UNIVERSITY OF OULU

FACULTY OF INFORMATION TECHNOLOGY AND ELECTRICAL ENGINEERING

Antti Luukkonen

EXPLORING REMOTE PHOTOPLETHYSMOGRAPHY SIGNALS FOR DEEPFAKE DETECTION IN FACIAL VIDEOS

Master's Thesis
Degree Programme in Computer Science and Engineering
June 2023

Luukkonen A. (2023) Exploring Remote Photoplethysmography Signals for Deepfake Detection in Facial Videos. University of Oulu, Degree Programme in Computer Science and Engineering, 58 p.

ABSTRACT

With the advent of deep learning-based facial forgeries, also called "deepfakes", the field of accurately detecting forged videos has become a quickly growing area of research. For this endeavor, remote photoplethysmography, the process of extracting biological signals such as the blood volume pulse and heart rate from facial videos, offers an interesting avenue for detecting fake videos that appear utterly authentic to the human eye.

This thesis presents an end-to-end system for deepfake video classification using remote photoplethysmography. The minuscule facial pixel colour changes are used to extract the rPPG signal, from which various features are extracted and used to train an XGBoost classifier. The classifier is then tested using various colour-to-blood volume pulse methods (OMIT, POS, LGI and CHROM) and three feature extraction window lengths of two, four and eight seconds.

The classifier was found effective at detecting deepfake videos with an accuracy of 85 %, with minimal performance difference found between the window lengths. The GREEN channel signal was found to be important for this classification.
Keywords: Deepfakes, Remote Photoplethysmography, Deepfake Detection

Luukkonen A. (2023) Etäfotoplethysmografian Hyödyntäminen Syväväärennösten Tunnistamiseen. Oulun yliopisto, Tietotekniikan tutkinto-ohjelma, 58 s.

TIIVISTELMÄ

Syväväärennösten eli syväoppimiseen perustuvien kasvoväärennöksien yleistyessä väärennösten tarkasta tunnistamisesta koneellisesti on tullut nopeasti kasvava tutkimusalue. Etäfotoplethysmografia (rPPG) eli biologisten signaalien kuten veritilavuuspulssin tai sykkeen mittaaminen videokuvasta tarjoaa kiinnostavan keinon tunnistaa väärennöksiä, jotka vaikuttavat täysin aidoilta ihmissilmälle.

Tässä diplomityössä esitellään etäfotoplethysmografiaan perustuva syväväärennösten tunnistusmetodi. Kasvojen minimaalisia värimuutoksia hyväksikäyttämällä mitataan fotoplethysmografiasignaali, josta lasketuilla ominaisuuksilla koulutetaan XGBoost-luokittelija. Luokittelijaa testataan usealla eri värisignaalista veritilavuussignaaliksi muuntavalla metodilla sekä kolmella eri ominaisuuksien ikkunapituudella.

Luokittelija pystyy tunnistamaan väärennetyn videon aidosta 85 % tarkkuudella. Eri ikkunapituuksien välillä oli minimaalisia eroja, ja vihreän värin signaalin havaittiin olevan luokittelun suorituskyvyn kannalta merkittävä.

Avainsanat: Syväväärennökset, etäfotopletysmografia, syväväärennösten tunnistaminen

TABLE OF CONTENTS

FOREWORD

I want to thank the University of Oulu CSE research unit for the generous scholarship that allowed me to focus on this thesis, I am grateful for getting to work on this interesting subject with amazing people. I feel like there was so much more that could have been done on the subject had I the time and resources to do it, but for now it is time for me to lay down this work and turn my focus elsewhere.

I want to thank my supervisor Miguel Bordallo López for both his guidance on the thesis, and the assistance and work on getting the thesis done and reviewed in the short time I had left. I would also like to thank Jukka Komulainen for acting as the secondary supervisor of my thesis and for his flexibility on the short review time. I would especially like to thank Constantino Álvarez Casado for his excellent guidance and invaluable moral support during the writing of the thesis.

I would like to thank all my friends for their support. I would especially like to thank my fellow student Jouni Saari for carrying me through the low points of my studies and for being my personal calendar when my own time-keeping was lacking.

I would like to thank all my sisters and brothers for all their support and for being the best family one could ask for. Particularly, I would like to thank my brothers Jaakko and Arttu for being my best friends through my life and the best playtesters a young hobbyist game developer could have.

I would like to thank my parents for letting my childlike interest in technology flourish, and for the never-ending support and encouragement. My childhood has been a happy and secure one and there is nothing I could write that could scratch the feeling of gratefulness I have for everything.

Last but certainly not the least, I would like to thank my wife Inka for everything. None of this would be possible without her.

Finally, I dedicate this thesis to our son, whose smile has been my light in the countless days I have spent staring at the screen.

Oulu, June 14th, 2023

Antti Luukkonen

LIST OF ABBREVIATIONS AND SYMBOLS

AAM	Active Appearance Models
AC	Alternating Current
AE	Autoencoder
AGRD	Adaptive Green-red difference
AI	Artificial Intelligence
AROI	Adaptive Region of Interest
AUC	Area Under the Curve
BCG	Ballistiocardiographic motion
BP	Blood Pressure
BPM	Beats Per Minute
BSG	Ballistiography
BVP	Blood Volume Pulse
CHROM	Chrominance
CNN	Convolutional Neural Networks
cPPG	Contact Photoplethysmography
CPU	Central Processing Unit
CSE	Computer Science and Engineering
DAN	Deep Alignment Network
DC	Direct Current
DFDC	DeepFake Detection Challenge
DFT	Discrete Fourier Transform
DL	Deep Learning
DNN	Deep Neural Networks
DRMF	Discriminative Response Map Fitting
ECG	Electrocardiogram
ERT	Ensemble of Regressor Trees
FFT	Fast Fourier Transform
FPR	False Positive Rate
GAN	Generative Adversarial Networks
GBDT	Gradient Boosted Decision Tree
GPU	Graphical Processing Unit
GRD	Green-red difference
HOG	Histogram of Oriented Gradients
HR	Heart Rate
HRV	Heart Rate Variability
ICA	Independent Component Analysis
IT	information technology
KNN	K-Nearest Neighbors
LBF	Local Binary Features
LBP	Local Binary Patterns
LE	Laplacian Eigenmap
LED	Light-emitting Diode
LGI	Local Group Invariance
LSTM	Long Short-Term Memory

MAE	Mean Average Error
ML	Machine Learning
MMST	Motion-Magnified dual-Spatial-Temporal attentional network
MMST	Motion-Magnified dual-Spatial-Temporal Representation
OMIT	Orthogonal Matrix Image Transformation
PBV	Normalized Blood-Volume Pulse Vector
PCA	Principal Component Analysis
POS	Plane Orthogonal-to Skin
PPG	Photoplethysmography
PR	Pulse Rate
PRV	Pulse Rate Variability
PSC	Projection vector based on Spectral Characteristics
RGB	Red, Green and Blue
ROI	Region of Interest
ROC	Receiver Operating Characteristics
rPPG	Remote Photoplethysmography
SFFS	Sequential Floating Forward Selection
SNR	Signal-to-noise Ratio
SPE	Stochastic Proximity Embedding
SSA	Singular Spectrum Analysis
SSD	Single-Shot Multibox Detector
SVD	Singular Value Decomposition
STFM	Spatio-Temporal Filter Module
STFT	Short Time Fourier Transform
SVM	Support Vector Machine
SVR	Support Vector Regressor
TPR	True Positive Rate
2SR	Spatial Subspace Rotation
2DCNN	Two-Dimensional Convolutional Neural Network
3DCNN	Three-Dimensional Convolutional Neural Network
Hz	Hertz

1. INTRODUCTION

The invention of deep learning-based image and video manipulation techniques, also known as *deepfakes*, has shaken up the field of multimedia forensics [1]. The manipulation of multimedia assets, especially videos, has previously been an arduous and time-consuming task, requiring the skills of an experienced video editor. However, deepfakes have shifted the manipulation process to machines, bringing the tools to create believable fake media to everyone with access to a computer. While widely available deepfake technology is excellent for purposes like digital art, photography, and movie production, it can also be used for malicious purposes, like fake news campaigns and blackmailing. An example of malicious use of deepfakes is the case of European politicians duped into a video conference with a deepfake impersonation of Vitali Klitschko, the mayor of Kyiv at the time [2]. The inability to trust the authenticity of media assets also can reduce the public trust in journalism, including credible and reliable sources.

The problems caused by the progression of deepfake technology have caught a lot of media attention and have resulted in a rise of interest in multimedia forensics research from both the academy and from major information technology (IT) companies and agencies [1]. Various methods of deepfake detection have been developed, and deepfake reference datasets have been composed.

This thesis concerns the use of remote photoplethysmography in deepfakes and deepfake detection. Photoplethysmography (PPG) is an optical measurement system for detecting blood volume changes in the microvascular bed of tissue. [3]. Basic PPG technology utilizes a few optoelectronic components: a light source and a photodetector to measure the variations in light intensity of the tissue. This variation can be used to extract signals such as the blood volume pulse (BVP). These signals and their features can be used to classify the video as real or fake using signal processing and machine learning techniques.

This thesis reviews state-of-the art research on deep learning-based and conventional rPPG extraction methods, with a special focus on conventional methods. A standard methodology for a conventional rPPG extraction scheme is explained in detail, briefly examining the current research and standard methods on each step. A brief review of deepfakes and deepfake generation methods is conducted. Deepfake detection research is studied, including deepfake detection datasets and deepfake detection methods, with a focus on rPPG-based detection methods. Finally, current research on PPG modification and synthetic generation of PPG signals is studied.

