EL-FDL: Improving Image Forgery Detection and Localization via Ensemble Learning

Bin Wang[1,2], Feifan Wang[1,2], Jingge Wang[3], Haonan Yan[1], Shaopeng Zhou[2,4], and Chaohao Li[2,4](✉)

[1] Hangzhou Research Institute, Xidian University, Xian, China
wangbin02@xidian.edu.cn, 22151214404@stu.xidian.edu.cn

[2] Zhejiang Key Laboratory of Artificial Intelligence of Things (AIoT) Network and Data Security, Hangzhou, China
{abnerzhou,lchao}@zju.edu.cn

[3] The University of Sydney Camperdown, Hangzhou, NSW, Australia
jwan0689@uni.sydney.edu.au

[4] Zhejiang University, Hangzhou, China

Abstract. The widespread dissemination of diverse forgery images has profoundly impacted social life. Thus, image forgery detection techniques are becoming increasingly urgent. Existing models are usually trained to detect certain types of forgery images, leading to an insufficient generalization in detecting various forgery images (e.g., copy-move, splicing, inpainting). In this paper, we conducted extensive testing on SOTA models and revealed the limitations of individual models, including 1) insufficient generalization capability and 2) high false positive rates for pristine images. To address the above issues, we propose EL-FDL, a method based on stacking ensemble learning, which enhances the detection and localization abilities by integrating the output of heterogeneous SOTA models. Extensive experimental results demonstrate that our proposed EL-FDL significantly improved: +16.4% in detection, +11.1% in localization, and overall false positive rate decreased by at least -21.0% across the test dataset.

Keywords: forgery detection · ensemble learning · image forensics · decision fusion

1 Introduction

With the proliferation of image editing tools and image generation technologies, producing forgery images has become increasingly simple yet difficult to detect. This threatens personal privacy and risks public safety and social stability. For instance, spreading misinformation on social media, engaging in identity deception, and fabricating news have become prevalent tactics in online criminal

This work is supported by China NSFC Grant nos. 92167203, 92267204

M. Wand et al. (Eds.): ICANN 2024, LNCS 15017, pp. 248–262, 2024.
https://doi.org/10.1007/978-3-031-72335-3_17

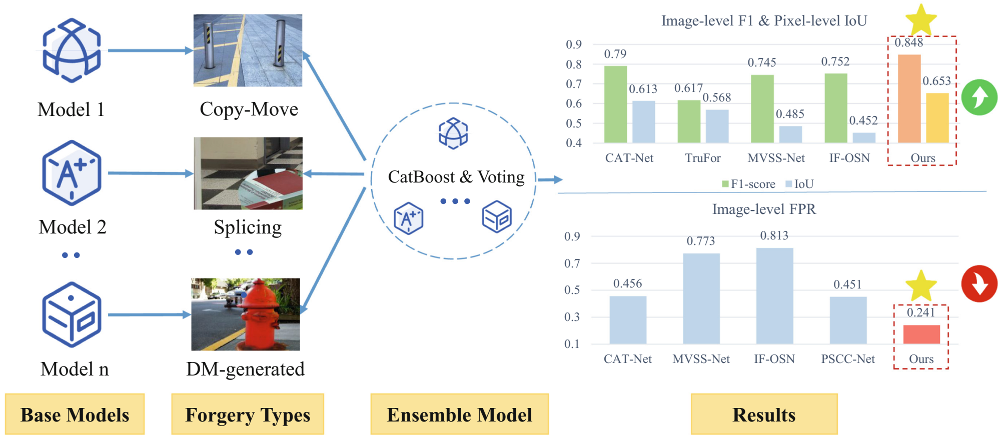

Fig. 1. An overview of EL-FDL. It uses stacking and voting ensemble learning methods to combine heterogeneous models for detecting different types of forgery images, thereby improving the detection performance while reducing the false positive rate.

activities. Moreover, the unethical utilization of manipulated images to deceive the public and undermine competitors' interests has become widespread, particularly in political elections and business competitions. Therefore, the threat of image forgery technology is fatal to various aspects of human life.

The main types of image forgery include manual tampering and artificial intelligence generation. The former includes copy-move, splicing, object-removal. The latter involves generative forgery images generated by artificial intelligence models, e.g., generative adversarial networks (hereafter referred to as GAN) and diffusion models [11,20].

Previous work has devoted much effort to addressing various forgery image detection. For instance, prior studies utilize fully convolutional networks for pixel-level semantic segmentation to detect splicing images [2]. Meanwhile, others employ SRM high-pass filters to emphasize edge features or region similarity matching for detecting copy-move forgeries [32]. For the GAN-generated images, extracting the features of the image block by deep CNN or replacing the discriminator of the GAN framework was applied in the detection [18]. Others detect DM-generated images by exploring forensic traces left by the diffusion model [4]. These methods and models show unsatisfactory performance when detecting forgery image types without pre-training.

To this end, we conducted a series of experiments on existing SOTA models and gained three insights. Firstly, the detection performance of the models varies notably depending on the type of forgery dataset. Then, the performance of the model varies even for the same type of forgery yet different datasets. In addition, we also find that the models exhibit high false positive rates on pristine images, which is not practical in real scenarios and increases maintenance costs. In summary, it can be described as 1) insufficient generalization, 2) limited robustness, and 3) high false positive rates.

