applied sciences

MDPI

Article

Tampered and Computer-Generated Face Images Identification Based on Deep Learning

L. Minh Dang [1], Kyungbok Min [1], Sujin Lee [2], Dongil Han [1] and Hyeonjoon Moon [1,*]

[1] Department of Computer Science and Engineering, Sejong University, Seoul 143-747, Korea; danglienminh93@gmail.com (L.M.D.); clintminjsr@gmail.com (K.M.); dihan@sejong.ac.kr (D.H.)
[2] Da Vinci Software Education Institute, ChuangAng University, Seoul 143-747 Korea; genegraphy@sogang.ac.kr
* Correspondence: hmoon@sejong.ac.kr

Received: 4 December 2019; Accepted: 8 January 2020; Published: 10 January 2020

Abstract: Image forgery is an active topic in digital image tampering that is performed by moving a region from one image into another image, combining two images to form one image, or retouching an image. Moreover, recent developments of generative adversarial networks (GANs) that are used to generate human facial images have made it more challenging for even humans to detect the tampered one. The spread of those images on the internet can cause severe ethical, moral, and legal issues if the manipulated images are misused. As a result, much research has been conducted to detect facial image manipulation based on applying machine learning algorithms on tampered face datasets in the last few years. This paper introduces a deep learning-based framework that can identify manipulated facial images and GAN-generated images. It is comprised of multiple convolutional layers, which can efficiently extract features using multi-level abstraction from tampered regions. In addition, a data-based approach, cost-sensitive learning-based approach (class weight), and ensemble-based approach (eXtreme Gradient Boosting) is applied to the proposed model to deal with the imbalanced data problem (IDP). The superiority of the proposed model that deals with an IDP is verified using a tampered face dataset and a GAN-generated face dataset under various scenarios. Experimental results proved that the proposed framework outperformed existing expert systems, which has been used for identifying manipulated facial images and GAN-generated images in terms of computational complexity, area under the curve (AUC), and robustness. As a result, the proposed framework inspires the development of research on image forgery identification and enables the potential to integrate these models into practical applications, which require tampered facial image detection.

Keywords: image manipulation; deep learning; imbalanced dataset; image forgery detection; GAN; tampered image

1. Introduction

A social networking service (SNS) provides various online services for its users to connect with friends, families, classmates, and other people who share similar hobbies, careers, and backgrounds. According to today's trend, social networking is one of the easiest ways for a person to communicate with other people online, and it has transformed the way people create, maintain, and sustain their social information network [1]. An immense amount of data is uploaded on social networking sites, such as Facebook, YouTube, Instagram, and Twitter every day [2]. Compared to text materials and videos, photographs are a more straightforward means to convey information. Originally, most of the images uploaded on social network platforms are genuine because users capture life moments and share these moments on social networks. However, the threats of fake news have become more and more serious in recent years [3].

Image tampering is an effective technique that can be exploited to manipulate images. There are three standard techniques in image tampering, including copy-move, image splicing, and image retouching [4]. The copy-move method refers to the process of copying some parts from a source image and putting them into a target image, whereas the image splicing technique combines two or more images to create a composite image. On the other hand, the image retouching technique applies several computer vision (CV) technologies to create a new image by enhancing some features from the original image [5]. After conducting these techniques, enhancement of boundary, shape, scaling, and illumination for the created image is implemented to minimize the defects and make it more challenging to identify the tampered regions. To make it even more difficult, generative adversarial networks (GANs), a new branch of unsupervised learning artificial intelligence (AI) has emerged as a hot topic in recent years. It can generate photographs that have realistic characteristics and look superficially authentic to human observers [5,6]. Output images of these techniques spread like wildfire on the internet due to the development of social networks [6]. Figure 1 represents four digital image tampering examples, which concentrate especially on manipulating the facial parts. Even after a close inspection, there is a high possibility that observers are unable to detect the tampered regions. If these images are fed into conventional face detection and recognition frameworks, the faces from manipulated images are detected and recognized as authentic images [7]. The consequences become even more severe if manipulated images are used for commercial or political intentions.

Figure 1. Example of digital image forgery methods (**a**) copy-move forgery method, (**b**) image retouching method, (**c**) image splicing method, and (**d**) generative adversarial networks (GANs) method.

Even though researchers dedicated to conducting image tampering detection have increased sharply in recent years, many drawbacks still exist in previous frameworks. Existing models were designed to recognize specific characteristics of the dataset under consideration. For example, in error level analysis (ELA), the actual interpretation of the level of compression artifacts in a given segment of an image is biased, which can lead to inaccurate judgment [8]. The color filter array (CFA) method is vulnerable to images that were resampled onto a CFA and then re-interpolated. Moreover, pixels that are close to the digital sensor resolution limit can be a problem for CFA [9]. Double JPEG localization technique is vulnerable to tampered images that went through many post-processing

steps [10]. For GANs, each model generates new data instances on a particular topic, and the generated instances have different sizes and resolutions. In addition, GAN-generated images identification research is limited. As a result, conventional methods are inefficient and time-consuming because they depend heavily on hand-crafted features and manually selected machine learning (ML) algorithms. Deep learning has thrived recently as a replacement for traditional methods, because it has shown excellent performance in CV, including image tampering and GAN-generated mage detection [1]. Although deep learning automatically extracts abstract features instead of using hand-crafted features, it demands enormous computing power and a substantial amount of data [11]. Image tampering detection research also suffers from imbalanced dataset problem (IDP) [12], which leads to the poor performance of the models on the minority class. For example, IDP appears in the standard two-class classification when the number of tampered images (minority class) is significantly lower than the number of real images (majority class) [13]. As a consequence, ML algorithms classify the testing samples based on features extracted from the majority class and ignore features extracted from samples of the minority class [12]. There are four main techniques to cope with the IDP, namely: algorithm-based techniques [14,15], data-based techniques [16,17], cost-sensitive learning techniques [18,19], and ensemble-based techniques [13,20].

In this study, a customized convolutional neural network (CNN) for tampered face images detection (TFID), was introduced. It effectively identifies different types of tampered face images, verifies their genuineness, and performs well even under extreme IDP. Initially, face regions are detected and extracted from the datasets. Then they are used to train the TFID model under a balanced dataset scenario. After that, three extensions of the TFID model, which integrate three different IDP techniques into the existing TFID model were presented. Finally, several experiments were implemented to examine the performance of the proposed frameworks on two different datasets and various imbalanced dataset scenarios. In the first experiment, 4-fold cross-validation was implemented to evaluate the TFID's performance on a tampered face dataset. Next the proposed model was compared with the state-of-the-art SE-ResNet-50 model and VGG16 model [21]. After that, a second experiment was conducted to check the performance of different extensions of the TFID model on different balancing ratios between the real and the tampered class, ranging from 1/1 (balance dataset) to 1/100 (extremely imbalanced dataset). Finally, the proposed models were trained with other manipulated face datasets, and the performance was evaluated. The main contributions of the research are represented as follows:

1. An introduction of a TFID model that can effectively identify tampered face images and images generated by the computer.
2. Investigate the effectiveness of three different approaches that deal with the imbalanced dataset problem.
3. The ensemble-based extension of the TFID model achieves high performance on different imbalanced dataset scenarios.
4. The proposed models outperform existing models to identify tampered face images and GAN-generated face images.