Additionally, an end-to-end system for classifying videos using rPPG signals is introduced, using the *Face2PPG* [4] rPPG extraction pipeline to extract the signals, then using a gradient boosting library *XGBoost* [5] to classify the features extracted from the signals. The performance of the classifier is evaluated using three different window lengths and nine different feature sets using different rPPG to BVP conversion schemes. The results are analyzed and compared to other relevant research results.

2. RELATED WORK

This chapter includes the necessary background of rPPG-based deepfake detection technology. First, explanations of concepts in photoplethysmography (PPG), especially remote photoplethysmography (rPPG), are given, and state-of-the-art research in the area is reviewed. Second, research on deepfake technology and generation methods is reviewed. Third, the various state-of-the-art methods of detecting deepfakes are studied, particularly in rPPG-based studies. Finally, the current work on modifying the PPG signals and generating synthetic PPG signals are explored.

2.1. Photoplethysmography

Photoplethysmography is an optical technique for measuring blood volume changes in tissue. Due to its non-invasive, wearable implementation, the PPG method has gained popularity over traditional electrocardiogram (ECG) technology for measuring blood volume pulse (BVP) and heart rate (HR). Basic PPG technology utilizes a light source and a photodetector to measure the absorption of light by the blood in the tissue. The light source is emitted into the tissue and the reflected light is measured. The blood flow in the arteries and capillary bed network cause variance in the intensity of the absorbed light. From the measured BVP signal, HR information can be extracted from the peripheral pulse. Figure 1 shows

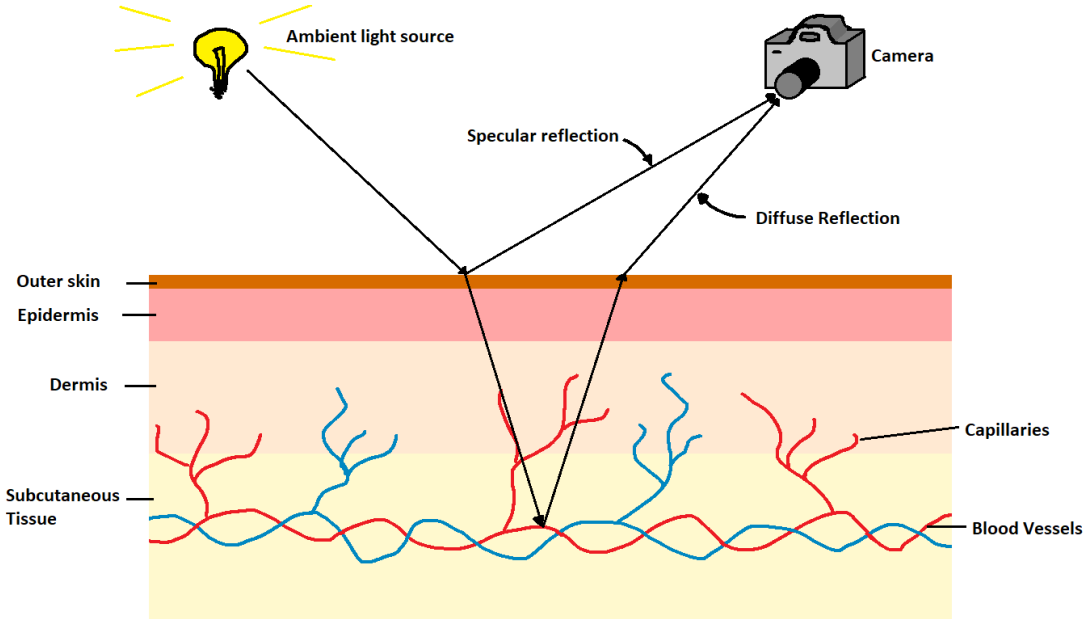


Figure 1. Light propagation in skin.

PPG measurement can be a valuable tool for monitoring BVP and HR as it is non-invasive, and the implementation can be small and lightweight. While the origins of the different components of the PPG signal are not fully understood, they can provide valuable information on the cardiovascular system [3][6].

The PPG waveform can be split into two components [3]. The pulsating component, often called the 'AC' component, has a frequency linked to the heart rate. The AC

component is superimposed onto a 'DC' component originating from the tissues and average blood volume. The DC component varies slowly due to various characteristics, including respiration and vasomotor activity. The AC and DC components can be extracted from the PPG with filtering and amplification.

2.2. Remote PPG Extraction

Remote PPG (rPPG) extraction refers to methods of extracting biosignals from the human body without physical contact, utilizing remote sensors such as cameras or radars. Classic rPPG methods use image and signal processing techniques for analyzing video, looking for slight color and illumination changes related to the BVP [7]. More recently, learning-based alternatives have been used to train models capable of heart rate estimation from a video containing a face [8].

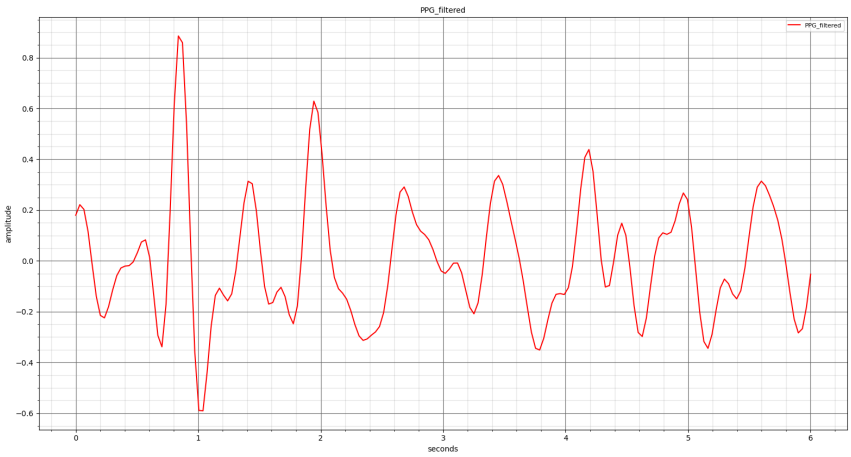


Figure 2. Example of a remotely extracted PPG signal.

Due to not needing direct contact to skin like common PPG or ECG systems, rPPG has found applications in many fields, such as fitness monitoring [9], stress monitoring [10], driver state monitoring [11][12] and more [13]. The pressure of contact PPG (cPPG) sensors on the skin is also shown to significantly impact the pulse arrival time detected by the sensor [14]. This could lead to detection of erroneous blood pressure variation when the pressure of the sensor is not constant. Remote PPG sensors can also alleviate the possible skin irritation caused by long term use of contact PPG sensors. As digital cameras are inexpensive and widely available, rPPG technique is easily scalable compared to traditional cPPG.

While having the advantage of not needing contact to the skin, rPPG has its challenges compared to the contact PPG methods. The signal-to-noise ratio (SNR) of the rPPG signal is much lower than that of a cPPG signal. While cPPG utilizes dedicated light sources and contact probes to enhance the signal quality, rPPG can only utilize ambient light. In addition, the camera used in rPPG measurement is usually located further away from the measured tissue, resulting in less accurate color measurement. Increased melanin content also decreases the diffusely reflected light from the skin, reducing the SNR of the rPPG signal [15]. This results in worse performance when extracting rPPG from subjects with darker skin tones.

Nowadays, there are two ways to extract physiological signals from videos: non-learning-based techniques that use signal processing, and deep learning approaches. Non-learning-based methods aim to extract physiological signals using computer vision and signal processing without learning from prior data. They usually follow a standard pipeline, which includes several steps, as shown in Figure 3. Learning-based methods try to learn how to extract physiological signals or parameters using deep learning models. They can try to extract the signals directly from the videos or replace some of the steps in the convolutional rPPG pipeline with deep learning methods.

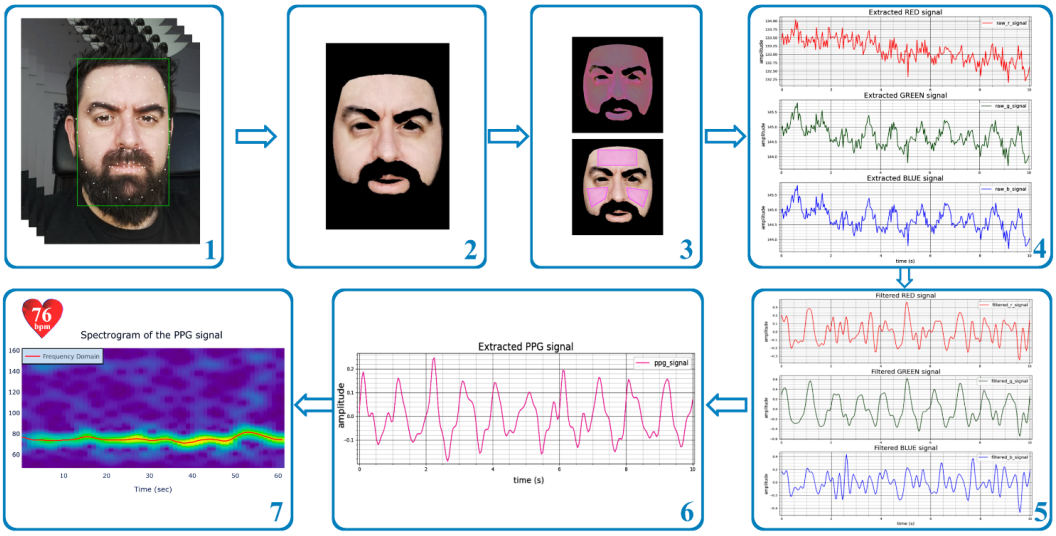


Figure 3. Example of a conventional rPPG method working principle (*Face2PPG* [4]).

