Image Forgery Detection using EfficientNets and Multi-attentional Methods at different levels of JPEG Compression

Marcelo Galindo 葛洋晴
Student ID: 108006205

Table of contents

01

Motivation and Objectives

Research motivations and purpose of the work.

Motivations

- Deep learning allows the creation of realistic images and videos.
- Threat to privacy, identity and trust.
- Social media amplifies it.
- Compression introduces noise and visual artifacts.
- Decrease in robustness for higher degree of compression

[1] Clark, B. (2018, February 21). Deepfakes algorithm nails Donald Trump in most convincing fake yet. TNW | Artificial-Intelligence. https://thenextweb.com/news/deepfakes-algorithm-nails-donald-trump-in-most-convincing-fake-yet

Objectives

Improve Robustness

Increase accuracy of the model under varying levels of JPEG compression (quality levels of 77, 60, 15, and 10).

Implement an efficient backbone

Exploring the use of EfficientNets (Tan & Le, 2021), as the primary detection model.

Contribute

To the field of digital Computer Vision by providing a more robust and accurate method for detecting image forgeries.

02

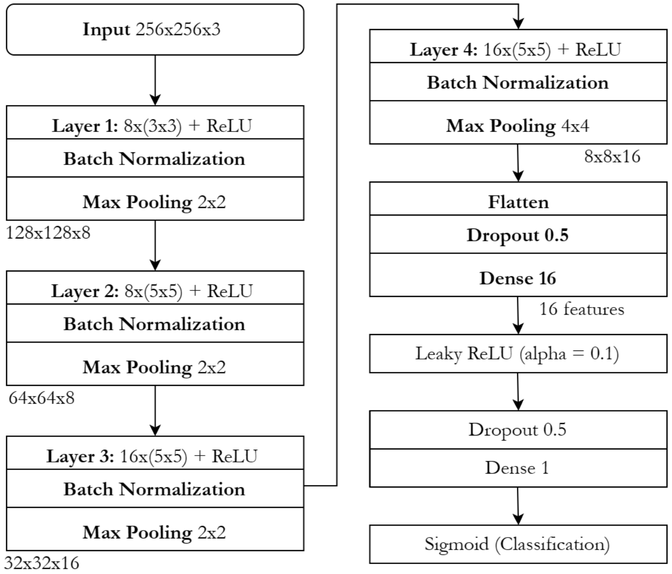
Current State of Related Research

The Current State of Related Research and Comparison, and Important Contributions of the Project.

Meso4 and MesoInception4

Limitations of Meso4:

- Issues in distinguishing between real and fake.
- Fails to adapt to unseen forgery types.
- Struggles with generalization



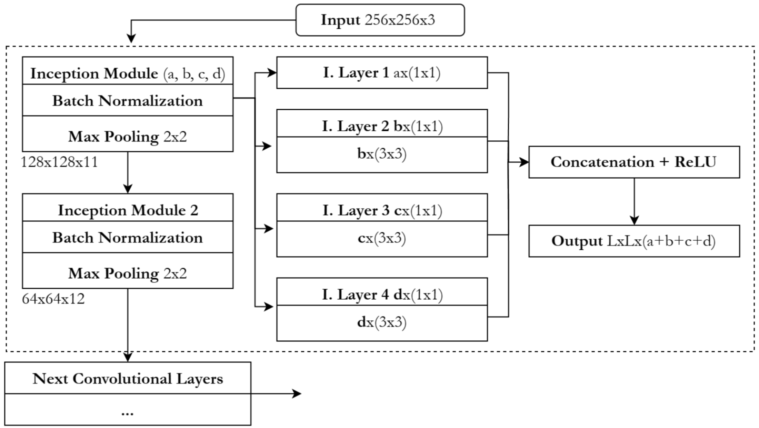


***Figure** Block Diagram of the Meso4 network Architecture and Inception Layer from Inception4.*

EfficientNet Principles

- Optimize both accuracy and computational efficiency (measured in FLOPS).

$$\mathrm{OPT(m)} = \mathrm{ACC}(m) \times \left(\frac{\mathrm{FLOPS}(m)}{T}\right)^{w}$$

With w: hyperparameter (define at -0.07).

Example:

$$\mathrm{OPT(m)} = \mathrm{ACC}(m) \times \left(\frac{2{,}000{,}000{,}000}{1{,}500{,}000{,}000}\right)^{-0.07}$$

Trade-off between accuracy and efficiency. By setting ACC as 85%, by scaling down, the optimization score for model m is approximately 0.831.