The remainder of the manuscript is organized as follows. The main datasets used in this study are shown in Section 2. Next, deep learning-based manipulated face detection frameworks are carefully described in Section 3. After that, Section 4 explains three main experiments, which are conducted to evaluate the proposed systems on both imbalanced and balanced dataset scenarios. Section 5 discusses the experimental results and presents some insights about the proposed model. Finally, the main contents of this study and future approaches are mentioned in Section 6.

2. Dataset

Two different datasets that are used to verify the performance of the proposed model are manipulated face (MANFA) dataset [13] and progressive growing of GANs (PGGAN) dataset [7]. MANFA dataset is used to check the performance of manipulated face images identification. On the

other hand, PGGAN is used to check whether a model can differentiate between real face images and computer-generated face images.

2.1. MANFA Dataset (Dataset 1)

MANFA dataset is a face image tampered dataset involves only face regions and dedicates to tampered face identification task [13]. Some of the images taken from MANFA dataset are shown in Figure 2. It includes a total of 204,200 face images (4200 images belong to the tampered class, and 200,000 images are from the real class) with various sizes from 82×82 to 1098×1098 and unconstrained conditions, such as illumination changes, background cluttered, and poses. In addition, MANFA contains faces with a wide range of features, such as gender, hair color, personal identities, ethnicities, ages, and glasses.

Figure 2. Examples of face images from the MANFA dataset. The first row shows tampered face images (red circles highlight manipulated regions) and the second row contains real face images.

2.2. PGGAN dataset (Dataset 2)

PGGAN dataset contains images generated by the PGGAN model that was proposed by Nvidia researchers [7]. It contains 5000 real face images taken from the CelebA dataset [11] and 5000 high-quality face images generated by the PGGAN model. The image size is 256×256 and the images are stored in a PNG format. Figure 3 shows two realistic images, which were created by the PGGAN model.

Figure 3. Two sample images (256×256) generated artificially using a Progressive Growing of GANs Dataset (PGGAN) model.

3. Methodology

This section thoroughly describes the full process of the proposed TFID and three extensions of TFID model. Before the proposed deep learning-based model is trained, face regions are localized and extracted from the datasets using facial landmark algorithm [22]. After that, the presented TFID framework is explained in Section 3.1, and three extensions of TFID model to deal with the IDP are discussed in Section 3.2. They are created by (1) adding XGBoost layers to the original TFID model, (2) controlling the class weight for each class, and (3) applying oversampling and undersampling techniques to the imbalanced dataset. These models are tested using different imbalanced factors and also compared with previous state-of-the-art models.

3.1. Tampered Face Identification Framework (TFID)

Figure 4 illustrates a complete architecture of the proposed TFID model, which shows the input size, the kernel size, and the output size. Overall, the proposed TFID accepts 256 × 256 images as input. Next, input data are passed through six convolutional layers (C1 to C6) and five max-pooling layers (M1 to M5), followed by batch normalization and one dense layer to give the final classification decision.

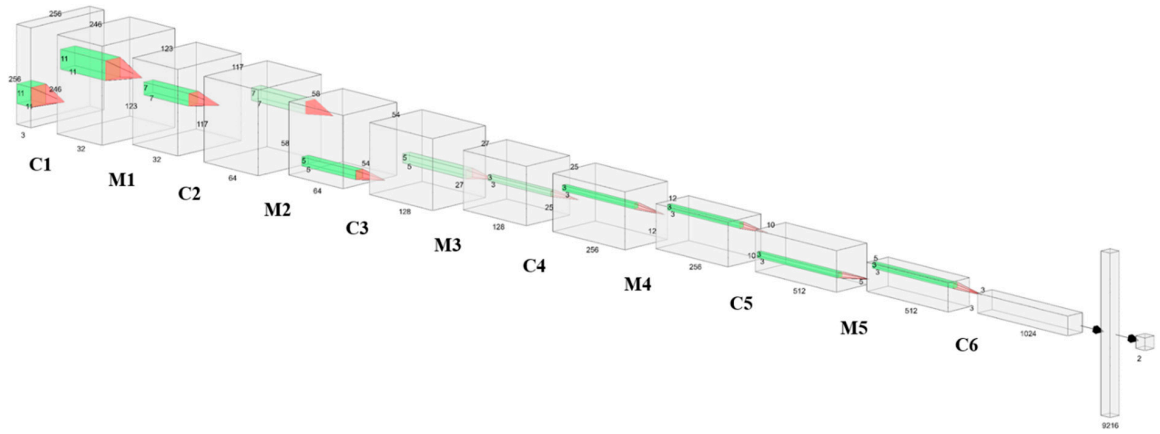

Figure 4. Tampered Face Identification Framework (TFID) architecture with a detailed configuration for each layer.

Initially, a suitable CNN structure must be figured out, and the selection of each layer, such as the convolutional layers and the max-pooling layers, and the dropout depends strongly on the experiments. By reviewing previous research and the dataset size [1,4], the model is configured to accept 256 × 256 images as the input. The TFID model contains six convolution layers that are in charge of extracting abstract features. Each convolutional layer requires a proper kernel size to manage the parameters effectively. Therefore, suitable kernel sizes for each convolutional layer that ranged from 11 to 3 are selected. The rectified linear unit (ReLU) nonlinearity function (f = max(0,x)) is used as the activation function for each convolutional layer. It was proved to prevent the overfitting problem more efficiently than the hyperbolic and sigmoid functions [23]. Next, max-pooling layers are added behind the convolutional layer to decrease the feature maps' spatial size and prevent overfitting issues. The output of a pooling layer is pooled or down-sample feature maps, which significantly reduces original features size. Finally, a dense layer that uses a softmax function is added to decide whether the input is a tampered or real image.

Table 1 presents detailed configurations and output of each layer in the TFID model. The dropout value is set to 0.2 for the first five dropout regularizations and 0.5 for the last dropout regularization.

Table 1. A detailed configuration and output of each layer in the proposed TFID framework.

TFID Framework		
Layer Name	**Configuration**	**Output (Rows, Cols, Channels)**
Input		256 × 256
Convolution_1	11 × 11	(246, 246, 32)
Maxpool_1	2 × 2	(123, 123, 32)
Dropout_1	Probability: 0.2	
Convolution_2	7 × 7	(117, 117, 64)
Maxpool_2	2 × 2	(58, 58, 64)
Dropout_2	Probability: 0.2	
Convolution_3	5 × 5	(54, 54, 128)
Maxpool_3	2 × 2	(27, 27, 128)
Dropout_3	Probability: 0.2	
Convolution_4	3 × 3	(25, 25, 256)
Maxpool_4	2 × 2	(12, 12, 256)
Dropout_4	Probability: 0.2	
Convolution_5	3 × 3	(10, 10, 512)
Maxpool_5	2 × 2	(5, 5, 512)
Dropout_5	Probability: 0.2	
Convolution_6	3 × 3	(3, 3, 1024)
BatchNorm		(3, 3, 1024)
Dropout_4	Probability: 0.5	
Flatten	Length: 16	(9216)
Dense	Length: 2	(2)

3.2. TFID Extensions for Imbalanced Dataset Problem (IDP)

As explained in the introduction section, the number of tampered images is insignificant compared to the massive number of real images in existing tampered face datasets. As a result, previous tampered face images and GAN-generated images identification frameworks have suffered significantly from the IDP [1,13]. This section demonstrates three well-known techniques to solve the IDP for TFID model, including ensemble-based technique, cost-sensitive learning technique, and data-based technique, as shown in Figure 5.

