Deep Learning Final Project

Few Shot Forgery Detection

Group 1
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

Outline

1. Goal
2. Dataset
3. Detection Pipeline
4. Proposed Pipeline
5. Training Phase
6. Testing Phase
7. Implementation Detail
8. Results
9. Conclusion

Goal

1. Classify if a video is real or fake

2. Disscuss transfer learning performance on few-shot sample finetuning

Dataset (FaceForensics++ & Celeb-DF)

Face Forensics++
Fixed Dataset




Celeb-DF
Dataset to be finetuned

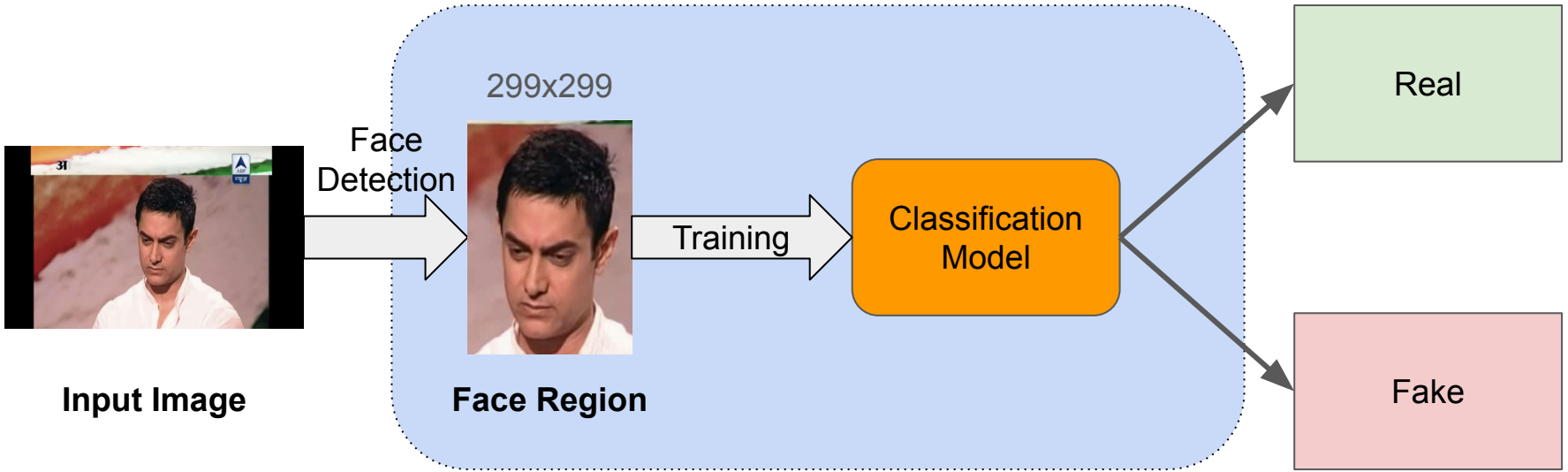


Dataset Comparison

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

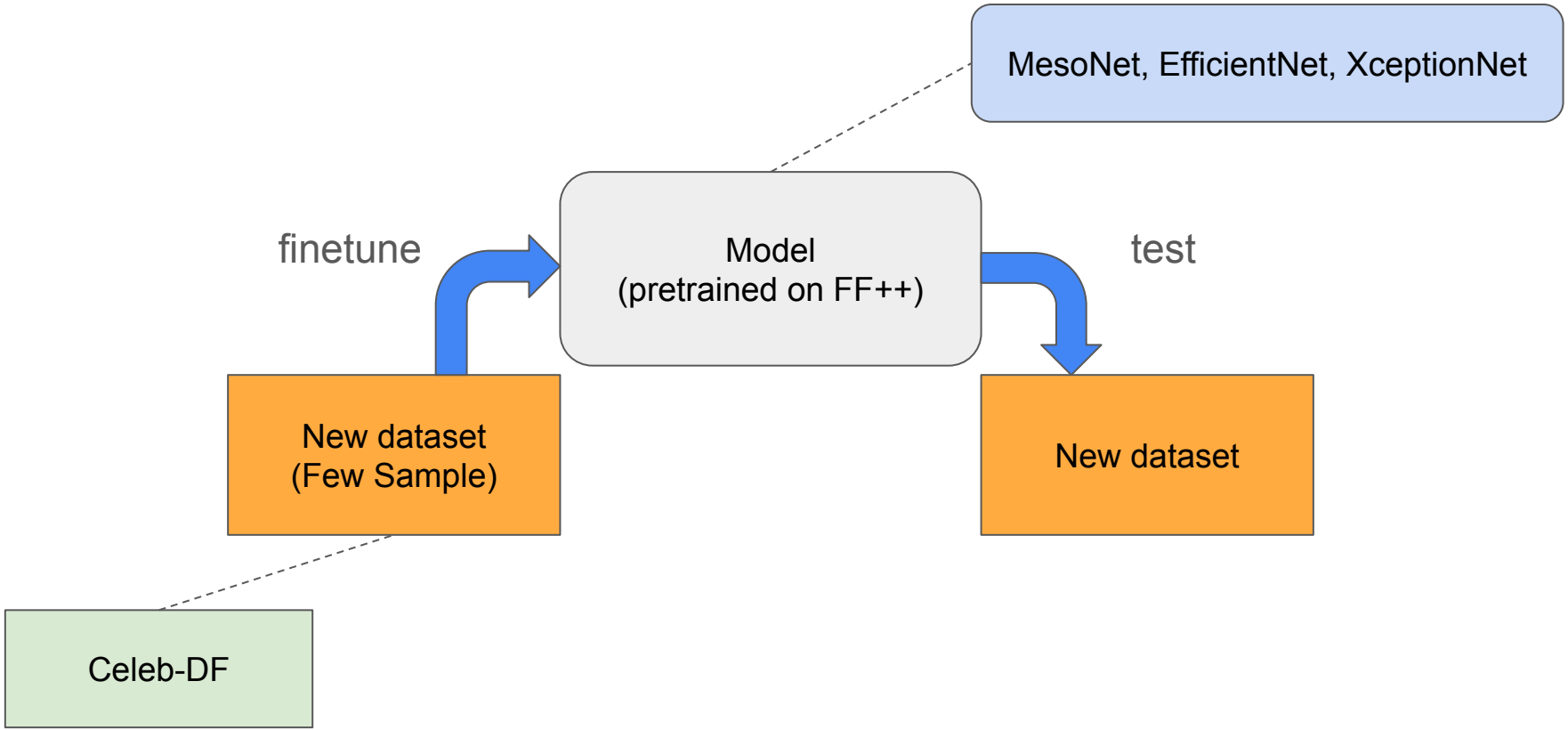
Table 1. *Basic information of various DeepFake video datasets. ∗: the original videos in DF-TIMIT are from Vid-TIMIT dataset.*

Detection Pipeline



* Face Detection done by dlib
* Classification Model: MesoNet, XceptionNet, EfficientNet

