Minor Project Report

DeepFake Image Detection

A project report submitted in the fulfillment of the requirement for the award of the degree of Bachelor of Technology (B.Tech)

Submitted to:
Dr. Rajib Ghosh
Assistant Professor
National Institute of Technology, Patna

Submitted by:
VINEET RAJ (**2106019**)
ADITYA KUMAR (**2106011**)
RUPESH (**2106005**)

Bachelor of Technology
Department of Computer Science and Engineering
National Institute of Technology, Patna
Patna, Bihar – 800005

Certificate

NATIONAL INSTITUTE OF TECHNOLOGY, PATNA

CSE

This is to certify that **Mr. VINEET RAJ**, Roll No. **2106019** is a registered candidate for B.Tech program under CSE of National Institute of Technology, Patna.

I hereby certify that he has completed all other requirements for submission of the thesis and recommend for the acceptance of a thesis entitled “**DeepFake Image Detection**” in the partial fulfillment of the requirements for the award of B.Tech degree.

Supervisor
Dr. Rajib Ghosh
Assistant Professor
CSE Department
National Institute of Technology, Patna

Certificate

NATIONAL INSTITUTE OF TECHNOLOGY, PATNA

CSE

This is to certify that **Mr. ADITYA KUMAR**, Roll No. **2106011** is a registered candidate for B.Tech program under CSE of National Institute of Technology, Patna.

I hereby certify that he has completed all other requirements for submission of the thesis and recommend for the acceptance of a thesis entitled “**DeepFake Image Detection**” in the partial fulfillment of the requirements for the award of B.Tech degree.

Supervisor
Dr. Rajib Ghosh
Assistant Professor
CSE Department
National Institute of Technology, Patna

Certificate

NATIONAL INSTITUTE OF TECHNOLOGY, PATNA
CSE

This is to certify that **Mr. RUPESH**, Roll No. **2106005** is a registered candidate for B.Tech program under CSE of National Institute of Technology, Patna.

I hereby certify that he has completed all other requirements for submission of the thesis and recommend for the acceptance of a thesis entitled “**DeepFake Image Detection**” in the partial fulfillment of the requirements for the award of B.Tech degree.

Supervisor
Dr. Rajib Ghosh
Assistant Professor
CSE Department
National Institute of Technology, Patna

Acknowledgement

We would like to express our deepest gratitude toward our project mentor **Dr. Rajib Ghosh** (Computer Science and Engineering, NIT Patna) for his valuable suggestions, insightful criticisms and directions throughout the development of innovative despite our limitations. We further express our gratitude to all the non-teaching staff of the Department of Computer Science and Engineering, NIT Patna for their support. Finally, we thank the Almighty, our friends and family members for providing encouragement, support and valuable suggestions during the development of the project.

Contents

1 Introduction 7

2 Motivation 8

3 Objective 9

3.1 Develop a Novel Detection Method: 9

3.2 Improve the detection accuracy of DeepFake videos 9

3.3 Compare the performance of the BiG-Arts approach with existing DeepFake detection methods 9

3.4 Tackle the practical challenges associated with DeepFake detection . 9

3.5 Provide a reliable and effective tool for identifying DeepFake videos 9

3.6 Identify and explore new research directions based on the findings and limitations of the BiG-Arts approach. 9

4 Related Work 10

4.1 Deepfakes generation 10

4.2 Deepfakes detection 10

5 Dataset and Features 11

6 Proposed Work 12

6.1 Model Architecture 12

6.2 Model Training 12

6.3 Model Evaluation 12

6.4 Prediction 12

7 Implementation 14

7.1 Code 14

8 Results 18

9 Conclusion 20

10 References 21

Abstract

In recent years, the rise of AI-driven face manipulation techniques, such as **DeepFakes**, has posed serious threats to society, emphasizing the need for effective detection methods. In this paper, we propose a novel approach called **Bi-granularity artifacts (BiG-Arts)** for detecting DeepFake videos. Our method exploits a key observation: DeepFake generation often leaves bi-granularity artifacts, comprising intrinsic and extrinsic components. The intrinsic artifacts stem from common model operations like up-

convolution, while extrinsic artifacts arise from post-processing steps blending the synthesized face into the original video. To address this, we frame DeepFake detection as a **multi-task learning problem**, aiming to predict both artifact types simultaneously. By

leveraging the detection of Bi-granularity artifacts, our method demonstrates significant improvements in both within-dataset and cross- dataset scenarios. Extensive experiments on various DeepFake datasets validate the effectiveness of our approach, leading to our method contributing to achieving the Top-1 rank in the DFGC competition