Generally, the non-learning-based rPPG methods work in a few steps: face detection and alignment, region of interest (ROI) detection and selection, BVP extraction and calculation of vital parameters, such as heart rate.

A standard non-learning-based pipeline for recovering physiological signals from videos comprises of six sequentially connected main modules. The first module, **Face detection and alignment**, involves the detection and alignment of the face in every frame. This step includes face detection to locate the face coordinates in the frame and face alignment to detect several facial points in the detected face. The second module is the **ROI selection**, which selects the regions of interest of the face based on a color skin segmentation or selection of patches based on face location or the coordinates of the landmarks. The third module, **RGB extraction**, extracts the raw signal from a window of several RGB frames, where the value at every sample is computed using the mean value of the pixels contained in the mask or patches selected in the previous step. The fourth module, **Pre-processing**, filters the extracted raw RGB signal to adequately prepare it for the band of interest. Filtering is a critical factor in properly recovering the BVP signal, as remote PPG signals usually present trends and noise in the frequency spectrum of the camera sample rate. The fifth module, **RGB to PPG transformation (rPPG)**, is where the RGB filtered signal is transformed to a PPG signal using a transformation method. This module is one of the core blocks in this topic since it is the component that transforms RBG signals into physiological signals. The last module, **Frequency analysis**, performs spectrum analysis of the rPPG and reference ground-truth (BVP or ECG) signals to estimate the heart rate.

2.2.1. Face Detection

Face detection is a crucial task in computer vision that aims to locate the presence of human faces in digital images or videos. It is an essential component of many facial analysis tasks, such as face recognition, facial expression recognition, head pose estimation, and anti-spoofing. The primary goal of face detection is to determine whether there are any faces in an image and estimate their locations and sizes accurately [16].

Over the past few decades, the study of face detection has been a topic of intense research. The complexity of face detection arises from the vast variations in scale, orientation, skin color, facial expression, lighting conditions, occlusions, complex background, and other factors that can impede the accuracy of face detection algorithms [16].

Significant advances have been made in the field of face detection, with the earliest face detection methods being rule-based and relying on hand-crafted features to find faces [17]. These methods were highly dependent on the quality of the input images and performed poorly in challenging environments. One of the earliest and most influential methods for face detection is the Viola-Jones algorithm, which uses a cascade of boosted classifiers to detect faces efficiently [17].

These methods paved the way for numerous other algorithms, such as Haar-like features, Local Binary Patterns (LBP) [18], Histogram of Oriented Gradients (HOG) [19], or Convolutional Neural Networks (CNN) [20]. These algorithms have been highly successful in detecting faces in images and videos, although numerous challenges still remain to increase the robustness in real-world scenarios.

While any face detection algorithm can be used in rPPG implementations, a frequently used method is the Viola-Jones algorithm, particularly because of its long-time availability in the OpenCV [21] computer vision library [22]. Alternative methods include an algorithm to detect skin regions [23] using a neural network-based classifier. More recent approaches [4] make use of more modern face detection algorithms, such as the single shot multibox detector (SSD) [24].

2.2.2. Face Alignment

Face alignment is an essential step in the field of computer vision that has gained significant attention in the past few decades. This process involves the detection of facial landmarks, which are predefined key points of a face in a given image or video, as shown in Figure 4. Facial alignment is critical for various applications, such as face recognition, facial expression recognition, face modeling, and facial attribute computing and others [25][26]. Face alignment has become a popular research topic because it plays a vital role in improving the accuracy and robustness of facial analysis systems. With accurate facial landmark detection, facial features can be extracted, and specific regions of interest (ROIs) can be selected for analysis. This is especially important in rPPG systems, as it allows for more accurate and robust tracking of the ROI, even in motion.

The majority of face alignment methods use machine learning to predict facial landmarks. In recent years, deep learning has emerged as the state-of-the-art method

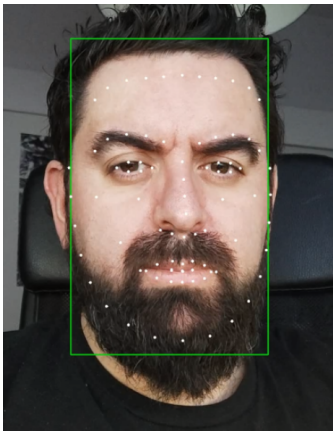


Figure 4. Facial detection and landmarks detection performed with *Face2PPG* [4]).

for facial alignment. This technique has shown superior performance compared to traditional machine learning approaches, as it can learn complex patterns and features from large amounts of training data. However, deep learning requires computationally intensive inference.

For use cases that require faster computation times, Ensemble of Regression Trees (ERT) [27] and Local Binary Features (LBF) [28] can achieve an impressive performance of more than 1000 frames per second with 194 points [25]. In time critical applications, such as potential real-time rPPG systems, fast face alignment is necessary.

Face alignment methods can be classified into two categories: *generative* and *discriminative* methods [25].

- *Generative methods* formulate facial alignment as an optimization problem to find the shape and appearance parameters that generate an appearance model that best fits the test face. Examples of a generative method are active appearance models (AAM) [29].

- *Discriminative methods*, on the other hand, directly infer the target location from the facial appearance. These methods teach independent local detectors or regressors for each facial point and employ a global shape model to regularize their predictions. Alternatively, they teach a vectorial regression function to infer the entire face shape, where the shape constraint is implicitly encoded. A few examples of discriminative methods include discriminative response map fitting (DRMF) [30], Ensemble of Regression Trees (ERT) [27], Local Binary Features (LBF) [28], and deep learning-based methods such as Deep Alignment Network (DAN) [31].

2.2.3. ROI Selection

ROI detection is generally accomplished by color-based and patch-based selection methods. The thickness of the skin differs across the face, and so measures from different areas give differing diffusion reflection information [32]. A common method

is to select and track a preselected region or multiple regions of the face [4]. Choosing only suitable regions of the face improves the performance of the extraction over selecting the entire face [33]. Some implementations choose the whole face [34][35], but as the forehead and cheek regions usually provide stronger rPPG signals compared to other regions of the face [32], many methods only select them as the ROI [36][37][4].

Some methods use skin detection to discard non-skin regions, as they do not usually contain rPPG information [38]. Skin detection is particularly effective in removing noise caused by head rotations [39].

When selecting rectangular regions of a face in their study, Wang [40] concluded that selecting the best 1/4 to 1/2 of total regions is favorable for heart rate extraction. Selecting too few regions will be subject to quantization noise and too many will suffer from uneven illumination interference.

Po et al. [38] propose an adaptive ROI (AROI) approach for dynamically selecting ROI. In their method, the face is divided into regions and the rPPG signal of each region is extracted. Each regions signals SNR is then calculated and applied to a SNR map indicating the quality of the rPPG signal in the region. Suitable regions are then selected by applying mean-shift clustering and adaptive thresholding to the SNR maps. Kumar et. al. [41] use a similar method of ROI selection based on a "goodness" metric. Po et al. also explore a method that uses skin detection masking to select the skin regions of the full face. In their experiments, they compare the methods to conventional fixed ROI methods and report that the AROI method improved the rPPG signal quality and achieved better HR measurement accuracy than the other methods of ROI selection.

Figure 5 shows the different approaches for ROI selection in literature.

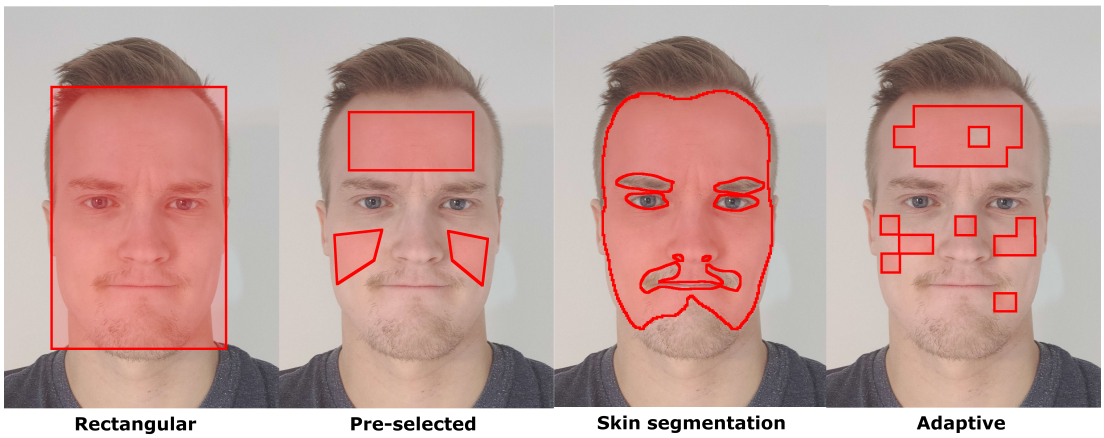


Figure 5. Different ROI selection approaches.

2.2.4. RGB Extraction

The core principle of rPPG methods is to extract color information from the face. A common method to achieve this is to extract the RGB signal from the skin by averaging the skin pixel values of the ROI. This method can be done separately for every ROI.

Some methods use a window of several frames over which the mean RGB is calculated [4].