We propose a practical framework for applying Ensemble Learning to image Forgery Detection and Localization (EL-FDL) to tackle the above issues, as shown in Fig. 1. EL-FDL is based on a stacking ensemble, which fuses the heterogeneous models via CatBoost [22] and Voting. For example, Model 1 excels in detecting copy-move, while Model 2 specializes in splicing, ..., and Model N is expert at detecting DM-generated. First, we process the original datasets using these base models and obtain their outputs. Then, we use Catboost to train the meta-model of the detection module, while using voting to perform decision fusion of the localization module. Finally, as shown in Fig. 1, we obtain image-level detection and pixel-level localization results with higher performance and lower false positive rates.

Our experimental results show that EL-FDL has notable improvements over the SOTA models. On these test datasets, CASIA, Columbia, Coverage, Defacto-inpainting, and CocoGlide, EL-FDL gained average improvements: 1) +18.1% and +14.8% in image-level ACC and F1-score, while reducing the FPR at least -21.0%, and 2) +9.3% and +12.9% in pixel-level F1-score and IoU. Our contributions are as follows:

1. We proposed an ensemble learning-based framework, integrating various heterogeneous detection models via Catboost and Voting.
2. We conducted extensive experiments that reveal the limitations of existing models, including insufficient generalization and high false positive rates.
3. The proposed EL-FDL presents better performances than the existing SOTA method over the test datasets with average gains being +16.4% in detection, +11.1% in localization, and -21.0% reduction in overall FPR.

2 Related Work

2.1 Forensic Detection and Localization

Traditional forgery detection techniques show poor performance with the development of manually tampered images towards deep forgery images. Therefore, to better address the challenges posed by forgery images, more effective deep-learning methods have been continuously proposed and widely applied.

Detection. Deep learning methods are usually analyzed by extracting various high-dimensional features (e.g., noise, frequency domain, pixels) in the forgery images. Chengbo Dong et al.(2022) develop multi-scale supervised and noisy views and explicitly extract boundary artifacts to learn the operational detection function [5]. In addition, to improve the robustness of image forgery detectors, Haiwei Wu et al.(2022) decompose noise on social networks into two parts [31], predictable noise and invisible noise, and model them separately then increase the accuracy of forgery detection. To further improve the forgery detection confidence, Guillaro et al. (2023) extracted traces in the image using learned noise-sensitive fingerprints, and introduced confidence map and integrity score mechanisms for image forgery detection [12].

Localization. Most SOTA model deep-learnings of forgery localization are inspired by semantic segmentation [12], but there are also unsupervised classification methods using contrastive learning. Xiuli Bi et al.(2019) make the features of image tampering regions more visible through residual propagation and feedback processes in CNN [1]. Kwon et al.(2022) deal with forgery localization by considering spliced objects of different shapes and sizes [14]. It uses dual JPEG detection of pre-trained DCT streams and JPEG artifacts to locate tampered regions. On the other hand, Haiwei Wu et al.(2023) use pixel-level contrastive learning to extract high-level features of an image [30]. The ReLoc framework [33] uses an image recovery module to identify and locate corrupted regions and further restore distorted tampered images.

Despite the good performance of these models, the challenges and limitations of detecting various forgeries remain. Therefore, this paper introduces ensemble learning to cope with the challenge.

2.2 Ensemble Learning

Ensemble learning is an effective method to improve model performance by combining various individual models [10]. Currently, ensemble learning has been employed in a wide range of applications (e.g., healthcare, speech systems, intrusion detection, image recognition) [23,25–27]. Ensemble learning algorithms are at the heart of ensemble models and have been continuously developed. It can be broadly categorized into bagging, boosting, and stacking.

Boosting is an approach to convert weak models into a model with better generalization [10]. Its optimized algorithms are now widely used in various ensemble applications. AdaBoost [8] and GBDT [9] stand out as two mainly recognized boosting algorithms. Improved algorithms have been proposed by scholars in recent years, such as XGBoost [3], LightGBM [19], CatBoost [22], and NgBoost [7]. Stacking employs meta-learning techniques to improve model performance by fusing the outputs of heterogeneous models. Rajani et al.(2016) [24] introduced auxiliary features, confidence scores, and provenance, enhancing model accuracy.

Therefore, to address the challenge that a single model cannot detect various forgery images, we adopt a stacking-based ensemble approach combined with boosting to integrate multiple heterogeneous models for forgery image detection and localization.

3 Motivation

3.1 Design Goal

The design goal of EL-FDL is to **promote forgery detection performance while reducing the false positive rates**. Selecting the appropriate models and algorithms for the ensemble is the core of our work. Therefore, we have taken the following two paths to achieve our goals:

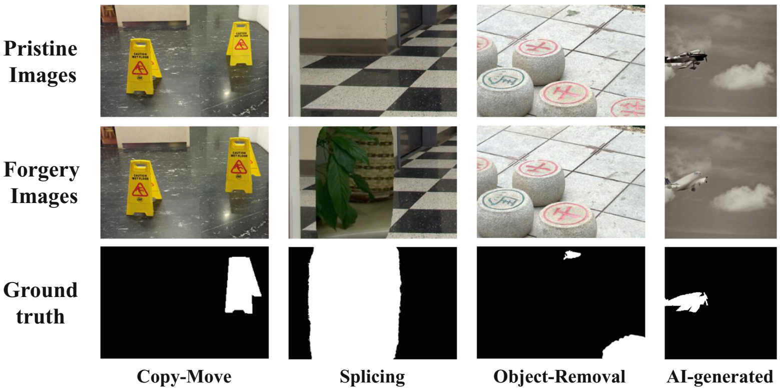

Fig. 2. Examples of different types of forgery images.