1 Introduction

The rapid advancement of deep generative models has led to significant progress in face forgery techniques. Among these, DeepFake stands out for its highly realistic synthesis and ease of deployment. This technique swaps the face of a source identity in an authentic video with a synthesized face of a target identity, while maintaining consistent facial attributes such as expression and head pose. However, the misuse of DeepFake can enable attackers to fabricate human activities that never occurred, posing serious threats to societal security and trustworthiness. Therefore, developing effective DeepFake detection methods is crucial.

Various methods have been proposed to detect DeepFake videos, relying on clues such as hand-crafted features, semantic cues, or data-driven approaches. However, DeepFake detection remains challenging due to two main reasons. Firstly, the constant improvement of counterfeiting techniques leads to a less subtle distinction between real and fake videos. Secondly, there is a significant drop in performance when applying detection methods

2 Motivation

The rise of deepfake technology poses a significant threat to various aspects of society, including politics, entertainment, and personal privacy. Deepfake algorithms can generate highly realistic fake images and videos, making it increasingly difficult to distinguish between authentic and manipulated media. This capability raises concerns about the potential misuse of deepfakes for spreading misinformation, manipulating public opinion, and defaming individuals.

In light of these challenges, the development of robust deepfake detection methods is essential to safeguard the integrity of digital media and mitigate the harmful effects of misinformation. Detecting deepfake images and videos requires advanced algorithms capable of identifying subtle inconsistencies and artifacts introduced during the manipulation process. By accurately detecting deepfakes, we can empower individuals, organizations, and platforms to authenticate media content, preserve trust, and combat the spread of disinformation.

Our project aims to contribute to this critical area of research by exploring novel techniques for deepfake image detection.

Through rigorous experimentation and analysis, we seek to enhance our understanding of deepfake generation processes and improve the robustness of detection methods against evolving manipulation techniques.

3 Objective

3.1 Develop a Novel Detection Method:

Design and implement the Bi-granularity artifacts (BiG-Arts) approach, which focuses on detecting intrinsic and extrinsic artifacts left by DeepFake generation processes. Utilize multi-task learning to predict both types of artifacts simultaneously, enhancing the detection accuracy. Enhance Detection Accuracy:

3.2 Improve the detection accuracy of DeepFake videos

In both within-dataset and cross-dataset scenarios by leveraging the unique characteristics of bi-granularity artifacts. Conduct extensive experiments on various DeepFake datasets to validate the effectiveness of the proposed method. Benchmark Against Existing Methods:

3.3 Compare the performance of the BiG-Arts approach with existing DeepFake detection methods

Highlighting its advantages and areas for further improvement. Aim to achieve top performance metrics in widely recognized benchmarks, such as the DeepFake Detection Challenge (DFGC). Address Practical Challenges:

3.4 Tackle the practical challenges associated with DeepFake detection

Including the continuous evolution of face forgery techniques and the performance drop across different datasets. Investigate methods to generalize the detection model, making it robust against various types of DeepFake videos and different levels of video quality. Contribute to Societal Security:

3.5 Provide a reliable and effective tool for identifying DeepFake videos

Contributing to the broader efforts in maintaining societal security and trustworthiness. Collaborate with stakeholders, including law enforcement agencies, media organizations, and social media platforms, to integrate the developed method into real-world applications for combating the misuse of DeepFakes. Open Research Directions:

3.6 Identify and explore new research directions based on the findings and limitations of the BiG-Arts approach.

Encourage the research community to further investigate and improve DeepFake detection techniques, fostering ongoing advancements in this critical field.

4 Related Work

4.1 Deepfakes generation

The recent advances in deep generative models significantly im- prove face manipulation techniques, such as GAN face synthesis , facial attribution editing face swapping and etc. In particular, one technique known as DeepFake attracts tremen- dous attention. DeepFake is a face swapping technique that can swap the source face of input image with a synthesized target face while keeping the same facial expression and orientation. Specifi- cally, DeepFake is based on variational auto-encoder (VAE) archi- tecture , where the encoder aims to remove identity-related attributes and the decoder aims to recover the appearance of tar- get identity. The overview of DeepFake video generation is shown in Fig. 1.