Proposed Pipeline

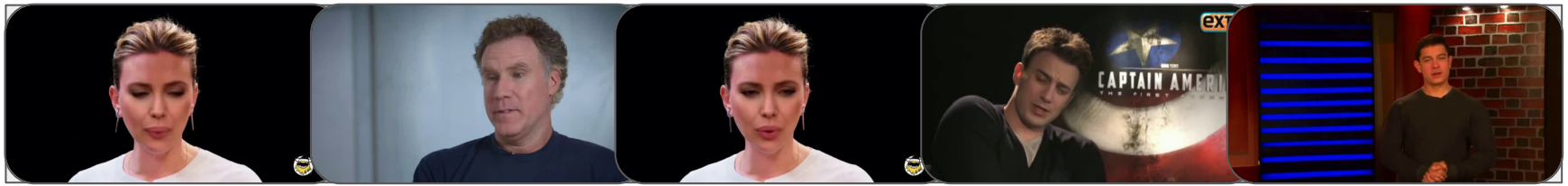


Training Phase

Fake

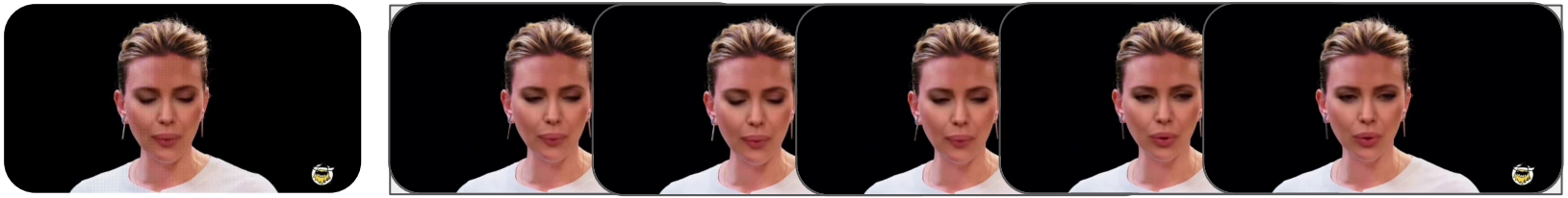
Real

Dataloader: random split the fake/real video frame



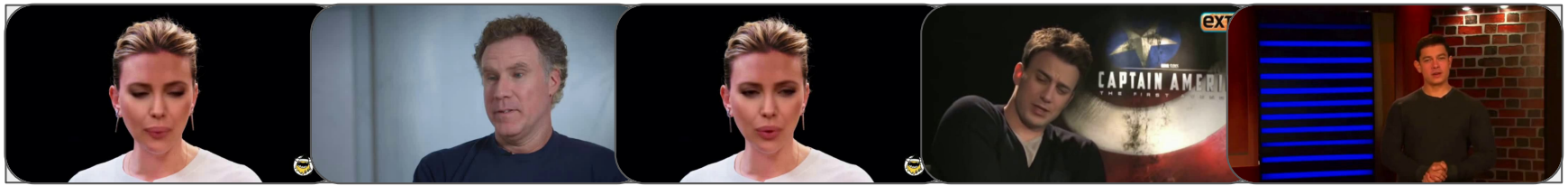
Training Phase

Fake

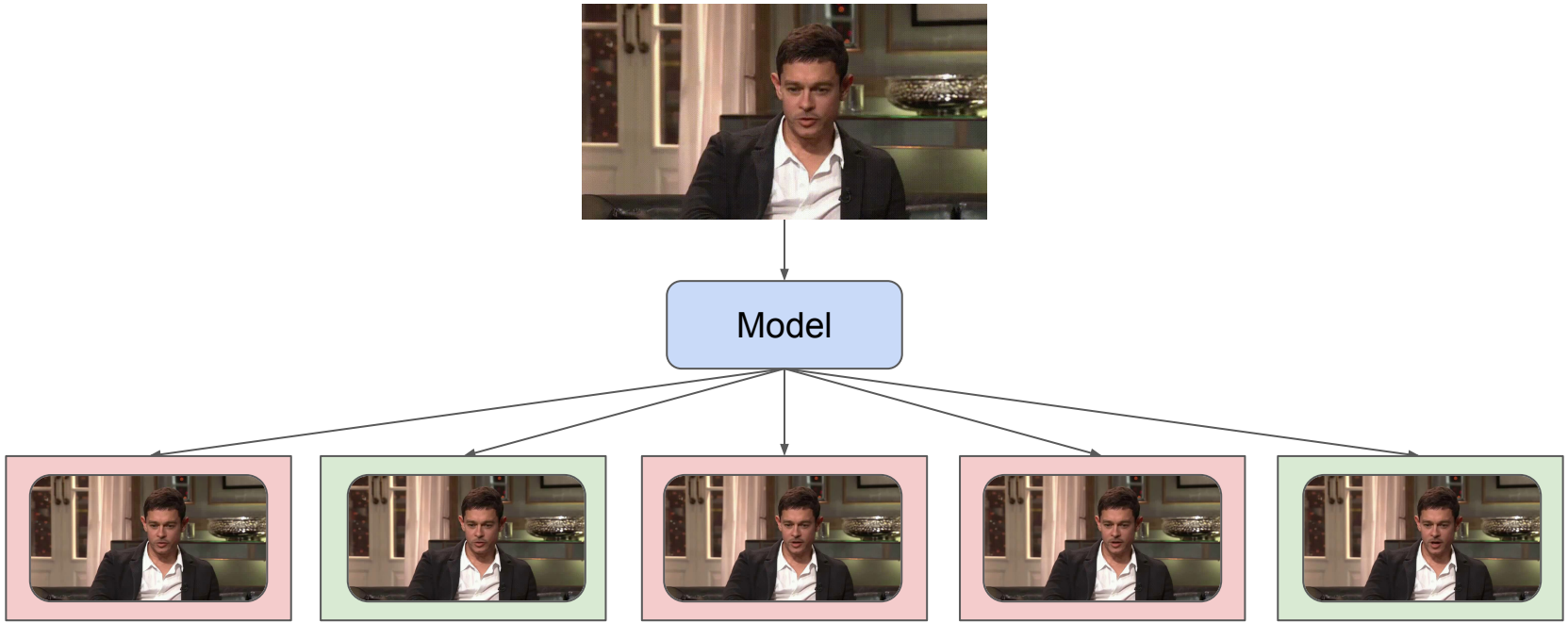


Real

Dataloader: random split the fake/real video frame



Testing Phase



We classify a video to be real/fake based on the average predicted label

Testing Phase

Model

We classify a video to be real/fake based on the average predicted label

Implementation Detail

model pretrained on face forensics ++ 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