1. Select models: We experimented with existing models on various forgery datasets and chose the appropriate models for fusion by analyzing and comparing the performance of the models.
2. Select algorithms: We tried to apply various algorithms to the stacking ensemble and find apt algorithms to enable the ensemble model to have better detection performance in the face of multiple types of forgery images.

3.2 Threat Model

The attacker aims to generate forgery images(e.g., generate fake faces by GAN) for profit. Figure 2 shows examples of forgery types, and the definitions are also given below.

1. Copy-move forgery is a type of manipulation where a portion of an image is copied and pasted into the same image to form a new segment, which blends in with the surrounding area to create the illusion of a more complex scene.
2. Splicing forgery involves taking two or more images and merging them, to produce a realistic composite image.
3. Object-removal involves removing an object or region from an image and then replacing it with new content that is generally consistent with the surrounding area, intending to remove any information that could reveal the existence of the removed object or region.
4. AI-generated employs generative models such as GAN and diffusion models to create realistic but synthetic images for deceptive purposes.

Attackers often use one or more forgery image techniques for benefit fraud, and traditional detection models make it difficult to detect multiple forgeries simultaneously. Therefore, to address this challenge, we proposed EL-FDL, which aimed to detect various forgery types by an ensemble model.

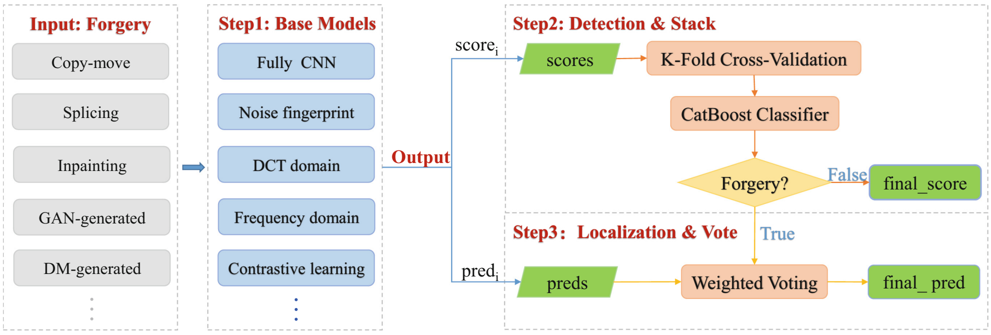

Fig. 3. The framework of EL-FDL. A variety of forgery images were first input into the base models. Base models will output the detection and localization results, namely $score_i$ and $pred_i$. The $scores$ will be first input into the detection module and given the final detection result $final\_score$ by the CatBoost classifier. Then, depending on the detection result, $preds$ will be input into the localization module and given the final localization result $final\_pred$ by voting.

4 Method

To tackle the challenge of detecting various types of forgery, we proposed an ensemble learning-based image forgery detection framework, EL-FDL.

This Section presents an overview of EL-FDL, illustrated in Fig. 3. Subsequent subsections will provide the details of each component. Initially, forgery images, are inputted. N base models, denoted as $\{L_1, L_2, \ldots, L_N\}$, are employed to detect forgery images. Each base model produces corresponding detection results, represented as $score_i$ or $pred_i$, where $score_i$ represents the forgery detection score output by base model L_i in the range $[0, 1]$. A score closer to 1 indicates a higher likelihood of forgery, while a score closer to 0 suggests a more pristine image. The $pred_i$ represents the predicted mask for locating forged regions, where 0 denotes real pixels and 1 denotes forged pixels. Next, the scores from all base models are concatenated into a matrix, denoted as $scores[N, 1]$, and fed into the image forgery detection module. This module outputs the final image forgery detection result in the range [0, 1]. If the result indicates pristine, the final detection score, denoted as $final\_score$, is outputted. If the result suggests forgery, the $pred_i$ values are concatenated into a matrix $preds[N, H, W]$, where each $pred_i$ has been pre-normalized to a size of 1024×1024. Finally, $preds[N, H, W]$ is inputted into the image forgery localization module, generating the final result, $final\_score$ for detection and $final\_pred$ for localization.

4.1 Base Models Layer

While single classifiers often exhibit excellent performance, they frequently suffer from drawbacks such as overfitting and insufficient generalization. In this module, we have aggregated various methods for simple and deep fake image detection and localization, aiming to empower ensemble learning models to cover an adequate range of image tampering types. We have standardized the input and output interfaces of all models, allowing for the detection of an image by simply inputting the detection image. Depending on the output type of each model, some models output only the confidence score, some only the localization mask, and some output both. These models are then concatenated in the order of their inputs, ultimately outputting $scores[N, 1]$ and $preds[N, H, W]$ for subsequent forgery detection and localization modules.