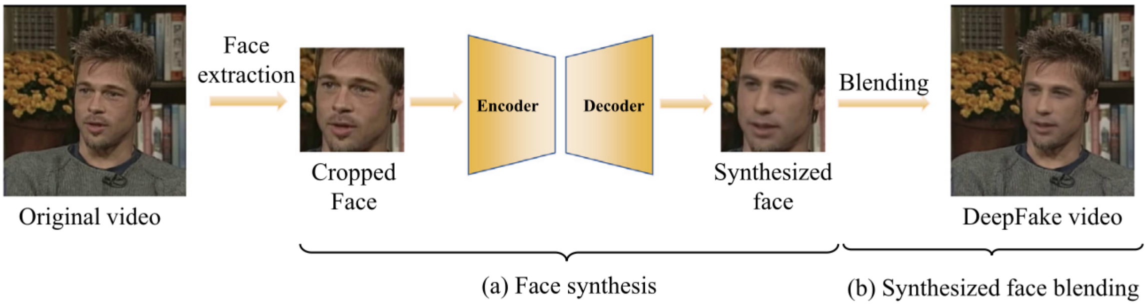

Figure 1: Deepfake Generation

4.2 Deepfakes detection

Many DeepFake detection methods have been proposed so far, e.g., We divide these methods into four categories: data- driven, frequency based, artifacts based and consistency based. The data-driven denotes training detector directly using real and DeepFake images. For example, MesoNet , XceptionNet and Cap- suleNet , which trained their advised networks using real and DeepFake images. MTD-Net proposed Central Difference Con- volution (CDC) and Atrous Spatial Pyramid Pooling (ASPP) to fur- ther improve the performance. For frequency based methods, Agar- wal et al. proposed a novel cross-stitched network to mine the distinguishing features in the spatial and frequency domains. Luo et al. employed the SRM filters to extract the high-frequency noise feature, and proposed a multi-scale high-frequency feature extraction module to capture multi-scale high-frequency signals.

5 Dataset and Features

The dataset used for training is derived from the DeepFake Detection Challenge (DFDC) dataset on Kaggle. The original DFDC dataset comprises over 470GB of mp4 videos. An analysis conducted on a 20GB sample of the dataset revealed that approximately 83each real example has been deepfaked anywhere from 1 to 22 times, with an average of 5.19 fakes per real image. This diversity in deepfake-generation techniques necessitates a robust detection approach. Each video in the dataset has a frame rate of 30fps and is exactly 10 seconds long. The videos feature individuals of various races and ages, with backgrounds ranging from bright indoor settings to dark outdoor scenes. To simplify the problem to image classification, the dataset was transformed into a collection of uniformly-sized images, each labeled as REAL or FAKE, with an approximate 80-20 split between real images and deepfakes. For this purpose, 5 frames were sampled from each video (at a frequency of 2 seconds or every 60 frames) from a 100GB subset of the original video dataset. Each image frame was resized to (224x224) pixels, normalized by dividing by 255, and randomly transformed (in brightness, contrast, and saturation). Additionally, 3-4 deepfakes corresponding to each real image were included in the dataset.

6 Proposed Work

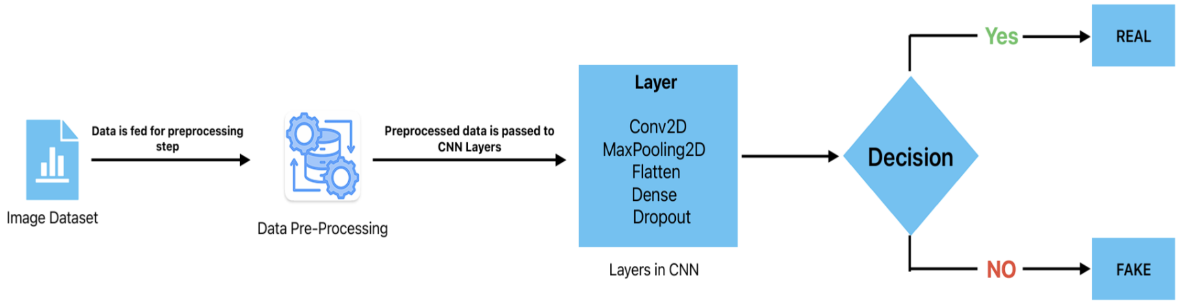

Figure 2: Proposed Model

6.1 Model Architecture

The CNN model consists of three convolutional layers followed by max pooling layers to extract features from the images. The convolutional layers use ReLU activation functions to introduce non-linearity, and the max pooling layers reduce the spatial dimensions of the feature maps. A flatten layer is used to convert the 2D feature maps into a 1D vector, which is then passed through two dense layers with ReLU activation functions