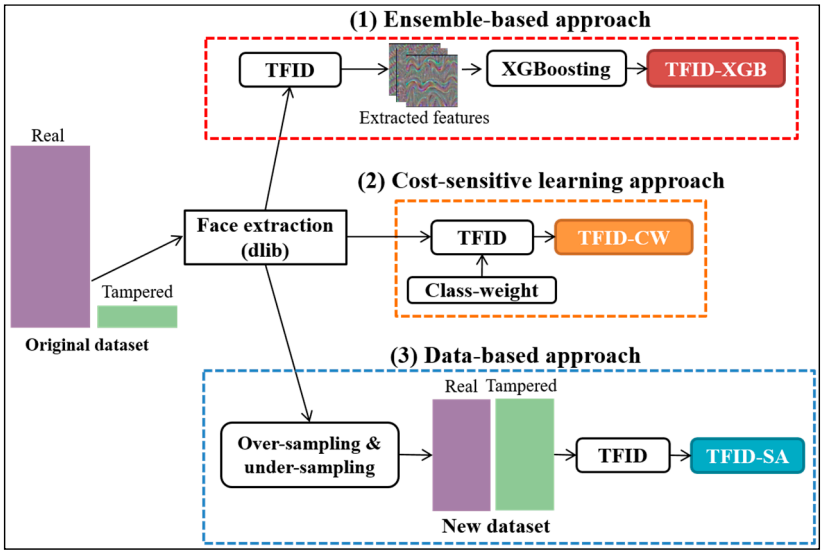

Figure 5. Implementation of three different approaches to deal with the imbalanced data problem (IDP) for the proposed TFID model.

In the first approach, the softmax layer that determines the multi-class probabilities for each test sample is replaced with an XGBoost boosting function. For the cost-sensitive learning approach, a class weight is assigned for each class. Finally, an over-sampling approach is applied to the minority class (tampered images). In contrast, an under-sampling method is conducted on the majority class (real images) to obtain a new balanced dataset. A detailed explanation for each approach is described as follows.

3.2.1. Ensemble-Based Technique

A gradient boosting tree is a learning approach created explicitly for preventing the IDP, where the final classifier is built from a collection of weak classifiers. Initially, a simple classifier is trained to classify the training dataset, and incorrectly classified samples are recorded. Then, the next classifier is trained and forced to fix the wrong predictions of the previous classifier based on correct class labels. After that, many weak classifiers are constructed to fix prediction errors that previous trees made.

Extreme gradient boosting (XGBoost) is a lightning-fast and robust implementation of the gradient boosting algorithms [24]. It tackles potential information loss when a new tree is created, which is one of the major drawbacks of gradient boosted trees. XGBoost analyzes the distribution of features across all data points and uses this information to reduce the search space of the possible feature splits. The equation for XGBoost is described as follows:

$$Obj = L + \Omega \tag{1}$$

where the loss function L controls the predictive power of XGBoost. Ω is the regularization used to control the overfitting problem [25]. The regularization component Ω is set based on the number of observers and the prediction threshold of the observers in the ensemble model. The loss function L can be either the root mean squared error (RMSE) for the regression analysis, the log loss for binary classification, or the mlogloss for multi-class classification.

3.2.2. Cost Sensitive-Based Technique

Traditional ML models assume that all misclassification errors carry the same cost, which leads to poor performance in IDP. In contrast, cost-sensitive models establish fixed and unequal misclassification costs between classes. The classification cost is based on a cost matrix $\lambda_{c_1 c_2}$, which expresses the cost of categorizing a sample from a class c_1 to class c_2. This matrix is normally represented in terms of average misclassification costs. The diagonal elements in the matrix are set to 0 to indicate accurate classification. The conditional risk Cr for making decision α_i is defined as:

$$Cr(\alpha_i|x) = \sum_i \lambda_{c_1 c_2} P(v_j|x) \tag{2}$$

The equation shows that the probability of class i is based on fixed misclassification costs, and the uncertainty about the true class of x is indicated by the posterior probabilities. The goal of cost-sensitive learning is to reduce the misclassification cost by outputting the class v_j with the minimum conditional risk Cr.

3.2.3. Data-Based Technique

Resampling is a well-known data-based method that attempts to balance the class distribution to deal with the IDP [12]. It includes over-sampling, under-sampling, and hybrid techniques, as represented in Figure 6.

- Under-sampling technique constructs a new subset from the original dataset by omitting some of the samples of the majority class.

- Over-sampling method creates a new superset from the original dataset by duplicating some of the samples of the minority class or generating new samples based on the original samples from the minority class.
- Hybrid technique is a method that involves both oversampling and under-sampling techniques.

In this study, a hybrid data-based approach is implemented to prevent the drawbacks of applying only over-sampling approach or over-sampling approach [16,17].

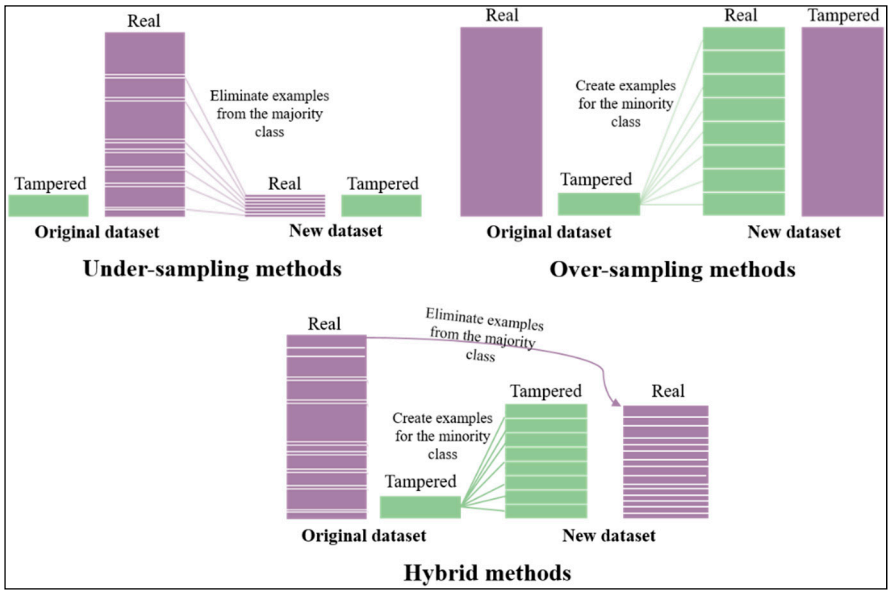

Figure 6. Illustration of data-based approaches, which include under-sampling, over-sampling, and hybrid methods.