EfficientNets Principles

$$\begin{aligned} \textit{Depth} \quad (d)&: \alpha^{\phi} \\ \textit{Width} \quad (w)&: \beta^{\phi} \\ \textit{Resolution} \quad (r)&: \gamma^{\phi} \end{aligned}$$

Computational cost (measured in FLOPS) increases approximately by 2^{ϕ}

EfficientNet	Width	Depth	Resolution	Dropout	FLOPS
B0	1.0	1.0	224	0.2	0.39B
B1	1.0	1.1	240	0.2	0.70B
B2	1.1	1.2	260	0.3	1.0B
B3	1.2	1.4	300	0.3	1.8B
B4	1.4	1.8	380	0.4	4.2B
B5	1.6	2.2	456	0.4	9.9B
B6	1.8	2.6	528	0.5	19B
B7	2.0	3.1	600	0.5	37B
B8	2.2	3.6	672	0.5	89.5B
L2	4.3	5.3	800	0.5	-

EfficientNets

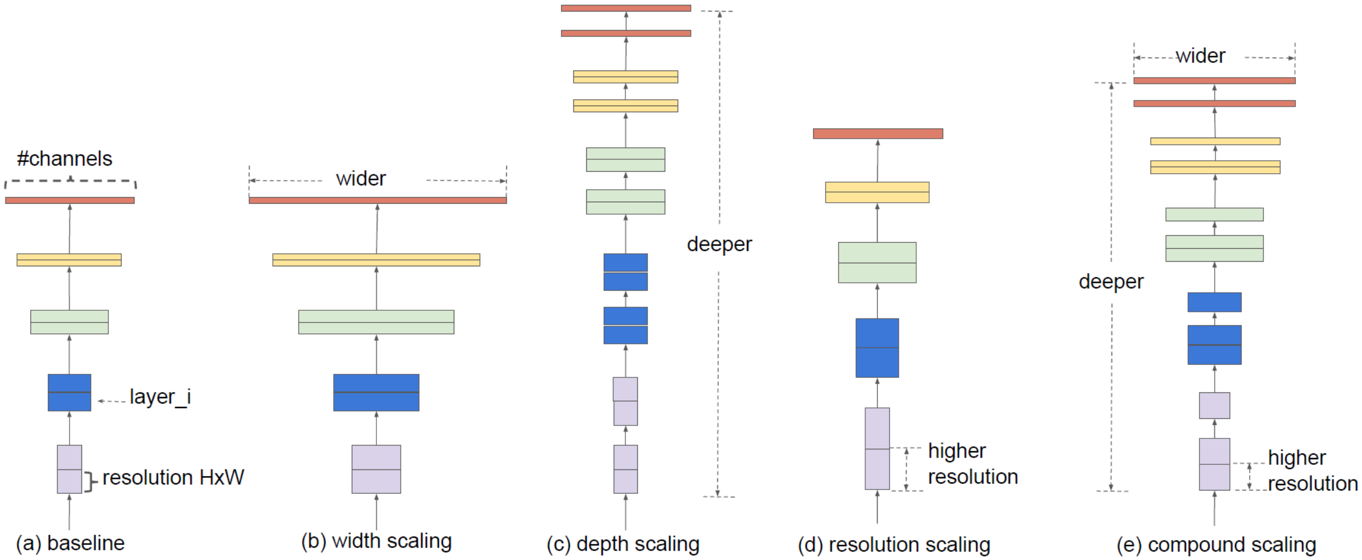


Figure 2. **Model Scaling.** (a) is a baseline network example; (b)-(d) are conventional scaling that only increases one dimension of network width, depth, or resolution. (e) is our proposed compound scaling method that uniformly scales all three dimensions with a fixed ratio.

[2] Tan, M., & Le, Q. V. (2020). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks (arXiv:1905.11946). arXiv. https://doi.org/10.48550/arXiv.1905.11946

Comparison results

Detector	Backbone	FaceForensics++						
		FF-c23	FF-c40	FF-DF	FF-F2F	FF-FS	FF-NT	Avg
Meso4	MesoNet	0.6077	0.5920	0.6771	0.6170	0.5946	0.5701	0.6097
MesoIncept	MesoNet	0.7583	0.7278	0.8542	0.8087	0.7421	0.6517	0.7571
EffNetB4	Efficient	0.9567	0.8150	0.9757	0.9758	0.9797	0.9308	0.9389
EffNetB4*	Efficient	*	*	0.9806	0.9870	0.9708	0.9531	0.9729
Detectors		Overall Gain						
EffNetB4 vs EffNetB4*		*	*	+0.4%	+1%	−0.8%	+2%	+3%
Meso4 vs EffNetB4*		37.29%	39.50%	29.86%	35.88%	37.62%	38.30%	36.32%
MesoIncept vs EffNetB4*		22.23%	25.92%	12.15%	16.71%	22.87%	30.14%	22.23%

03

Research Methodology

Design Principles, Research Methods and Steps.

