Representative Forgery Mining for Deep Fake Detection

Name:	**Akash Kumar Singh**
Registration No./Roll No.:	19023
Institute/University Name:	IISER Bhopal
Stream:	Data Sci. and Engg.
Date & Time of Submission:	11:55 PM, November 22, 2022

1 Introduction

Face modification technology is advancing quickly, making it easier than ever to create false faces. This facilitates the rapid distribution of fake facial photos on social media, and as more sophisticated techniques emerge, it gets harder for people to tell the difference between the two. Unfortunately, while a simple CNN-based false face detector may detect phony faces with sufficient accuracy, its perception of forgery may differ from that of humans.

Humans typically detect representative forgery throughout the entire face, but vanilla CNN-based detectors tend to check forgeries from a limited portion of the face.

Instead of overfitting the forgeries, which are primarily beneficial in lowering the bi-classification loss function on the training set, detectors should pay more attention to the forgeries, which can considerably represent the corresponding manipulation approach.

In this study, the limited-attention issue is addressed by improving training data while it is being trained using the attention-based data augmentation technique Representative Forgery Mining (RFM).

2 Methods

In this project we have worked on this paper[1]. Representative Forgery Mining (RFM), an attention-based data augmentation technique, is suggested using fake face detection as a binary classification problem. The following two elements make up its composition:

2.1 Forgery Attention Map (FAM)

- A tracer method called FAM is used to precisely identify the area of the face that a detector is sensitive to, and to use that information as a guide for data augmentation.
- The most sensitive region is defined as the region where perturbation has the most critical impact on detection results.
- Each value in the FAM exactly identifies the detector's sensitivity to the relevant image pixel. FAM Map can be formulated as

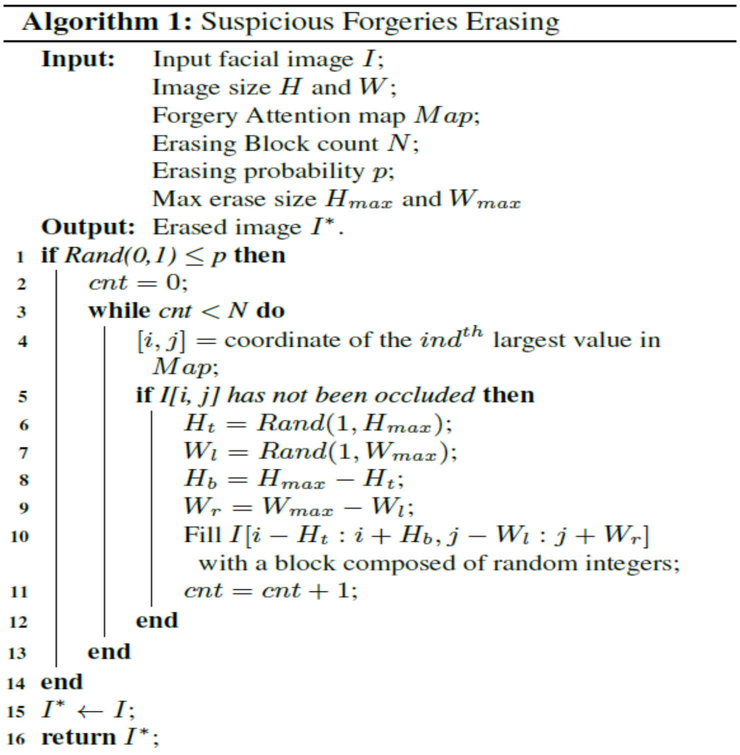
  $$\begin{aligned}\mathrm{Map}_I &= max(abs(\nabla_I O_{fake} \nabla_I O_{real})\\ &= \max(\nabla_I(abs(O_{fake} O_{real})))\end{aligned}$$

  where the function max(·) calculates the maximum value along channel axis and the function abs(·) obtains the absolute value of each pixel.