4. Experimental Results

The experimental results section describes all experiments that were conducted and obtained results on two different datasets. The first dataset is the manually collected and evaluated MANFA dataset [13], and the second dataset is the PGGAN dataset [7]. All experiments are implemented on an NVIDIA DIGITS toolbox with a pre-installed Ubuntu 16.04. It contained an Intel® Core i7-5930K processor, four 3072 CUDA cores, four Titan X 12GB GPUs, and 64GB of DDR4 RAM.

Section 4.1 explains the evaluation metrics used in this research, including AUC score, precision, and recall. Then, the first experiment is carried out to validate the proposed model performance on a balanced dataset as shown in Section 4.2. A visualization of detected tampered regions is implemented in Section 4.3 to explain why the model classifies an input image as manipulated image. After that, the performance of applying three different approaches to the proposed model for solving the IDP was described in Section 4.4. Section 4.5 shows the performance of the proposed model on a GAN dataset.

4.1. Evaluation Metrics

The prediction output of the system for an input image is either real or tampered, so it is a binary classification problem. The performance of the system is usually represented in a confusion matrix, which is given in Table 2.

After the confusion matrix was constructed, accuracy, precision, and recall are computed to investigate the proposed model performance. Accuracy refers to the proportion of correctly classified samples (TP and TN) among the total samples in the test dataset. Accuracy cannot provide a thorough evaluation of the model. Therefore, precision and recall are two widely used additional measurements.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{3}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{4}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{5}$$

where TP, FN, FP, and TN are the corresponding true positive, true negative, false positive, and false negative values for one class against the other class, which were taken from the confusion matrix.

Based on the acquired values from the confusion matrix, true-positive rate (TPR) and false-positive rate (FPR) measurements are computed. The TPR is the same as recall, whereas FPR is shown in the following equation:

$$\text{FPR} = \frac{\text{FP}}{\text{TN} + \text{FP}} \tag{6}$$

Table 2. A confusion matrix for the binary classification results of one class against the other class.

		Prediction	
		Tampered	**Real**
Actual	Tampered	TP (True positive)	FN (False negative)
	Real	FP (False positive)	TN (True negative)

A receiver operating characteristic (ROC) curve [26] is illustrated with TPR against the FPR for separate cut-off points. Moreover, in the multi-class classification, every false prediction is an FP for a class, and every single negative is an FN for a class. Each point on the curve depicts a sensitivity/specificity set that correlates with a particular decision threshold. The area under the ROC curve, or AUC [26], is usually applied to estimate the performance of the proposed classification model. If a ROC curve for class 1 (C1) has a higher AUC value than class 2 (C2), then the proposed classifier C1 is considered to achieve a better performance than C2.

4.2. Balanced Dataset Experiment

The initial experiment is conducted to examine the performance of TFID model on the balanced MANFA dataset (Dataset 1) for the tampered face images identification task. The training dataset contains 4200 tampered images and 4200 real images, which were randomly taken from the original MANFA dataset. 4-fold cross-validation is then implemented on the extracted dataset by dividing it into four subsets, and each subset contains 2100 images. The number of tampered and real images are shown in Table 3. For each fold, three subsets are used as the training dataset, and the remaining subset is used for testing purposes. Within the training dataset, 80% of the training data are used to train the proposed model, and the rest of the images are utilized as a validation dataset to validate the trained model.

Table 3. The numbers of real and tampered images in four subsets.

	Subset 1	**Subset 2**	**Subset 3**	**Subset 4**
Real	1047	1051	1053	1049
Tampered	1044	1048	1052	1056

Face regions are first localized and extracted based on a python implementation of a facial landmark algorithm proposed by [22]. This study applied a 5-point facial landmark (2 points for the left eye, 2 points for the right eye, and 1 point for the nose) because it has been proved to be 8–10% faster than the original 68-point detector [27]. Then, localized face images are rotated and aligned to frontal to remove pose changes. The facial landmark algorithm is implemented with dlib library version 19.18.0. Next, OpenCV library version 4.1.1 is used to perform the face rotation and resize all images to 256 × 256. Python programming language and Keras neural-network library are

used to implement the proposed models. The optimization function in the proposed TFID is Adam optimization with the learning rate is set to 0.001 initially as recommend by [28] for model with a small number of convolutional layers. The batch size is set to 32 because for Adam optimizer the smaller batch size can increase the test accuracy [29]. The model is trained through 50 epochs. The validation accuracy, validation loss, training accuracy, and training loss for each fold are provided in Figure 7.

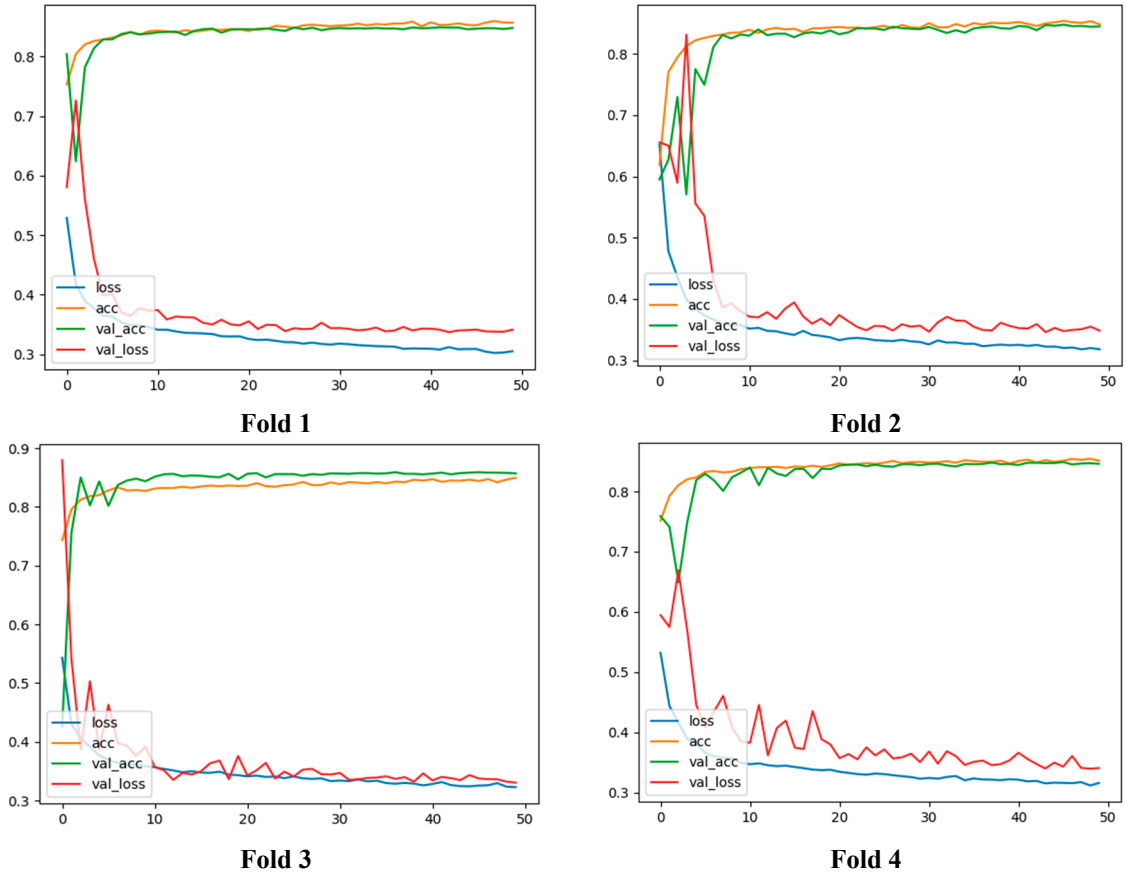

Figure 7. Training accuracy, training loss (loss), validation accuracy (val_acc), and the validation loss (val_loss) during the training of TFID model using four-fold cross-validation.