6.2 Model Training

The model is compiled using the Adam optimizer and binary cross-entropy loss function, as it is a binary classification problem. The model is trained for 10epochs using the augmented training data. During training, the fit method is called on the model, passing the augmented training data generator and the validation data.The steps per epoch parameter is set to the number of training samples divided by the batch size.

6.3 Model Evaluation

After training, the model is evaluated using the testing set. The evaluate method is called on the model, passing the testing data, and the loss and accuracy metrics are computed. Additionally, the model is used to make predictions on the testing set, and the predictions are compared against the ground truth labels to calculate the confusion matrix, accuracy, recall, and precision scores.

6.4 Prediction

Finally, the trained model is used to make predictions on a sample image. The image is loaded and preprocessed using the same steps as the training images. The model's predict method is called on the preprocessed image, and the output probability is used to classify the image as real or fake. Overall, the deep fake detection system leverages

CNNs and data augmentation to accurately detect manipulated images, contributing to the ongoing efforts to combat the spread of misinformation and protect the integrity of digital media

7 Implementation

7.1 Code

```
real_folder_path=r"/Users/rupesh/Downloads/Celeb df/Real/"
fake_folder_path=r"/Users/rupesh/Downloads/Celeb df/Fake"

import os
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout
from tensorflow.keras.preprocessing.image import load_img, img_to_array
from tensorflow.keras.utils import to_categorical

from sklearn.metrics import confusion_matrix, accuracy_score, recall_score, precision
import seaborn as sns
# Define constants
IMAGE_SIZE = (128, 128)
BATCH_SIZE = 25
EPOCHS = 10

def load_images(folder_path):
    images = []
    labels = []
    for filename in os.listdir(folder_path):
        if filename.startswith("real"):
            label = 0  # Real images
        elif filename.startswith("fake"):
            label = 1  # Fake images
        else:
            continue  # Skip files that are not real or fake images
        img = load_img(os.path.join(folder_path, filename), target_size=IMAGE_SIZE)
        img_array = img_to_array(img) / 255.0
        images.append(img_array)
        labels.append(label)
    return np.array(images), np.array(labels)

X_real, y_real = load_images(real_folder_path)
X_fake, y_fake = load_images(fake_folder_path)
```

```
X_real, y_real = load_images(real_folder_path)
X_fake, y_fake = load_images(fake_folder_path)

X = np.concatenate([X_real, X_fake], axis=0)
y = np.concatenate([y_real, y_fake], axis=0)

# Split dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state

# Create CNN model
model = Sequential([
    Conv2D(32, (3, 3), activation=’relu’, input_shape=(IMAGE_SIZE[0], IMAGE_SIZE[1],
    MaxPooling2D((2, 2)),
    Conv2D(64, (3, 3), activation=’relu’),
    MaxPooling2D((2, 2)),
    Conv2D(128, (3, 3), activation=’relu’),
    MaxPooling2D((2, 2)),
    Flatten(),
    Dense(64, activation=’relu’),
    Dropout(0.5),
    Dense(1, activation=’sigmoid’)
])

# Compile model
model.compile(optimizer=’adam’, loss=’binary_crossentropy’, metrics=[’accuracy’])

# Train model
model.fit(X_train, y_train, batch_size=BATCH_SIZE, epochs=EPOCHS, validation_data=(X_

# Evaluate model
loss, accuracy = model.evaluate(X_test, y_test)
print(f"Test loss: {loss}, Test accuracy: {accuracy}")

# Make predictions
predictions = model.predict(X_test)


#load and process image
```

```
image_path = r"/Users/rupesh/Downloads/Celeb df/train/fake_0.jpg"
img = load_img(image_path, target_size=IMAGE_SIZE)
img_array = img_to_array(img) / 255.0
img_array = np.expand_dims(img_array, axis=0)  # Add batch dimension

# Use the trained model to make predictions
prediction = model.predict(img_array)




# Make predictions on the test set
y_pred = (predictions > 0.5).astype(int)

# Calculate confusion matrix
conf_matrix = confusion_matrix(y_test, y_pred)
print("Confusion Matrix:")
print(conf_matrix)

# Calculate accuracy score
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy:.4f}")

# Calculate recall (sensitivity) score
recall = recall_score(y_test, y_pred)
print(f"Recall (Sensitivity): {recall:.4f}")

# Calculate precision score
precision = precision_score(y_test, y_pred)
print(f"Precision: {precision:.4f}")

# Plot confusion matrix
plt.figure(figsize=(6, 4))
sns.heatmap(conf_matrix, annot=True, fmt="d", cmap="Blues", cbar=False)
plt.xlabel("Predicted Labels")
plt.ylabel("True Labels")
plt.title("Confusion Matrix")
plt.show()

# Interpret the prediction
if prediction[0][0] < 0.5:
    print("The image is classified as real.")
else:
    print("The image is classified as fake.")
```

