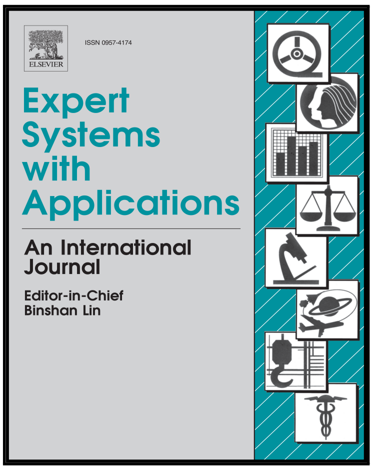


Face Image Manipulation Detection based on a Convolutional Neural Network

L. Minh Dang , Syed Ibrahim Hassan , Suhyeon Im ,
Hyeonjoon Moon

PII: S0957-4174(19)30235-0
DOI: https://doi.org/10.1016/j.eswa.2019.04.005
Reference: ESWA 12587

To appear in: *Expert Systems With Applications*

Received date: 13 August 2018
Revised date: 4 April 2019
Accepted date: 4 April 2019

Highlights

- Proposing MANFA - a customized CNN model for manipulated face detection.
- Integrating XGBoost, and AdaBoost with MANFA to cope with the extreme imbalanced dataset.
- Proposing a manually collected dataset (8,950 images) for altered face detection

Face Image Manipulation Detection based on a Convolutional Neural Network

L. Minh Dang, Syed Ibrahim Hassan, Suhyeon Im, Hyeonjoon Moon*

Department of Computer Science and Engineering, Sejong University, Seoul, Republic of Korea;

danglienminh93@gmail.com; ibrahimhassanshah@gmail.com; thehyeon@nate.com; hmoon@sejong.ac.kr

***** Corresponding Author**,** E-mail: hmoon@sejong.ac.kr

Keywords: image manipulation; deep learning; AdaBoost; XGBoost; imbalanced dataset; boosting.

Abstract. Facial image manipulation is a particular instance of digital image tampering, which is done by compositing a region from one facial image into another facial image. Fake images generated by facial image manipulation now spread like wildfire on news websites and social networks, and are considered the greatest threat to press freedom. Previous research relied heavily on handcrafted features to analyze tampered regions which were inefficient and time-consuming. This paper introduces a framework that accurately detects manipulated face image using deep learning approach. The original contributions of this paper include 1) A customized convolutional neural network model for Manipulated Face (MANFA) identification; it contains several convolutional layers that effectively extract features of multi-levels of abstraction from a tampered region. 2) A hybrid framework (HF-MANFA) that uses Adaptive Boosting (AdaBoost) and eXtreme Gradient Boosting (XGBoost) to deal with the imbalanced dataset challenge. 3) A large manipulated face dataset that is manually collected and validated. The results from various experiments proved that proposed models outperformed existing expert and intelligent systems which were usually used for the manipulated face image detection task in terms of area under the curve (AUC), computational complexity, and robustness against imbalanced datasets. As a result, the presented framework will motivate the finding of a more powerful altered face images detection method and encourages the integration of the proposed model in applications that have to deal with manipulated images regularly.

1. Introduction

Social networking is the way social media sites offer services for their users to connect with friends, family, classmates, customers, and clients. It serves various objectives, such as social purposes, educational purposes, and business purposes. It is a trend that has been considered promising for the development of civil society (Andreassen, Torsheim, & Pallesen, 2014). Every day, people upload an enormous amount of multimedia contents on popular social websites such as Instagram, YouTube, Facebook, and Twitter. Among them, uploading a photo (Hu, Manikonda, & Kambhampati, 2014) is a faster way of conveying information than writing some text, and easier to approach than video. In case of photos uploaded on social website, most of them are genuine because they capture the moments from people’s lives and are being shared as a part of people’s social experiences. However, in the era of fake news (Gu, Kropotov, & Yarochkin, 2017), more and more manipulated images have appeared on the internet, particularly ones involving facial regions.

One of the most common image manipulation techniques is splicing (Asghar, Habib, & Hussain, 2017), which is a process of taking one part of the face from a source image and injecting it into a target image. To make it even harder for viewers to detect the tampered regions, a correction of the shape, boundary, illumination, and scaling are carried out. Given the advances in computer vision in recent years, it is easy for anyone to try face manipulation with a low budget using mobile applications (Zhou, Han, Morariu, & Davis, 2017) or open-source softwares. Some digital image forgery examples, especially in the facial parts, are shown in Fig. 1. Even after a close inspection of Fig. 1 (c) and (d) from DSI-1 dataset (De Carvalho, Riess, Angelopoulou, Pedrini, & de Rezende Rocha, 2013), there is a high chance that people mistakenly identify a fake image as an original one. Moreover, when manipulated face images were examined on previous face recognition techniques (Moon & Phillips, 2001; Phillips, Moon, Rizvi, & Rauss, 2000; Zhou et al., 2017) detected and recognized (Lee & Lee, 2016) the genuine face and the fake face belong to the same identity.

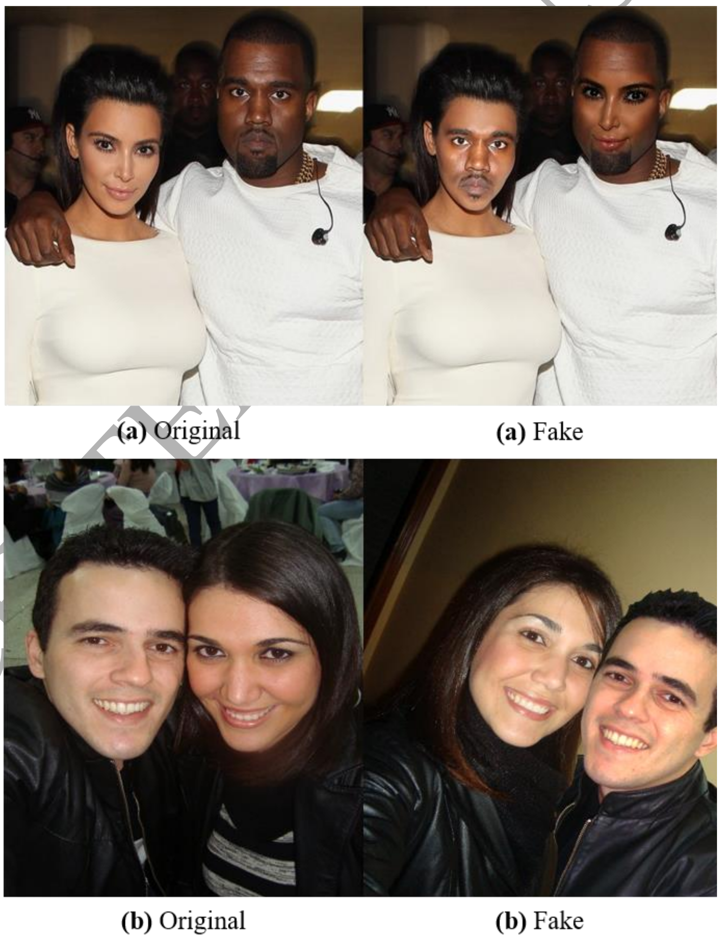


Fig. 1. Images (a) original and (a) fake depict original and fake images downloaded from the internet, whereas images (b) original and (b) fake represent original and fake images taken from DSI-1 dataset.

The consequences would become even more severe if manipulated face images were used for commercial or political motives. Although the identification of image tampering has become an active research topic since the last decade, numerous limitations still exist in current approaches because they focused only on specific evidence that existed in a dataset and ignored other pieces of evidence (Amerini, Uricchio, Ballan, & Caldelli, 2017; Barni et al., 2017; Ferrara, Bianchi, De Rosa, & Piva, 2012; Yao, Wang, Zhang, Qin, & Wang, 2017; Zeng, Zhan, Kang, & Lin, 2017). For example, error level analysis (ELA) fails to detect manipulated images which are carefully edited or generated without lossy compression (PNG image). Color filter array (CFA) works only on original size images whereas double JPEG localization technique is susceptible to image editing; this method fails if many image post-processing steps are implemented. Moreover, traditional approaches (Bappy, Roy-Chowdhury, Bunk, Nataraj, & Manjunath, 2017; Cristin, Ananth, & Raj, 2018) depended heavily on handcrafted features, which were inefficient and time-consuming because usually suitable features and classification algorithms were manually determined based on conducting extensive experiments.

On the other hand, a potential replacement for traditional methods which has thrived recently is deep learning; it has shown excellent performance in image classification tasks, including manipulated facial image detection (Bappy et al., 2017; Dang et al., 2018; Nguyen, Thai, Nguyen-Xuan, & Lee, 2018; Zhou et al., 2017). The central concept of deep learning is to perform both features extraction and classification within one model; it automatically extracts abstract features without a requirement of manually crafted features. However, compared to traditional machine learning approaches, it requires a considerable amount of data and computing power to perform well.

One of the biggest challenges this topic and many other research topics (such as electricity pilferage, fraudulent transactions in banks, and identification of rare diseases) need to deal with is that the number of samples belonging to one class is significantly lower than those belonging to other classes. Two main approaches are usually applied to solve the problem; the first approach uses data level approach or resampling techniques including under-sampling and over-sampling (Yu, Zhou, Tang, & Chen, 2018) with the primary goal is to either increases the frequency of the minority class or decreases the frequency of the majority class, so an equal number of instances for both the classes is roughly obtained. The other one is called algorithmic ensemble methods (Wu, Jing, Shan, Zuo, & Yang, 2017), its primary target is to enhance the classifier’s performance by constructing several two-stage classifiers from original data and then adding up their predictions.

