大学共同利用機関法人 情報・システム研究機構
国立情報学研究所
National Institute of Informatics

National University
SOKENDAI
The Graduate University for Advanced Studies

Modular Convolutional Neural Network for Discriminating between Computer-Generated Images and Photographic Images

Huy H. Nguyen*, Ngoc-Dung T. Tieu, Hoang-Quoc Nguyen-Son, Vincent Nozick, Junichi Yamagishi, and Isao Echizen

International Conference on Availability, Reliability and Security

August 27-30, 2018

Hamburg, Germany

Outline

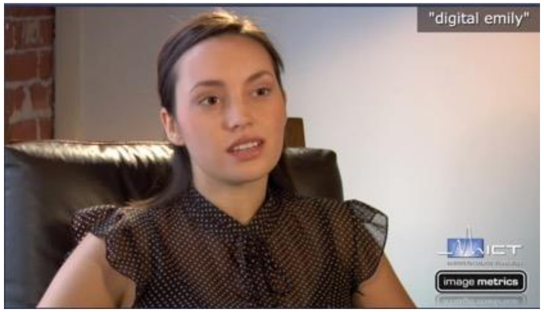
1. Motivation
2. Related Work
3. Proposed Method
4. Evaluation
5. Conclusion & Future Work

1. Motivation

1. Motivation

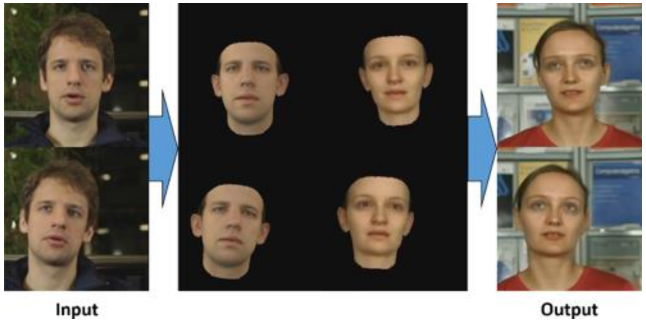
1.1. Hard level for attackers

Hard <Requirements for attackers to perform spoofing attacks> **Easy**



The Digital Emily Project [1] (2008)

Face2Face: Real-time face capture & reenactment [2] (2016)



DeepVideo Portraits = Face2Face + head poses [3] (2018)

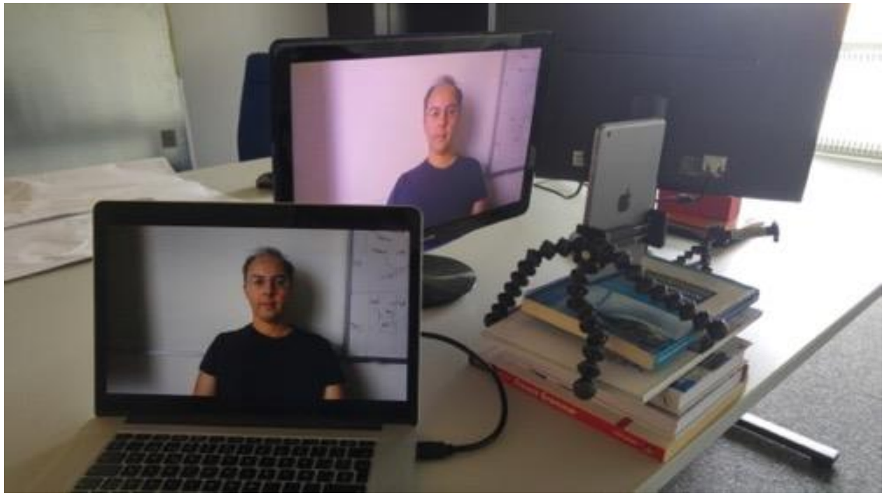
[1] SIGGRAPH 2008 Expo / SIGGRAPH 2009 Computer Animation Festival / SIGGRAPH 2009 Courses / CVMP 2009 / IEEE CG&A 2010.