2.2.5. Pre- and Post-Processing

Proper pre- or post-processing of the RGB signal is essential for accurate extraction of BVP signals in remote PPG. Filtering is a critical factor in preparing the raw RGB signal for the band of interest, which is typically between 0.75 and 4.0 Hz for heart-related signals. The raw RGB signal often presents trends and noise in the frequency spectrum of the camera sample rate, and filtering helps remove these artifacts. Some common filtering methods used for RGB processing include detrending, bandpass filtering, or a moving average filter. Additionally, some methods also normalize the RGB values over a temporal interval to further enhance signal quality [15]. Overall, proper RGB signal preprocessing can significantly improve the accuracy and reliability of remote PPG systems [4].

For some RGB to PPG transformation methods, Unakafov [42] suggests filtering after the RGB to PPG conversion. In the study, postprocessing instead of preprocessing performed better with GREEN, GRD, CHROM, and POS methods, while preprocessing was preferred for aGRD and ICA. The RGB to PPG conversion methods are explained in the next section.

2.2.6. RGB to PPG Transformation

Multiple methods of transforming the RGB signal to PPG have been proposed in scientific literature. This section contains brief descriptions of the methods and results of several comparisons.

- The GREEN [36] method estimates the rPPG by the green signal alone. The green signal is chosen as it contains the strongest plethysmographic signal, corresponding to an absorption peak of oxyhaemoglobin [43]. This is a popular approach due to its simplicity.
- The Green-red difference (GRD) [44] method uses the difference of the PPG-favoured green signal and the red signal, which is considered to contain artefacts.
- Adaptive green-red difference (AGRD) [45] is an improved version of GRD with adaptive color difference operation applied to improve motion robustness.
- Principal Component Analysis (PCA) [46][47] is a blind source separation (BSS) method, which finds the components that explain the maximum amount of variance possible, called *principal components*. These components are obtained by first computing the covariance matrix of the RGB vector, then calculating the eigenvectors and eigenvalues of the covariance matrix. These eigenvectors correspond to the principal components and eigenvalues to the amount of variance explained by the component, i.e., the significance of the principal component. The principal components are sorted in the order of significance, and the RGB data is then cast to the principal components.

- Independent Component Analysis (ICA) [48][34] is a BSS method that tries to find independent sub-parts that make up the set of original signals. ICA assumes that the RGB data is a mixture of different sources and aims to find a demixing matrix $\mathbf{W}$ that maximizes the non-gaussianity of each source [35]. ICA aims to solve the equation:

$$\hat{\mathbf{x}}(t) = \mathbf{W}\mathbf{c}(t), \tag{1}$$

  where $\hat{\mathbf{x}}(t)$ is the estimation of the underlying source signals that make up the RGB signal. In practice, iterative methods are used, a common method being the JADE algorithm [49].

- Laplacian Eigenmap (LE) [50] uses the Laplacian-Beltrami operator in manifold to discover the embedded low-dimensional data from the three-dimensional RGB signal, while maintaining the distance relation of any two points.

- Stochastic proximity embedding (SPE) [51] generates an one-dimensional Euclidean embedding out of the RGB signal, where the similarities between the related observations are preserved. It achieves this by using a self-organizing scheme that attempts to bring each sum-of-squares error function to zero.

- Normalized Blood-Volume Pulse Vector (PBV) [52] uses the characteristic wavelength dependency of the PPG signal to estimate the pulse signal from time-sequential RGB data.

- Singular Spectrum Analysis (SSA) [40] is a spectrum estimation method for extracting oscillatory components, such as heart rate, from time series.

- CHROM [15] removes the specular reflection component from the RGB data, leaving only the color difference, i.e., the chrominance signals. The rPPG is then estimated from the chrominance signals.

- Spatial Subspace Rotation (2SR) [53] constructs a spatial subspace in the RGB space and measures the rotation angle of the spatial subspaces between subsequent frames for pulse extraction.

- Plane Orthogonal-to Skin (POS) [54] uses a plane orthogonal to the skin-tone in the temporally normalized RGB space for rPPG extraction.

- Local Group Invariance (LGI) [55] uses features invariant with respect to the action of a differentiable local group of local transformations to re-arrange the signal to a more concentrated distribution.

- Orthogonal Matrix Image Transformation (OMIT) [4] uses matrix decomposition techniques. It leverages reduced QR factorization and Householder Reflections to generate an orthogonal matrix capturing variations in RGB data. By projecting the input data onto a subspace orthogonal to the dominant variations, OMIT extracts the PPG signal.

- Projection vector based on Spectral Characteristics (PSC) [13] defines a projection vector to exploit the spectral characteristics of rPPG signals and limit the projection vector to a constraint plane, similar to the POS method.

In their study, Van Es et al. [56] compared different rPPG methods including GRD, AGRD, PCA, ICA, LE, SPE, CHROM, and POS in terms of pulse rate (PR) and pulse rate variability (PRV). They recommended using POS or CHROM. However, the OMIT or LGI methods were not included in the comparison.

Similarly, de Haan et al. [15] compare CHROM based methods to BSS methods and a method roughly equivalent to GRD, in videos containing motion. They conclude that the chrominance-based methods work better, and that the BSS methods suffer from not being able to distinguish the pulse signal from periodic motion distortion.

In another study on the effectiveness of rPPG methods, Haugg et al. [57] evaluated the performance of POS, LGI, CHROM, OMIT, GREEN, ICA, PCA, and PBV. They found LGI, POS, and OMIT to be the best methods overall. POS had the best results in a gym video activity with a lot of movement and indoor lighting, as well as in a video with a rotating face. In natural lighting and a relatively static video of a person talking, CHROM provided the best results. For BPM estimation, POS was concluded to be the best rPPG method in the study.

2.2.7. Frequency Analysis

One of the main goals of rPPG signal processing is to estimate the heart rate. To achieve this, frequency analysis is commonly used to analyze the spectrum of the rPPG signal. One straightforward method is to calculate the Fourier transform or spectral density of the signal and detect peaks from it.

A common way to perform Fourier transform is to use the Short Time Fourier Transform (STFT) which calculates the Fourier transform over successive overlapping windows of the signal, allowing for a more localized frequency analysis. On the other hand, spectral density estimation is often done using Welch's method, which divides the signal into overlapping segments and computes the periodogram of each segment, resulting in a smoothed estimate of the power spectrum.

2.2.8. Deep Learning-Based Remote Photoplethysmography Methods

Deep learning-based rPPG methods, especially CNN-based methods have started to gain attention since 2018. Deep learning is common in face detection and alignment methods. This section briefly reviews the rPPG approaches where deep learning is also applied in other steps of the process, such as PPG or HRV feature extraction. Deep learning-based methods can be categorized into two groups: combinations of conventional and deep learning methods, and end-to-end deep learning methods [8].

Combination of Conventional and Deep learning methods

EVM-CNN [58], uses conventional methods for tracking facial landmarks (LBF) and ROI selection. Then the RGB signals extracted from the ROIs are fed to a spatial

decomposition and temporal filtering module, to obtain so-called feature images. These features imagers are then fed to a CNN that is used to estimate heart rate.

Luguev et al. [59] developed a method for HRV measurements, where a 3D convolutional neural network (3D-CNN) is used for pulse signal extraction. Raw video sequences are fed to the 3D-CNN without any face detection, face alignment, preprocessing or ROI selection. The mean absolute error of the signal is used as the loss function for the model. The HRV features are then extracted from the output signal using conventional methods.

In their work, Zhan et al. [60] study CNN-based PPG signal extraction. They address four questions:

- Does the CNN learn PPG, ballistocardiographic motion caused by blood pulsation (BCG) or a combination of both?
- Can the finger oximeter be used as the reference for CNN training?
- Does CNN learn the spatial context information of the measured skin?
- Is CNN robust to motion and how is this motion-robustness achieved?

A CNN-PPG model and four experiments are used to answer these questions. From their experiments, they conclude that the trained model does indeed measure PPG instead of BCG. The results on the finger oximeter indicate that because of the physiological delay between the reference finger oximeter and video data, the CNN may not learn the correct match between the video and the reference PPG signal. This can be somewhat alleviated with phase correction, but even then the results are worse than when using a rPPG signal extracted with a conventional rPPG method as reference. However, by filtering out the high-frequency harmonics of the phase-corrected finger-PPG signal, the signal outperforms the conventional rPPG signal as a reference signal. However, they note that there is no motion in the used PURE and HNU datasets, which can affect the results. They also conclude that the training of the CNN on different objects (e.g., face or palm) does not make a difference, confirming that the spatial context information is not exploited by the CNN. The experiments on motion robustness of the CNN show that the accuracy of the CNN trained on clean videos is close to 0% when tested on noisy videos. However, when trained on noisy data, the accuracy is close to 100%, suggesting that the CNN can differentiate between the intensity variations caused by blood absorption and motion. Additionally, they experiment and present results that prior-knowledge, such as POS method, in combination with the CNN can improve the CNN-PPG.

End-to-end Deep learning methods

Chen et al. [61] developed an end-to-end method for heart rate extraction using deep CNN named *DeepPhys*. DeepPhys uses a motion representation algorithm based on skin reflection model and an attention mechanism based on appearance information to guide the motion estimation.

Deep PPG [62] by Reiss et al. uses a CNN-model to estimate the heart rate from PPG. The PPGs are first segmented into sliding windows and fast Fourier transform (FFT) is applied to each window. The resulting time-frequency spectra are the cut to

the 0-4 Hz band of interest and z-normalization is performed. The resulting spectra are used as input for the deep learning model.