The training accuracy and the validation accuracy increase dramatically to over 79%, whereas training loss and the validation loss decline significantly to 32% after the 7th epoch. During the remaining epochs, the training accuracy and the validation accuracy rose steadily and reach a peak of 83%. Robust results are observed in fold 3 regarding validation accuracy and validation loss. In contrast, other folds fluctuate in validation accuracy and validation loss.

The proposed model is also compared with pre-trained SE-ResNet-50 model and VGGFace model, which have achieved state-of-the-art performance on VGGFace2 dataset [3]. The reason these two models were selected is that they are trained on huge dataset related to human facial features. Therefore, human face features help the pre-trained models optimized faster on MANFA dataset, which is also related to the human face. We set the hyper-parameters as suggested by [3,30] for pre-trained VGG16 and SE-ResNet-50 models to enable a good trade-off between bias and variance. A performance comparison between TFID, VGG16, and SE-ResNet-50 models are shown in Table 4.

In general, all three models performed well on the balanced MANFA dataset. The obtained results showed that the VGG16 model achieved an accuracy of 81%, precision of 78%, recall of 84%, and an AUC value of 0.83, while the TFID model obtained a higher accuracy of 83%, precision of 81%, recall of 89%, and an AUC value of 0.86. On the other hand, the pre-trained SE-ResNet-50 model witnessed the highest classification performance with an accuracy of 84.7%, precision of 82%, recall of 91%, and

an AUC value of 0.89. The classification performance of the proposed model is comparable to the state-of-the-art VGG16 and SE-ResNet-50 models. Therefore, the TFID model has the potential to deal with a tampered face images identification task. Based on the result on Table 4, TFID and SE-ResNet-50 models are used in the next experiment because they performed better than the VGG16 model.

Table 4. Performance of the proposed model and two state-of-the-art models on the MANFA dataset.

Model	Accuracy	Precision	Recall	AUC
TFID	0.83 ± 0.02	0.81 ± 0.01	0.89 ± 0.02	0.86 ± 0.03
VGG16	0.81 ± 0.08	0.78 ± 0.04	0.84 ± 0.03	0.83 ± 0.04
SE-ResNet-50	0.84 ± 0.04	0.82 ± 0.03	0.91 ± 0.02	0.89 ± 0.05

4.3. Visualization of the Proposed Model Prediction

A class activation map (CAM) is usually applied to illustrate how AI models classify a test image based on the learned weights. It projects class-specific weights of the softmax function output back to feature maps of the last convolutional layer to highlight crucial manipulated regions. Tampered regions in Figure 8a–d are highlighted in the CAM images, which are shown in Figure 8(a1–d1). The CAM visualization results in Figure 8 confirm that the proposed TFID model correctly identifies manipulated images based on manipulated traits.

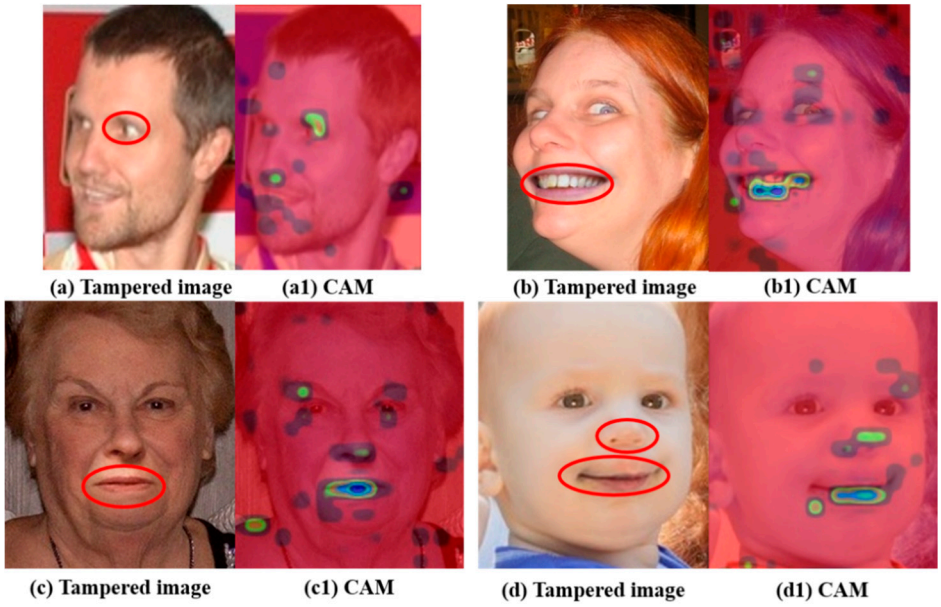

Figure 8. TFID model interpretation via class activation mapping (CAM). For each case, a left image shows tampered image with highlighted manipulated region(s) (red circle), while the corresponding right image represents the left image CAM.

4.4. Imbalanced Dataset Experiment

In this section, an experiment is conducted to evaluate the performance of three different extensions of the TFID model to deal with the IDP. They include XGBoost from the ensemble-based approach, class weight from cost-sensitive learning approach, and data-based approach.

The proportion of tampered images to real images ranging from 1/1 (balanced dataset) to 1/100 (highly imbalanced dataset) is applied to the MANFA dataset. A total of 2000 tampered images and 200,000 real images are chosen from the MANFA dataset. Table 5 depicts the number of real and tampered face images for each imbalanced case.

For the ensemble-based approach, the output 9216 feature vectors from the flatten layer are extracted, whereas 2048 feature vectors are extracted from the SE-ResNet-50 model. After that, XGBoost

classifier is trained based on these extracted features. The learning rate for XGBoost is set to 0.1, the number of trees to fit is 100, and the maximum tree depth for base learners is 3.

Table 5. The number of tampered and real images on various settings of the imbalanced dataset.

	1/100	1/50	1/25	1/10	1/5	1/1
Real	200,000	100,000	50,000	20,000	10,000	2000
Tampered	2000	2000	2000	2000	2000	2000

For the cost-sensitive learning-based approach, every sample from the tampered class is considered as n instances of the real class. Therefore, a classifier is forced to treat the tampered class and the real class equally. This assumption is implemented by using the *class_weight* parameter from Keras library, which assigns a higher loss to the tampered class to make the classifier focus more on samples from tampered class. The *class_weight* for each class was set different according to the proportion of tampered images to real images. For example, when the proportion of tampered images to real images is 1/100, *class_weight* is set 100 for the tampered class, while *class_weight* for the real class is 1 to force the model to treat every instance of the tampered class as 100 instances of the real class. On the other hand, when the proportion of tampered images to real images is 1/1, which indicate a balanced dataset, *class_weight* is fixed to 1 for both tampered class and real class.