[2] Thies, Justus, et al. "Face2Face: Real-time face capture and reenactment of RGB videos." CVPR 2016.

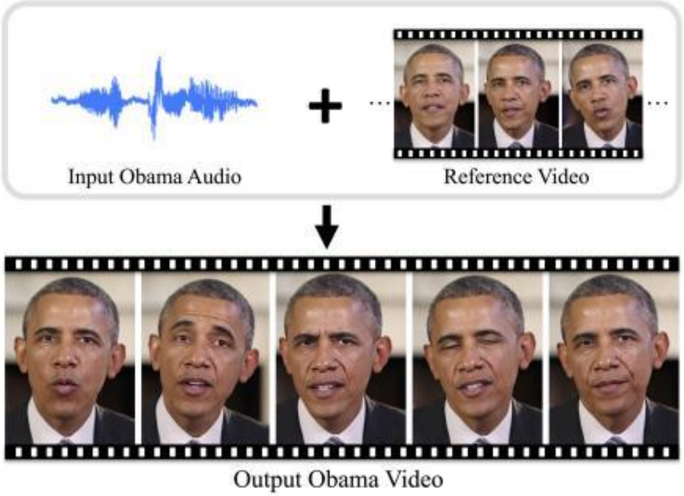
[3] Kim, Hyeongwoo, et al. "DeepVideo Portraits." SIGGRAPH 2018.

1. Motivation

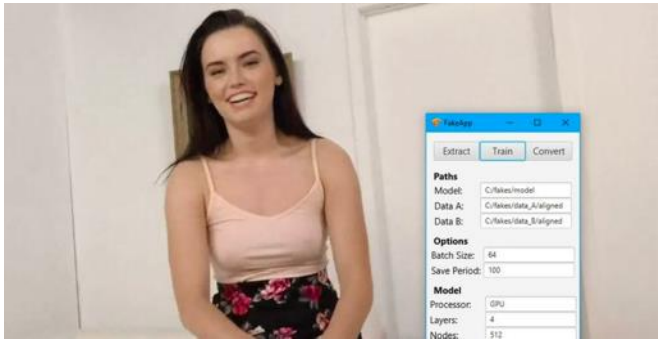
1.2. Examples



A presentation attack to break a facial authentication system [4]



Creating fake news/ Impersonation [5]



Pornography DeepFake[6]

[4] Costa-Pazo, Artur et al. "The replay-mobile face presentation-attack database." BIOSIG 2016.
[5] Suwajanakorn, Supasorn et al. "Synthesizing obama: learning lip sync from audio." TOG 36.4 (2017): 95.
[6] Brandon, John "Terrifying high-tech porn: Creepy 'deepfake' videos are on the rise". Fox News. Retrieved 2018-02-20.

1. Motivation

1.3. Computer-Generated Images (CGIs) vs Photographic Images (PIs)

There is a continuous competition between attackers and defenders.

→ CGI-PI discriminators need to be regularly updated to deal with:

- New kind of attacks
- Better quality of CGIs
- Larger amount of data

2. Related Work

2. Related Work

Spoofing /forgery detection

- Using wavelet/wavelet-like transformations or differential images.
- Using the intrinsic properties of image acquisition devices.
- Using texture information.
- Using statistical analysis (independently or jointly with other methods).
- Using convolutional neural network (CNN) as classifier of hand-crafted features / automatic feature extractor + classifier.