4.2 Forgery Detection Module

After the detection of the base model layer, we get the input $scores[N, 1]$ of the image forgery detection module, which represents the image-level forgery detection results of image x by N base models. It is only necessary to input the $scores[N, 1]$ into the meta-model, which the ensemble learning algorithm trains, to get the final forgery detection results.

Ensemble Method. The weak classifiers generated by bagging and boosting are isomorphic since the model structure varies across detection base models. Therefore, we cannot train the meta-model directly with the original dataset by bagging or boosting. Instead, we need to use the classification results of the base models as the training set to train the meta-model, which is called stacking. The integration by stacking does not affect the detection performance of the base models and produces a more powerful image detection meta-model.

Classification Algorithm. The choice of machine learning algorithms greatly impacts the generalization performance of Stacking. We are concerned that the number of features in the feature matrix of the stacking constructed meta-model is equal to the number of base models. However, there are usually only a few base models, so the feature matrix of the meta-model is often insufficient. When the meta-model is trained directly using traditional machine learning algorithms, the problem of overfitting often occurs.

To solve this problem, we introduce K-Fold Cross-Validation [29] to expand the feature matrix of the meta-model when training the model to improve its generalization performance. At the same time, we utilize the CatBoost [22] algorithm based on the GBDT [9] framework with ordered boosting and prediction shift mechanism and thus solve the problem of overfitting. Algorithm 1 presents workflow details in our ensemble framework.

4.3 Forgery Localization Module

Since the generalization of a single model is limited, we introduce a voting mechanism for localization ensemble. Traditional voting follows the majority rule to reduce variance but ignores the differences in model performance, so we introduce weight parameters to improve the voting method.

Algorithm 1. CatBoost application in EL-FDL

```
Input: Detection results and the labels$\{(\mathbf{x}_i, y_i)\}_{i=1}^n, I, \alpha, L, m, Mode$;
$\theta_t \leftarrow$ random permutation of $[1, n]$ for $t = 0..m$;
$M_0(i) \leftarrow 0$ for $i = 1..n$;
if $Mode = Plain$ then
    $M_t(i) \leftarrow 0$ for $t = 1..m, i : \theta_t(i) \leq 2^{j+1}$;
end if
if $Mode = Ordered$ then
    for each $j \in [1, \lceil \log_2 n \rceil]$ do
        $M_{t,j}(i) \leftarrow 0$ for $t = 1..m, i = 1..2^{j+1}$;
    end for
end if
for each $k \in [1, I]$ do
    $T_k, \{M_t\}_{t=1}^m \leftarrow BuildTree(\{M_t\}_{t=1}^m, \{(x_i, y_i)\}_{i=1}^n, \alpha, L, \{\theta_i\}_{i=1}^m, Mode)$;
    $leaf_0(i) \leftarrow GetLeaf(\mathbf{x}_i, T_k, \theta_0)$ for $i = 1..n$;
    $grad_0 \leftarrow CalcGradient(L, M_0, y)$;
    for each $leaf\ j\ in\ T_k$ do
        $b_j^k \leftarrow -avg(grad_0(i)$ for $i : leaf_0(i) = j)$;
    end for
    $M_0(i) \leftarrow M_0(i) + \alpha b_{leaf_0(i)}^k$ for $i = 1..n$;
end for
return $F(x) = \sum_{k=1}^I \sum_j \alpha b_j^k \mathbb{1}_{\{GetLeaf(\mathbf{x}, T_k, ApplyMode) = j\}}$;
```

The input of the localization module is denoted as $pred[N, H, W]$. For an image x, each pixel point is characterized by N channels, where the number of base models determines the value of N. Meanwhile, each base model is assigned a weight θ based on its performance in weighted voting. Usually, better-performing models receive a higher weight. Finally, we obtain the final prediction $final\_pred$ by calculating the weighted average of θ and $pred_i$.

$$final\_pred = \frac{1}{\sum_{i=1}^N \theta_i} \sum_{i=1}^N \theta_i \times pred_i \tag{1}$$

5 Experiment

5.1 Experiment Setting

Unless otherwise noted, we used the same experimental setup and equipment to test the performance of the EL-FDL.

Training Datasets. We train the forgery detection model of EL-FDL using the 10000 forgery images of datasets tampCOCO [15], and the 2000 pristine images of IMD2020 [21]. We expanded IMD2020 to 10,000 images to ensure a balanced distribution of positive and negative samples.

Testing Datasets. We selected three commonly used public forgery image datasets, namely CASIA [6], Columbia [13], and Coverage [28]. These datasets contain copy-move and splicing, totaling 1203 forgery and 283 pristine images. Additionally, we included Defacto-inpainting including 20000 Object-Removals images from the DEFACTO [17], and CocoGlide [12] generated by diffusion models with 512 forgery and 512 pristine images.

Evaluation Metrics. We adopt the following metrics throughout the evaluation. **Accuracy(ACC)**: It characterizes the rate at which the classifier correctly classifies all samples, including positive and negative examples. **F1-score**: It is a metric that considers both precision and recall and provides a weighted harmonic mean of both measures to evaluate the accuracy of imbalanced datasets. **False Positive Rate (FPR)**: It characterizes the rate at which the classifier incorrectly classifies negative samples as positive. **Intersection over Union (IoU)**: It is a metric used in object detection that measures the overlap between the predicted and the ground-truth bounding boxes. It is calculated by dividing the intersection area between the two boxes by the union of both areas.