Dataset: FaceForensics++

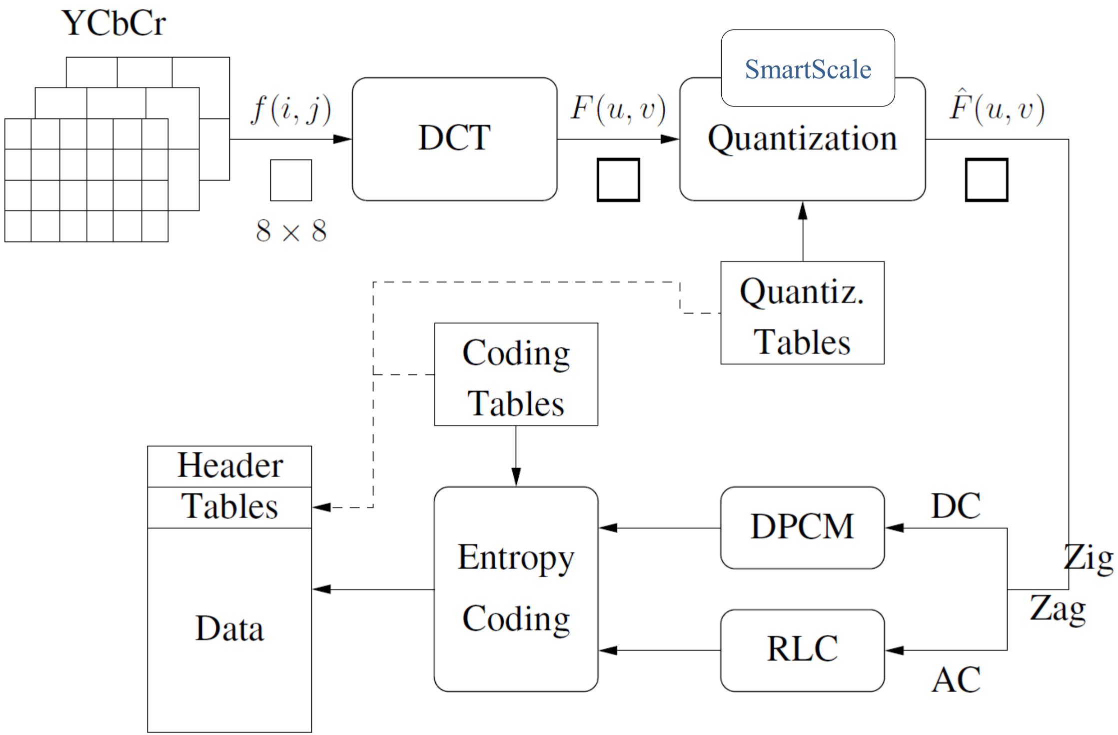
	Dataset: FaceForensics++					
Material	Original	FF-DF	FF-F2F	FF-FS	FF-NT	Total
Videos	1000	1000	1000	1000	1000	5000
Frames	32	32	32	32	32*	160,000

**NeuralTextures to achieve Data Augmentation more frames were extracted after.*

Dataset: FaceForensics++

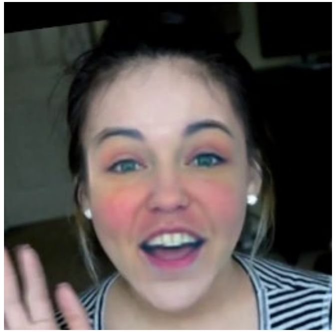
Original *Deepfake* *Face2Face* *FaceSwap* *NeuralTextures*

JPEG Compression with MozJPEG



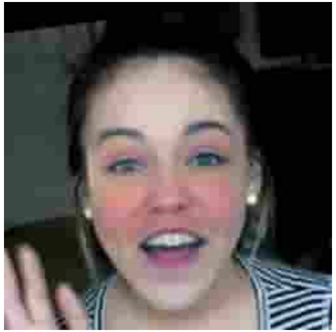
[1] Li, Z.-N., Drew, M. S., & Liu, J. (2021). Fundamentals of Multimedia. Springer International Publishing. https://doi.org/10.1007/978-3-030-62124-7, [4] Libjpeg-turbo | About / A Study on the Usefulness of DCT Scaling and SmartScale. (n.d.). Retrieved May 27, 2024, from https://libjpeg-turbo.org/About/SmartScale