Meta-rPPG [63] is an end-to-end deep learning approach for rPPG estimation. It consists of a convolutional encoder for feature extractor, a bidirectional LSTM for rPPG estimation and a shallow Hourglass network as synthetic gradient generator for transductive learning. Transductive learning is a method of coping with unforeseeable distributional changes during deployment by taking unlabeled samples during testing for a self-supervised weight adjustment. This provides fast adaptation to the distributional changes.

Špetlík et al. [64] proposed a two-step CNN to estimate the heart rate. The first step extracts the rPPG signal from a sequence of images using a CNN, and the second step estimates the HR from the signal using another CNN.

Yu et al. [65] propose a two-stage end-to-end method for recovering a rPPG from highly compressed videos. It uses a spatio-temporal video enhancement network (STVEN) for enhancing the video, then an rPPG network (rPPGNet) for extracting the rPPG signal. The rPPGNet can function independently for extracting rPPGs from videos.

PhysNet by Yu et al. [66] uses spatio-temporal networks for reconstructing rPPG signals from videos. They experiment on different spatio-temporal models, such as 3DCNN, temporal encoder-decoder 3DCNN (3DCNN-ED), 2DCNN, LSTM, bidirectional LSTM and convolutional LSTM. Of these models 3DCNN-ED is found to give the best performance.

HeartTrack [67] uses a spatio-temporal attention CNN that takes a time-domain discrete derivative of the video in ROI, a ROI mask, and the frames of the video in the ROI area to extract a rPPG signal. The heart rate is then estimated using a one-dimensional CNN. *HeartTrack* also uses synthetic PPG signals to pre-train the heart rate estimator CNN.

Boutsefsaf et al. [68] also use synthetic PPG for training set augmentation, but they try to apply the signals to videos. A 3D-CNN structure used to extract the heart rate was then trained on the augmented dataset.

DeeprPPG by Liu et al. [69] is a lightweight rPPG estimation network based on spatiotemporal convolutions. Additionally it uses an adaptive spatiotemporal rPPG aggregation strategy to obtain robust rPPG signal from multiple input skin regions.

Shortcomings of deep learning-based methods

Deep learning-based methods show exceptional results in extracting the rPPG signal and heart rate from videos. They are often complete end-to-end systems with minimal intermediate steps. However, this black-box behavior presents challenges when a clear understanding of the underlying mechanisms is preferred, such as in critical applications like healthcare. Furthermore, the need for massive amounts of diverse training data required to train robust and generalized solutions is an important problem for deep learning-based methods.

2.3. Deepfakes

Deepfakes are images and videos manipulated using advanced deep learning tools. They are a type of synthetic media created using machine learning algorithms, particularly generative adversarial networks (GAN) or autoencoders (AE), which can create hyper-realistic media that appears to be genuine. In recent years, deepfakes have become a central problem due to the ease with which they can be generated, distributed and used to manipulate public opinion [70].

Deepfake technology enables easy creation of realistic manipulated media, provided one can access large amounts of data. This is because deepfakes rely on training algorithms on large datasets of genuine media in order to generate fake media that convincingly resembles the real thing. The quality of deepfakes has improved significantly in recent years, making it difficult to distinguish them from authentic media. Figure 6 shows an example of a fabricated deepfake video.

Deepfakes have non-malicious use cases, for example in movie production, photography, and video games. In these fields, deepfakes can be used to create new content or enhance existing media. For instance, deepfakes can be used to create realistic special effects or to modify scenes that are otherwise difficult or impossible to film.

(a) Original (b) Deepfake

Figure 6. Example of a deepfake from the *FaceForensics++* [71]) dataset.

2.3.1. Problems of Deepfakes

Deepfakes can also be used for malicious purposes, such as blackmailing and fake-news campaigns to manipulate public opinion. Deepfakes can be used to create fake political speeches, interviews, or endorsements, which can then be circulated on social media platforms to spread misinformation. This can have serious consequences, as it can undermine public trust in political institutions and the media.

Inability to trust the authenticity of media may also reduce trust in journalism, including serious and reliable sources. As deepfake technology becomes more sophisticated, it becomes easier to create fake news stories that look and sound

convincing. This can make it difficult for people to distinguish between genuine and fake news, leading to a loss of trust in the media as a whole.

2.3.2. Historical Context

Media manipulation is not a new problem. Image manipulation has been carried out since photography was born. One well-known example is the iconic portrait of Abraham Lincoln, which was revealed to be two pictures stitched together [72]. Research of multimedia forensics has been going on for at least 15 years as well. With the advent of photo-editing software, manipulation of videos and photos has become commonplace. However, deepfake technology has made manipulation easier and faster, increasing the availability of malicious manipulation. Deepfake applications, such as FakeApp [73], FaceSwap [74], and ZAO [75] are easily accessible and usable by people with no major technological skillset required.

2.3.3. The Race Between Manipulation and Forensics

Deepfakes have gained a lot of attention, which has resulted in more attention in research of multimedia forensics. Multimedia manipulation and forensics are in a continuous race, and deep learning based deepfakes require multimedia forensics research to come up with new solutions. As the technology used to create deepfakes becomes more advanced, so must the technology used to detect and prevent them.

2.4. Deepfake Generation Methods

Deepfake generation is achieved through the utilization of deep neural networks that reconstruct entirely new, yet convincingly realistic, faces based on the underlying characteristics of the original face [1]. There are three common models used in deepfake creation: autoregressive models, autoencoders, and generative adversarial networks (GANs), with GANs being the most prominent approach. Visual deepfake methods can be categorized into different types, including reenactment, face synthesis, face attribute manipulation, and face-swapping [76].

Face synthesis methods aim to synthesize entire faces without having a specific target subject in mind. Essentially, the generated persona does not exist in the real world. This technique can have significant implications in industries such as video games and modeling. An example of a state-of-the-art face synthesis method is StyleGAN [77].

Facial attribute manipulation involves modifying various facial characteristics such as hairstyles, eye color, wrinkles, skin color, age, and gender. There are several methods available for face attribute manipulation, including StarGAN [78], BeautyGAN [79], and SC-FEGAN [80]. A popular tool for face attribute manipulation is FakeApp [73].

Face-swapping techniques involve switching faces from one individual to another while preserving the expression of the original face. State-of-the-art face-swapping

methods include FSNET [81] and FaceShifter [82]. Multiple face-swapping applications, such as ZAO [75], DeepFaceLab [83], and FaceSwap [74], are readily available for users.

Reenactment techniques involve transferring facial expressions or body motion from one person to another. This technique was popular even before the advent of deepfakes. Reenactment methods can be further categorized into facial expression transfer with neural textures, typical facial expression reenactment (Face2Face), and body motion reenactment (puppet-master). An example of a well-known Face2Face reenactment method is FaceSwapNet [84].

2.5. Deepfake Detection

In recent years, various approaches have been developed to combat the threat deepfakes [85]. To achieve this, deepfake detection methods and datasets have been developed. Deepfake detection aims to distinguish between real and manipulated video content, relying on the use of deep learning and computer vision algorithms to detect inconsistencies in videos that may indicate that they are deepfakes. Additionally, deepfake datasets have been composed for training these methods and for benchmarking their performance.

The importance of deepfake detection has not gone unnoticed, with large technological companies such as Amazon, Meta, and Microsoft hosting deepfake detection challenges (DFDC) [86][87] to push forward the innovation of deepfake detection technology.

Development of deepfake detection technology is critical: in 2020, Korshunov et al. [88] found that human subjects classified 75.5% of good quality deepfakes incorrectly. However, human subjects were more accurate in classifying videos correctly than the deepfake detection algorithms. This highlights the need for continued research and development in this field. The use of deepfake detection methods in conjunction with human evaluation is the surest way to detect fakes, as deepfake detection algorithms may have trouble with videos that humans can easily distinguish as fake, while performing well on videos that are difficult for humans. With the ever-improving quality of deepfakes, deepfake detection is a constant race to stay on top of upcoming deepfake technologies [89].

2.5.1. Deepfake Detection Datasets

In recent years, various approaches have been developed to combat the threat of deepfakes [85]. To achieve this, deepfake detection methods and datasets have been developed. It is imperative to have a good and representative dataset for evaluating detection performance [1]. The dataset should contain subjects of varying gender, ethnicity, pose, synthesis method, and more to evaluate as comprehensively as possible.

To aid in the research and development of better deepfake detection methods, various deepfake detection datasets have been composed. These datasets consist of a large number of labeled videos featuring numerous individuals, with some of the videos modified using deepfake technology. The videos are used to train machine learning

or deep learning models that aim to classify them correctly. For evaluating deepfake detection performance, it is recommended to validate across multiple datasets. This approach takes into account the generalization ability of the detection methods.

The current video deepfake detection datasets are summarized in Table 1.

Database Name	**Real**	**Fake**	**Actors**
UADFV (2018) [90]	49	49	-
Deepfake-TIMIT (DF-TIMIT) (2018) [91]	320	620	32
Fake Faces in the Wild (FFW) (2018) [92]	0	150	-
FaceForensics (2018) [93]	1,004	1,004	1,004
FaceForensics++ (2019) [71]	1,000	4,000	977
Google DeepfakeDetection (DFD) (2019) [94]	363	3,068	28
DFDC preview (2019) [86]	1,131	4,113	66
Celeb-DF (2019) [95]	408	795	13+250
FakeET (FE) (2020) [96]	331	480	40
Deep Fakes (DFS) (2020) [97]	142	142	142
Celeb-DF v2 (2020) [95]	890	5,639	13+59
Deepfake Detection Chal. (DFDC) (2020) [87]	23,654	104,500	960
DeeperForensics-1.0 (DF-1.0) (2020) [98]	50,000	10,000	100
WildDeepFake (WDF) (2021) [99]	3,805	3,509	-
ForgeryNet (2021) [100]	99,630	121,617	5,400+
KoDF (2021) [101]	62,166	175,776	403
FakeAVCeleb (2021) [102]	500	19,500	500

Table 1. List of deepfake video datasets [1][85][89].