We utilize ACC, F1-score, and FPR to evaluate the image-level forgery detection ability. Meanwhile, employing F1-score and IoU to evaluate the pixel-level localization ability, with a preference for lower FPR indicating better performance and higher scores in other metrics being favorable. All metrics default to a threshold of 0.5 for consistency.

Implementation Details. We primarily utilized PyTorch deep learning framework to ensemble all methods, and the scikit-learn machine learning framework for training EL-FDL. 2 Xeon Gold 6226R(16C/2.90Ghz) and 2 NVIDIA RTX 3090 24GB were used for all experiments.

5.2 Experiment Results

We conducted extensive ensemble learning experiments for image-level forgery detection and pixel-level localization. After preliminary models and algorithms testing, we finally selected the following model for ensemble [5,12,14,16,30,31], and used CatBoost as the main ensemble algorithm. Detailed analysis of the experimental results will be provided in the subsequent sections.

Detection. Table 1 lists the ACC and F1-score for image-level forgery detection on the test dataset. We can observe that a single model might excel on one dataset, as seen in Trufor on Columbia, and PSCC-Net on CocoGide. However, their performance may need improvement on other datasets. For example, TruFor achieves only 11.8% and 20.8% and PSCC-Net only 1.6% and 2.7% on Defacto. In addition, CAT-Net is good at detecting splicing tampering types, so it performs well in CASIA, and Columbia datasets containing splicing. All the above results indicate the limitations of the generalization of a single model. In

Table 1. Image-level Accuracy and F1-score perfomance of image forgery detection. The best results are in **bold** and the second best results are in underlined.

Method	CASIA		Coverage		Columbia		Defacto		CocoGlide		Avg	
	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1
CAT-Net [14]	0.908	0.951	0.635	0.640	0.802	0.831	**0.964**	**0.981**	0.520	0.548	0.766	0.790
TruFor [12]	0.680	0.810	0.690	0.581	**0.986**	**0.986**	0.118	0.208	0.644	0.498	0.624	0.617
MVSS-Net [5]	0.615	0.762	0.550	0.681	0.664	0.747	0.790	0.883	0.539	**0.654**	0.632	0.745
IF-OSN [31]	0.758	0.862	0.510	0.657	0.526	0.677	0.888	0.941	0.567	0.622	0.650	0.752
PSCC-Net [16]	0.372	0.541	0.550	0.297	0.504	0.667	0.016	0.027	**0.660**	0.528	0.420	0.412
EL-FDL(ours)	**0.961**	**0.980**	**0.975**	**0.975**	0.690	0.699	0.930	0.964	0.621	0.620	**0.835**	**0.848**

Table 2. Image-level forgery detection on some representative test images. ✓ represents a correct judgment, and ✗ represents an incorrect judgment.

Forgery Type	Image	Ground-truth	CAT-Net	TruFor	MVSS-Net	IF-OSN	PSCC-Net	EL-FDL(ours)
Splicing	Fig b	Fig b	✗	✓	✓	✓	✗	✓
Copy-Move	Fig b	Fig b	✓	✗	✗	✓	✓	✓
Inpainting	Fig b	Fig b	✗	✗	✓	✓	✗	✓
DM-generated	Fig b	Fig b	✓	✗	✓	✗	✗	✓

contrast, EL-FDL performs better on various datasets in different and same-type datasets. It outperforms the best-performing CAT-Net on average, improving ACC and F1 scores by +6.9% and +5.8%, respectively. Table 2 gives typical cases that cannot be fully detected by a single model but can be done by EL-FDL, which shows a better generalization of EL-FDL.

FPR (False Positives Rate). False positive rate is often focused on in statistics but is easily overlooked in machine learning due to the focus on accuracy. The false positive rate is significant in practical applications of forgery image detection, and an increase in the labor cost accompanies an increase in the false positive rate. From table reftab:image fpr, we observe that CAT-Net, MVSS-Net, and IF-OSN, which show high performance in the table (reftab:image ACC and F1), but have high FPR of 45.6%, 77.3% and 81.3%, respectively. Meanwhile, as can be seen from the data with * in the table, TruFor has a lower FPR of 11.7%, which is because its confidence map and integrity score mechanism reduce misclassification of the pristine image. However, the mechanisms also result in many forgery images not being successfully detected, so its accuracy remains low despite the low FPR. In contrast, the average FPR of EL-FDL is only 24.1%, which is much lower than all models except TruFor. Furthermore, we notice that the FPR of different models on different datasets varies(e.g., CAT-Net and MVSS-Net have relatively low FPR on the Columbia dataset, while PSCC-Net has relatively low FPR on Coverage and CocoGlide). Therefore, EL-FDL fuses the decision results of multiple models, tends to balance the performance of each model, and reduces the problem of a high positivity rate caused by a single model (Table 3).

Table 3. Image-level False Positives Rate performance in forgery detection.

Method	Columbia	Coverage	CocoGlide	Avg
CAT-Net [14]	0.377	0.380	0.611	0.456
TruFor [12]	0.036 *	0.150 *	0.166 *	0.117 *
MVSS-Net [5]	0.667	0.860	0.793	0.773
IF-OSN [31]	0.940	0.920	0.578	0.813
PSCC-Net [16]	0.984	**0.190**	**0.180**	0.451
EL-FDL	**0.005**	0.340	0.377	**0.241**

Table 4. Pixel-level F1-score and IoU performance of image forgery localization.