	1%-shot	5%-shot	10%-shot	50%-shot	100%-shot
Epoch	100	100	100	50	30

Result - MesoNet

In-dataset

FF++(c23)				
Accuracy	F1-Score	Recall	Precision	AUC
0.884	0.8131	0.8038	0.8237	0.9191

Cross-dataset

Celeb-DF(v2)				
Accuracy	F1-Score	Recall	Precision	AUC
0.8475	0.6748	0.7488	0.6467	0.8273

Result - 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

Result - Xception

In-dataset

FF++(c23)				
Accuracy	F1-Score	Recall	Precision	AUC
0.966	0.9471	0.9488	0.9454	0.9935

Cross-dataset

Celeb-DF(v2)				
Accuracy	F1-Score	Recall	Precision	AUC
0.9133	0.6393	0.6031	0.7713	0.821

Result - 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

Result - EfficientNetB4

In-dataset

FF++(c23)				
Accuracy	F1-Score	Recall	Precision	AUC
0.95	0.9244	0.9388	0.9118	0.9812

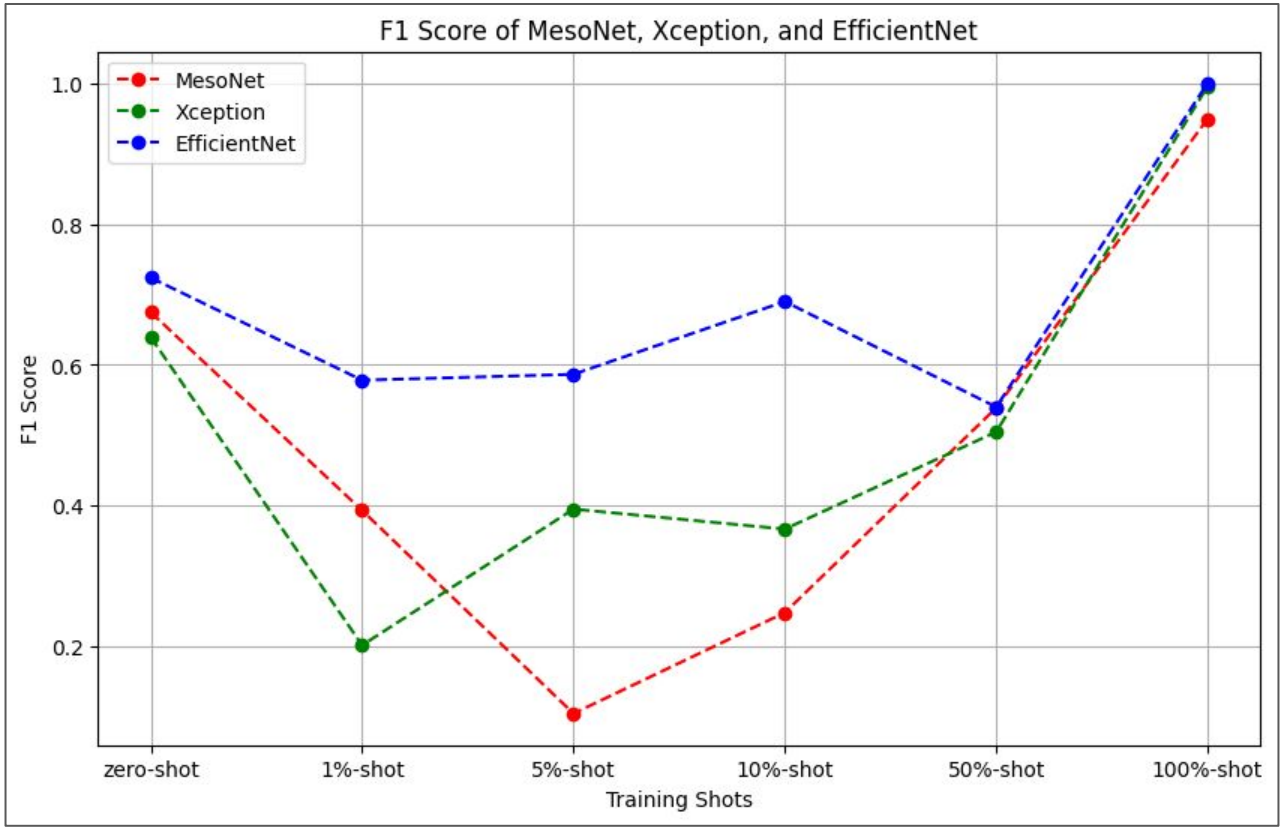
Cross-dataset

Celeb-DF(v2)				
Accuracy	F1-Score	Recall	Precision	AUC
0.8989	0.7238	0.7393	0.711	0.8351