As a result, there is an urgent need to develop a deep learning based expert system which can automatically and efficiently detect manipulated face images, validate their genuineness, and cope with imbalanced dataset scenarios. In this paper, we propose MANFA - a customized convolutional neural network (CNN) model for manipulated face detection to avoid focusing on specific manipulated traits and achieve robust manipulation detection; it is inspired by recent studies on CNNs that revealed the possibility to analyze multiple tampered pieces of evidence (Barni et al., 2017; Zhou et al., 2017). Hybrid MANFA or HF-MANFA that integrates a boosting technique into MANFA to overcome the imbalanced dataset scenario is proposed. HF-MANFA detects face regions in the image and uses them as input data. Next, boosting algorithms are used to extract features and identify manipulated facial images efficiently. Finally, three experiments are conducted to check HF-MANFA robustness on both balanced and imbalanced datasets. In the first experiment, HF-MANFA is compared with a state-of-the-art deep learning model VGG-Face and MANFA for tampered facial images detection on a balanced case. After that, the second test is implemented to check HF-MANFA performance under various balancing ratios between normal and fake facial images (from 1:1 to 1:100). Finally, the performance of the proposed model is evaluated on other manipulated face datasets, namely “SwapMe and FaceSwap” datasets.

With the proposed model, we attempt to find answers to the below questions, using results we have acquired from numerous experiments as a foundation:

1. What is the performance of the deep learning based MANFA and HF-MANFA on the balanced dataset scenario?
2. Does the proposed HF-MANFA model perform well under different imbalanced dataset scenarios?
3. Will proposed models outperform the performance of a state-of-the-art model regarding both accuracy and computational complexity and?

By answering these questions, main contributions of the research are pointed out:

1. A proposal of a large manipulated face dataset which was collected and validated manually.
2. A proposal of MANFA and HF-MANFA models to effectively classify manipulated face dataset.
3. The state-of-the-art performance on the imbalanced dataset is achieved by using the proposed HF-MANFA model.
4. The integration of an ensemble approach into MANFA model brings a robust performance on various imbalanced dataset scenarios.
5. The proposed model outperforms existing models in detecting manipulated face region.

The rest of the paper is divided as follows. Section 1 introduces a problem statement for this research. In section 2, we thoroughly survey previous approaches that studied manipulated image detection and the imbalanced data problem. The proposed tampered face detection frameworks will be explained carefully in section 3. In section 4, various experiments will be implemented to test our proposed models on both imbalanced and balanced cases. After several experiments, section 5 discusses experimental results and provides some comments about the overall performance of the proposed model. Finally, in section 6, we summarize and discuss future approaches.

2. Related work

2.1. Manipulated face image detection

There are emerging research activities on image manipulation detection and localization. Prior research are categorized based on image features, such as double JPEG localization (Barni et al., 2017), local noise estimation (Zeng et al., 2017), pattern analysis (Peng, Wang, Dong, & Tan, 2017), illumination model (Cristin et al., 2018), Color Filter Array (CFA) , and steganalysis feature classification (Holub & Fridrich, 2015). Besides, many CNN-based frameworks (J. Chen, Ou, Chi, & Fu, 2017; Zhou et al., 2017) have achieved state-of-the-art results recently.

The concept behind local noise level estimation based techniques is to highlight differences between image global noise level (normal part) and local noise level (manipulated part) to reveal manipulated regions. For example, Yao (Yao et al., 2017) used a color image as input and then computed noise level function (NLF) to disclose noise level inconsistency in different regions from the manipulated image. The proposed approach had higher accuracy in data fitting. Moreover, the model did not require a massive amount of data for training, and optimized parameters such as NLF and CRF were applied simultaneously in the Bayesian inference. Zheng (Zeng et al., 2017) estimated a level of block-wise local noise because they assumed that manipulated regions and normal regions in an altered image had distinctive noise levels. The proposed model yielded good results, even when noise level between the manipulated region and the original region was unnoticeable. However, this approach performs poorly when post-processing techniques, such as image blending and filtering are applied to lower the inconsistency between global noise and local noise.

Double JPEG localization techniques can be categorized as either non-aligned double JPEG compression or aligned double JPEG compression (Guo, Liu, & Wu, 2013), classification decision is determined by checking whether quantization factors align well after applying double JPEG compression to an image. This method depends on the concept that background regions go through JPEG compression two times while manipulated regions do not. An example of this is (Amerini et al., 2017) applied multi-domain convolutional neural networks to detect double JPEG compression. Reported results showed that when a spatial domain was used directly or was combined with a frequency domain produced higher performance. Barni (Barni et al., 2017) investigated the performance of CNN for aligned and non-aligned double JPEG compression detection. CNN with self-learned features outperformed state-of-the-art methodologies in all conducted test scenarios. The weakness of this approach is that it mainly relies on the double JPEG assumption, and it is also susceptible to post-processing techniques.

Color filter array (CFA) analysis approaches assume that a CFA pattern is distinguishable between altered regions and genuine regions since different imaging devices or manipulating processes generate low-level artifacts. By detecting CFA patterns for a manipulated image, it has the potential to differentiate authentic regions and manipulated regions. For example, (Ferrara et al., 2012) presented a framework which calculated the filter pattern of the camera based on the assumption that difference of prediction error between CFA absent regions (manipulated regions) and CFA present regions (authentic regions) was different. After training a Gaussian mixture model

(GMM) classifier, manipulated areas could be detected. Although the proposed method included CFA aware steganalysis features, they also added a second stream that searched for additional pieces of evidence. Nevertheless, the hypothesis could be wrong if a tampered region had an identical CFA pattern or an image is rescaled which eliminated original CFA information and added new noise.

Illumination based method's goal is to detect illumination inconsistencies between manipulated regions and authentic regions. For instance, a splicing face (from another image) and an original face (from the same image) will have separate lighting orientations. (Peng et al., 2017) presented a reflection model which incorporated facial texture information and non-convex geometry, which was more appropriate for genuine faces. As a result, this technique was more effective and robust for image forgery detection which was verified through various experiments. (Cristin et al., 2018) exposed forgery by applying the illumination texture descriptor and trained a support vector neural network (SVNN) classifier. The experiment was conducted on two datasets and evaluated using training percentage and k-fold cross-validation. The model achieved an accuracy of approximately 95%. Although approaches based on the lighting environment are useful for detecting photographic composites of faces, they perform poorly on images with a complex scene.

Steganalysis approaches extract various low-level features which can become a local descriptor for the image. By analyzing co-occurrence statistics of nearby noise residual pixels acquired from numerous linear and non-linear filters. (Farooq, Yousaf, & Hussain, 2017) presented a spatial rich model (SRM) and combined it with textural features like local binary pattern (LBP). Experimental results proved that co-occurrence matrices using both BEST-q-CLASS feature selection procedures and LBP obtained the highest accuracy of 98.4%. (Holub & Fridrich, 2015) proposed a new feature set for steganalysis on JPEG images. They named it DCTR because the features were extracted from noise residuals obtained using the 64 dual-clutch transmission bases. The feature had very low dimensionality (8,000) which led to remarkably low computational complexity while achieving a reasonable detection rate among other JPEG algorithms. The steganalysis-based approaches deliver excellent performance on tampered region detection because these methods use a set of low-level features. However, it cost much time to analyze and pick the appropriate features set.

Deep learning has been extensively applied in various computer vision topics recently because of its promising performance. For example, (Zhou et al., 2017) designed a two-stream manipulated face detection technique, they extracted tampering artifacts, hidden noise residual features and trained on GoogLeNet. The results showed that their model outperformed original features detection because CNN learned both hidden noise residual features and tampering artifacts. On the other hand, (Bappy et al., 2017) employed a more complex CNN-LSTM model which captured discriminative features between manipulated and non-manipulated regions. The framework was capable of detecting different types of image manipulations, including copy-move, removal, and splicing. Even though deep learning architectures have been applied in some of the manipulated facial image detection research, models proposed in this research area were mainly pre-trained models with a few or no modification in the CNN structure. Moreover, datasets were small, and these models were ineffective against the imbalanced dataset problem.

2.2. Imbalanced data problem

For an imbalanced dataset problem, there are two primary approaches: data level techniques (Yu et al., 2018) and algorithmic ensemble techniques (Fernández et al., 2018). The data level approach tries to balance samples between classes before feeding them into a classifier; it includes over-sampling and under-sampling. It is not affected by a learning algorithm being used, so most of the studies have followed this approach. (He & Garcia, 2008) presented a structured review of metrics and algorithm-level approaches, they also did some experiments on unbiased classifiers by changing sampling frequency. However, there are some disadvantages in data level techniques; under-sampling techniques can discard potentially useful information which could be crucial for building classifiers' rule whereas over-sampling techniques creates the likelihood of overfitting since it replicates the minority class events.

Recently, ensemble approaches (Wu et al., 2017) have drawn considerable interests. The main idea is to train several models and combine their classification results to yield a single class label which will lead to higher accuracy. They are categorized into bagging based and boosting based approaches. In bagging approach, 'n' different bootstrap training samples with replacement are generated. After that, each bootstrapped training samples were trained separately and then aggregating the predictions at the end. In case of boosting approach, three sequential ensemble algorithms that have gained tremendous popularity recently are AdaBoost (Freund & Schapire, 1997), gradient boosting (Friedman, 2002), and XGBoost (T. Chen & Guestrin, 2016). Recently, several research studies have been done to improve the existing ensemble approaches; (Lu, Ke, Zhang, Mei, & Xu, 2017) proposed an improved algorithm designed for solving the binary imbalanced classification problem namely IW-ELM (weighted extreme learning machine). There were three major steps: train k weighted ELM classifiers, remove unusable classifiers, and determine the final result based on majority voting of remaining classifiers. Simulation results demonstrated that IWELM achieved higher accuracy compared to other ELM based algorithms. (Ren et al., 2018) proposed an ensemble-based approach called Gradual Resampling Ensemble (GRE) for learning different kinds of concept drifts from imbalanced data, the results showed that GRE achieved high performance. Besides, class-weight learning approach is also used frequently; it assigns misclassification costs to data from each class and forces a classifier to concentrate on the minority classes. (Khan, Hayat, Bennamoun, Sohel, & Togneri, 2018) introduced a cost-sensitive deep neural network; it automatically extracted robust feature that represented both majority and minority classes. Obtained results on six public classification datasets proved that customized cost functions worked well on the majority as well as on the minority classes in the dataset.