Method	CASIA		Coverage		Columbia		Defacto		CocoGlide		Avg	
	F1	IoU	F1	IoU	F1	IoU	F1	IoU	F1	IoU	F1	IoU
CAT-Net[14]	0.844	0.636	0.607	0.459	**0.852**	0.824	**0.702**	**0.585**	0.581	0.562	0.717	0.613
TruFor[12]	0.833	0.623	0.640	0.712	0.641	0.799	0.533	0.073	0.655	0.635	0.660	0.568
MVSS-Net[5]	0.707	0.397	0.628	0.667	0.695	0.744	0.495	0.010	0.597	0.609	0.624	0.485
IF-OSN[31]	0.741	0.465	0.526	0.504	0.606	0.708	0.484	0.010	0.624	0.575	0.596	0.452
PSCC-Net[16]	0.715	0.408	0.616	0.610	0.311	0.283	0.495	0.024	0.656	0.640	0.559	0.393
FOCAL[30]	**0.859**	**0.706**	0.670	**0.799**	0.690	0.842	0.622	0.210	0.533	0.595	0.675	0.630
EL-FDL (Hard)	0.802	0.578	**0.750**	0.713	0.754	0.872	0.548	0.088	**0.711**	0.619	0.713	0.574
EL-FDL (Weighted)	0.855	0.673	0.718	0.747	0.729	**0.879**	0.686	0.322	0.672	**0.643**	**0.732**	**0.653**

Localization. We conducted ensemble experiments for forgery localization using hard and weighted voting methods. Table 4 illustrates the pixel-level F1-score and IoU. In terms of average scores, Hard Voting has significantly outperformed all models except CAT-Net with an F1-score of 71.3%, and the IoU of 57.4% also shows improvement compared to other models such as TruFor, MVSS-Net, and IF-OSN. However, it is noteworthy that weaker models can still influence the hard voting method on specific datasets, such as Defacto, where most models underperform. In contrast, weighted voting allows us to make nuanced adjustments to models' weights, appropriately increasing the voting weight of outstanding models. Ultimately, the model outperformed all single models with 73.2% of F1 and 65.3% of IoU.

5.3 Ablation Study

Classification Algorithmic Comparison. To validate the machine learning classification algorithm chosen for the stacked ensemble model, we compared the commonly used classification algorithms KNN, DT, LogiR, and RF; all models were trained using default parameters. Table 5 indicates the results. On average scores, GBDT outperforms most other models with an ACC of 81.1% and an F1-score of 82.9%, while having the lowest FPR of 14.6%. Its ACC and F1-score

Table 5. Classification algorithmic comparison ablation results: Image-level ACC, F1-score, and FPR. The max ACC and F1 are in **bold** and the min FPR are underlined.

Alg.	CASIA		Columbia			Coverage			Defacto		CocoGlide			Avg		
	ACC	F1	ACC	FPR	F1	ACC	FPR	F1	ACC	F1	ACC	FPR	F1	ACC	FPR	F1
RF	0.878	0.935	**0.981**	0.025	**0.980**	0.680	0.190	0.584	0.688	0.815	**0.636**	0.339	0.533	0.773	0.185	0.769
DT	0.904	0.950	0.939	0.039	0.938	0.608	0.290	0.567	0.758	0.862	0.556	0.438	0.553	0.753	0.256	0.774
KNN	0.833	0.909	0.939	0.016	0.936	0.615	0.165	0.503	**0.864**	**0.927**	0.562	0.299	0.492	0.763	0.158	0.753
LogiR	**0.940**	**0.970**	0.970	0.016	0.959	**0.685**	0.370	**0.671**	0.830	0.907	0.589	0.439	0.584	**0.803**	0.275	**0.818**
GBDT	0.911	0.953	0.961	0.011	0.960	0.660	0.160	0.585	0.778	0.875	0.620	0.195	**0.643**	0.787	0.122	0.803

are 4.8% and 7.6% higher than the commonly used KNN algorithm, respectively. Although the ACC and F1-score of GBDT are not as good as those of LogiR, its FPR is much lower than that of LogiR. Therefore, given its performance metrics, it proves that our chosen GBDT algorithm has advantages for the training of the EL-FDL.

GBDT Optimisation. Optimization of the GBDT algorithm was attempted as the model is prone to overfitting due to the high learning capacity of GBDT. We finally used the GBDT-optimised Catboost [22] and compared it with several other GBDT-optimised algorithms, ngboost [7], lightgbm [19] and xgboost [3]. Table 6 shows the performance of each optimization algorithm on the test dataset. Compared to GBDT, XgBoost, LightGBM, and CatBoost, all have improved detection performance, but the false positive rates have also increased. Among these algorithms, CatBoost has the largest improvement in ACC and F1-score of 5.2% and 4.5% respectively, and the smallest increase in FPR of 11.9%. On the contrary, NgBoost shows a decrease of about 10% in detection and FPR compared to GBDT. It is demonstrated through experimental results that CatBoost has the greatest performance improvement in detection enhancement and false positive suppression. Meanwhile, NgBoost can significantly reduce the FPR, but a severe decline in detection capability accompanies it.

