*, pp. 6105–6114. PMLR (2019)
28. Verdoliva, L.: Media forensics and deepfakes: an overview. *IEEE J. Sel. Top. Sig. Process.* **14**(5), 910–932 (2020)
29. Wan, J., He, X., Shi, P.: An iris image quality assessment method based on Laplacian of Gaussian operation. In: *MVA*, pp. 248–251 (2007)
30. Wang, B., Wu, X., Tang, Y., Ma, Y., Shan, Z., Wei, F.: Frequency domain filtered residual network for deepfake detection. *Mathematics* **11**(4), 816 (2023)
31. Xia, Z., Qiao, T., Xu, M., Zheng, N., Xie, S.: Towards DeepFake video forensics based on facial textural disparities in multi-color channels. *Inf. Sci.* **607**, 654–669 (2022)
32. Yang, X., Li, Y., Qi, H., Lyu, S.: Exposing GAN-synthesized faces using landmark locations. In: *Proceedings of the ACM Workshop on Information Hiding and Multimedia Security*, pp. 113–118 (2019)