2.6. Deepfake Detection Methods

Deepfake detection can be classified into five different categories: *general network-based methods*, *temporal consistency-based methods*, *visual artefacts-based methods*, *camera fingerprints-based methods*, and *biological signals-based methods* [89]. This section reviews the five different categories, with a special focus on biological signals-based methods, particularly photoplethysmography-based methods.

2.6.1. General Network-Based Methods

The first category of deepfake detection methods is general network-based methods. These methods use face images extracted from the detected video to train a detection network. The trained network is then applied to all frames of the video, and predictions are calculated by averaging or voting strategies. These methods are highly dependent on neural networks and do not require specific distinguishable features. However, the downside of these methods is that they tend to overfit on specific datasets.

2.6.2. Temporal Consistency-Based Methods

Temporal consistency-based methods exploit the unique feature of time continuity in videos. Deepfake algorithms tend to cause inconsistencies between adjacent frames, which can be exploited in detecting deepfakes. Temporal inconsistency can be especially noticed in the shift of face position and video flickering. Compared to general network-based approaches, temporal consistency methods improve the detection performance by taking the continuity of the video into account. However, many temporal consistency models tend to destroy the spatial structure of the original frames when extracting temporal features. Some models avoid destroying the spatial features, but instead have excessive parameters that make it easier to overfit on the training dataset.

2.6.3. Visual Artefacts-Based Methods

Visual artefacts-based methods identify discrepancies caused by the blending operation of the deepfake generation process. These methods detect artefacts such as face warping, blending boundary artefacts, and head pose inconsistency. As they target more general artefacts, they tend to have better generalization performance than other methods. However, as deepfake technology progresses, these artefacts are gradually disappearing, making these methods less effective.

2.6.4. Camera Fingerprints-Based Methods

Camera fingerprints-based methods aim to recognize when face and background images are recorded with different devices. Different cameras leave different traces in the captured images, and these traces can be used for detection. However, camera fingerprint estimation requires a large number of images captured by different types of cameras, and there would be a decrease in accuracy when detecting images captured by unknown cameras. Image postprocessing can also negatively affect the performance of these methods. Recent research also shows that images can be generated with simulated camera fingerprints, deceiving methods that rely on camera fingerprints.

2.6.5. Biological Signals-Based Methods

Biological signals-based methods use the hidden biological signals of faces to detect deepfakes. Deepfake forgery algorithms have a hard time synthesizing these signals, making them a promising avenue for detection. There are two main approaches to detect deepfakes with biological signals: eye blinking-based and heart rate-based. Eye blinking methods use the abnormal blinking frequency of fake videos to detect them. However, recent deepfake technology has solved the problem of blink frequency, rendering this method no longer applicable for current deepfake detection tasks.

In the following section, we will focus on photoplethysmography-based methods, and discuss their potential as a reliable and accurate method for detecting deepfakes.

2.6.6. Photoplethysmography-Based Methods

Current deepfake technology has trouble fabricating the hidden heart rate of the face. Photoplethysmography-based approaches use the heart rate from the BVP signal extracted by rPPG methods to detect deepfake videos. In recent years, there have been several studies regarding deepfake detection using rPPG signal data.

Fernandes et al. [103] predict the heart rate of deepfakes by using neural ordinary differential equations [104]. However, no actual deepfake detection was explored.

FakeCatcher [97], uses a combination of the green channel and chrominance-based PPG, both taken from three face regions: forehead, the left cheek and and the right cheek. From these signals two classifiers are trained: A support vector machine (SVM) classifier, and a convolutional neural network (CNN) classifier. For the SVM classifier, transformations are applied to the PPGs and then signal characteristic feature sets are extracted from these transformed signals. The CNN is trained on PPG maps. The PPG maps are created from the green PPG extracted from the forehead region of interest, mapped into 32 subsections. The subsections are then combined into a single column of the PPG map, with each column representing a frame of the signal. The results of the SVM classifier and CNN classifier are then aggregated for a final result. FakeCatcher reports high detection accuracy even for low-quality videos, with 94,65% accuracy on the FaceForensics++ [71] dataset and 91,50% on the Celeb-DF [95] dataset.

DeepRhythm [105], uses a motion-magnified spatial-temporal representation (MMSTR) of the videos and classifies them with a dual-spatial-temporal attentional network. The MMSTR-method first removes the eyes and the background via landmark detection. Second, it performs a motion magnification algorithm on the resulting face. Finally, the motion magnified face is divided into ROI blocks and the RGB values of each block are averaged. In the resulting motion-magnified spatial-temporal (MMST) map, each column represents a frame of the video, and each row represents the motion-magnified temporal variation of one ROI block of the frame. The dual-spatial attention combines pre-selected ROI selection with an adaptive ROI selection CNN model. The dual-temporal attention uses a Long short-term memory (LSTM) model to predict the significance of the frame in fake detection from the MMST map, and combines it with a per-frame classification of the frame significance using the Meso-4 [106] architecture. The combined frame significances are used to weigh the frames contribution to the final classification, with more significant frames having a higher contribution. Finally, the attentions are combined to create an attentional MMST map, which is fed to the final deepfake detection model, which uses ResNet18 [107] for classification. *DeepRhythm* reports a 98% accuracy on the FaceForensics++ dataset, and a 64.1 % accuracy on the DFDC dataset.

DeepFakesON-Phys [108] inspired by *DeepPhys* [61] combines two CNN branches, a motion model and an appearance model, into a Convolutional Attention Network (CAN). For the motion model, the difference of the frame and the previous frame are normalized and fed as input. For the appearance model, the input is the frame normalized to zero mean and unitary standard deviation. The attention mask coming from the appearance model is shared with the motion model at two different points of the CAN. The final output layer of the motion model is the final output of the entire CAN, depicting the probability of the face being real. The *DeepFakesON-Phys* reports accuracy results of 98.7% for the Celeb-DF dataset, and 94.4% for the DFDC dataset.

Boccignone et al. [109] detect deepfakes by extracting the PPG from 100 ROI patches in the videos, in overlapping time-windows. From the PPG windows, 12 intra-patch BVP complexity measures and 8 inter-patch coherence measures are extracted as features. Using sequential floating forward selection (SFFS) [110] these features are reduced to 4 complexity features and 2 inter-patch coherence features. A SVM classifier is then used to classify the features into real or fake, with video-level predictions made by picking the most predicted label. Boccignone et al. report an accuracy of 94.45% on the FaceForensics++ dataset.

Liang et al. [111] propose a two-stage network for deepfake detection. They use a three-dimensional spatial-temporal map to represent the PPG signal, with one axis corresponding to the frames of the video, one to the combination of ROIs used, and one to the color channels. The videos are fed into the model in overlapping windows. The spatial-temporal maps are then filtered in the spatial-temporal filter module (STFM) to produce a one-dimensional PPG signal. Finally, in their adjacency interaction module, they measure the linear similarity error of the overlapping parts of the windows as additional supervisory information to help the STFM filter out the interference. use a convolutional feature extraction network and a bidirectional LSTM model to transfer information from adjacent windows to each other. Lian et al. report a multi-category deepfake categorization accuracy of 98.33% on the six categories of the FaceForensics++ dataset, showing an impressive ability to categorize the used deepfake method due to the unique rhythmic patterns of the different methods.

Wu et al. [112], use a similar spatial-temporal maps as Liang et al. [111] to represent the PPGs. They however improve on their method by Using a mask-guided local attention module that guides the attention of the spatial-temporal map by using a mask to select more significant parts of the map. The attention focused spatial-temporal maps are then fed to a temporal transformer module which exploits the long-distance information between adjacent video clips. Wu et al. report their method to have results of 99,38% in multi-category deepfake categorization on the six categories of the FaceForensics++ dataset.

Jeon et al. [113] explore a method of extracting rPPG signal from both the face and the neck region and using the band of interest (0-4 Hz) of the discrete fourier transform (DFT) of the two signals as feature vectors. L2 norm calculation is then performed on the DFT feature vectors. They report that the L2 norm distributions vary greatly between real and fake videos, giving a d-prime value of 2.32, showing clear difference in the distributions. However, the experiment was conducted using a small database (50 real, 50 fake) of high-resolution (1920x1080) videos, diminishing the reliability of the results.

Remote photoplethysmography-based detection methods show a good performance on various datasets, with fairly good generalization. However, the problem with rPPG methods is that the quality of the rPPG signal suffers in bad quality videos. Current deepfake algorithms do not account for biological signals. However, research of biologically plausible generative models is considered *FakeCatcher*. In the future, deepfake technology may accomplish realistic synthesized biological signals. Additionally, methods for modifying the rPPG signal of videos have been researched [114][115], or even completely synthesizing videos from an image and a rPPG signal [116], opening the door for spoofing the rPPG deepfake detectors.

2.6.7. Summary of Deepfake Detection

As the field of deepfakes and deepfake detection evolves, numerous different solutions are developed, each with their own strengths and weaknesses. It is therefore beneficial to look into using multiple methods in conjunction, to achieve better generalized results. A mix of different detectors will have a better result than a single detector, with the drawback of processing power and complexity. It is important to note that there is no be-all end-all solution for deepfake detection. Instead, the solutions must be carefully considered for each use case.