Table 6. GBDT optimization ablation results: Image-level ACC, F1-score and FPR.

Alg.	CASIA		Columbia			Coverage			Defacto		CocoGlide			Avg		
	ACC	F1	ACC	FPR	F1	ACC	FPR	F1	ACC	F1	ACC	FPR	F1	ACC	FPR	F1
GBDT	0.911	0.953	0.961	0.011	0.960	0.660	0.160	0.585	0.778	0.875	0.620	0.195	**0.643**	0.787	0.122	0.803
XgBoost	0.965	0.982	0.972	0.010	0.972	0.680	0.340	0.686	0.840	0.913	0.613	0.406	0.621	0.814	0.252	0.835
LightGBM	**0.973**	**0.986**	0.959	0.038	0.958	**0.695**	0.350	**0.708**	0.842	0.914	0.608	0.443	0.628	0.815	0.277	0.839
NgBoost	0.826	0.905	0.915	0.005	0.906	0.650	0.030	0.485	0.348	0.515	**0.625**	0.062	0.455	0.673	0.032	0.653
CatBoost	0.961	0.980	**0.975**	0.005	**0.975**	0.690	0.340	0.699	**0.930**	**0.964**	0.621	0.377	0.620	**0.835**	0.241	**0.848**

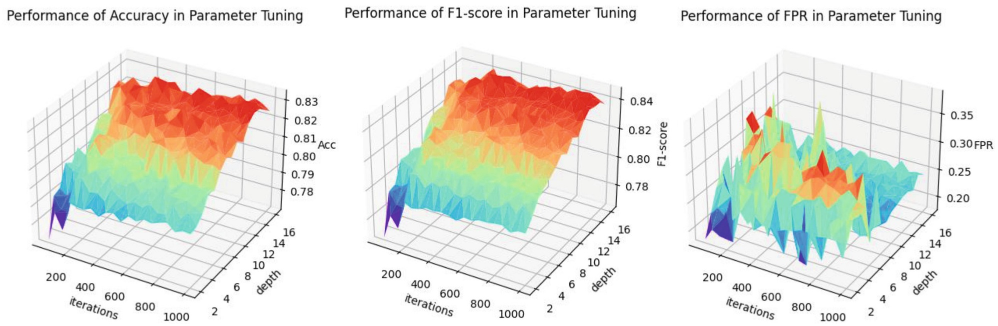

Fig. 4. Ablation results for fine-tuned *iteration* and *depth*: ACC, F1-score and FPR.

Parameter Finetune. The model parameters have a decisive impact on the model's performance. Therefore, we employed the Grid Search algorithm for optimal model parameters. CatBoost has many default parameters, such as $iterations$, $grow_{policy}$, $depth$, $min\_data\_in\_leaf$,max_leaves, etc. Through a series of experiments, we found that $iterations$ and $depth$ parameter values of the CatBoost model directly impacted the model's performance compared to other parameters. Specifically, we tried $iterationss = [50, 1001, 50]$ and $depth = [2, 17, 1]$. For each parameter set, we used 5-fold Cross-Validation to finetune the model. Figure 4 shows the tuning results. From the formula, $W = ACC + F1 - \frac{1}{2} \times FPR$, we determine that the model's performance was relatively optimal when $itrations = 300$ and $depth = 14$. In this case, the model's mean ACC and F1-score reached 83.5% and 84.8%, respectively, while the mean false positive rate was 24.1%.

6 Conclusion

In this paper, we propose EL-FDL, an efficient ensemble framework for image forgery detection and localization. We conducted extensive experiments that reveal the limitations of the existing SOTA methods, including insufficient generalization and high false positive rates. To tackle these limitations, we introduced stacking and Catboost for forgery detection and weighted voting for region localization. Our experimental results show that our ensemble model can better handle various types of forgery images compared to existing SOTA models, mainly in terms of +21.6 and +18.4%+ in average image-level ACC and F1-score, +9.3% and 12.9% in average pixel-level F1-score and IoU, and −21% decrease in FPR.