The most appropriate technique for solving the imbalanced class problem depends heavily on the characteristics of the imbalanced dataset (Khoshgoftaar, Fazelpour, Dittman, & Napolitano, 2015). Relevant evaluation parameters should be carefully examined during model selection to choose the most suitable technique for the imbalanced dataset.

3. Methodology

In this section, a full process of proposed MANFA and HF-MANFA models is explained thoroughly. It includes 1) Implementation details of the MANFA model, 2) Implementation of HF-MANFA model by adding AdaBoost, and XGBoost layers to deal with imbalanced dataset. Before the proposed models are trained, faces are first detected from input images; next, all face images are fed into several models, then through some convolutional layers, abstract features are extracted. Finally, a softmax layer is trained to classify an input image into a “fake” or “normal” face image. Furthermore, we attempt to cope with the imbalanced scenario by replacing MANFA’s softmax layer with XGBoost or AdaBoost layer to build the HF-MANFA model. All the models are tested under various imbalanced dataset environments and compared with other state-of-the-art models. The proposed MANFA framework is described in Section 3.1 whereas HF-MANFA framework is discussed in Section 3.2.

3.1. MANFA model

In this part, we explain in detail a manipulated image detection framework called MANFA. First of all, we have to figure out a suitable CNN architecture, and it depends heavily on practical points. Based on image input size (64x64) and dataset size. We found out that five convolution layers were adequate to handle the categorization problem. Once the number of convolution layers was set, we then examined the best kernel size to map (64x64) input to (1x2) output. In order to control the parameters’ flow, each layer needs to have an appropriate kernel size. Fig. 2 explains each layer with their corresponding input, kernel, and output size. The model maps an input 64×64 image into two output nodes, one node for “normal” and the other for “fake.” Overall, the proposed MANFA contains five convolutional layers (Convolution1 to Convolution5), four max-pooling layers (Max-Pool1 to Max-Pool4) followed by two dense layers. The final output from the dense layer includes two classes “normal” or “fake.” In the proposed model, rectified linear unit (ReLU) nonlinearity function ($f = max(0, x)$) is applied to the activation layer. It was proved to have higher fitting abilities than hyperbolic or sigmoid function (Glorot, Bordes, & Bengio, 2011). A max pooling layer is usually attached after the convolutional layer to lower the spatial size of feature maps and prevent the overfitting problem.

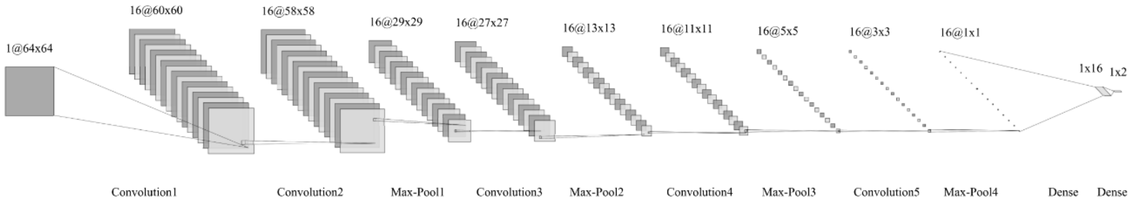


Fig. 2. MANFA architecture with detailed configuration for each layer.

In our work, Keras library which is a well-known open source neural network library written in Python (Chollet, 2015) was used to construct and train the proposed model. The optimization algorithm used in MANFA is Adam optimization (Kingma & Ba, 2014) with a learning rate of 0.001 through 50 epochs, and a batch size of 32**.** Table 1 shows a configuration of dropout layers in the MANFA model; the dropout rate was set to 0.2 across all the dropout layers.

Table 1. A configuration of MANFA framework. The activation function is hidden for compactness.

MANFA		
Layer	**Configuration**	**Output (rows, cols, channels)**
Input		64x64 – Gray Image
Convolution_1	5x5 – 16 kernels	(60, 60, 16)
Convolution_2	3x3 – 16 kernels	(58, 58, 16)
Maxpool_1	2x2	(29, 29, 16)
Dropout_1	0.2: Probability	(29, 29, 16)
Convolution_3	3x3 – 16 kernels	(27, 27, 16)
Maxpool_2	2x2	(13, 13, 16)
Dropout_2	Probability: 0.2	(13, 13, 16)
Convolution_4	3x3 – 16 kernels	(11, 11, 16)
Maxpool_3	2x2	(5, 5, 16)
Dropout_3	Probability: 0.2	(5, 5, 16)
Convolution_5	3x3 – 16 kernels	(3, 3, 16)
Maxpool_4	2x2	(1, 1, 16)
BatchNorm		(1, 1, 16)
Dropout_4	Probability: 0.2	(1, 1, 16)
Flatten	Length: 16	(16)
Dense	Length: 2	(2)

3.2. HF-MANFA model

Through various experiments, we observed that the model's performance declined significantly when it was trained on an imbalanced dataset. As a result, we propose one more framework that is more powerful in dealing with manipulated face detection in imbalanced dataset scenarios, namely HF-MANFA (Fig. 3).

In the training phase, face regions are first detected and extracted by a facial landmark detector (Kazemi & Sullivan, 2014) and fed into MANFA model. We remove the last output dense layer, which is responsible for extracting features from the flattened layer of MANFA model and giving a final classification decision. For each image, 16 feature vectors extracted from trained convolutional neural network are then trained on both XGBoost and AdaBoost. They are machine learning algorithms which can effectively deal with the imbalanced dataset.

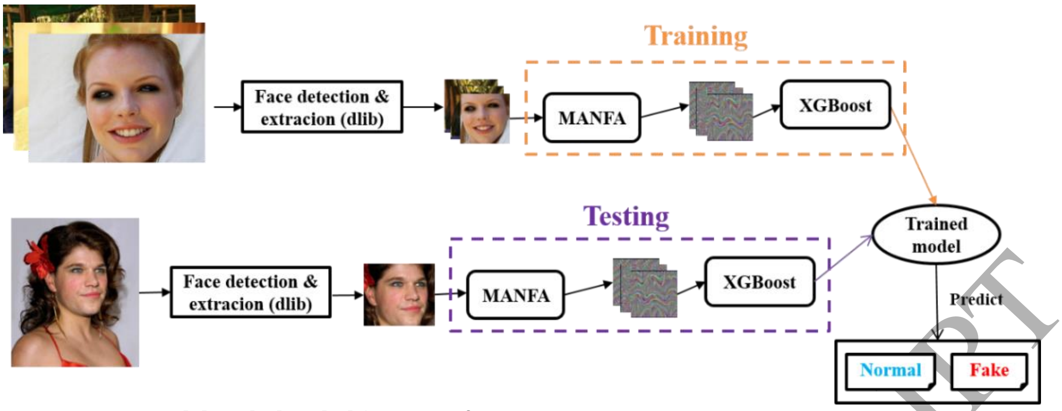


Fig. 3. HF-MANFA model with detailed layers' information.

In the testing phase, face regions are acquired from test images and then fed into the MANFA model to extract feature vectors. Extracted features are then classified by trained XGBoost and AdaBoost models to predict whether an image is a fake or normal.

A. Imbalanced data problem

In practical application, fake face dataset is an imbalanced data problem where the number of fake images is minor compared to normal images. To simulate this situation, we establish a new dataset which has a tiny number of fake images in contrast with the dominance of normal images. ϑ represents the dataset, and ϑ_{sm} and ϑ_{vm} indicate the small minority class of fake images, and the vast majority class of normal images, respectively. A dataset balancing ratio br_{ϑ} is computed as:

$$br_{\vartheta} = \frac{|\vartheta_{sm}|}{|\vartheta_{vm}|} \tag{1}$$

where $|.|$ represents cardinality of a set. In the problem of the imbalanced dataset, the more extreme the balancing ratios become, the faster the minor class accuracy drops. Resampling technique converts ϑ into a new dataset ϑ_{re} such that $br_{\vartheta} < br_{\vartheta_{re}}$ while ensemble techniques have a different approach. For the ensemble classifier in a binary classification, $C(\vartheta_{sm}, \vartheta_{vm})$ is described as a cost of majority class samples being classified as minority class samples whereas $C(\vartheta_{vm}, \vartheta_{sm})$ indicates a cost of the remaining cases. The motivation of ensemble approaches is to generate a model with the lowest misclassification cost which is calculated as follows:

$$Cost = C(\vartheta_{sm}, \vartheta_{vm}) \times FN + C(\vartheta_{vm}, \vartheta_{sm}) \times FP \tag{2}$$

where *FP* and *FN* are the numbers of false positive and false negative samples respectively.

In the imbalanced class issue, it is mandatory to focus on the accuracy of correctly classified minor samples. For example, in a testing set, normal samples occupy 95% while fake samples hold only 5% of the entire dataset. If a model predicts all of the samples belong to the normal class, the accuracy is considered to be 95%. In this case, we can be easily fooled by the high performance of the system. That is the reason why a receiver operating characteristic (ROC) curve has been commonly used along with accuracy so that we can perceive the performance of the system thoroughly. To compare two or more models, an area under the ROC curve (AUC) is usually applied as a primary evaluation protocol for classification measurement. The higher the value of AUC is, the better performance the model achieves.

B. Gradient boosting for the imbalanced data problem

Gradient boosting is a learning algorithm designed explicitly for regression and classification problems, which constructs a model from a collection of weak prediction models (decision trees). It begins with a simple decision tree. After training the tree, it is recorded for which samples the tree makes classification mistakes. After that, a second tree is created and trained to evaluate prediction outputs from previous trees and tries to improve predictions based on correct class labels. Then, another tree is generated which tries to estimate the error of its preceding tree and so on.