2. Related Work

State-of-the-art spoofing /forgery detections [7]

- Hand-crafted feature + SVM (Fridrich & Kodovsky 2012)
- Hand-crafted feature + CNN (Cozzolino et al. 2017)
- CNN with ordinary layers (Bayar and Stamm 2016)
- CNN + statistical pooling (Rahmouni et al. 2017)
- Pre-trained [VGG19 + AlexNet] (Raghavendra et al. 2017)
- Two-stream network and a pre-trained GoogleLeNet Inception V3 (Zhou et al. 2017)
- Transfer learning of XceptionNet (Rössler et al. 2018)

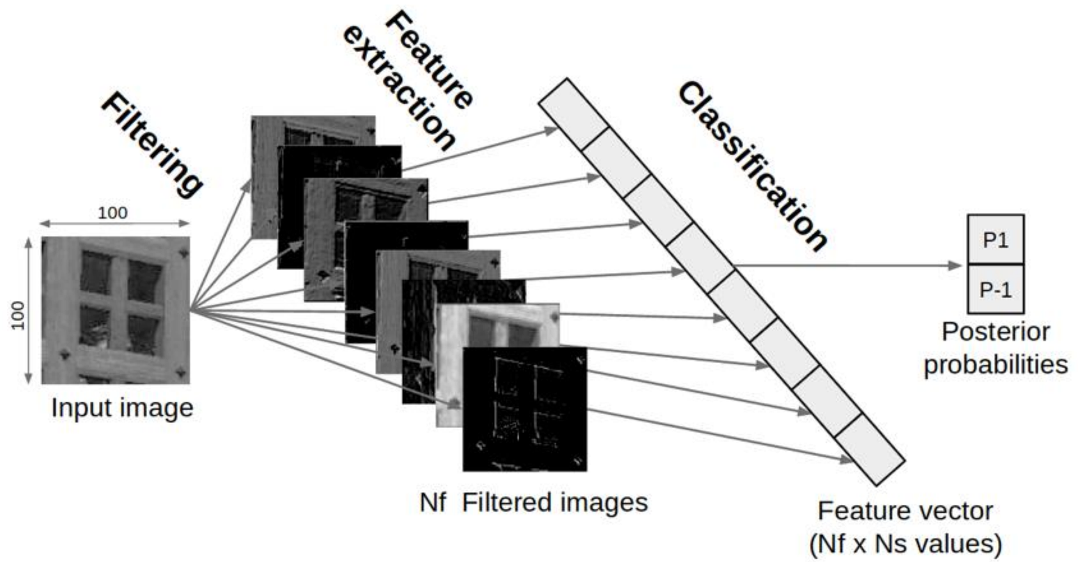
[7] Rössler, Andreas, et al. "FaceForensics: A Large-scale Video Dataset for Forgery Detection in Human Faces." arXiv preprint, 2018.

2. Related Work

Spoofing /forgery detection

Rahmouni et al. 2017 [8] :

- Using CNN filters.
- Each filter ends with a statistical pooling layer, which calculates mean, variance, min and max of the filtered image.



[8] Rahmouni et al. "Distinguishing Computer Graphics from Natural Images Using Convolution Neural Networks." WIFS 2017.

3. Proposed Method

3. Proposed method

3.1. Overview

- Split input image into small patches → Take advantage of the details (e.g. noise, patterns) which are different between CGIs and PIs

- Leverage well-trained network on a large-scale dataset
- Pre-trained, keeping fixed