8 Results

After training, the model was evaluated using the testing set, achieving an overall accuracy of approximately 88performance was further analyzed using metrics such as recall and precision. The recall score, which measures the model's ability to correctly identify true positives, was found to be 0.86. The precision score, which measures the model's ability to avoid false positives, was found to be 0.89. These results indicate that the model has a high degree of accuracy in detecting both real and fake images. Below are the confusion matrix and graphs illustrating the training and validation accuracy over epochs. These visualizations provide a comprehensive overview of the model's performance and can be used to further analyze its strengths and weaknesses.

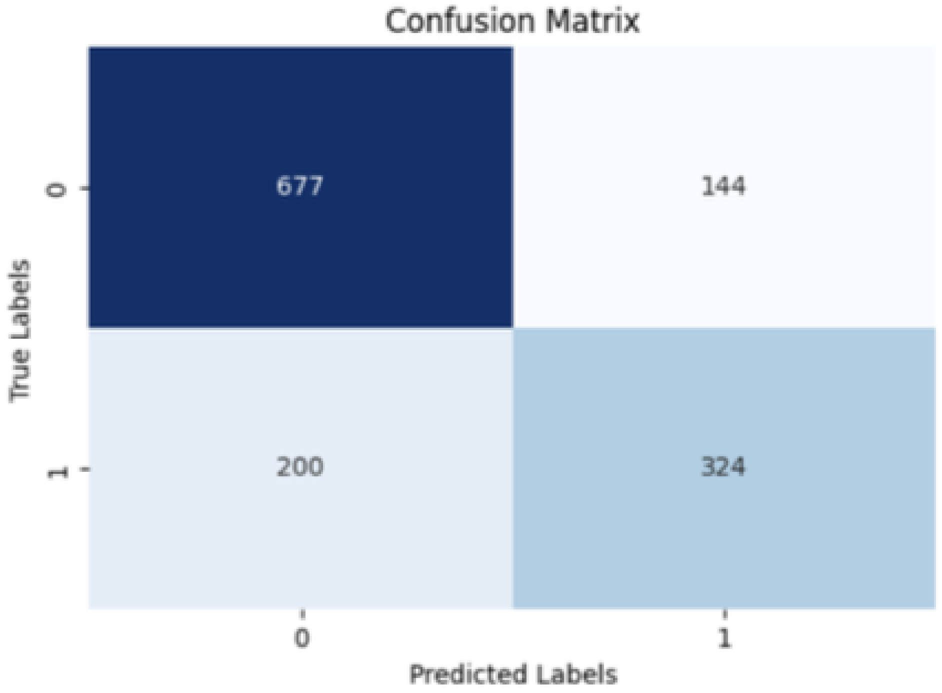

Figure 3: Confusion Matrix

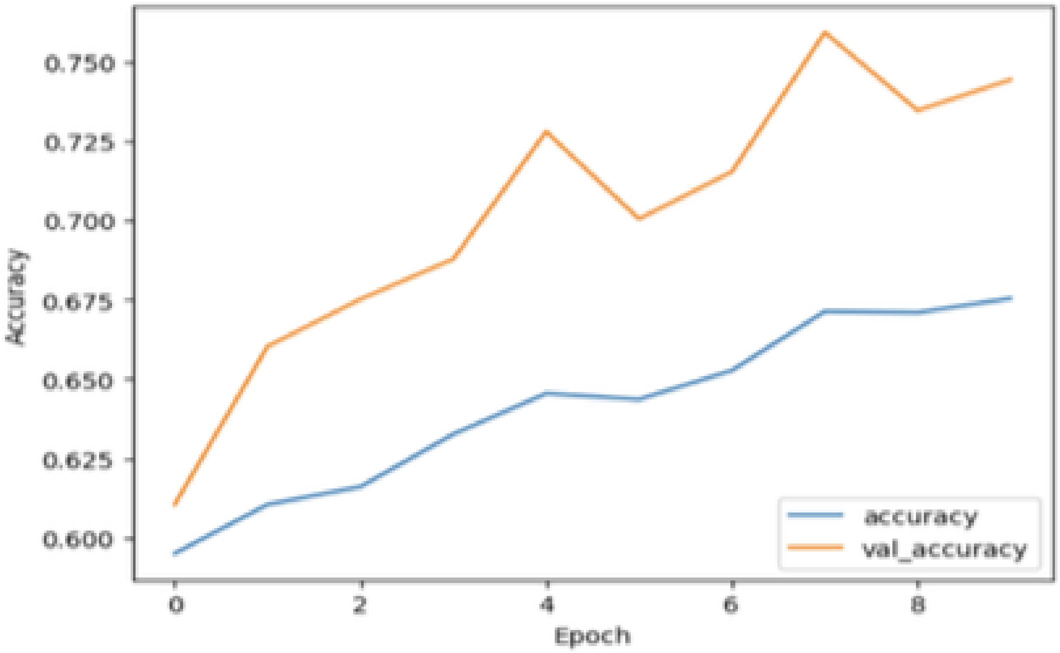

Figure 4: Accuracy v/s Epoch

9 Conclusion

In conclusion, the deep fake detection system presented in this report demonstrates promising results in detecting manipulated images. The system's high accuracy, robustness, and potential for further improvement make it a valuable tool in the fight against misinformation and fake media. By leveraging advanced machine learning techniques and data augmentation, this system represents a significant step forward in ensuring the integrity and authenticity of digital content. Further research and development in this field will continue to advance the capabilities of deep fake detection systems, ultimately contributing to a more trustworthy digital media environment.

The deep fake detection system has shown promising results, but there are several areas for future improvement. One area is the exploration of more advanced data augmentation techniques to further enhance the model's performance. Techniques such as mixup augmentation, CutMix, and RandAugment could be explored to generate more diverse training samples and improve the model's ability to generalize to unseen data.