Extreme gradient boosting (XGBoost) is one of the most famous algorithms in supervised learning these days; it is also one of the most common implementations of the gradient boosting technique. It was proposed by (T. Chen & Guestrin, 2016), the algorithm was fast compared to other gradient boosting approaches.

$$Obj\ = L + \ \Omega \tag{3}$$

Where L is a loss function which is responsible for predictive power, and Ω is a regularization component which manages the overfitting and simplicity of the model. The regularization component Ω is determined by the number of leaves and prediction threshold allocated to leaves in the tree ensemble model. A loss function L can be either Logloss for binary classification, mlogloss for multi-class classification, or Root Mean Squared Error for regression.

Besides, Adaptive Boosting, short for AdaBoost (Freund & Schapire, 1997) is another well-known machine learning algorithm that effectively solves the dataset imbalanced scenario; so it is also applied to learn MANFA model's extracted features. Given a training set $TS = (x_1, y_1) \dots (x_m, y_m)$, where $x_i \in X, y_i \in Y = \{-1, +1\}$; it manages to preserve a set of weights or a distribution on the training set. AdaBoost attempts to discover a weak hypothesis $X\ \rightarrow (-1, +1)$. The final output is based on the majority vote from all hypotheses:

$$H(x)\ = sgn(\sum_{t=1}^{T} \alpha_t \mathrm{h}_t) \tag{4}$$

where T is the total number of the weak hypotheses and α_t is a weight assigned to h_t.

4. Experimental Results

The experimental section represents all experiments that were carried out on two datasets to address research questions introduced in the introduction (Section 1). One of the datasets is the proposed MANFA dataset, the other was suggested by (Zhou et al., 2017). The metric applied throughout this research is the AUC score.

First experiment's (Section 4.3) primary purpose is to check proposed models' performance in a balanced dataset scenario, whereas second experiment (Section 4.4) validates different models' performance in the imbalanced dataset scenario. Then, proposed models are compared with the state-of-the-art model on the "SwapMe and FaceSwap" dataset in the third experiment (Section 4.5). Finally, we also evaluate the computational time of different models on MANFA dataset (Section 4.6).

4.1. Evaluation metric

In linear classification, classifier performance can be portrayed on a confusion matrix, as shown in Table 2. From the confusion matrix, two major evaluation metrics are computed, including true positives rate (TPR) and false positives rate (FPR); they are described below:

$$\begin{aligned} TPR &= \frac{TP}{TP+FN} \\ FPR &= \frac{FP}{FP+TN} \end{aligned} \tag{5}$$

where TP, FN, FP, and TN are metrics taken from the confusion matrix in Table 2. In a ROC curve (Bradley, 1997), a TPR is represented in a function of a FPR for separate cut-off points. Each point on the curve depicts a sensitivity/specificity pair corresponding to a specific decision threshold.

After the ROC curve is generated, AUC (Bradley, 1997) or the area under the ROC curve is utilized to compare two classifiers. If an arbitrary ROC curve (E1) occupies a larger AUC than other ROC curve (E2), then the classifier of E1 is considered to obtain better performance than the classifier of E2.

Table 2. A confusion matrix for summarizing classification results.

		Prediction	
		Normal	CG
Actual	Normal	TP (True positive)	FN (False negative)
	CG	FP (False positive)	TN (True negative)

4.2. Dataset

A. MANFA dataset

Although there are many public image manipulation datasets, they are not appropriate for manipulated face detection. For example, Columbia image splicing dataset (Ng, Hsu, & Chang, 2009) and CASIA dataset (Dong, Wang, & Tan, 2013) are huge, but most of the images do not contain human faces. Besides, (De Carvalho et al., 2013) proposed a high-resolution DSI-1 dataset for face manipulation. However, the number of images was limited with only 25. As a result, in this work, we propose a dataset called MANFA which is particularly collected for the task of altered face identification.

The dataset contains 21,000 face images with unconstrained conditions such as pose, background cluttered, illumination changes, and so forth; it contains faces in a wide range of ethnicities, genders, personal identities, glasses, ages, and facial hair. Initially, 4,200 images labeled as "fake" and 7,450 images labeled as “normal” were gathered. To analyze proposed models under extreme imbalanced dataset scenario, we increased the number of normal images by adding 192,550 images from CelebA dataset (Amerini et al., 2017) to the “normal” class, so the total number of images belonging to “normal” class increased from 7,450 to 200,000. Image size varies from 82x82 to 1098x1098. Fig. 4 shows four samples of face images in our database. The MANFA dataset has some advantages: 1) It is a facial image manipulation dataset that contains only face regions and is uniquely generated for manipulated face detection task. 2) Tampering quality is excellent which make some fake images look real.

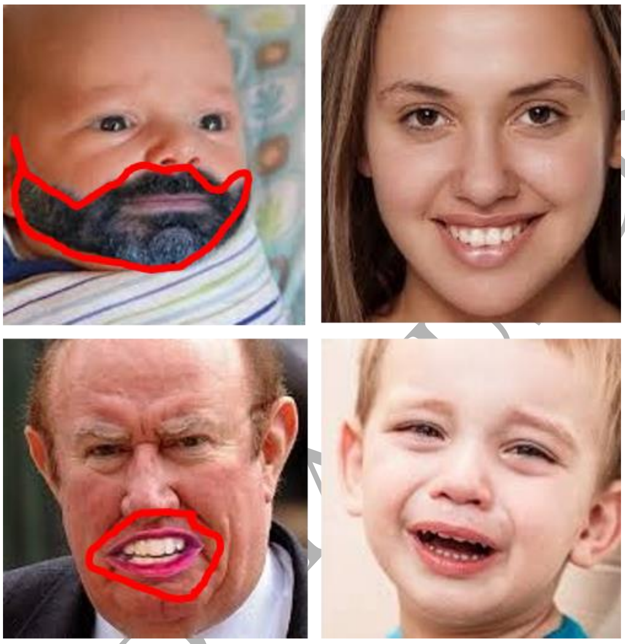


Fig. 4. Examples of face images from our MANFA dataset. Left column: Fake, red line indicates the manipulated region; Right column: Normal faces.

B. “SwapMe and FaceSwap” dataset

Zhou proposed this dataset in (Zhou et al., 2017), it was generated by using an iOS application called SwapMe and an open-source application called FaceSwap. A source face image and a target face image were fed into the programs, and they automatically replaced the target face with the source face. After that, a few post-processing processes, such as resizing, blending and boundary-blurring, were applied which made it hard to distinguish between the tampered image and an authentic image visually.

The training set contains 705 fake faces and 1,400 normal faces while the testing set has 900 normal faces and 300 manipulated faces. Fig. 5 represents sample images from both SwapMe and FaceSwap dataset. This dataset will be used to examine the effectiveness of our model against dataset from other research.

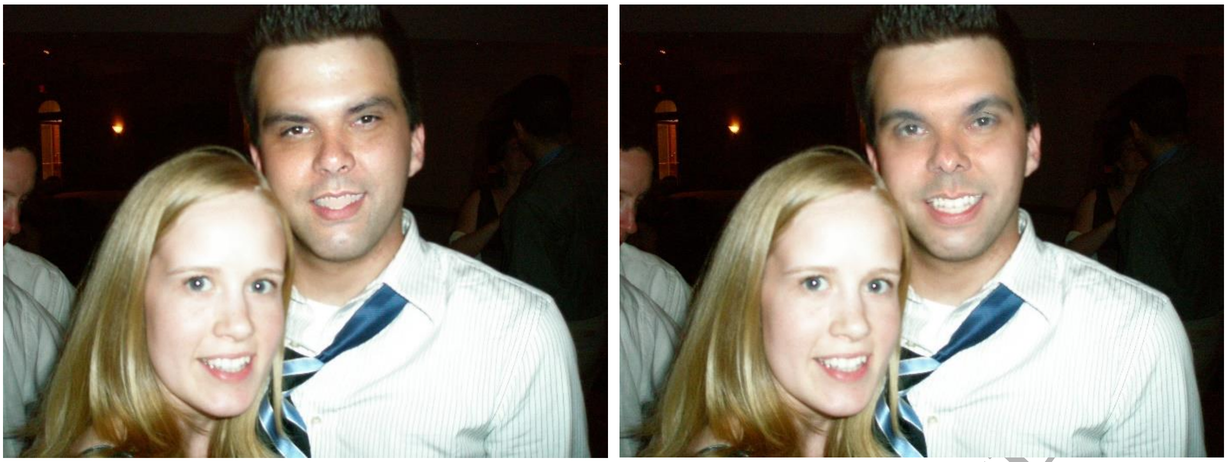


Fig. 5. An example of face images originated from “SwapMe and FaceSwap” dataset. Left column: Tampered face image from “SwapMe”; Right column: Tampered face image from “FaceSwap”.