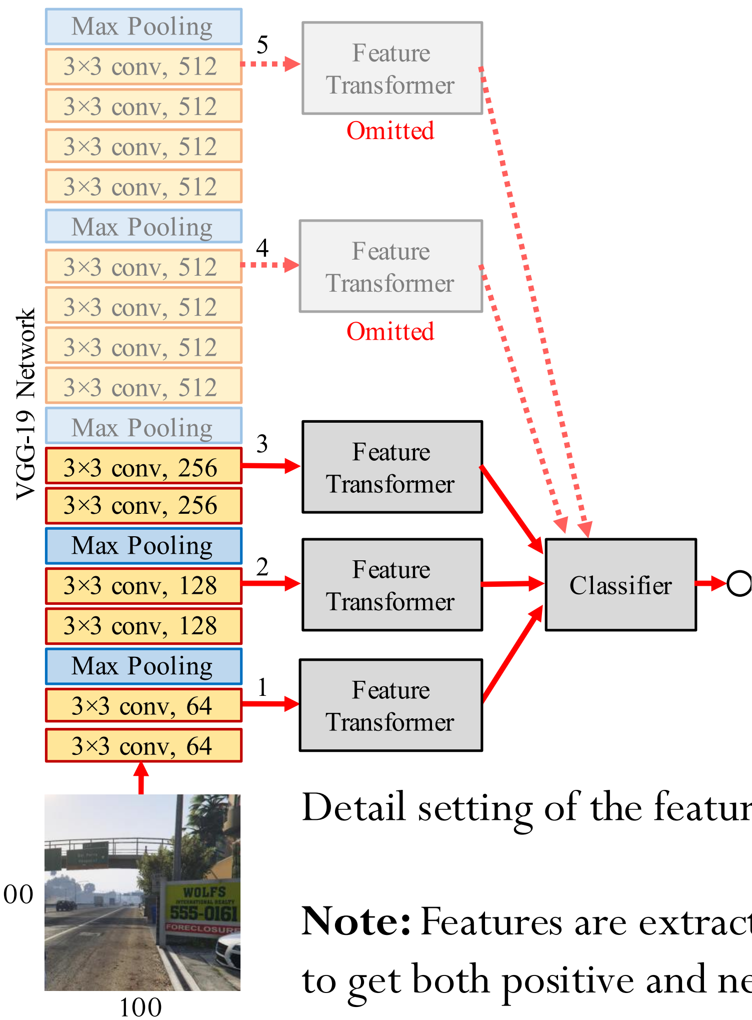
- Transforming features extracted by the feature extractor into useful features for CGI-PI classification.
- Need training

- Classifying CGI-PI
- Training with many classifier algorithms, then choosing the best one

- Aggregate the results

3. Proposed method

3.2. Feature extractor



Detail setting of the feature extractor

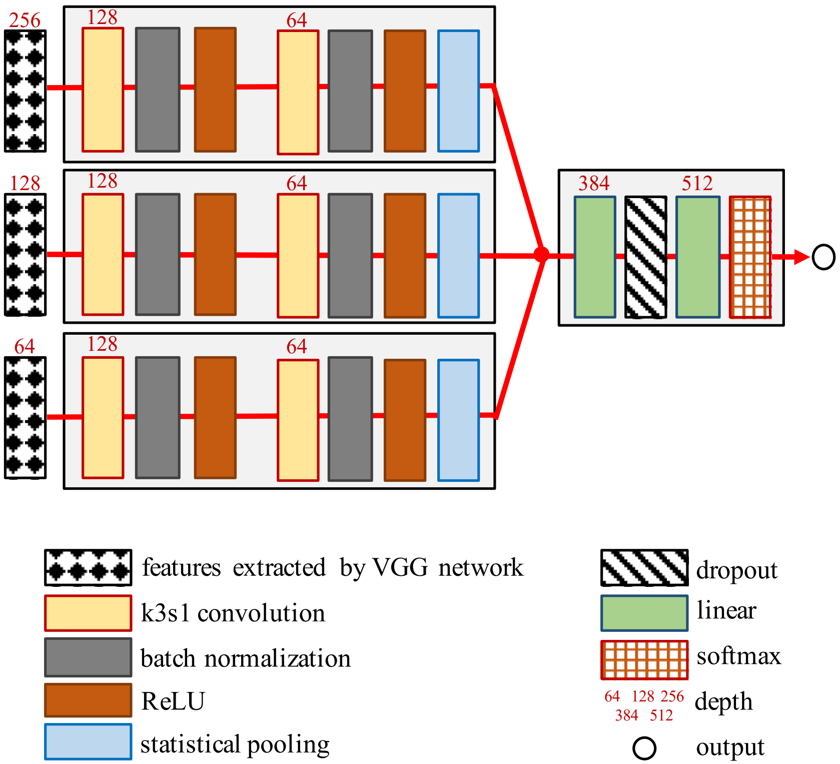
Features	Accuracy
1	95.40
1+2	97.60
1+2+3	**97.70**
1+2+3+4	96.50
1+2+3+4+5	96.10

Accuracy on Patch-100-Full dataset

Note: Features are extracted before ReLU layers to get both positive and negative components.

