Few Shot Forgery Detection

余振揚 NM6121030 機器人　　徐仁瓏 RE6121011 數據所
梁菁芸 Q36134182 電通所　　鄭翊宏 P76121657 資工所

https://github.com/LittleFish-Coder/few-shot-forgery-detection

I. Introduction

Our goal is to identify whether a video is real or fake. We use three models for transfer learning and observe the effects of finetuning in a few-shot setting. We use two datasets: FaceForensics++ and Celeb-DF (Figure 1). The latter contains 59 different faces and has more realistic Deepfake effects, while the former includes more fake videos that can be identified. All three models have been pretrained on FaceForensics++, and we will use Celeb-DF for finetuning.

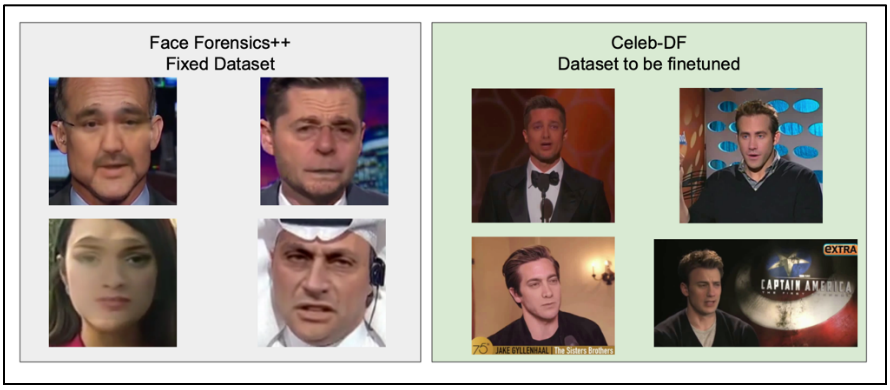


Figure1: Dataset (FaceForensics++&Celeb-DF)

Dataset	# Real		# DeepFake		Release Date
	Video	Frame	Video	Frame	
UADFV	49	17.3k	49	17.3k	2018.11
DF-TIMIT-LQ	320^*	34.0k	320	34.0k	2018.12
DF-TIMIT-HQ			320	34.0k	
FF-DF	1,000	509.9k	1,000	509.9k	2019.01
DFD	363	315.4k	3,068	2,242.7k	2019.09
DFDC	1,131	488.4k	4,113	1,783.3k	2019.10
Celeb-DF	590	225.4k	**5,639**	2,116.8k	2019.11

Table 1. *Basic information of various DeepFake video datasets. *: the original videos in DF-TIMIT are from Vid-TIMIT dataset.*

Table1: Dataset Comparison

II. Methodology

A. Detection Pipeline

First, we perform face detection on the input image to extract the facial region. Next, we crop the image to different sizes based on the requirements of different models. Finally, we classify the cropped images using the classification model to determine whether the video is real or fake.

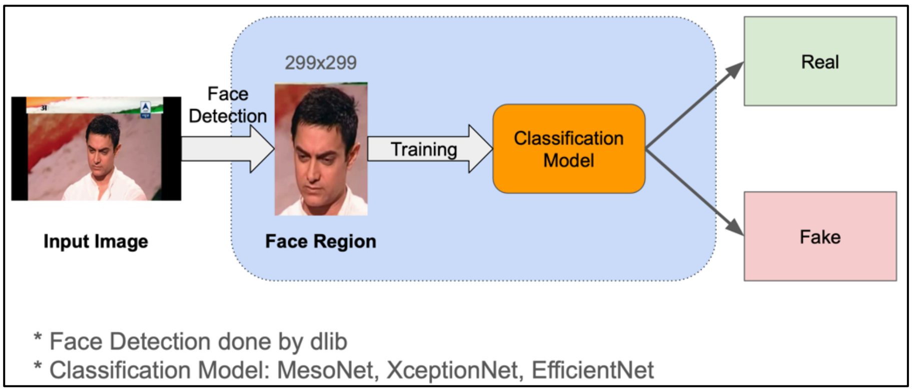


Figure2: Detection Pipeline

B. Proposed Pipeline

For the three models pretrained on FaceForensics++, we will finetune them using the new dataset.

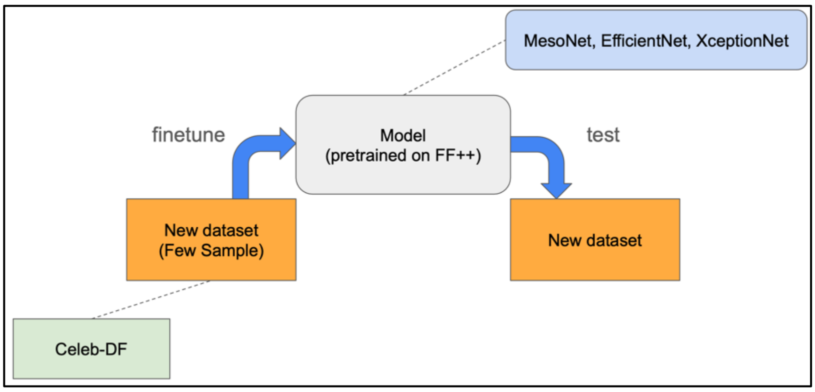


Figure3: Proposed Pipeline

III. Experiment

A. Training Phase

We first split the video into frames of images array, and then label the images with the same label as the video.