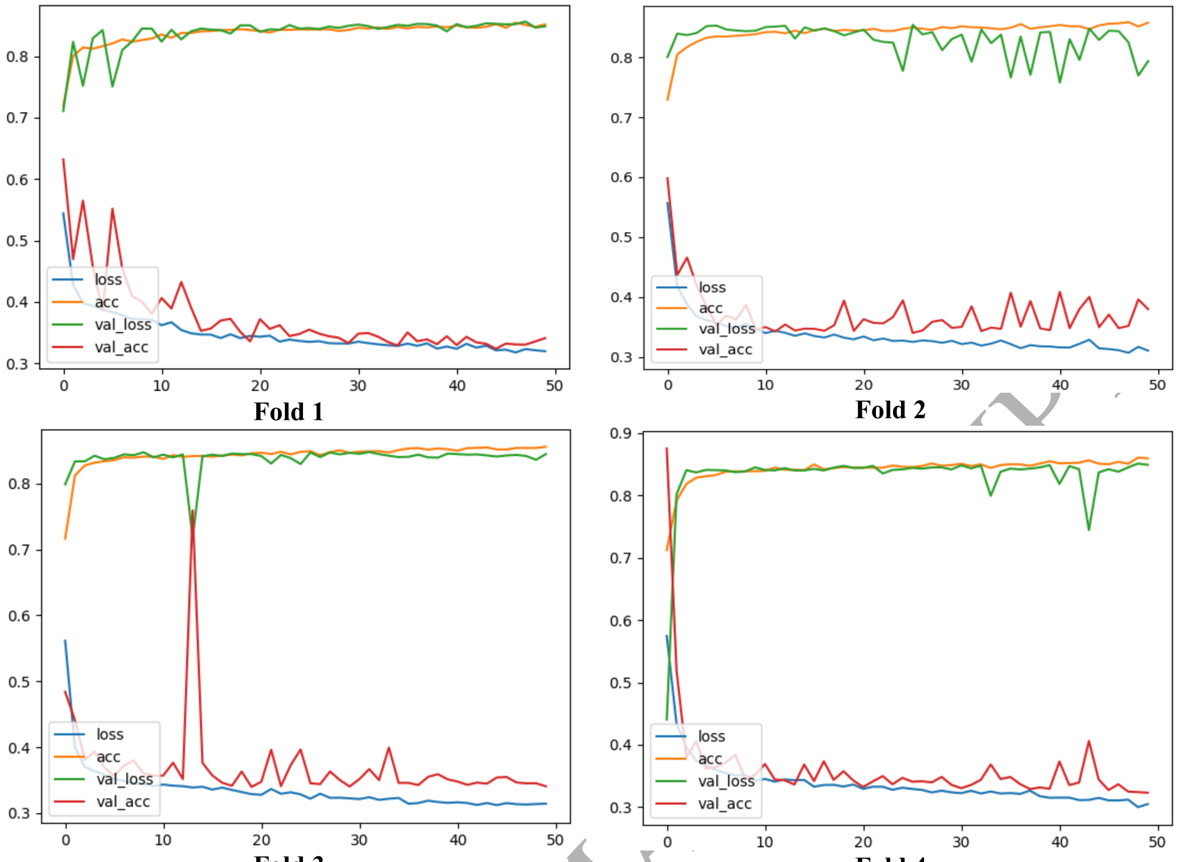
4.3. Balanced scenario experiment

First of all, we evaluate the performance of MANFA model in a balanced scenario by comparing it with other deep learning models. Human faces are first detected by using the facial landmark framework proposed by (Kazemi & Sullivan, 2014) and then resized to 64×64. To eliminate pose changes, face images are rotated and aligned to frontal.

Table 3. The numbers of "fake" and "normal" images in four subsets.

	Subset 1	**Subset 2**	**Subset 3**	**Subset 4**
Fake	1,049	1,053	1,043	1,055
Normal	1,051	1,047	1,057	1,045

We randomly split the original dataset into four subsets. Each subset contains a total of 2,100 samples. The distribution of “fake” and “normal” images are shown in Table 3. Cross-validation is then implemented on each subset. Each time, three subsets are fed into the model for training while the other subset is used for testing purpose. In the training process, 80% of training data is used by the convolutional neural model to learn and update weight whereas the remaining 20% will be used as validation data to fine-tune parameters. Full validation accuracy and loss for each fold are given in Fig. 6. The model is trained using Adam optimization (Kingma & Ba, 2014) through 50 epochs with a batch size of 32.



Fold 1

Fold 2

Fold 3

Fold 4

Fig. 6. Training accuracy (acc), training loss (loss), validation accuracy (val_acc), validation loss (val_loss) for four-fold cross validation through 50 epochs.

As shown in Fig. 6, training and validation accuracy grow dramatically to over 80%. On the other hand, training and validation loss drops significantly to 35% after the 5th epoch. For the rest of the training process, training and validation accuracy gradually increase and reach a peak at 86%. In contrast, training and validation loss slowly decrease and hit bottom at 32%. Within four folds, fold 1 yields the best results concerning validation accuracy and validation loss as well as model stabilization. On the other hand, other folds have some fluctuations in validation accuracy and validation loss.

Class activation map (CAM) is used to show that the proposed model successfully justified weight to classify manipulated image. It projects class-specific weights of the output classification layer back to feature maps of the last convolutional layer (Convolution_5), thus highlighting important regions for predicting a particular class. As shown in Fig. 7, manipulated regions from the left column were correctly highlighted in CAM images which proved that the model could classify manipulated images based on identifying manipulated regions.

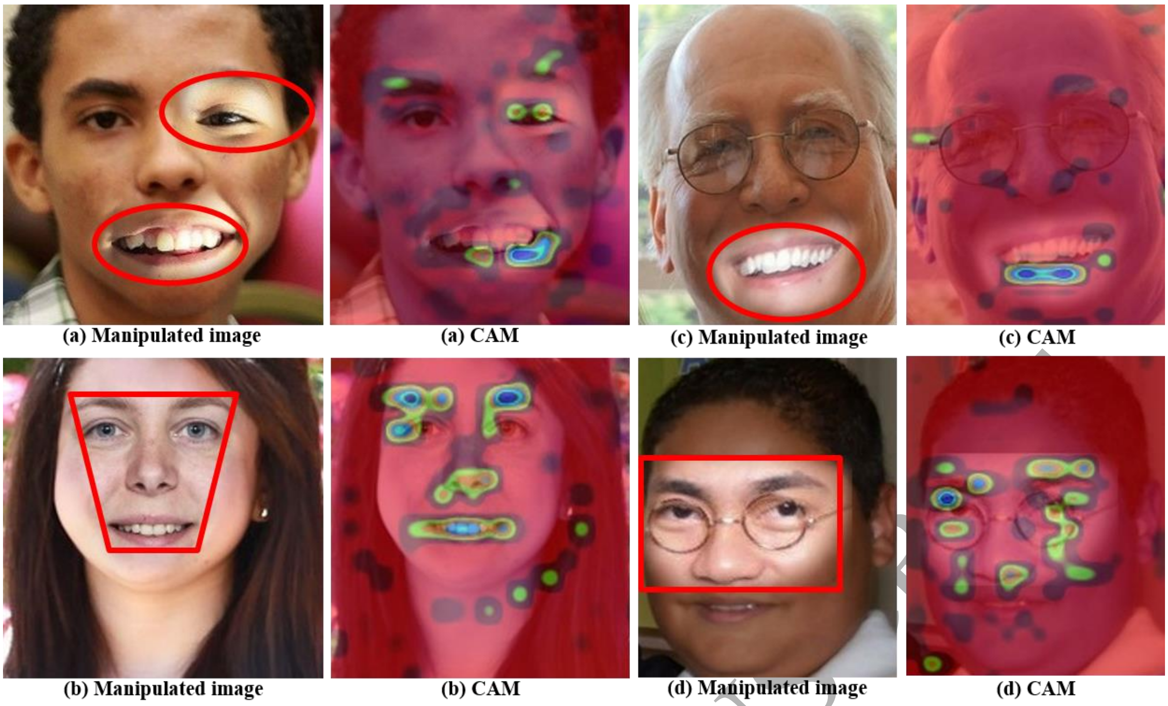


Fig. 7. The proposed system interpretation via class activation mapping (CAM). For each case, a left image shows face with manipulated region(s) (red shapes), whereas a corresponding right image represents class activation map of the manipulated face image that is classified by MANFA model.

The convolutional neural network extracts different types of features on each layer, and layers which are closer to the output layer learn more abstract concepts. In the next section, we remove the last dense layer and extract output feature map from the final hidden layer of the model from fold 1. Features are then trained by AdaBoost and XGBoost. We also implement transfer learning using a pre-trained Oxford VGGFace model (Parkhi, Vedaldi, & Zisserman, 2015) for manipulated face image classification problem. We use the pre-trained model as fixed feature extraction, and then use extracted features to feed the classifier. Then similar to MANFA model, features are extracted from VGGFace model then trained by AdaBoost and XGBoost.

Performance comparison of MANFA, ADA-MANFA, XGB-MANFA, VGG-Face, ADA-VGGFace, and XGB-VGGFace models (different classifier and features type) are shown in Table 4. It is noticeable that MANFA model achieved good performance at 84.7% in terms of accuracy and AUC at 0.81, but the pre-trained VGG-Face obtained a slightly better performance with the AUC value at 0.89. Moreover, ADA-MANFA with AdaBoost classifier yielded even better accuracy at 85.4% with AUC value at 0.89, while XGB-MANFA achieved an accuracy of 87.1% with AUC value at 0.9 which outperformed both MANFA and pre-trained VGG-Face models. However, after we applied AdaBoost and XGBoost classifiers to VGGFace model, ADA-VGGFace got the accuracy of 94.5% and AUC value at 0.89, and XGB-VGGFace obtained the highest accuracy of 95.1% and the highest AUC value at 0.91. In this balanced dataset experiment, XGB-VGGFace achieved the highest accuracy and AUC.

Results suggest that batch normalization and dropout layers which were placed after each convolution layer reduced not only overfitting data during training but also increased the robustness of manipulated face detection. Hybrid models produced a significantly better performance, which proved that they are a suitable approach for coping with the imbalanced dataset problem. There was a minor difference regarding performance between AdaBoost and XGBoost, so more experiments are required to indicate which algorithm is more suitable for MANFA dataset.

Table 4. Performance of different models on MANFA dataset.

Model	Feature	Classifier	Accuracy (%)	AUC
VGGFace	Raw pixels	Softmax	82.5±0.02	0.89±0.009
ADA-VGGFace	Raw pixels	AdaBoost	94.5±0.006	0.89±0.002
XGB-VGGFace	Learned feature	XGBoost	**95.1±0.004**	**0.91±0.001**
MANFA	Raw pixels	Softmax	84.7±0.14	0.81±0.006
ADA-MANFA	Raw pixels	AdaBoost	85.4±0.003	0.89±0.004
XGB-MANFA	Learned feature	XGBoost	87.1±0.008	0.90±0.005

4.4. Imbalanced scenario experiment

In the imbalanced experiment, the dataset is reorganized with balancing ratio range from 1:1 (the number of fake faces is equal to the number of normal faces) to 1:100 (the number of fake faces is a hundred times less than the number of normal faces).