[1] https://arxiv.org/pdf/2104.06609

2.2 Suspicious Forgery Erasing (SFE)

- To purposefully occlude the Top-N sensitive facial regions, enabling the detector to investigate representative forgeries from the disregarded facial region.
- The erasing method, SFE, achieves dynamic refinement by masking the Top-N sensitive face regions determined by FAM. The algorithm of SFE is shown in figure1.



```
Algorithm 1: Suspicious Forgeries Erasing
Input:  Input facial image $I$;
        Image size $H$ and $W$;
        Forgery Attention map $Map$;
        Erasing Block count $N$;
        Erasing probability $p$;
        Max erase size $H_{max}$ and $W_{max}$
Output: Erased image $I^*$.
if $Rand(0,1) \leq p$ then
    $cnt = 0$;
    while $cnt < N$ do
        $[i, j]$ = coordinate of the $ind^{th}$ largest value in $Map$;
        if $I[i, j]$ has not been occluded then
            $H_t = Rand(1, H_{max})$;
            $W_l = Rand(1, W_{max})$;
            $H_b = H_{max} - H_t$;
            $W_r = W_{max} - W_l$;
            Fill $I[i - H_t : i + H_b, j - W_l : j + W_r]$
              with a block composed of random integers;
            $cnt = cnt + 1$;
        end
    end
end
$I^* \leftarrow I$;
return $I^*$;
```

Figure 1: algorithm of SFE

By Combining these two elements, RFM is able to detect representative forgery without well-designed supervision and achieve State-Of-The-Art performance on the tested datasets. Figure 2 shows the procedure of RFM.

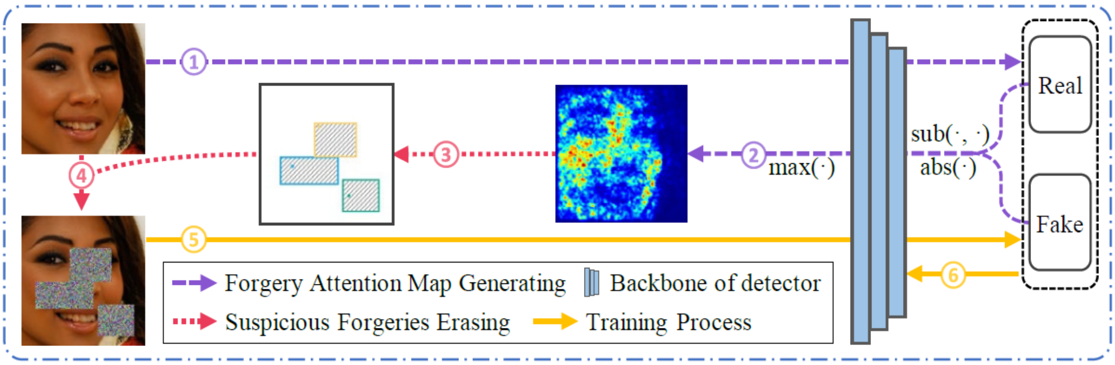


Figure 2: The procedure of RFM

3 Experimentation and Baseline Results

3.1 Dataset

- Experiments are performed on three well-known datasets: DFFD, FaceForencics++ and Celeb-DF.
- DFFD contains 58,703 real facial images and 240,336 fake facial images both one-stage and two-stage manipulation techniques are used to generate fake faces.
- According to the number of manipulation technical stages, DFFD is divided into two groups, Group A and Group B.
- Group A contains fake faces generated by two-stage techniques such as FaceSwap, Deepfakes and Face2Face, Group B is composed of the one-stage techniques such as FaceAPP, StarGAN, PGGAN and Style-GAN. The images in GroupA are collected from FaceForensics++.
- Celeb-DF uses second generation manipulation technology, which generates fake face through a improved two-stage technique.
- Celeb-DF contains 590 real videos collected from YouTube video clips of 59 celebrities and 5,639 high-quality fake videos of celebrities generated using improved synthesis process.
- For fake face detection, facial images are extracted from the key frames of videos.

3.2 Experiment Settings

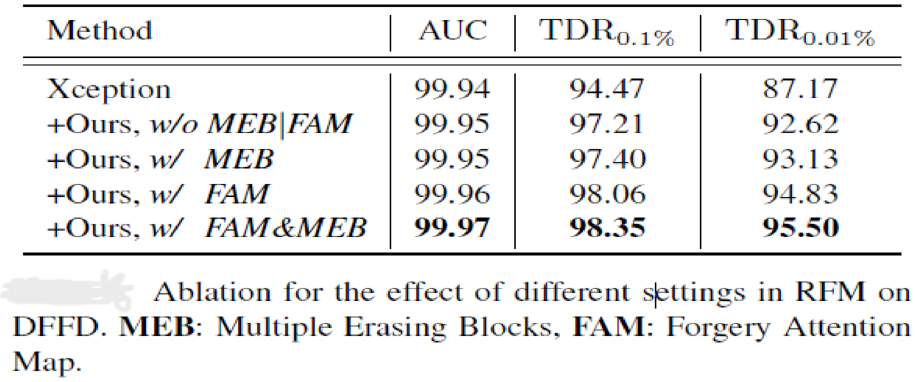
- Resized the aligned facial images into a fixed size of 256 × 256.
- Applied random and center cropping into training and testing process to resize the images to 224 × 224, respectively.
- Moreover, each image is flipped horizontally with a probability of 50% during training.
- All the detectors are trained by using Adam optimizer with fixed learning rate of 0.0002.
- The size of mini-batch is set to 16, and each mini-batch consists of 8 real and 8 fake facial images.
- Hyper-parameters N, p, Hmax and Wmax are implicitly set as 3, 1.0, 120 and 120, respectively.
- Detection performance is reported through the **evaluation metrics** such as Area Under Curve (AUC) of ROC, True Detect Rate (TDR) at False Detect Rate (FDR) of 0.01%, and TDR at FDR of 0.1%.