3. Proposed method

3.3. Feature transformers & classifier



Detailed settings of feature transformers and MLP classifier

Statistical pooling:

- Mean:

$$\mu_k = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} I_{kij}$$

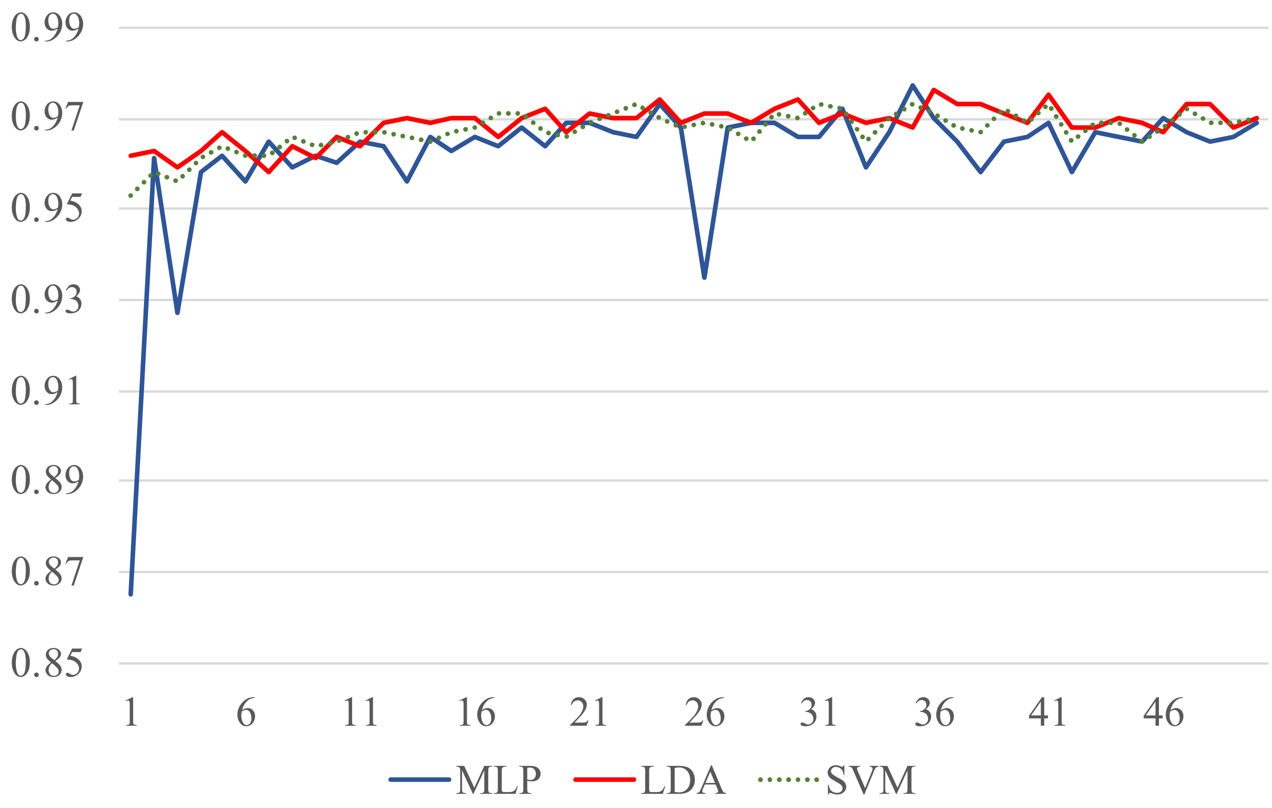
- Variance:

$$\sigma_k^2 = \frac{1}{H \times W - 1} \sum_{i=1}^{H} \sum_{j=1}^{W} \left(I_{kij} - \mu_k\right)^2$$

k: layer index
I: 2-D filter array
H: height of the filter
W: width of the filter