The initial experiment with a balancing ratio of 1:10 was used to observe the performance of class-weight and ensemble approaches in dealing with imbalanced dataset. Data used in this experiment was originated from MANFA dataset which contains a total of 4,200 images from the fake class, and 42,000 randomly selected images from the normal class. Then, we divided the dataset into four subsets, each subset contains a total of 11,550 samples. The distribution of "normal" and "fake" images are shown in Table 5.

Table 5. The number of "fake" and "normal" images in each subset with a 1:10 balancing ratio.

	Subset 1	Subset 2	Subset 3	Subset 4
Fake	1,045	1,053	1,051	1,051
Normal	10,505	10,497	10,499	10,499

Two classifiers used in this experiment are the proposed MANFA class and state-of-the-art VGGFace classifiers. We implement VGGFace classifier by following a transfer learning procedure. For class-weight approach, every instance of "fake" class is treated as 10 instances of "normal" class to force two classifiers to treat the "fake" and "normal" classes equally, we achieve this setting by applying *class_weight* parameter in keras library which set higher loss value to the minority class to help classfiers concentrate more on "fake" class. For ensemble approach, after training two classifiers with the 1:10 imbalanced dataset, output features map from the hidden layer before the final dense layer of both MANFA and VGGFace models are extracted. Then, features are trained in the AdaBoost and XGBoost classifiers. Finally, eight models including MANFA, CW-MANFA (class-weight based MANFA classifier), ADA-MANFA, XGB-MANFA, VGGFace, CW-VGGFace (class-weight based VGGFace classifier), ADA-VGGFace, and XGB-VGGFace are evaluated to discover the best model. As shown in Fig. 8, in 1:10 imbalanced dataset, in which the number of fake faces is tiny compared to the number of normal images, MANFA and VGGFace models get low AUC values at an average of 0.768 and 0.734, respectively. It means that these two models performed poorly on imbalanced dataset because they misclassified almost all fake images. In contrast, take fold 1 as an example, CW-MANFA and CW-VGGFace achieved AUC value of 0.863 and 0.864, respectively, which means class-weight approach can solve class imbalanced issue. However, ADA-VGGFace and especially XGB-VGGFace performed even better with AUC values of 0.911 and 0.936, respectively whereas XGB-MANFA achieved the highest AUC of 0.944. Through the observed results, both class-weight and ensemble approaches improved the performance of MANFA and VGGFace on imbalanced dataset significantly. Besides, ensemble approach, especially XGB is more suitable for our dataset as it obtained the highest AUC value of 0.944.

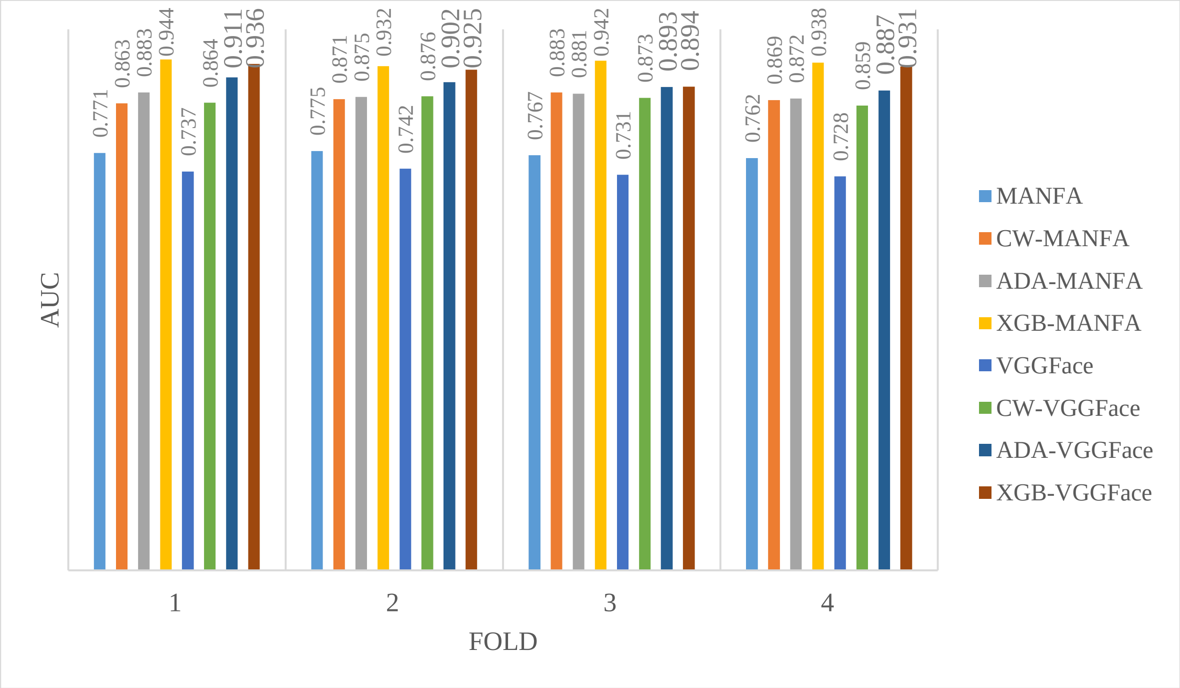


Fig. 8. A comparison in terms of AUC for various methods through 4 folds on the imbalanced dataset where balancing factor of fake faces is minor (No. fake faces / No. normal faces = 1:10).

Next, we examine the proposed model in all possible balancing factors. Table 6 shows the number of normal and fake images for eleven cases of the imbalanced scenario. A total of 2,000 fake images and 200,000 normal images were taken from MANFA dataset and used in this experiment.

Table 6. The number of "fake" and "normal" images on various balancing ratios.

	1:100	1:10	2:10	3:10	4:10	5:10	6:10	7:10	8:10	9:10	10:10
Fake	2,000	2,000	2,000	2,000	2,000	2,000	2,000	2,000	2,000	2,000	2,000
Normal	200,000	20,000	10,000	6,666	5,000	4,000	3,333	2,857	2,500	2,222	2,000

As shown in Fig. 9, MANFA and VGGFace models were affected by imbalanced data when the number of fake faces is minimal compared to the number of normal faces. AUC value of MANFA and VGGFace models drop dramatically to 0.59 and 0.62 when balancing factor is equal to 1:100. They fluctuate and become stable when balancing factor increases. As a result, MANFA and

VGGFace models mostly concentrated on the majority class to obtain the highest result and ignored the minority one. On the other hand, AUC values obtained from hybrid models were more stable and always over 0.8 compared to that from original models. Among hybrid models, XGB-VGGFace and especially XGB-MANFA obtained higher AUC when they were evaluated on extreme imbalanced dataset scenarios. For 1:100 balancing ratio, XGB-VGGFace got 0.89 AUC whereas XGB-MANFA obtained 0.9 AUC. We notice that XGB-MANFA and XGB-VGGFace kept fluctuating around 0.87 and 0.95, XGB-MANFA worked better than XGB-VGGFace when balancing ratios were from 1/100 to 4/10. However, when XGB-VGGFace performed better for remaining balancing factors. On average, XGB-MANFA achieved a slightly better performance compared to XGB-VGGFace. Based on the results, we conclude that XGB-MANFA outperforms other models and achieves more stable results which AUC values always remains over 0.89 even in the most imbalanced case (balancing factor 1:100).

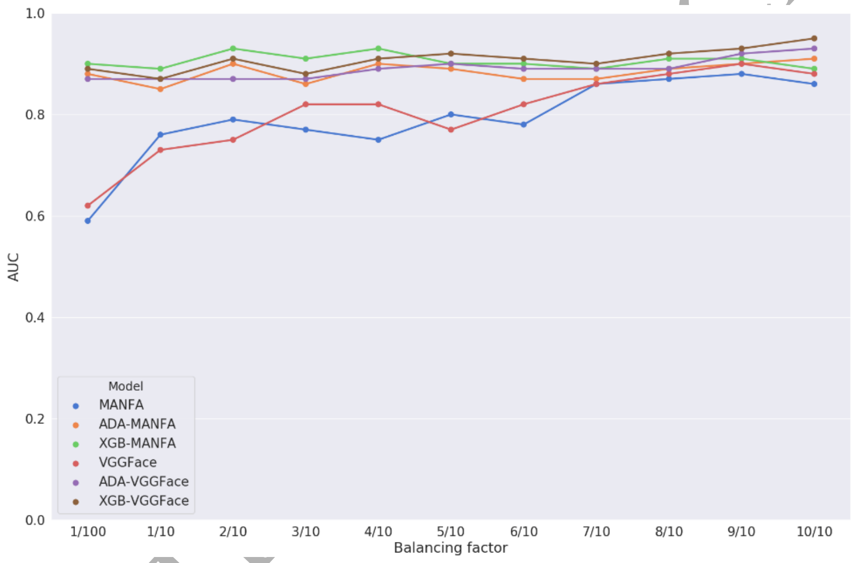


Fig. 9. AUC values of various methods on imbalanced dataset scenarios, where the number of fake samples is minor compared to the number of normal samples (No. fake / No. normal =1:100, 1:10, 2:10, 3:10…10:10).

4.5. Performance on the "SwapMe and FaceSwap" dataset

In the last experiment, the performance of the proposed model was evaluated on other manipulated face dataset which was named "SwapMe and FaceSwap" dataset collected by (Zhou et al., 2017). As described in (Zhou et al., 2017), the dataset contains 705 manipulated and 1,400 authentic faces for training, and 900 authentic faces and 300 tampered faces for testing. Although authors shared the dataset, they did not provide genuine images that were used for training, so as a replacement, we randomly selected 1,400 authentic faces image from our MANFA dataset.