For the data-based approach, data augmentation transformation, including horizontal flip, horizontal and vertical shift, brightness, zooming, noise addition, random rotation within 10 degrees is implemented. After that, it is integrated into the python imbalanced-learn library to create a balanced batch generator, which ensures that the number of samples per class always follows a balanced distribution.

Finally, eight models, including TFID, data-based TFID classifier (TFID-SA), cost-sensitive learning-based TFID classifier (TFID-CW), ensemble-based TFID classifier (TFID-XGB), SE-ResNet-50, cost-sensitive learning-based SE-ResNet-50 classifier (SE-ResNet-50-CW), data-based SE-ResNet-50 classifier (SE-ResNet-50-SA), and ensemble-based SE-ResNet-50 classifier (SE-ResNet-50-XGB) are implemented. The performance of each model on various imbalanced dataset settings are shown in Figure 9.

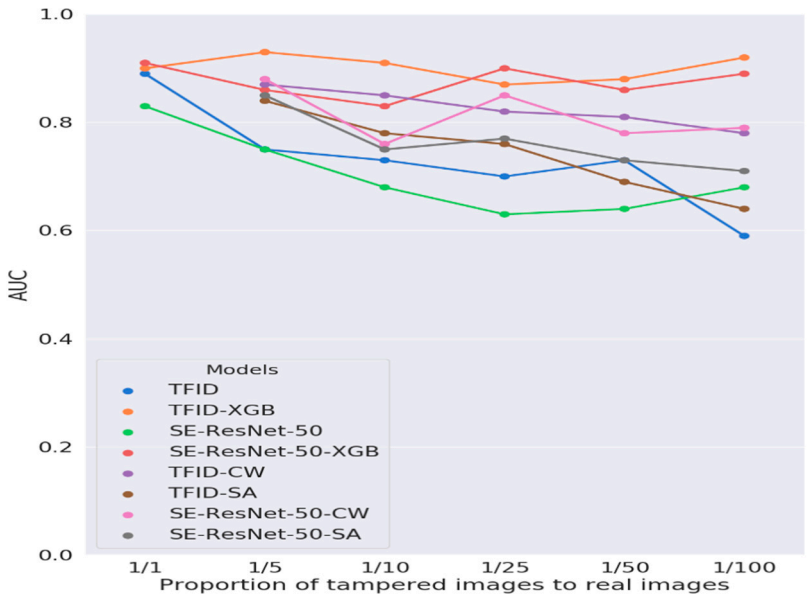

Figure 9. Area under the curve (AUC) values of different TFID extensions on different imbalanced dataset scenarios, where the number of tampered images is minor compared to the number of real images (No. tampered / No. real = 1/1, 1/5, 1/10, 1/25, 1/50, 1/100).

When the proportion of the real images to the tampered images is 1/1 (balanced dataset), all models achieved an AUC value of over 0.8. Moreover, the TFID-XGB and the SE-ResNet-50-XGB models reached a slightly higher performance compared to TFID and SE-ResNet-50. The AUC values became lower when the proportion of the real images to the tampered images increased because TFID and SE-ResNet-50 models focused on the features from the majority class and overlooked features from the minority class.

The effect IDP can be observed under the extreme setup when the proportion of the real images to the tampered images was 1/100. The AUC values of the TFID and the SE-ResNet-50 models plummeted to 0.59 and 0.6, respectively. However, the results obtained from ensemble-based models (XGB) were more robust and remained over 0.8 compared to the other extensions of TFID. Among two ensemble-based models, TFID-XGB achieved an AUC value of 0.92, and SE-ResNet-50-XGB reached an AUC value of 0.88 with the highly imbalanced ratio of 1/100. In addition, the cost-sensitive learning approach, which includes TFID-CW and SE-ResNet-50-CW models, also witnessed a high AUC value between 0.76 and 0.88. We noticed that the data-based hybrid models (TFID-SA and SE-ResNet-50-SA) performance decreased gradually as the number of real images increased, and the TFID-SA and SE-ResNet-50-SA reached their lowest AUC value at 0.64 and 0.71, respectively, when the proportion was 1/100. The main reason that led to the poor performance of the data-based approach is that the generated images using the augmentation technique were just the extension of the original images. Thus, it can lead to the overfitting problem [31,32].

After calculating the AUC value, macro-precision, macro-recall, and macro-f1 are computed. These measurements are usually computed when we want to evaluate the performance of the system on different datasets. Moreover, these measures are invariant with respect to the IDP. Macro-precision and macro-recall are computed by averaging the precision and recall of a classifier on different datasets. Table 6 shows the computed macro-precision, macro-recall, and macro-f1 of 8 different models in different imbalanced dataset settings.

Table 6. Macro-precision, macro-recall, and macro-f1 of eight different classifiers on different IDP scenarios.

Model	Macro-Precision	Macro-Recall	Macro-F1
TFID	0.768 ± 0.04	0.79 ± 0.04	0.778
TFID-CW	0.808 ± 0.0.3	0.886 ± 0.02	0.845
TFID-SA	0.818 ± 0.05	0.83 ± 0.01	0.823
TFID-XGB	0.875 ± 0.01	0.9 ± 0.02	0.887
SE-ResNet-50	0.8 ± 0.02	0.82 ± 0.03	0.81
SE-ResNet-50-CW	0.82 ± 0.02	0.84 ± 0.02	0.86
SE-ResNet-50-SA	0.83 ± 0.03	0.81 ± 0.05	0.82
SE-ResNet-50-XGB	0.9 ± 0.01	0.91 ± 0.01	0.904

The highest macro-f1 value belongs to SE-ResNet-50-XGB model, whereas the proposed TFID-XGB model achieves the macro-f1 value of 0.887. Obtained results confirm that the ensemble-based approach using XGB is the most effective way to deal with the IDP.

In the previous section, the ensemble-based extensions of TFID and SE-ResNet-50 models outperformed other approaches because the AUC values always remained over 0.8, even in the most imbalanced scenario. This experiment is conducted to compare these models in terms of computational complexity to check which model requires the lowest testing time and which model demands the highest testing time. The testing time per image of eight different models, including TFID, TFID-CW, TFID-XGB, TFID-SA, SE-ResNet-50, SE-ResNet-50-CW, SE-ResNet-50-XGB, SE-ResNet-50-SA on MANFA dataset are shown in Figure 10 (the proportion of tampered images to real images is 1/10).

As shown in Figure 10, the obtained results confirm that the testing time per image of SE-ResNet-50, SE-ResNet-50-SA and SE-ResNet-50-CW models is about 3 s. In addition, SE-ResNet-50-XGB requires 4.2 s per image, which is the longest time among the extensions of SE-ResNet-50 models. In contrast,

TFID and TFID-CW models have the shortest testing time per image (about 0.8 s). The testing time per image for TFID-XGB model is longer at 1.5 s. The ensemble-based models require more computing power because the tampered features must be extracted from the TFID or the SE-ResNet-50 model. Then those features are fed into the XGBoost classifier for classification. However, it is a fair tradeoff because the model performance is significantly increased.