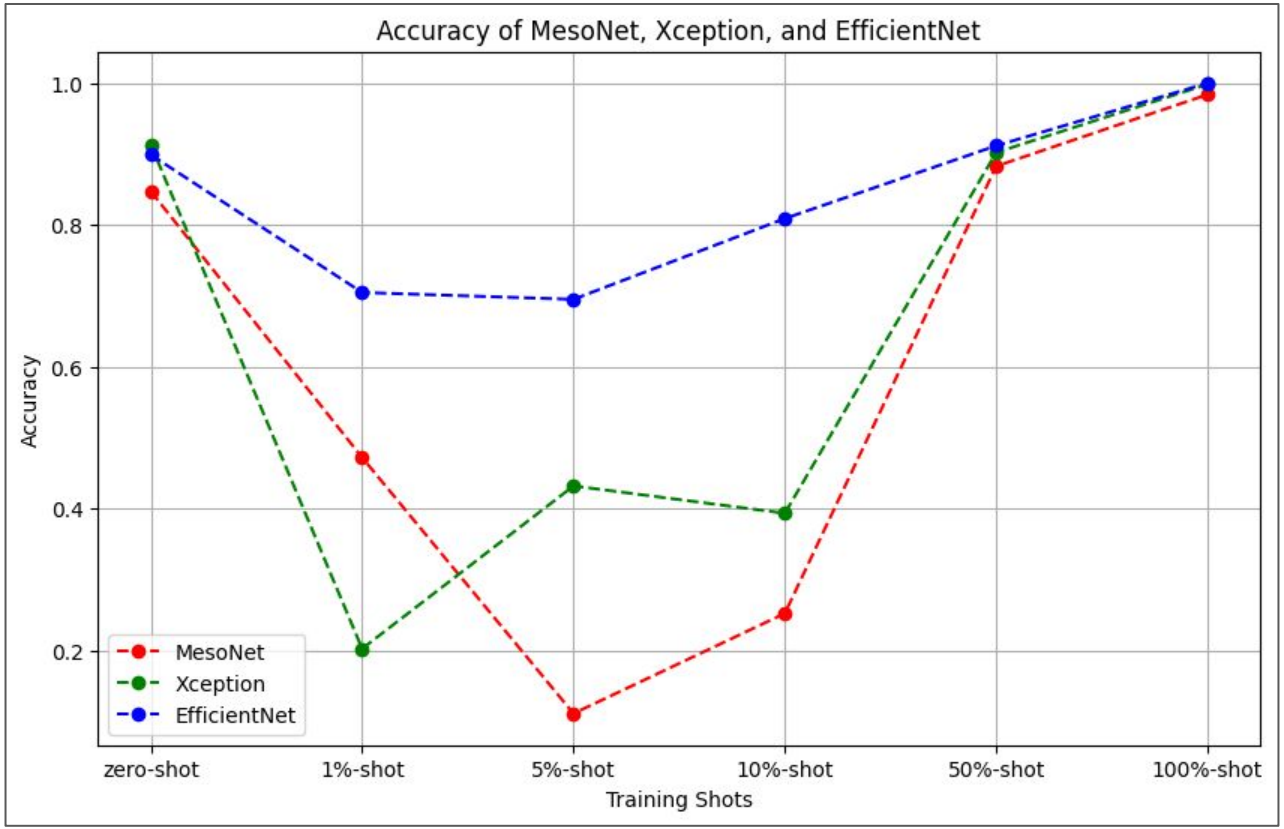
Result - EfficientNetB4

	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

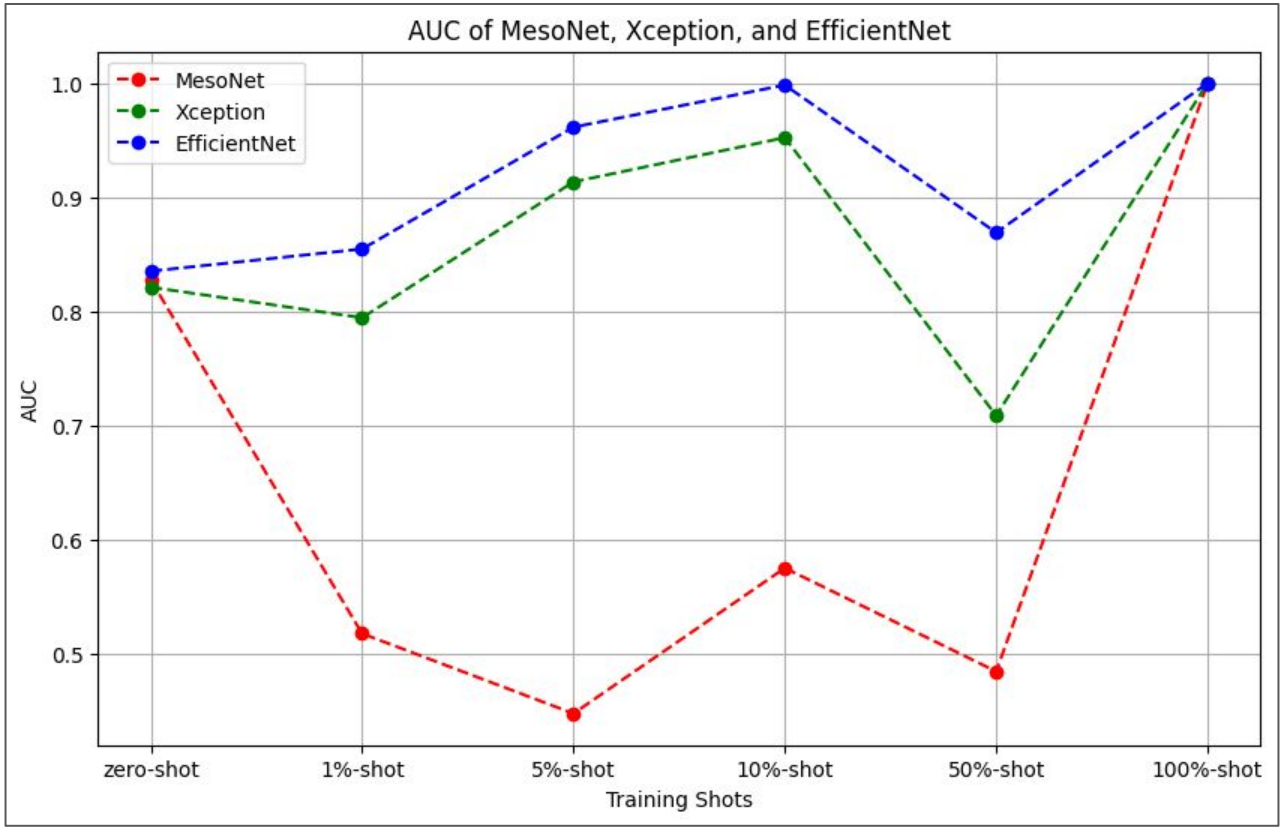
Model Comparison on F1-Score