In (Zhou et al., 2017), authors proposed a two-stream framework for classifying whether a face image was real or fake. One stream was a classifier trained on GoogleNet while the other stream was a patch triplet stream which extracted steganalysis features by using triplet loss. After training, this stream showed that a pair of patches originated from the same image was closer in embedding space while the distance between a pair of patches from two different images was immense.

In previous experiments, we proved that MANFA model worked well with XGBoost classifier and obtained the highest AUC value. As a result, in this experiment, we retrain XGB-MANFA model using the new dataset with a balancing ratio of 5:10. All face regions are first detected by using dlib library, then are resized to 64x64. Next, a model is trained using the learning rate of 0.001 through 50 epochs and a batch size of 32 which was similar to previous experiments.

By observing results shown in Table 7 and Fig. 10, XGB-MANFA yields AUC of 0.934 which means the model correctly predicted whether an image belongs to major genuine face class and especially minor tampered face class. It also achieved higher AUC value compared to the best AUC at 0.927 reported in (Zhou et al., 2017), when both GoogLeNet classification stream and patch triplet stream were used.

Table 7. AUC values reported in (Zhou et al., 2017) compared with our model.

Method	AUC
Face classification stream	0.854
Patch triplet stream	0.875
Two-stream network	0.927
XGB-MANFA (Ours)	**0.934**

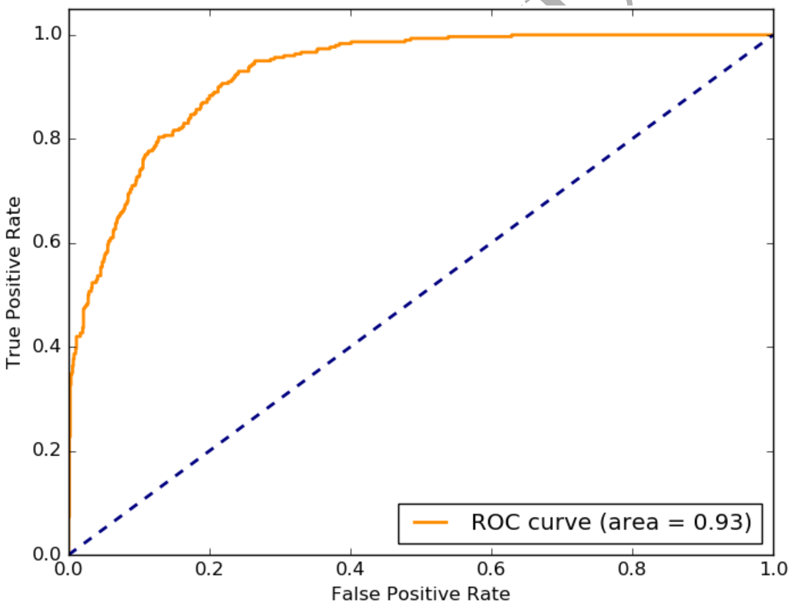


Fig. 10. AUC value of XGB-MANFA on “SwapMe and FaceSwap” dataset.

4.6. Computational complexity

The system used in this research was NVIDIA DIGITS toolbox. All experiments were implemented on a Linux machine with a pre-installed Ubuntu 14.04; it used Intel® Core i7-5930K processor, four 3,072 Cuda cores, four Titan X 12GB GPUs, and 64GB of DDR4 RAM.

To decide which model requires lowest and which model requires highest computational complexity, we first compare the training and validation time among VGG-Face, CW-VGGFace, ADA-VGGFace, XGB-VGGFace, MANFA, CW-MANFA, ADA-MANFA, and XGB-MANFA on MANFA dataset with the balancing ratio of 1:10 (Section 4.4). Results in Fig. 11 show that VGGFace and CW-VGGFace requires approximately 45 minutes whereas ADA-VGGFace and XGB-VGGFace models require 59 minutes and 52 minutes, respectively. On the other hand, with a simpler architecture, MANFA and CW-MANFA takes the least time at 20 minutes, while ADA-MANFA and XGB-MANFA need a longer computational time at 33 minutes and 27 minutes, respectively. Hybrid models are expected to have longer processing time because manipulated facial features must be extracted from the MANFA or VGGFace model, and then these features are fed to AdaBoost and XGBoost classifier so that it will take more computational time. However, as shown in previous experiments, it is an acceptable tradeoff because the model's performance was improved significantly.

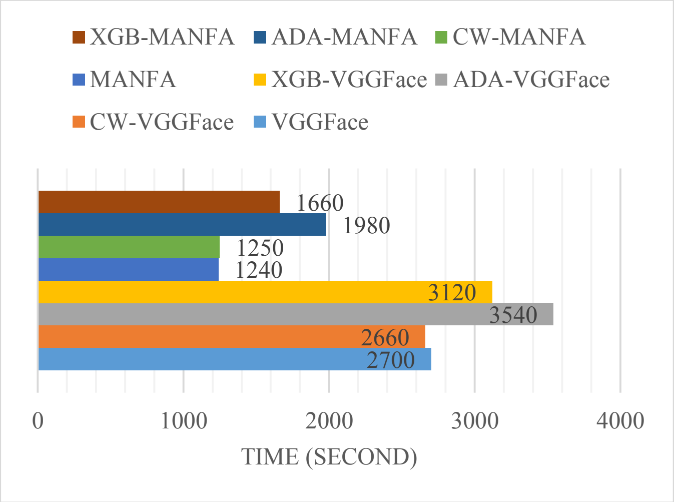


Fig. 11. Computational complexity of different models on MANFA dataset.

Table 8 shows multiple variables that can be used to check the model's computational complexity. Test time/batch indicates the time required to run a batch during the testing phase. The number of parameters shows the total number of trainable parameters for each model, and multiply-and-accumulate (MAC) that represents both multiply and addition functions. As observed from Table 8, VGG-Face and CW-VGGFace models which have the highest number of convolutional layers requires 8 seconds to perform one testing batch, and it has 27 million trainable parameters and 15.5 billion MAC operations. ADA-VGGFace and XGB-VGGFace require longer test time per patch at 13 seconds and 11 second, respectively.

On the contrary, MANFA and CW-MANFA model contain five convolutional layers with a total of 11 thousand trainable parameters and 1.4 billion MAC operations, so testing time is faster at 1 second per batch. Finally, HF-MANFA models (including ADA-MANFA and XGB-MANFA) has the same number of trainable parameters and MAC operations as MANFA model, because only the softmax classifier was replaced. However, ADA-MANFA and XGB-MANFA need more testing time per batch at 3s and 2s, respectively.

Table 8. Some statistics to evaluate the computational complexity of the proposed model.

Model	Test time/batch	Number of parameters	MACs
MANFA	1s	11 thousand	1.4 billion
CW-MANFA	1s	11 thousand	1.4 billion
ADA-MANFA	3s	11 thousand	1.4 billion
XGB-MANFA	2s	11 thousand	1.4 billion
VGGFace	8s	27 million	15.5 billion
CW-VGGFace	8s	27 million	15.5 billion
ADA-VGGFace	13s	27 million	15.5 billion
XGB-VGGFace	11s	27 million	15.5 billion

5. Discussion

As discussed in the introduction section, three key research questions need to be answered based on results obtained from various experiments. The first question was about the performance of MANFA and HF-MANFA models on a balanced dataset. Through results obtained from section 4.3, we concluded that although MANFA and XGB-MANFA got high accuracy and AUC value. XGB-VGGFace was better when it was trained on balanced dataset, and achieved the state-of-the-art performance of 95% with AUC value at 0.91. The second question asked about the performance of HF-MANFA model under different configurations of the imbalanced dataset. As shown in previous section, XGB-MANFA outperformed other models and achieved a stable result with AUC value at over 0.89 even in the severe balancing factor of 1:100; this indicates that the modified version of the MANFA model with XGBoost classifier achieved a robust performance on various imbalanced dataset scenarios. Lastly, we raised a question about the performance of the proposed model compared with the state-of-the-art model, our XGB-MANFA model achieved the AUC value at 0.934, while the two-stream network achieved the AUC value of 0.927.

Through these answers, we prove that the proposed MANFA model (especially XGB-MANFA) is effective in detecting whether an image is manipulated or normal image. We also figure out that VGGFace model (especially XGB-VGGFace) achieves better performance than XGB-MANFA when the balancing ratio is in (5:10, 6:10, 7:10, 8:10, 9:10 and 10:10). However, VGGFace models require more computational power, so they need significantly more time for training and testing. We also solved the training problem that this research and many other similar research topics faced which was imbalanced dataset scenario. It has an excellent possibility to reduce labor cost in preventing the thriving of manipulated face images. As a result, it guarantees the rights and legitimacy of the press. The proposed model can detect images edited manually by a human or automatically by a computer. Therefore, it also plays a significant role in digital image security.

6. Conclusion and future work

This study proposed an expert system which was able to identify whether an image is original or has been altered. Several methods were carried out to raise the performance of the system. We collected a huge manipulated face dataset, namely MANFA, to test the proposed model performance. We also proposed a customized deep learning model that was superior in detecting altered face images, and we further revised the structure of MANFA to create XGB-MANFA which achieved the state-of-the-art performance in imbalanced dataset scenarios. With the AUC value of up to 93.4% in classification results, our proposed model surpassed the best-known result to us by approximately 6%.

Taking into consideration end-to-end feature, high performance, a flexibility of the model, and no need for specialized tools or expert knowledge. The proposed model shows several superiorities over existing expert and intelligent systems that are now usually used for the task of manipulated face image detection. Given more data and further research on good network architecture, the proposed model could eventually substitute current standard algorithms.