3.3 Baseline Results

Ablation Study on DFFD & Celeb-DF: Separately conducted experiments on DFFD to investigate how Forgery Attention Map (FAM) and Multiple Erasing Blocks (MEB) boost detection performance. They compared RFM with well-known erasing methods such as Adversarial Erasing (AE) and Random Erasing (RE). The results are shown in Figure 3, where "FAM-MEB" is the original setting, "w/ MEB" denotes placing the anchors of SFE randomly, "w/ FAM" denotes only occluding the Top-1 sensitive region under the guidance of FAM, and "w/o MEB—FAM" denotes using a single erasing block to occlude a random region of the face.

4 Reproduced Results

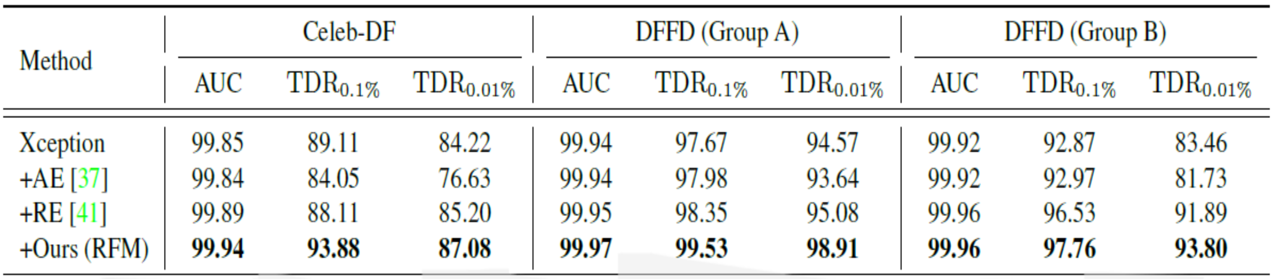
The reproduced results corresponding to above Figure 3 and 4 is in figure 5 and 6.



Method	AUC	TDR$_{0.1\%}$	TDR$_{0.01\%}$
Xception	99.94	94.47	87.17
+Ours, *w/o MEB|FAM*	99.95	97.21	92.62
+Ours, *w/ MEB*	99.95	97.40	93.13
+Ours, *w/ FAM*	99.96	98.06	94.83
+Ours, *w/ FAM&MEB*	**99.97**	**98.35**	**95.50**

Ablation for the effect of different settings in RFM on DFFD. **MEB**: Multiple Erasing Blocks, **FAM**: Forgery Attention Map.

Figure 3:



Method	Celeb-DF			DFFD (Group A)			DFFD (Group B)		
	AUC	TDR$_{0.1\%}$	TDR$_{0.01\%}$	AUC	TDR$_{0.1\%}$	TDR$_{0.01\%}$	AUC	TDR$_{0.1\%}$	TDR$_{0.01\%}$
Xception	99.85	89.11	84.22	99.94	97.67	94.57	99.92	92.87	83.46
+AE [37]	99.84	84.05	76.63	99.94	97.98	93.64	99.92	92.97	81.73
+RE [41]	99.89	88.11	85.20	99.95	98.35	95.08	99.96	96.53	91.89
+Ours (RFM)	**99.94**	**93.88**	**87.08**	**99.97**	**99.53**	**98.91**	**99.96**	**97.76**	**93.80**

Figure 4: Comparison of RFM with well-known erasing methods and state of the art on DFFD and Celeb-DF.We use Xception as the backbone in the first cell and use Patch as the backbone in the second cell.