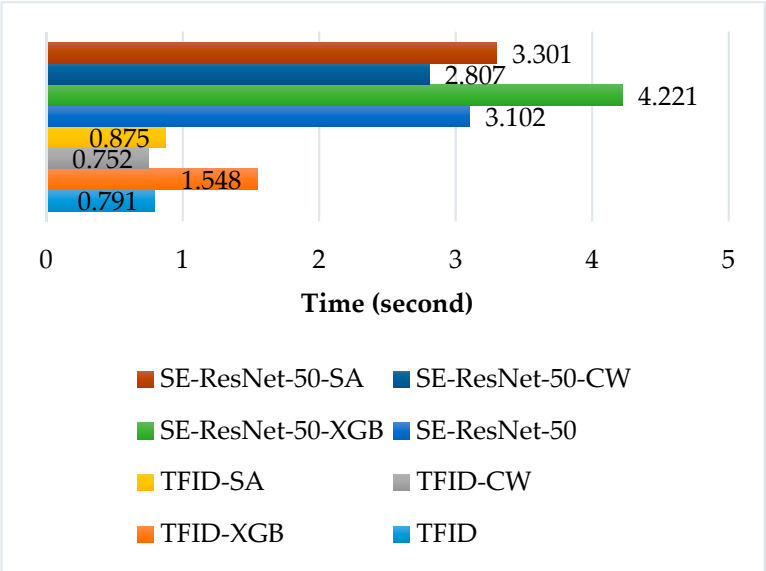

Figure 10. The computational complexity of the different models on the MANFA dataset.

4.5. Performance on PGGAN Dataset

Previous experiments were conducted on MANFA dataset and proved the TFID ability in identifying manipulated face images and solving the IDP with three extensions of the TFID. This experiment verifies whether the proposed model can effectively detect GAN-generated images from PGGAN dataset [7] similar to what it has achieved on tampered face images.

The PGGAN dataset is configured similar to [1]. The training dataset contains 3750 pairs of real-GAN-generated face images, and the validation dataset has 1250 pairs of real-GAN-generated face images. The parameters of the models in this section were set similar to previous experiments. The classification results are shown in Table 7.

Table 7. The AUC values on the GAN dataset proposed by [7].

Model	Accuracy	Precision	Recall	AUC
SE-ResNet-50	0.94	0.83	0.87	0.894
TFID	0.89	0.82	0.85	0.874
SE-ResNet-50-XGB	0.93	0.89	0.93	0.953
TFID-XGB	0.91	0.87	0.88	0.914

Overall, all models show high accuracy of over 89%, precision and recall of over 80%, and AUC values of above 0.87. The results indicate that the models could correctly classify whether an image is real or is generated by GAN. In addition, it is noticeable that ensemble-based TFID-XGB and SE-ResNet-50-XGB models achieve better performance in terms of AUC compared to the original TFID and SE-ResNet-50 models. Among the four models, SE-ResNet-50-XGB has the highest classification accuracy of 93% and an AUC value of 0.953, while the proposed TFID-XGB reaches the accuracy of 91% and an AUC value of 0.914.

5. Discussion

The obtained results from the first experiment (Section 4.2) proved that the proposed TFID model performed well on the balanced dataset through the four-fold cross-validation with an accuracy of 83% and an AUC value of 0.89. Although the SE-ResNet-50 model achieved higher accuracy compared to the TFID with a classification accuracy and an AUC value of 84.7 and 0.89, respectively, the TFID model showed its potential in tampered face identification. The accuracy of the two mentioned models stayed under 85%, and it could not improve to above 85%, because the limited number of the tampered dataset (4200 images) made the model stop learning useful features. This issue can be solved in the future by expanding the tampered dataset or pre-train the model with denoising criteria to force convolutional layers to learn important general features that are useful for reconstructing the input signal.

In the second experiment (Section 4.3), six new datasets (1/1, 1/5, 1/10, 1/25, 1/50, 1/100) were created based on the original TFID dataset by changing the proportion of the tampered images to real images. After that, these datasets were used to train two original models (TFID and SE-ResNet-50), and six extensions of TFID model (TFID-SA, TFID-CW, TFID-XGB, SE-ResNet-50-SA, SE-ResNet-50-CW, and SE-ResNet-50-XGB) to investigate the performance of the eight models on the IDP. The TFID-XGB performance exceeded other models and achieved robust results with different settings. Moreover, the AUC value was at 0.92, and it was 1/100 even in the most severe imbalanced scenario. Therefore, it showed that the TFID-XGB classifier obtained a robust performance on the IDP. Although the proposed TFID-XGB model achieved equivalent performance to the state-of-the-art pre-trained SE-ResNet-50-XGB model, it outperformed the SE-ResNet-50-XGB model in terms of computational complexity. With a simpler architecture, TFID-XGB model requires significantly lesser testing time but achieves similar performance to the SE-ResNet-50-XGB model. Therefore, it is a better choice for practical applications that are sensitive to the computing power.

Finally, we applied the proposed model to try to identify the GAN-generated images, which is an emerging method in image forgery. The TFID-XGB classifier and the SE-ResNet-50-XGB model performed well on the GAN-generated dataset.

Through numerous experiments, the proposed framework proved to have the potential to be applied in practical applications to reduce the labor cost in manually checking the increasing number of manipulated images. The proposed model can identify images, which are forged manually by humans or generated automatically by a computer. Therefore, it also plays an important role in digital image security.

6. Conclusions and Future Work

In this research, a deep learning-based system, which can detect whether an image is original or has been manipulated, was introduced. Several methods were conducted to improve the performance of the proposed model. We also concentrated on the imbalanced dataset problem by applying three different approaches, which included the ensemble-based method (XGBoost), the hybrid data-based method, and the cost-sensitive learning method (class weight), to the proposed model to create three new extensions. The TFID-XGB obtained state-of-the-art results in different imbalanced dataset scenarios with the highest AUC value of 0.92. Moreover, our model can detect images generated entirely with a computer using a trending model, such as the generative adversarial network.

Our proposed model is flexible, has computational efficiency, and robustness against an imbalanced dataset problem. Therefore, it is superior over existing expert and intelligent systems, which are usually applied to the task of tampered face image detection. Given that more training data is collected and further development on the CNN architecture is conducted, the proposed model could eventually replace the current standard algorithms.

In the future, some issues must be addressed to improve the model's performance. Firstly, the proposed model was trained only on RGB images, so it is necessary to investigate other color channels or environments to figure out potential features for the tampered image identification. Secondly, several pre-processing techniques, such as whitening transformation and rescaling need to be implemented to

improve the model performance. Finally, there are many object detection and localization studies, such as SSD and YOLO3, which achieved quite impressive performances in recent years. The proposed model only identified tampered face images, and it cannot localize the tampered regions. Therefore, the integration of localization will allow the proposed model to point out the extract location of the tampered regions in the image.