In the future, several related issues should be studied to improve the performance of the model. Firstly, our model focused only on extracting features from RGB color channel, and it would be better if we consider potential features when a manipulated image is exposed under other channels or environments. Secondly, images were directly fed into the deep learning model without any pre-processing; it is worth applying several pre-processing techniques, such as image whitening transformation, augmentation to increase the model performance. Thirdly, current model achieved the state-of-the-art performance on detecting manipulated face image. However, it fails to detect image generated entirely by computer (using the trending Generative Adversarial Network); it is worth considering the identification of computer-generated face image in the future. Finally, the proposed model only detected face image manipulation without localizing manipulated regions, there have been many notable works on fast object detection and localization, such as SSD, YOLOv3, that must be applied into our model; the localization module will help pinpoint the extract location of manipulated regions in the image.

Acknowledgments

This work was supported by Korea Institute of Planning and Evaluation for Technology in Food, Agriculture, Forestry and Fisheries(IPET) through Agri-Bio Industry Technology Development Program, funded by Ministry of Agriculture, Food and Rural Affairs(MAFRA) (316033-04-2-338 SB030).

AUTHORSHIP STATEMENT

Manuscript title: Face Image Manipulation Detection based on a Convolutional Neural Network

All persons who meet authorship criteria are listed as authors, and all authors certify that they have participated sufficiently in the work to take public responsibility for the content, including participation in the concept, design, analysis, writing, or revision of the manuscript. Furthermore, each author certifies that this material or similar material has not been and will not be submitted to or published in any other publication before its appearance in the *Expert Systems with Applications*.

Authorship contributions

Please indicate the specific contributions made by each author (list the authors' initials followed by their surnames, e.g., Y.L. Cheung). The name of each author must appear at least once in each of the three categories below.

Category 1

Conception and design of study: L.M. Dang;

acquisition of data: L.M. Dang, S.H. Im;

analysis and/or interpretation of data: L.M. Dang, S.I. Hassan.

Category 2

Drafting the manuscript: L.M. Dang, H.J. Moon, S.I. Hassan;

revising the manuscript critically for important intellectual content: S.H. Im, H.J. Moon.

Category 3

Approval of the version of the manuscript to be published (the names of all authors must be listed):

L.M. Dang, S.I. Hassan, S.H. Im, H.J. Moon.

References

Amerini, I., Uricchio, T., Ballan, L., & Caldelli, R. (2017). *Localization of jpeg double compression through multi-domain convolutional neural networks.* Paper presented at the Proc. of IEEE CVPR Workshop on Media Forensics.

Andreassen, C. S., Torsheim, T., & Pallesen, S. (2014). Predictors of use of social network sites at work-a specific type of cyberloafing. *Journal of Computer-Mediated Communication, 19*(4), 906-921.

Asghar, K., Habib, Z., & Hussain, M. (2017). Copy-move and splicing image forgery detection and localization techniques: a review. *Australian Journal of Forensic Sciences, 49*(3), 281-307.

Bappy, J. H., Roy-Chowdhury, A. K., Bunk, J., Nataraj, L., & Manjunath, B. (2017). *Exploiting spatial structure for localizing manipulated image regions.* Paper presented at the Proceedings of the IEEE International Conference on Computer Vision.

Barni, M., Bondi, L., Bonettini, N., Bestagini, P., Costanzo, A., Maggini, M., . . . Tubaro, S. (2017). Aligned and non-aligned double JPEG detection using convolutional neural networks. *Journal of Visual Communication and Image Representation, 49*, 153-163.

Bradley, A. P. (1997). The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern recognition, 30*(7), 1145-1159.

Chen, J., Ou, Q., Chi, Z., & Fu, H. (2017). Smile detection in the wild with deep convolutional neural networks. *Machine vision and applications, 28*(1-2), 173-183.

Chen, T., & Guestrin, C. (2016). *Xgboost: A scalable tree boosting system.* Paper presented at the Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining.

Chollet, F. (2015). Keras: Deep learning library for theano and tensorflow. *URL: https://keras. io/k, 7*(8).

Cristin, R., Ananth, J. P., & Raj, V. C. (2018). Illumination-based texture descriptor and fruitfly support vector neural network for image forgery detection in face images. *IET Image Processing*.

Dang, L. M., Hassan, S. I., Suhyeon, I., kumar Sangaiah, A., Mehmood, I., Rho, S., . . . Moon, H. (2018). UAV based wilt detection system via convolutional neural networks. *Sustainable Computing: Informatics and Systems*.

De Carvalho, T. J., Riess, C., Angelopoulou, E., Pedrini, H., & de Rezende Rocha, A. (2013). Exposing digital image forgeries by illumination color classification. *IEEE Transactions on Information Forensics and Security, 8*(7), 1182-1194.

Dong, J., Wang, W., & Tan, T. (2013). *Casia image tampering detection evaluation database.* Paper presented at the Signal and Information Processing (ChinaSIP), 2013 IEEE China Summit & International Conference on.

Farooq, S., Yousaf, M. H., & Hussain, F. (2017). A generic passive image forgery detection scheme using local binary pattern with rich models. *Computers & Electrical Engineering, 62*, 459-472.

Fernández, A., García, S., Galar, M., Prati, R. C., Krawczyk, B., & Herrera, F. (2018). Cost-Sensitive Learning *Learning from Imbalanced Data Sets* (pp. 63-78): Springer.

Ferrara, P., Bianchi, T., De Rosa, A., & Piva, A. (2012). Image forgery localization via fine-grained analysis of CFA artifacts. *IEEE Transactions on Information Forensics and Security, 7*(5), 1566-1577.

Freund, Y., & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences, 55*(1), 119-139.

Friedman, J. H. (2002). Stochastic gradient boosting. *Computational Statistics & Data Analysis, 38*(4), 367-378.

Glorot, X., Bordes, A., & Bengio, Y. (2011). *Deep sparse rectifier neural networks.* Paper presented at the Proceedings of the fourteenth international conference on artificial intelligence and statistics.

Gu, L., Kropotov, V., & Yarochkin, F. (2017). The fake news machine: How propagandists abuse the internet and manipulate the public. *pdf] Trend Micro, 81*, 1073547711.1497355570-1028938869.1495462143.

Guo, J.-M., Liu, Y.-F., & Wu, Z.-J. (2013). Duplication forgery detection using improved DAISY descriptor. *Expert Systems with Applications, 40*(2), 707-714.

He, H., & Garcia, E. A. (2008). Learning from imbalanced data. *IEEE Transactions on Knowledge & Data Engineering*(9), 1263-1284.

Holub, V., & Fridrich, J. (2015). Low-complexity features for JPEG steganalysis using undecimated DCT. *IEEE Transactions on Information Forensics and Security, 10*(2), 219-228.

Hu, Y., Manikonda, L., & Kambhampati, S. (2014). *What We Instagram: A First Analysis of Instagram Photo Content and User Types.* Paper presented at the Icwsm.

Kazemi, V., & Sullivan, J. (2014). *One millisecond face alignment with an ensemble of regression trees.* Paper presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

Khan, S. H., Hayat, M., Bennamoun, M., Sohel, F. A., & Togneri, R. (2018). Cost-sensitive learning of deep feature representations from imbalanced data. *IEEE transactions on neural networks and learning systems, 29*(8), 3573-3587.

Khoshgoftaar, T. M., Fazelpour, A., Dittman, D. J., & Napolitano, A. (2015). *Ensemble vs. data sampling: Which option is best suited to improve classification performance of imbalanced bioinformatics data?* Paper presented at the Tools with Artificial Intelligence (ICTAI), 2015 IEEE 27th International Conference on.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Lee, S., & Lee, C. (2016). Multiscale morphology based illumination normalization with enhanced local textures for face recognition. *Expert Systems with Applications, 62*, 347-357.

Lu, C., Ke, H., Zhang, G., Mei, Y., & Xu, H. (2017). An improved weighted extreme learning machine for imbalanced data classification. *Memetic Computing*, 1-8.

Moon, H., & Phillips, P. J. (2001). Computational and performance aspects of PCA-based face-recognition algorithms. *Perception, 30*(3), 303-321.

Ng, T.-T., Hsu, J., & Chang, S.-F. (2009). Columbia image splicing detection evaluation dataset: apos.

Nguyen, T. N., Thai, C. H., Nguyen-Xuan, H., & Lee, J. (2018). Geometrically nonlinear analysis of functionally graded material plates using an improved moving Kriging meshfree method based on a refined plate theory. *Composite Structures, 193*, 268-280.

Parkhi, O. M., Vedaldi, A., & Zisserman, A. (2015). *Deep face recognition.* Paper presented at the BMVC.

Peng, B., Wang, W., Dong, J., & Tan, T. (2017). Optimized 3D lighting environment estimation for image forgery detection. *IEEE Transactions on Information Forensics and Security, 12*(2), 479-494.

Phillips, P. J., Moon, H., Rizvi, S. A., & Rauss, P. J. (2000). The FERET evaluation methodology for face-recognition algorithms. *IEEE Transactions on pattern analysis and machine intelligence, 22*(10), 1090-1104.

Ren, S., Liao, B., Zhu, W., Li, Z., Liu, W., & Li, K. (2018). The Gradual Resampling Ensemble for mining imbalanced data streams with concept drift. *Neurocomputing, 286*, 150-166.

Wu, F., Jing, X.-Y., Shan, S., Zuo, W., & Yang, J.-Y. (2017). *Multiset Feature Learning for Highly Imbalanced Data Classification.* Paper presented at the AAAI.

Yao, H., Wang, S., Zhang, X., Qin, C., & Wang, J. (2017). Detecting image splicing based on noise level inconsistency. *Multimedia Tools and Applications, 76*(10), 12457-12479.

Yu, L., Zhou, R., Tang, L., & Chen, R. (2018). A DBN-based resampling SVM ensemble learning paradigm for credit classification with imbalanced data. *Applied Soft Computing, 69*, 192-202.

Zeng, H., Zhan, Y., Kang, X., & Lin, X. (2017). Image splicing localization using PCA-based noise level estimation. *Multimedia Tools and Applications, 76*(4), 4783-4799.

Zhou, P., Han, X., Morariu, V. I., & Davis, L. S. (2017). *Two-stream neural networks for tampered face detection.* Paper presented at the 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). IEEE.