Compression examples

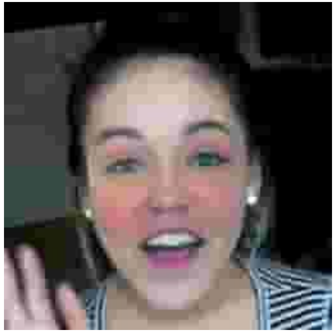


c23
PSNR: 33.580
SSIM: 0.879

c40
PSNR: 57.47
SSIM: 0.99



c85
PSNR: 34.086
SSIM: 0.842



c90
PSNR: 32.987
SSIM: 0.638

c95
PSNR: 30.277
SSIM: 0.603

EfficientNet-B4, main attention method

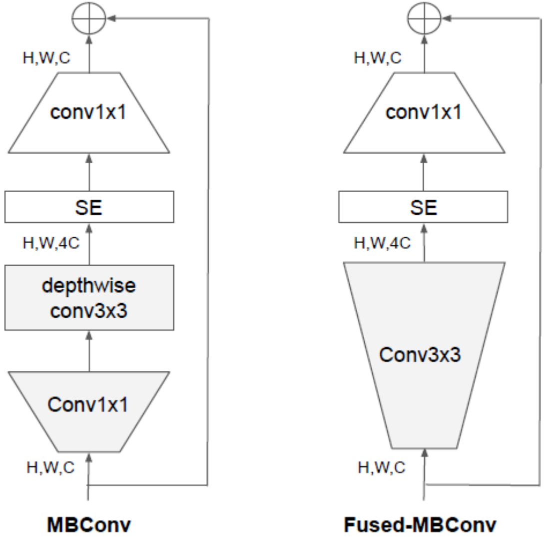


Figure 2. Structure of MBConv and Fused-MBConv.

[1] Tan, M., & Le, Q. V. (2021). *EfficientNetV2: Smaller Models and Faster Training* (arXiv:2104.00298). arXiv. http://arxiv.org/abs/2104.00298

04

Experiments and Results

Efficiency Evaluation and Results.

Training

Google Colaboratory as main environment:

- Interactive notebook capabilities.
- Google Colab's **L4 GPU** (53 GB RAM, 22.5 GB GPU).
- Limited amount of computer units and allowed runtime.
- Save a checkpoint every 500 iterations, mitigates the lost progress and to save disk space.
- 5 ~ 10 epochs per run.

End results

Compression	Dataset				
	FF-DF	FF-F2F	FF-FS	FF-NT	Avg
c23 & c40	0.987	0.986	0.992	0.953	0.9799
c85 & c90	0.883	0.930	0.916	0.718	0.8623
	After utilizing the methods				
c85 & c90	0.91	0.938	0.94	0.756	0.8864
Improvement	+3%	+0.91%	+2.64%	+5.43%	~ +3%

Compression	Dataset				
	FF-DF	FF-F2F	FF-FS	FF-NT	Avg
All datasets*	0.928	0.897	0.937	0.734	0.874

IMPORTANT CONTRIBUTIONS

1) *Enhanced Deepfake Detection Accuracy*
2) *Robustness Under Compression*
3) *Comprehensive Data Augmentation*
4) *Efficient Training and Checkpointing*

Q&A

1. What is the influence of compression rate on detection accuracy in your results?

End results

Compression	Dataset				
	FF-DF	FF-F2F	FF-FS	FF-NT	Avg
c23 & c40	0.987	0.986	0.992	0.953	0.9799
c85 & c90	0.883	0.930	0.916	0.718	0.8623
After utilizing the methods					
c85 & c90	0.91	0.938	0.94	0.756	0.8864
Improvement	+3%	+0.91%	+2.64%	+5.43%	~ +3%

Compression	Dataset				
	FF-DF	FF-F2F	FF-FS	FF-NT	Avg
All datasets*	0.928	0.897	0.937	0.734	0.874

Please describe how to integrate multi-attentional mechanisms to improve the model robustness? Any tradeoff exists?

EfficientNet-B4, main attention method

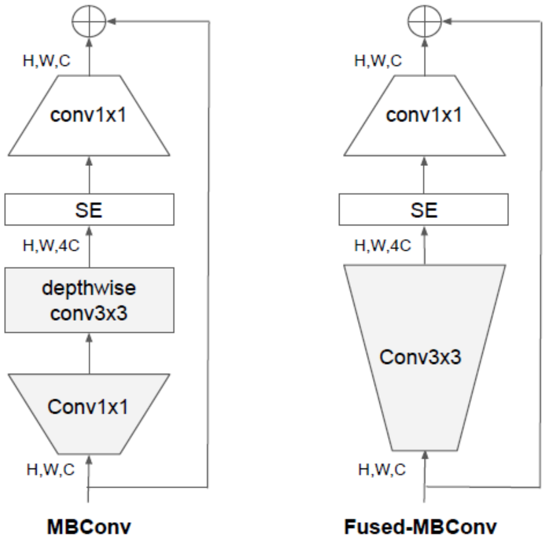


Figure 2. Structure of MBConv and Fused-MBConv.

[1] Tan, M., & Le, Q. V. (2021). *EfficientNetV2: Smaller Models and Faster Training* (arXiv:2104.00298). arXiv. http://arxiv.org/abs/2104.00298

Thanks for your attention

Let’s go to our Q&A

Email: chelogalma@gmail.com