Model Comparison on Accuracy



Model Comparison on AUC



Conclusion

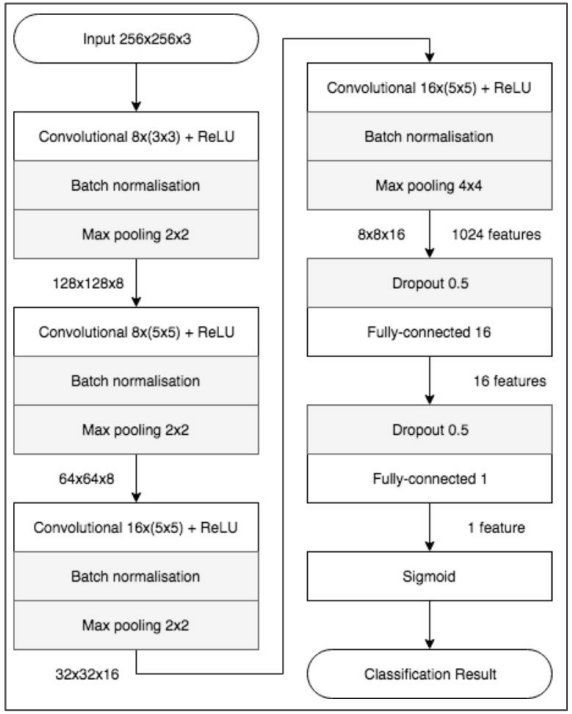
EfficientNet outperforms the other two models when testing few-shot samples