2.7. PPG Modification and Synthetic PPG Generation

Remote PPG systems, especially deep learning-based systems require large amounts of data for training the models. Moreover, the data needs to have some sort of reference PPG signals for the loss function. While various datasets such as the COHFACE [117] dataset exist, some research has been made in artifically modifying the PPG signals of existing videos. This method would enable the creation of artificial training videos with accurate rPPG signals. These systems could also be used to mask the heart rate from videos, for privacy reasons. However, the more nefarious use case of PPG modification would be to fool rPPG-based deepfake detection systems.

Bousefsaf et al. [68] propose a generation method of synthetic rPPG videos for creating training data for their model. They use a five-step procedure: A waveform model that is fitted to real rPPG signals using Fourier series, is used to construct a generic PPG wave. This wave is then used to create a two-second signal. A linear, cubic or quadratic tendency is added to the signal, with controlled frequency and amplitude. The resulting signal is then added to the video using vector repetition. Finally, random noise simulating the natural fluctuations due to camera noise is added to the video.

PulseEdit by Chen et al. [114] extracts the rPPG signal from the video, edits it by detrending and optimizing the similarity to the target rPPG signal, then adjusts the skin pixels of the face area. *PulseEdit* greatly increases the mean average error (MAE) on multiple rPPG methods when removing the rPPG signal. Additionally, PulseEdit was used to apply real PPG signals into deepfake videos, successfully reducing the accuracy of the *FakeCatcher* [97] rPPG-based deepfake detection system on the modified Celeb-DFv1 [95] dataset.

HeartTrack by Perepelkina et al. [67] is a rPPG extraction model that uses mathematically generated synthetic PPG signals to augment the training data of the heart rate estimation CNN. The mathematical model used to generate the signal considers the heart rate, heart cycle phase, breath cycle phase, magnitude of pulse signal, dicrotic pulse magnitude, breath signal magnitude, white noise, abd a standard deviation of the noise as parameters.

Wang et al. [116] generate entire synthetic faces as a means of rPPG dataset creation. They use real rPPGs that are applied into completely synthetic face videos generated from images. Results from experiments show that augmenting training data with synthetic videos can improve the performance on existing datasets.

Sun et al. present *PrivacyPhys* [115], a 3D-CNN based method to modify rPPG signals in facial videos. The method uses a visual fidelity term to ensure that the video is visually identical to the original, and a rPPG measurement term to ensure that rPPG measurement methods only capture the new rPPG signal, not the original one. The video is then updated using projected gradient descent to ensure that only the skin pixels are changed. Sun et al. report improved results and faster performance compared to *PulseEdit*.

While there are multiple studies on modifying the existing PPG of the videos, there are no studies which explicitly insert synthetic PPGs. However, it can be assumed that the process is as simple as changing the inserted PPG signals to synthetic ones.

3. IMPLEMENTATION

The implementation aims to create a classifier that can detect deepfake videos from authentic, using rPPG signals. The implemented classifier pipeline consists of a few steps, shown in Figure 7: signal extraction and preprocessing, windowing, feature extraction, training and finally validation. In this chapter the different steps of the implementation are explained in detail.

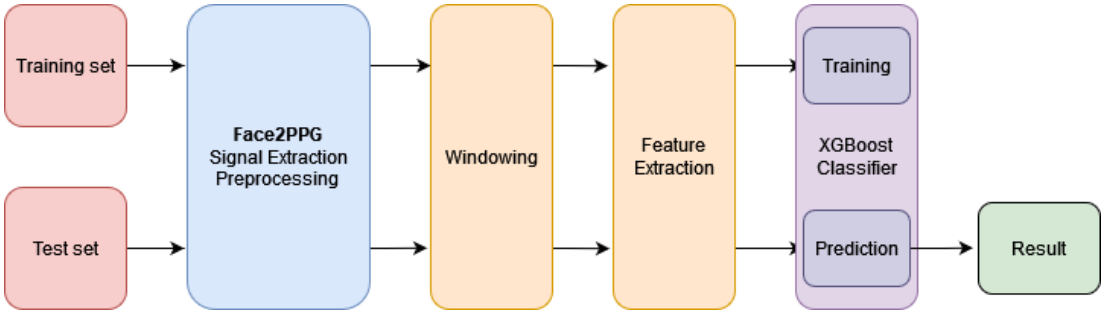


Figure 7. General working principle of the classifier.

3.1. Training Dataset

Training the classifier requires a sufficient amount of data, i. e. videos with real and fake faces. FaceForensics++ [71] was chosen as the dataset because of the unique actor per video, making it easier to avoid accidentally classifying by actor.

FaceForensics++ [71] is a dataset of facial forgeries consisting of five different face forgery methods, applied to 1000 video clips collected from 977 YouTube videos. The video quality varies from 480p to 1080p. The manipulation methods applied to these clips are *Face2Face* [118], *FaceSwap* [74], *FaceShifter* [82], *DeepFakes* [119] and *NeuralTextures* [120].

(a) Original (b) Deepfake

Figure 8. Example from the used *FaceForensics++* [71]) dataset.

The complete dataset consists of a 1000 video clips for each forgery method, with a total of 6000 videos. This equal distribution of videos is ideal for comparing classifier

performance against different forgery methods. However, for reduced processing times the dataset is reduced to only 250 real videos and 250 videos from the "DeepFakes" forgery method, with a total of 500 videos. The framerate of the videos is constant, so there is no need for standardization. Figure 8 shows an example from the used dataset.

3.2. PPG Extraction

In order to extract the rPPG signals from the facial videos, an unsupervised methodological pipeline called *Face2PPG* [4] was used. The unsupervised nature of the pipeline is beneficial, as the supervised rPPG extraction methods are not usually trained with faces that do not contain rPPG information and may hypothetically extract rPPG signals from faces where there are none, making classification of signals more difficult.

The *Face2PPG* pipeline follows the conventional rPPG method steps: face detection and alignment, ROI detection and selection, BVP extraction and finally spectral analysis and postprocessing to calculate parameters such as heart rate. However, for deepfake detection only the BVP pulse is needed, so the later steps are left out.

Face2PPG implements a robust deep-learning face detection method with the Single Shot Multibox Detection (SSD) network [24] and a deep learning facial landmark detector called Deep Alignment Network (DAN) [31]. These algorithms demonstrate a high performance even in challenging conditions [121]. These landmarks are used for a geometrical skin segmentation and normalization scheme using 85 facial landmark points for creating a facial mesh composed of 131 triangles. This facial mesh is then normalized to a frontal pose.

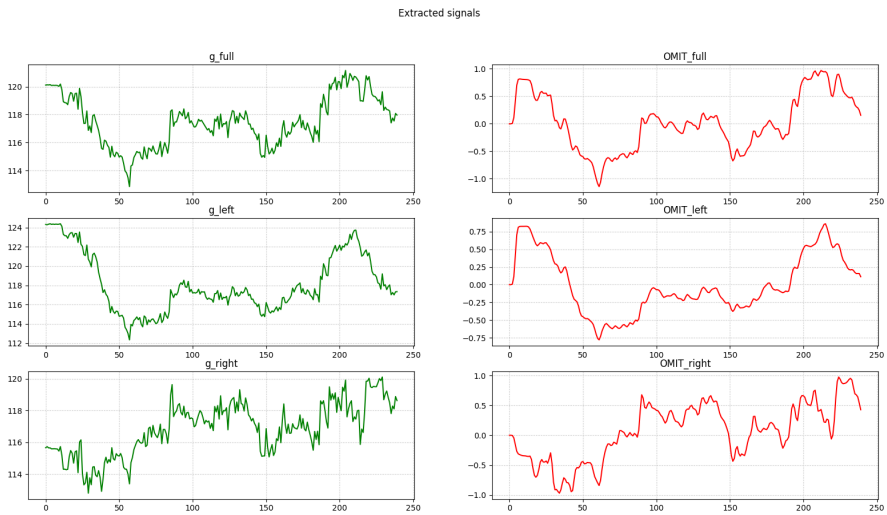


Figure 9. Example of the extracted rPPG signals using OMIT.

After the face normalization, the RGB signals are averaged from the entire face and the left and right sides of the face. To the three RGB signals, detrending and bandpass filtering is then applied to eliminate artifacts and to confine the signal to the frequency

band of interest. These filtered signals are then transformed to rPPG signals using Orthogonal Matrix Image Transform (OMIT) [4]. The resulting signals are then used for the classification and no further filtering is applied, to preserve the differing signal artifacts extracted from real and deepfake videos.

The resulting signals that are extracted are the green channel averages from the full face and the left and right sides (**gF**, **gL** and **gR**) and the transformed rPPG signals for the whole face and both sides using OMIT, POS, CHROM, or LGI method. This totals to six rPPG signals for each video. Figure 9 shows an example of the extracted signals when using OMIT.

3.3. Feature Extraction

To train the classifier model, various features are extracted from the rPPG signals, with 100 from the GREEN channel and 167 from the extracted rPPG signals. Additionally, the classifier was trained with features from every rPPG method, totaling 768 features. The extracted rPPG signals are first split into overlapping windows. The overlap for all scenarios is 10 frames, or 0.33 seconds, and the window length is tested with 4 and 8 seconds.

The extracted features can be classified into **intra-patch** and **inter-patch** features, depending on whether the features are computed from a single ROI patch or from multiple.

3.3.1. Intra-Patch Features

From each of the windowed signals, intra-patch features were extracted.