Author Contributions: Conceptualization, H.M. and D.H.; methodology, L.M.D.; validation, K.M., S.L. and H.M.; resources, D.H.; writing—original draft preparation, L.M.D.; writing-review and editing, K.M.; visualization, S.L.; supervision, H.M.; funding acquisition, H.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: This work was supported by Seoul Industrial-Academic-Cooperation Project (Artificial Intelligence Technology Commercialization Support Project) in 2019. An Empirical Study on Public Facilities Health Assessment System such as Tunnel by Automating 3D Drawing Generation with XAI-based Defect Detection and BIM Linkage (CY190003) and by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (2019-0-00136, Development of AI-Convergence Technologies for Smart City Industry Productivity Innovation).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Dang, L.; Hassan, S.; Im, S.; Lee, J.; Lee, S.; Moon, H. Deep learning based computer generated face identification using convolutional neural network. *Appl. Sci.* **2018**, *8*, 2610. [CrossRef]
2. Serrat, O. Social Network Analysis. In *Knowledge Solutions*; Springer: Singapore, 2017; pp. 39–43.
3. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018.
4. Carvalho, T.; Faria, F.A.; Pedrini, H.; Torres, R.D.S.; Rocha, A. Illuminant-based transformed spaces for image forensics. *IEEE Trans. Inf. Forensics Secur.* **2015**, *11*, 720–733. [CrossRef]
5. Antipov, G.; Baccouche, M.; Dugelay, J.L. Face aging with conditional generative adversarial networks. In Proceedings of the 2017 IEEE International Conference on Image Processing (ICIP), Beijing, China, 17–20 September 2017.
6. Karras, T.; Aila, T.; Laine, S.; Lehtinen, J. Progressive growing of gans for improved quality, stability, and variation. *arXiv* **2017**, arXiv:1710.10196.
7. Chauhan, D.; Kasat, D.; Jain, S.; Thakare, V. Survey on keypoint based copy-move forgery detection methods on image. *Procedia Comput. Sci.* **2016**, *85*, 206–212. [CrossRef]
8. Jeronymo, D.C.; Borges, Y.C.C.; dos Santos Coelho, L. Image forgery detection by semi-automatic wavelet soft-thresholding with error level analysis. *Expert Syst. Appl.* **2017**, *85*, 348–356. [CrossRef]
9. Liu, L.; Zhao, Y.; Ni, R.; Tian, Q. Copy-Move Forgery Localization Using Convolutional Neural Networks and CFA Features. *Int. J. Digit. Crime Forensics* **2018**, *10*, 140–155. [CrossRef]
10. Taimori, A.; Razzazi, F.; Behrad, A.; Ahmadi, A.; Babaie-Zadeh, M. A novel forensic image analysis tool for discovering double JPEG compression clues. *Multimed. Tools Appl.* **2017**, *76*, 7749–7783. [CrossRef]
11. Liu, Z.; Luo, P.; Wang, X.; Tang, X. Deep learning face attributes in the wild. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015.
12. Haixiang, G.; Yijing, L.; Shang, J.; Mingyun, G.; Yuanyue, H.; Bing, G. Learning from class-imbalanced data: Review of methods and applications. *Expert Syst. Appl.* **2017**, *73*, 220–239. [CrossRef]
13. Dang, L.M.; Hassan, S.I.; Im, S.; Moon, H. Face image manipulation detection based on a convolutional neural network. *Expert Syst. Appl.* **2019**, *129*, 156–168. [CrossRef]
14. Kwak, J.; Lee, T.; Kim, C.O. An incremental clustering-based fault detection algorithm for class-imbalanced process data. *IEEE Trans. Semicond. Manuf.* **2015**, *28*, 318–328. [CrossRef]
15. Vluymans, S.; Tarragó, D.S.; Saeys, Y.; Cornelis, C.; Herrera, F. Fuzzy rough classifiers for class imbalanced multi-instance data. *Pattern Recognit.* **2016**, *53*, 36–45. [CrossRef]
16. Abdi, L.; Hashemi, S. To combat multi-class imbalanced problems by means of over-sampling techniques. *IEEE Trans. Knowl. Data Eng.* **2015**, *28*, 238–251. [CrossRef]

17. Lin, W.C.; Tsai, C.F.; Hu, Y.H.; Jhang, J.S. Clustering-based undersampling in class-imbalanced data. *Inf. Sci.* **2017**, *409*, 17–26. [CrossRef]
18. Khan, S.H.; Hayat, M.; Bennamoun, M.; Sohel, F.A.; Togneri, R. Cost-sensitive learning of deep feature representations from imbalanced data. *IEEE Trans. Neural Netw. Learn. Syst.* **2017**, *29*, 3573–3587. [PubMed]
19. López, V.; Del Río, S.; Benítez, J.M.; Herrera, F. Cost-sensitive linguistic fuzzy rule based classification systems under the MapReduce framework for imbalanced big data. *Fuzzy Sets Syst.* **2015**, *258*, 5–38. [CrossRef]
20. Zhai, J.; Zhang, S.; Wang, C. The classification of imbalanced large data sets based on mapreduce and ensemble of elm classifiers. *Int. J. Mach. Learn. Cybern.* **2017**, *8*, 1009–1017. [CrossRef]
21. Parkhi, O.M.; Vedaldi, A.; Zisserman, A. Deep face recognition. *BMVC* **2015**, *1*, 6.
22. Wu, Y.; Ji, Q. Robust facial landmark detection under significant head poses and occlusion. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015.
23. Nair, V.; Hinton, G.E. Rectified linear units improve restricted boltzmann machines. In Proceedings of the 27th International Conference on Machine Learning (ICML-10), Haifa, Israel, 21–24 June 2010.
24. Chen, T.; Guestrin, C. Xgboost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016.
25. Zhou, P.; Han, X.; Morariu, V.I.; Davis, L.S. Two-stream neural networks for tampered face detection. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Honolulu, HI, USA, 21–26 July 2017.
26. Bradley, A.P. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognit.* **1997**, *30*, 1145–1159. [CrossRef]
27. Wu, Y.; Hassner, T.; Kim, K.; Medioni, G.; Natarajan, P. Facial landmark detection with tweaked convolutional neural networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *40*, 3067–3074. [CrossRef]
28. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
29. Hoffer, E.; Hubara, I.; Soudry, D. Train longer, generalize better: Closing the generalization gap in large batch training of neural networks. In *Advances in Neural Information Processing Systems*; NIPS: California, CA, USA, 2017; pp. 1731–1741.
30. Nguyen, T.N.; Lee, S.; Nguyen-Xuan, H.; Lee, J. A novel analysis-prediction approach for geometrically nonlinear problems using group method of data handling. *Comput. Methods Appl. Mech. Eng.* **2019**, *354*, 506–526. [CrossRef]
31. Nguyen, T.N.; Thai, C.H.; Luu, A.T.; Nguyen-Xuan, H.; Lee, J. NURBS-based postbuckling analysis of functionally graded carbon nanotube-reinforced composite shells. *Comput. Methods Appl. Mech. Eng.* **2019**, *347*, 983–1003. [CrossRef]
32. Nguyen, T.N.; Nguyen-Xuan, H.; Lee, J. A novel data-driven nonlinear solver for solid mechanics using time series forecasting. *Finite Elem. Anal. Des.* **2020**, *171*, 103377. [CrossRef]