We then randomize the order of the images and split the images into training set and validation set.

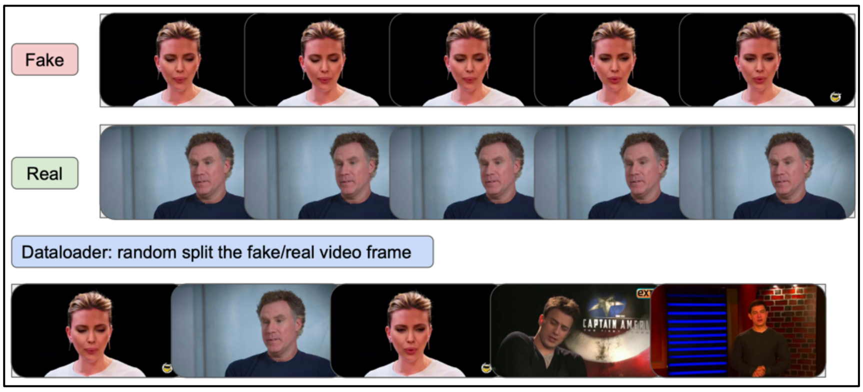


Figure4: Dataloader Design

B. Testing Phase

In the testing phase, we feed the testing video into the model and get the prediction of each frame. We then calculate the average prediction of all frames in the video and use it as the final prediction of the video.

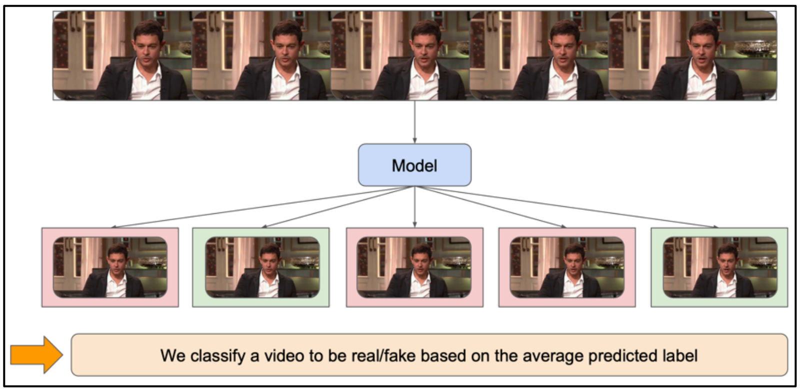


Figure5: Testing phase

*The model will assign a real or fake label to each frame. The label that appears more will be used to determine whether the video is classified as real or fake.

C. Implementation Detail

Model pretrained on FaceForensics++ c23

- models: MesoNet(2018), XceptionNet(2016), EfficientNet(2019)

Finetune 1%, 5%, 10%, 50%, 100% Celeb-DF

- Training set: 1%, 5%, 10%, 50%, 100%
- Validation set: 1%
- Testing set: all Celeb-DF official testing set

Hyperparameter

- Loss: Cross Entropy
- Optimizer: Adam
- Learning rate: 0.001
- Scheduler: StepLR, step size=5
- Epoch

Shot	1%	5%	10%	50%	100%
Epoch	100	100	100	50	30

Table 2: Epochs for different training dataset

IV. Results

A. Model Generalization on Pretrained-Model (zero-shot)

Model	Cross-Dataset	Accuracy	F1-Score	Recall	Precision	AUC
MesoNet	In-dataset	0.884	0.8131	0.8038	0.8237	0.9191
	Cross-dataset	0.8475	0.6748	0.7488	0.6467	0.8273
Xception	In-dataset	0.966	0.9471	0.9488	0.9454	0.9935
	Cross-dataset	0.9133	0.6393	0.6031	0.7713	0.821
EfficientNetB4	In-dataset	0.95	0.9244	0.9388	0.9118	0.9812
	Cross-dataset	0.8989	0.7238	0.7393	0.711	0.8351

B. MesoNet

	Celeb-DF(v2)				
	Accuracy	F1-Score	Recall	Precision	AUC
zero-shot	0.8475	0.6748	0.7488	0.6467	0.8273
1%-shot	0.4719	0.3935	0.5262	0.5091	0.5172
5%-shot	0.1108	0.1052	0.5089	0.5481	0.4471
10%-shot	0.252	0.2484	0.5717	0.5449	0.5747
50%-shot	0.8828	0.5391	0.5331	0.571	0.484
100%-shot	0.9839	0.9493	0.9153	0.9913	0.9998

