**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>

1. **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)

**一張含有 文字, 字型, 螢幕擷取畫面, 數字 的圖片

自動產生的描述**

Table1: Dataset Comparison

1. **Methodology**
2. **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.

一張含有 文字, 人的臉孔, 螢幕擷取畫面, 人員 的圖片

自動產生的描述

Figure2: Detection Pipeline

1. **Proposed Pipeline**

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

一張含有 文字, 螢幕擷取畫面, 字型, 行 的圖片

自動產生的描述

Figure3: Proposed Pipeline

1. **Experiment**
2. **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.

**一張含有 人的臉孔, 螢幕擷取畫面, 男人, 微笑 的圖片

自動產生的描述**

Figure4: Dataloader Design

1. **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.

**一張含有 人員, 人的臉孔, 螢幕擷取畫面, 服裝 的圖片

自動產生的描述**

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.

1. **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

1. **Results**
2. **Model Generalization on Pretrained-Model (zero-shot)**

**一張含有 文字, 螢幕擷取畫面, 數字, 字型 的圖片

自動產生的描述**

1. **MesoNet**

**一張含有 文字, 字型, 數字, 螢幕擷取畫面 的圖片

自動產生的描述**

1. **Xception一張含有 文字, 字型, 螢幕擷取畫面, 數字 的圖片

   自動產生的描述**
2. **EfficientNetB4**

**一張含有 文字, 字型, 數字, 螢幕擷取畫面 的圖片

自動產生的描述**

1. **Model Comparison on Different Metrics**

**F1-Score**

**一張含有 文字, 行, 繪圖, 圖表 的圖片

自動產生的描述**

**Accuracy**

**一張含有 行, 文字, 繪圖, 圖表 的圖片

自動產生的描述**

**AUC**

**一張含有 行, 圖表, 繪圖, 文字 的圖片

自動產生的描述**

1. **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.

1. **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](https://medium.com/ching-i/inception-%E7%B3%BB%E5%88%97-xception-fd2a4a4e7e82)

[4] EfficientNet: [https://medium.com/ching-i/efficientnet-論文閱讀-e828ac005ce8](https://medium.com/ching-i/efficientnet-%E8%AB%96%E6%96%87%E9%96%B1%E8%AE%80-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/>

1. **GitHub**

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