5 Our Methodology and Work

We developed three various strategies to improvise the suggested methodology and the authors' study work after producing the baseline results after implementing the paper and source code. The following are mentioned:

5.1 Novelty in Dataset: Faceshifter & Neural Texture

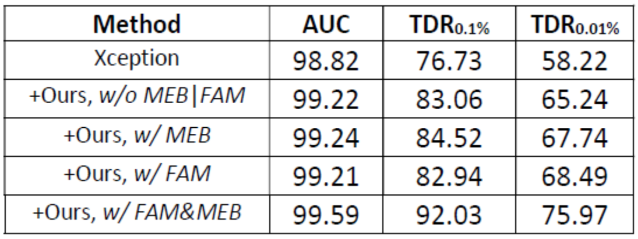
Separately conducted experiments to compare RFM on dataset FaceShifter and Neural Texture. The results obtained with the base model as Xception is shown in Table 1 below.

Method		Face Shifter		||	Neural Textures	
	AUC	TDR$_{0.1\%}$	TDR$_{0.01\%}$	AUC	TDR$_{0.1\%}$	TDR$_{0.01\%}$
Xception	99.65	96.47	90.65	97.15	78.04	59.90
+Ours (RFM)	99.64	96.02	89.42	97.78	86.43	69.76

Table 1:

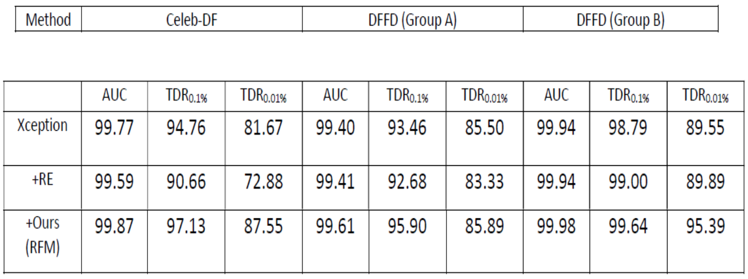
5.2 Novelty in Methodology: Gaussian Distribution

Separately conducted experiments on DFFD to compare Gaussian SFE with other erasing methods such as SFE(as in paper) and Random Erasing (RE). Here, values obtained after applying a gaussian filter are used to compose erasing blocks for each image. The results obtained is shown in Table 2 below.



Method	AUC	$\text{TDR}_{0.1\%}$	$\text{TDR}_{0.01\%}$
Xception	98.82	76.73	58.22
+Ours, *w/o MEB|FAM*	99.22	83.06	65.24
+Ours, *w/ MEB*	99.24	84.52	67.74
+Ours, *w/ FAM*	99.21	82.94	68.49
+Ours, *w/ FAM&MEB*	99.59	92.03	75.97

Figure 5:



Method	Celeb-DF			DFFD (Group A)			DFFD (Group B)		
	AUC	$\text{TDR}_{0.1\%}$	$\text{TDR}_{0.01\%}$	AUC	$\text{TDR}_{0.1\%}$	$\text{TDR}_{0.01\%}$	AUC	$\text{TDR}_{0.1\%}$	$\text{TDR}_{0.01\%}$
Xception	99.77	94.76	81.67	99.40	93.46	85.50	99.94	98.79	89.55
+RE	99.59	90.66	72.88	99.41	92.68	83.33	99.94	99.00	89.89
+Ours (RFM)	99.87	97.13	87.55	99.61	95.90	85.89	99.98	99.64	95.39

Figure 6:

Method	AUC	$\text{TDR}_{0.1\%}$	$\text{TDR}_{0.01\%}$
Xception + RFM (with Gaussian Occlusion)	99.20	85.02	66.60

Table 2:

5.3 Novelty in Model: Changed Xception to Efficient Net V2 S

The efficientnet-v2-s model is a small variant of the EfficientNetV2. EfficientNetV2 is a new family of convolutional networks that have faster training speed and better parameter efficiency than previous models. The results obtained is shown in Table 3 below.

Method	AUC	$\text{TDR}_{0.1\%}$	$\text{TDR}_{0.01\%}$
Efficient Net V2 S	97.91	69.79	55.87

Table 3:

6 Conclusion

We have reproduced the result which is very close to the paper. The RFM method of the author performs better than Xception on the new DataSet as we can see in the Novelty in DataSet section. We can also see that our new proposed method in the 'Novelty in Method' section of using Gaussian occlusion performs almost near to the base RFM reproduced result. If hyperparameters can be changed then it may outperform other techniques as well. Allthough the Efficient Net V2 S model doesn't perform well.

Code link and path of the script to run the code: run_models_check.sh