Guess:

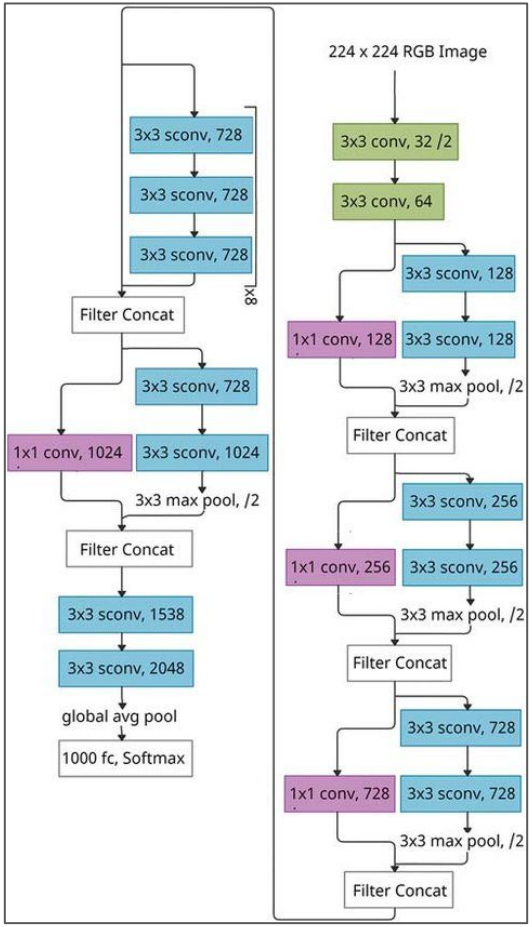
Since the difference between real and fake images is really small, it’s very hard to distinguish them by using simple convolutional feature extraction method.

We think Residual component plays an important role in this task

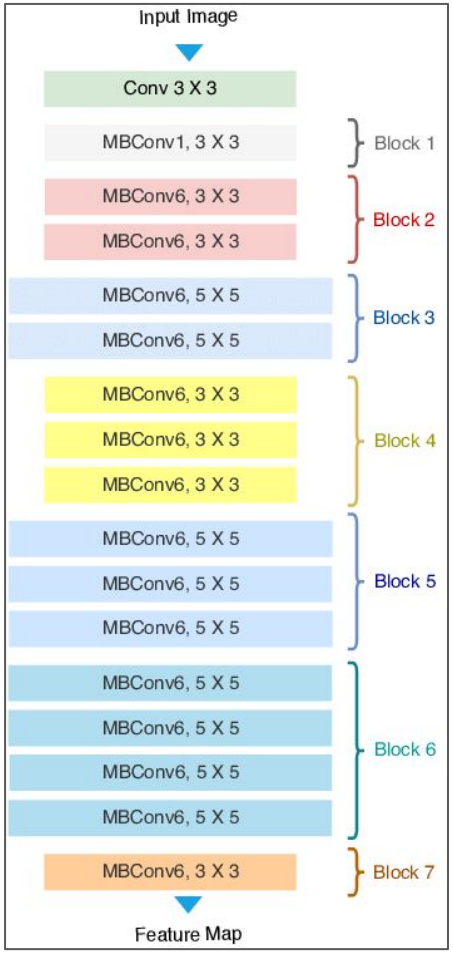
Model Comparison



MesoNet



XceptionNet



EfficientNet

Reference

dlib: https://github.com/davisking/dlib

MesoNet: https://github.com/DariusAf/MesoNet

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

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

Fail to apply on this project

MTCNN: https://github.com/ipazc/mtcnn

ViT (MARLIN): https://github.com/ControlNet/MARLIN

DFDC Dataset: https://ai.meta.com/datasets/dfdc/

Thanks !