3. Proposed method

3.4. Classifier



Learning curves of MLP, LDA and SVM on Patch-100-Full dataset

4. Evaluation

4. Evaluation

4.1. Datasets

Using only high-resolution images for training is not sufficient to counter real-world attacks.

→ We expanded the dataset proposed by Rahmouni et al. [8]

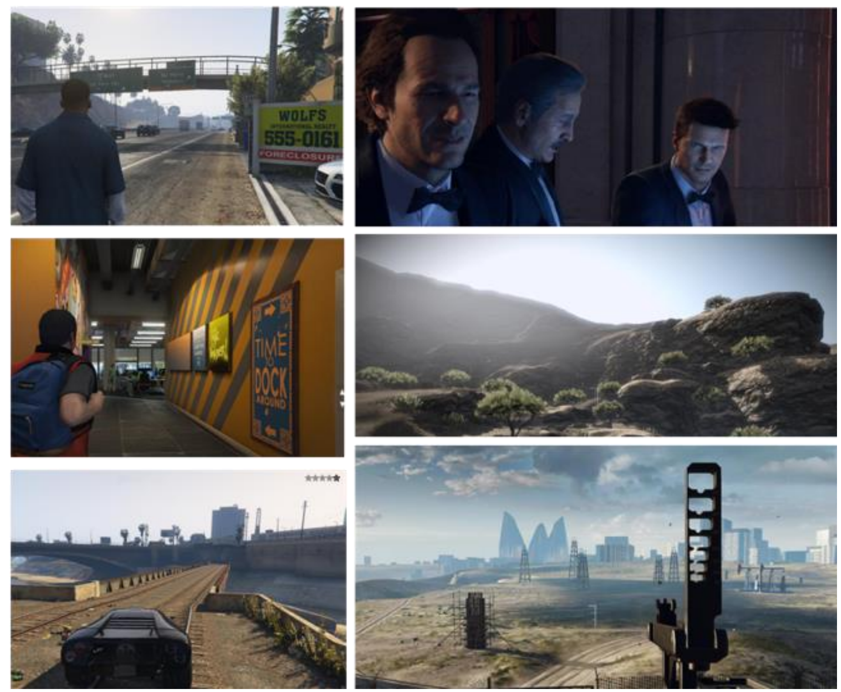
Name	No. for training	No. for valid.	No. for testing	Image size
Full-Size	2,520	360	720	High-resolution
Patch-100-Full	40,000	1,000	2,000	100 x 100
Patch-256-Full	40,000	1,000	2,000	256 x 256
Reduced-Size	2,520	360	720	360p
Patch-100-Reduced	40,000	1,000	2,000	100 x 100

[8] Rahmouni et al. "Distinguishing Computer Graphics from Natural Images Using Convolution Neural Networks." WIFS 2017.

4. Evaluation

4.1. Datasets

PIs [9]



CGIs [10]

[9] Dang-Nguyen et al. "RAISE: a raw images dataset for digital image forensics." MMSys 2015.
[10] M. Piaskiewicz. (2017, May) Level-design reference database. [Online]. Available: http://level-design.org/referencedb/.

4. Evaluation

4.2. Patch aggregation

Q: Why dividing images into patches?

A:

- Input images are usually large, but:
  - → We need to analyze the patterns, however, resizing images might destroy this information.
  - → GPU computation could not afford large images.
- Can do parallel computing by using patches as batch input.

Q: How to compute the final result?

A: Calculate the mean of probabilities of selected patches.