C. Xception

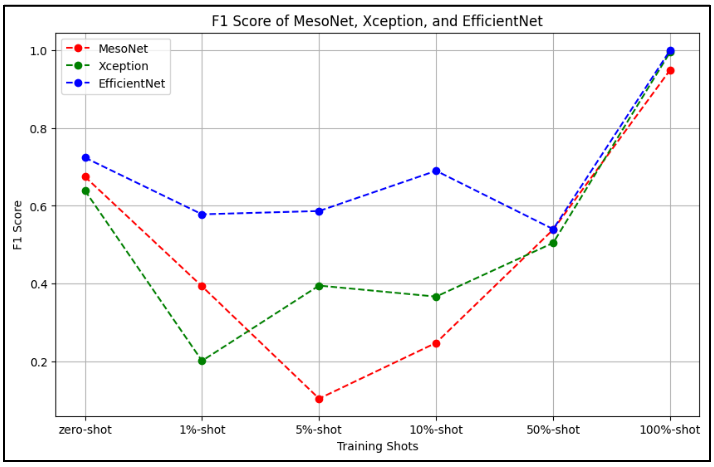
	Celeb-DF(v2)				
	Accuracy	F1-Score	Recall	Precision	AUC
zero-shot	0.9133	0.6393	0.6031	0.7713	0.821
1%-shot	0.2022	0.2021	0.5594	0.5531	0.7942
5%-shot	0.4318	0.3953	0.6786	0.5682	0.9134
10%-shot	0.3933	0.367	0.6649	0.5675	0.9522
50%-shot	0.9021	0.5049	0.5134	0.6205	0.7084
100%-shot	0.9984	0.9953	0.9915	0.9991	1

D. EfficientNetB4

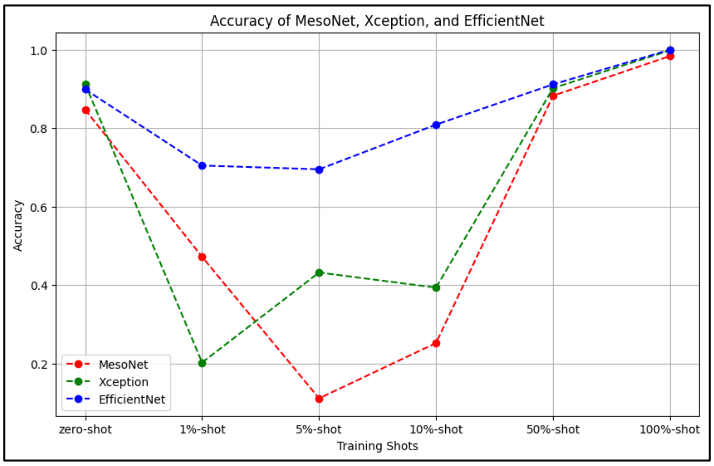
	Celeb-DF(v2)				
	Accuracy	F1-Score	Recall	Precision	AUC
zero-shot	0.8989	0.7238	0.7393	0.711	0.8351
1%-shot	0.7047	0.5783	0.761	0.5974	0.8545
5%-shot	0.695	0.5867	0.8164	0.6137	0.9614
10%-shot	0.809	0.69	0.8945	0.6657	0.9983
50%-shot	0.9117	0.5402	0.5339	0.9556	0.8689
100%-shot	1	1	1	1	1

E. Model Comparison on Different Metrics

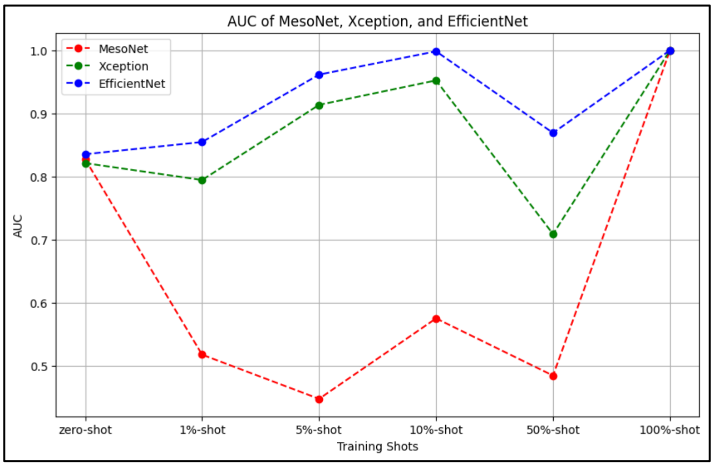
F1-Score



Accuracy



AUC



V. Conclusion

EfficientNet outperforms the other two models when testing few-shot samples.

Our assumption is that since the difference between real and fake images is small, it's very hard to distinguish them by using simple convolutional feature extraction method.

Also, we think Residual component plays an important role in this task.

VI. Reference

[1] dlib: https://github.com/davisking/dlib

[2] MesoNet: https://github.com/DariusAf/MesoNet

[3] XceptionNet: https://medium.com/ching-i/inception-系列-xception-fd2a4a4e7e82

[4] EfficientNet: https://medium.com/ching-i/efficientnet-論文閱讀-e828ac005ce8

Fail to apply on this project

[5] MTCNN: https://github.com/ipazc/mtcnn

[6] ViT (MARLIN): https://github.com/ControlNet/MARLIN

[7] DFDC Dataset: https://ai.meta.com/datasets/dfdc/

VII. GitHub

https://github.com/LittleFish-Coder/few-shot-forgery-detection