References

1. Bi, X., Wei, Y., Xiao, B., Li, W.: RRU-Net: the ringed residual u-net for image splicing forgery detection. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) (2019)
2. Chen, B., Qi, X., Wang, Y., Zheng, Y., Shim, H.J., Shi, Y.Q.: An improved splicing localization method by fully convolutional networks. IEEE Access **6**, 69472–69480 (2018)
3. 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 (2016). https://api.semanticscholar.org/CorpusID:4650265
4. Corvi, R., Cozzolino, D., Zingarini, G., Poggi, G., Nagano, K., Verdoliva, L.: On the detection of synthetic images generated by diffusion models. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1–5 (2023). https://doi.org/10.1109/ICASSP49357.2023.10095167
5. Dong, C., Chen, X., Hu, R., Cao, J., Li, X.: MVSS-Net: multi-view multi-scale supervised networks for image manipulation detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 1–14 (2022). https://doi.org/10.1109/TPAMI.2022.3180556
6. Dong, J., Wang, W., Tan, T.: CASIA image tampering detection evaluation database. In: IEEE China Summit International Conference on Signal and Information Processing, pp. 422–426. IEEE (2013)
7. Duan, T., Anand, A., Ding, D.Y., Thai, K.K., Basu, S., Ng, A., Schuler, A.: NGBoost: natural gradient boosting for probabilistic prediction. In: International Conference on Machine Learning, pp. 2690–2700. PMLR (2020)
8. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. J. Comput. Syst. Sci. **55**(1), 119–139 (1997)
9. Friedman, J.H.: Greedy function approximation: a gradient boosting machine. Ann. Stat. **29**, 1189–1232 (2001)
10. Ganaie, M.A., Hu, M., Malik, A.K., Tanveer, M., Suganthan, P.N.: Ensemble deep learning: a review. Eng. Appl. Artif. Intell. **115**, 105151 (2022)
11. Goodfellow, I., et al.: Generative adversarial nets. In: Neural Information Processing Systems (2014)
12. Guillaro, F., Cozzolino, D., Sud, A., Dufour, N., Verdoliva, L.: TruFor: leveraging all-round clues for trustworthy image forgery detection and localization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 20606–20615 (2023)
13. Hsu, Y., Chang, S.: Detecting image splicing using geometry invariants and camera characteristics consistency. In: IEEE International Conference on Multimedia and Expo, pp. 549–552. IEEE (2006)
14. Kwon, M.J., Nam, S.H., Yu, I.J., Lee, H.K., Kim, C.: Learning JPEG compression artifacts for image manipulation detection and localization. Int. J. Comput. Vision **130**(8), 1875–1895 (2022). https://doi.org/10.1007/s11263-022-01617-5
15. Lin, T.-Y., et al.: Microsoft COCO: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 740–755. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_48
16. Liu, X., Liu, Y., Chen, J., Liu, X.: PSCC-Net: progressive spatio-channel correlation network for image manipulation detection and localization. IEEE Trans. Circ. Syst. Video Technol. **32**, 7505–7517(2022)

17. Mahfoudi, G., Tajini, B., Retraint, F., Morain-nicolier, F., Dugelay, J.L., Pic, M.: DEFACTO: image and face manipulation dataset. In: 2019 27th European Signal Processing Conference (EUSIPCO), pp. 1–5 (2019). https://doi.org/10.23919/EUSIPCO.2019.8903181
18. Marra, F., Gragnaniello, D., Cozzolino, D., Verdoliva, L.: Detection of GAN-generated fake images over social networks. In: 2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR) (2018)
19. Meng, Q.: LightGBM: a highly efficient gradient boosting decision tree. In: Neural Information Processing Systems (2017)
20. Nichol, A., et al.: GLIDE: towards photorealistic image generation and editing with text-guided diffusion models. arXiv preprint arXiv:2112.10741 (2021)
21. Novozámský, A., Mahdian, B., Saic, S.: Imd2020: A large-scale annotated dataset tailored for detecting manipulated images. In: 2020 IEEE Winter Applications of Computer Vision Workshops (WACVW), pp. 71–80 (2020). https://doi.org/10.1109/WACVW50321.2020.9096940
22. Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A.V., Gulin, A.: CatBoost: unbiased boosting with categorical features. In: Advances in Neural Information Processing Systems, vol. 31 (2018)
23. Rai, H.M., Chatterjee, K.: Hybrid CNN-LSTM deep learning model and ensemble technique for automatic detection of myocardial infarction using big ECG data. Appl. Intell. **52**(5), 5366–5384 (2022)
24. Rajani, N.F., Mooney, R.J.: Stacking with auxiliary features. arXiv preprint arXiv:1605.08764 (2016)
25. Tama, B.A., Lim, S.: Ensemble learning for intrusion detection systems: a systematic mapping study and cross-benchmark evaluation. Comput. Sci. Rev. **39**, 100357 (2021)
26. Tanveer, M., Rashid, A.H., Ganaie, M., Reza, M., Razzak, I., Hua, K.L.: Classification of Alzheimer's disease using ensemble of deep neural networks trained through transfer learning. IEEE J. Biomed. Health Inform. **26**(4), 1453–1463 (2021)
27. Wang, B., Xue, B., Zhang, M.: Particle swarm optimisation for evolving deep neural networks for image classification by evolving and stacking transferable blocks. In: 2020 IEEE Congress on Evolutionary Computation (CEC), pp. 1–8. IEEE (2020)
28. Wen, B., Zhu, Y., Subramanian, R., Ng, T.T., Winkler, S.: Coverage – a novel database for copy-move forgery detection. In: IEEE International Conference on Image Processing (2016)
29. Wong, T.-T.: Performance evaluation of classification algorithms by k-fold and leave-one-out cross validation. Pattern Recogn. **48**(9), 2839–2846 (2015)
30. Wu, H., Chen, Y., Zhou, J.: Rethinking image forgery detection via contrastive learning and unsupervised clustering. arXiv preprint arXiv:2308.09307 (2023)
31. Wu, H., Zhou, J., Tian, J., Liu, J.: Robust image forgery detection over online social network shared images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 13440–13449 (2022)
32. Wu, Y., Abd-Almageed, W., Natarajan, P.: BusterNet: detecting copy-move image forgery with source/target localization. In: European Conference on Computer Vision (2018)
33. Zhuang, P., Li, H., Yang, R., Huang, J.: ReLoc: a restoration-assisted framework for robust image tampering localization. IEEE Transactions on Information Forensics and Security (2023)