4. Evaluation

4.2. Patch aggregation

Patch selection strategies:

Selecting all patches (left) vs. random sampling (right)

4. Evaluation

4.2. Patch aggregation

Classifier		MLP				LDA			
Patch size	**No. of patches**	**1**	**2**	**3**	**Avg.**	**1**	**2**	**3**	**Avg.**
100 x 100	10	99.31	99.72	99.86	99.63	99.86	99.31	99.72	99.63
	50	99.86	99.86	99.86	**99.86**	99.86	99.86	99.86	**99.86**
	100	99.86	99.86	99.86	**99.86**	99.86	99.86	99.86	**99.86**
	All				**99.86**				**99.86**
256 x 256	5	99.72	99.44	99.72	99.63	99.44	99.03	99.58	99.35
	10	100.0	99.72	100.0	**99.91**	99.86	99.58	99.72	99.72
	25	99.86	99.86	99.86	**99.86**	99.72	99.72	99.72	99.72
	All				**99.86**				99.72

4. Evaluation

4.3. Comparison

Case 1: High-resolution datasets

Method	Patch-100-Full	Patch-256-Full	Full-size
Rahmouni et al. - 100	86.10	x	96.94
Rahmouni et al. - 256	x	93.95	98.75
Proposed - MLP - 100	**96.55**	**x**	**99.86**
Proposed - LDA - 100	96.40	x	99.86
Proposed - MLP - 256	**x**	**98.70**	**99.72 - 100.0**
Proposed - LDA - 256	x	98.70	99.58 - 99.86

4. Evaluation

4.3. Comparison

Case 2: High- & low-resolution datasets

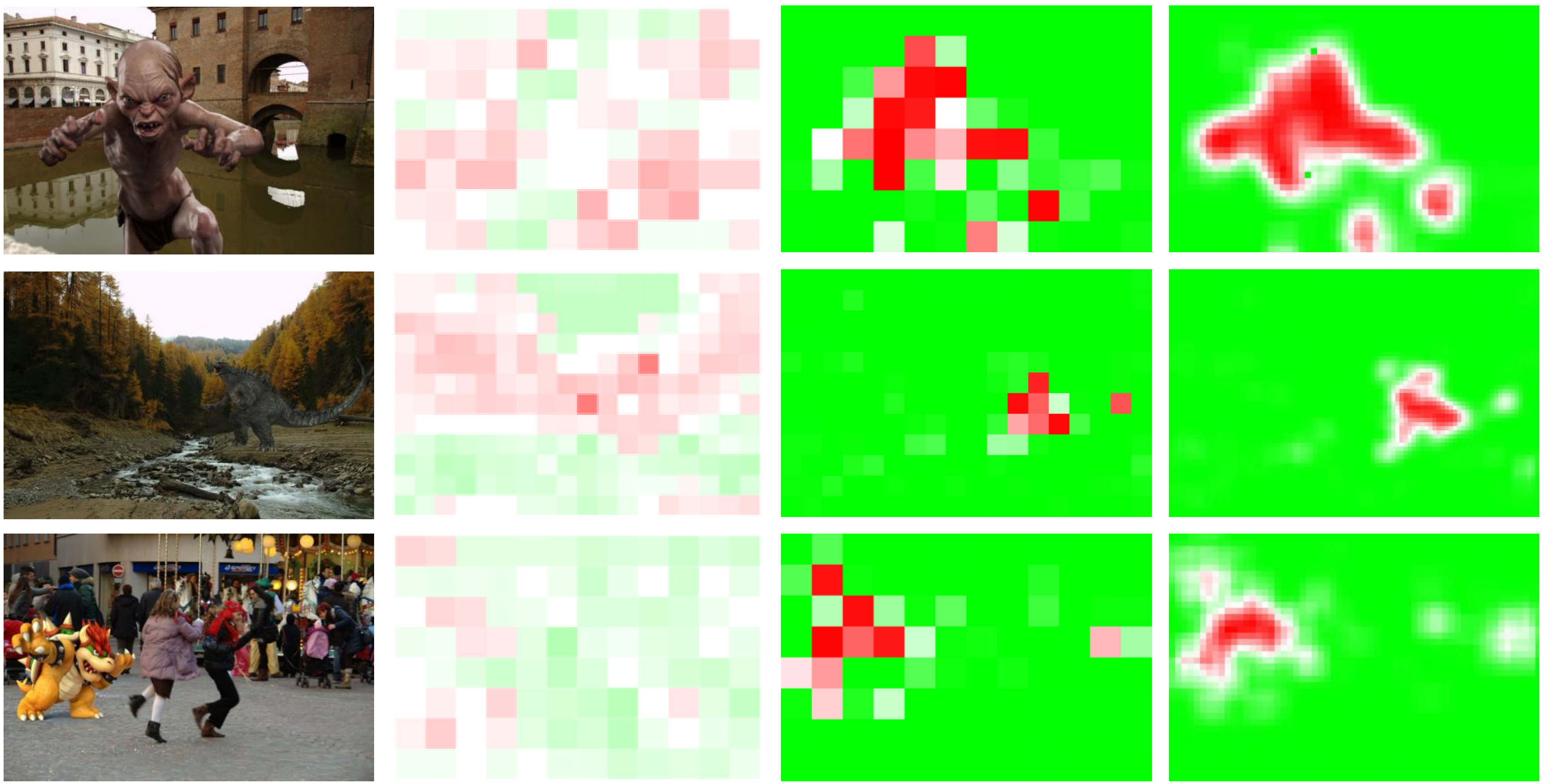
(1) Train on high-res datasets → test on both low- & high-res datasets