10 References

1. J. Thies, M. Zollhofer, M. Stamminger, C. Theobalt, M. Nießner, *Face2Face: real-time face capture and reenactment of RGB videos*, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2387–2395.

2. D. Afchar, V. Nozick, J. Yamagishi, I. Echizen, *MesoNet: a compact facial video forgery detection network*, in: 2018 IEEE International Workshop on Information Forensics and Security (WIFS), 2018, pp. 1–7, doi:10.1109/WIFS.2018.8630761.

3. A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, M. Nießner, *FaceForensics++: learning to detect manipulated facial images*, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 1–11.

4. F. Chollet, *Xception: deep learning with depthwise separable convolutions*, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1251–1258.

5. Y. Li, S. Lyu, *Exposing DeepFake videos by detecting face warping artifacts*, in: IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2019.

6. H.H. Nguyen, J. Yamagishi, I. Echizen, *Capsule-forensics: using capsule networks to detect forged images and videos*, in: ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2019, pp. 2307–2311, doi:10.1109/ICASSP.2019.8682602.

7. H. He, E.A. Garcia, *Learning from imbalanced data*, IEEE Trans. Knowl. Data Eng. 21 (9) (2009) 1263–1284.

8. D. Güera, E.J. Delp, *DeepFake video detection using recurrent neural networks*, in: 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), 2018, pp. 1–6, doi:10.1109/AVSS.2018.8639163.

9. X. Li, Y. Lang, Y. Chen, X. Mao, Y. He, S. Wang, H. Xue, Q. Lu, *Sharp multiple instance learning for DeepFake video detection*, in: Proceedings of the 28th ACM International Conference on Multimedia, 2020, pp. 1864–1872.