(2) Re-train on mixed datasets → test on both low- & high-res datasets

Method	Patch-100-Reduced	Reduced-Size	Patch-100-Full	Full-Size
Rahmouni et al. (1)	51.50	50.97	86.10	96.94
Proposed - MLP (1)	52.55	51.81	**96.55**	**99.86**
Proposed - LDA (1)	52.35	51.53	96.40	99.86
Rahmouni et al. (2)	60.45	79.72	81.20	95.00
Proposed - MLP (2)	88.60	96.67	93.40	97.64
Proposed - LDA (2)	**89.95**	**97.92**	**94.80**	**98.89**

4. Evaluation

4.4. Detecting image splicing



Original Rahmouni et al. Non-overlapped Overlapped

5. Conclusion & Future Work

5. Conclusion & Future Work

5.1. Conclusion

- The proposed method out-performed the method of Rahmouni et al. 2017 [8].
- Random sampling strategy is effective with large-scale images.
- The method can also be used to detect image splicing.
- Using only high-resolution images for training is not sufficient to counter real-world attacks.

[8] Rahmouni et al. "Distinguishing Computer Graphics from Natural Images Using Convolution Neural Networks." WIFS 2017.

5. Conclusion & Future Work

5.2. Future work

- Using adversarial training & evaluating with adversarial samples.
- Using more datasets: FaceForensics [7], 3D Mask Attach Dataset [11], ReplayAttack [12], Replay-Mobile [13].
- Using attention-based approach instead of patch aggregation.
- Comparing with more approaches.

[7] Rössler, Andreas, et al. "FaceForensics: A Large-scale Video Dataset for Forgery Detection in Human Faces." arXiv preprint, 2018.
[11] Erdogmus, Nesli, and Sebastien Marcel. "Spoofing in 2D face recognition with 3D masks and anti-spoofing with Kinect." BTAS 2013.
[12] Chingovska, Ivana et al. "On the effectiveness of local binary patterns in face anti-spoofing." BIOSIG 2012.
[13] Costa-Pazo, Artur, et al. "The replay-mobile face presentation-attack database." BIOSIG 2016.

Thank you for your attention