The statistical features were computed using the *Numpy Python* [122] library, and they include *min*, *max*, *mean*, *root mean square* (RMS), *variance* (VAR), *standard deviation* (STD), *power*, *peak*, *peak-to-peak* (PTP), *crest factor*. Additionally, the *SciPy* [123] library was used to compute the *skew* and *kurtosis* of the signal.

For computing the complexity of the time-series and to extract entropy and fractal features, *AntroPy* [124] package was used. The time-series and entropy features include the *zero-crossing rate*, *Hjorth mobility and complexity*, *spectral entropy*, *approximate entropy*, *sample entropy*, *permutation entropy*, *singular value decomposition entropy* [109]. The fractal features include *detrended fluctuation information*, *Katz fractal dimension*, *Petrosian fractal dimension* and *Higuchi fractal dimension* of the window [109].

For the non-GREEN signals, *Neurokit2* [125] and *HeartPy* [126] were used to compute various heart related features, including *heart rate* (HR), *breathing rate* (BR), *interbeat interval* (IBI), *standard deviation of NN intervals and differences* (SDNN, SDSD), *Poincaré plot information* (SD1, SD2, SD1SD2), *difference between RR intervals* (pNN20 and pNN50), *frequency domain components* (VLF, LF, HF, UHF), and *heart rate mean absolute deviation* (HR_MAD) [127][128].

3.3.2. Inter-Patch Features

The difference of the right and left sides of the intra-patch features are then computed as additional inter-patch features. Additionally, the *spectral similarity* and *mutual information* of the right and left signals are calculated using *Scikit-learn* [129] and *SciPy*. The mean and standard deviation of these values are then used as additional features [109].

3.4. PPG Classifier

For the classification of the feature set a Python library *XGBoost* [5] was used, along with the *Scikit-learn* [129] library. XGBoost is a gradient boosted decision tree (GBDT) machine learning library. GBDT is a decision tree ensemble learning algorithm for classification, meaning it combines multiple machine learning algorithms to obtain better results. In gradient boosting, the process of additively generating weak models is formalized as a gradient descent algorithm over an objective function. Each model is trained to correct the errors of previous models, gradually increasing prediction accuracy. In the implementation, the XGBoost classifier is used with the default parameters of the library.

4. EVALUATION

The classifier is evaluated by cross-validation, with the 500-video dataset split by stratified group k-fold split into six folds. The windows of a single video make up one group of the split. Each split, one fold is left as test data and others are used for training. The resulting window predictions are then pooled for each video to determine the per-video prediction by majority vote. The percentage of windows agreeing with the result is the *video prediction confidence*. For each rPPG method and window length, cross-validation is performed separately.

The results of the classifier are graphed on a receiver operating characteristic (ROC) curve, to illustrate the ability of the classifier. The ROC curve consists of the true positive rate (TPR) plotted against the false positive rate (FPR) at various threshold settings. From the ROC curve, the area under the curve (AUC) is calculated for a numerical metric of performance, invariant of the classification threshold. Additionally, the balanced accuracy of the system is calculated. All the metrics are averaged over the six folds of the cross-validation.

All of the ROC curves for the results listed are included in the appendix.

4.1. Results

This section lists all of the results for each window length.

4.1.1. 8 Second Window

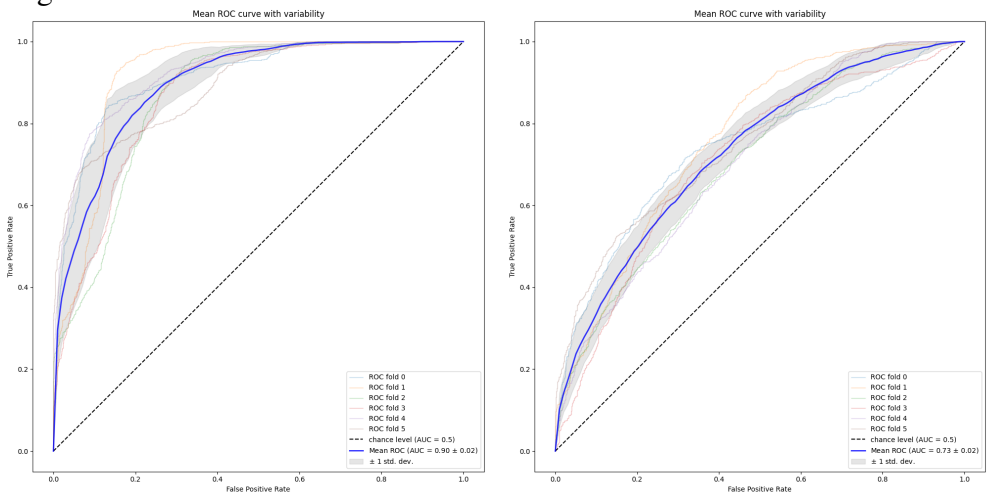
With the 8 second windows, the total number of windows in the dataset was 13291 windows. The results are listed in Table 2. The best results for each category are marked bold, excluding "All methods" and average file confidence.

rPPG method	window AUC %	window ACC %	video AUC %	video ACC	video conf. %
ALL	**90 ±2** %	**82** %	**85 ±4** %	85 %	94 %
OMIT+GREEN	89 ±5 %	**82** %	83 ±6 %	**87** %	95 %
POS+GREEN	**90 ±3** %	81 %	81 ±4 %	81 %	94 %
LGI+GREEN	**90 ±2** %	**82** %	**85 ±4** %	85 %	95 %
CHROM+GREEN	89 ±5 %	81 %	83 ±4 %	83 %	94 %
GREEN	89 ±4 %	81 %	84 ±4 %	84 %	95 %
OMIT	62 ±4 %	59 %	60 ±6 %	60 %	84 %
POS	64 ±5 %	61 %	64 ±5 %	64 %	85 %
LGI	73 ±2 %	66 %	68 ±4 %	68 %	89 %
CHROM	58 ±3 %	57 %	61 ±2 %	61 %	83 %

Table 2. Results for 8 second windows cross validation.

Figure 10 shows the ROC curves for the LGI + GREEN and LGI methods using 8 second windowing.

Figure 10. 8 second window ROC curve for LGI + GREEN and LGI



(a) 8 seconds, LGI + GREEN (b) 8 seconds, LGI

4.1.2. 4 Second Window

With the 4 second windows, the total number of windows in the dataset was 19251 windows. The results are listed in Table 3. The best results for each category are marked bold, excluding "All methods" and average file confidence.

rPPG method	window AUC %	window ACC %	video AUC %	video ACC	video conf. %
ALL	**89 ±3** %	**81** %	**86 ±4** %	**86** %	93 %
OMIT+GREEN	**88 ±3** %	**81** %	85 ±4 %	85 %	92 %
POS+GREEN	**88 ±3** %	80 %	84 ±5 %	84 %	91 %
LGI+GREEN	**88 ±3** %	**81** %	**86 ±5** %	**86** %	88 %
CHROM+GREEN	86 ±4 %	78 %	83 ±4 %	83 %	91 %
GREEN	85 ±3 %	78 %	81 ±4 %	81 %	94 %
OMIT	58 ±6 %	55 %	58 ±7 %	58 %	74 %
POS	63 ±2 %	59 %	61 ±4 %	61 %	73 %
LGI	73 ±7 %	67 %	69 ±6 %	69 %	81 %
CHROM	58 ±5 %	56 %	61 ±6 %	61 %	74 %

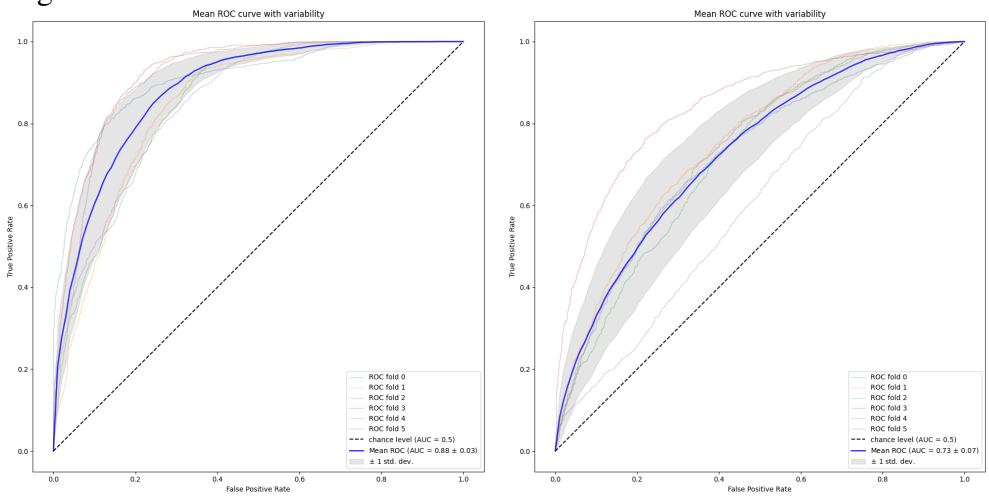
Table 3. Results for 4 second windows cross validation.

Figure 11 shows the ROC curves for the LGI + GREEN and LGI methods using 4 second windowing.

4.1.3. 2 Second Window

With the 2 second windows, the total number of windows in the dataset was 22225 windows. The results are listed in Table 4. The best results for each category are marked bold, excluding "All methods" and average file confidence.

Figure 11. 4 second window ROC curve for LGI + GREEN and LGI



(a) 4 seconds, LGI + GREEN (b) 4 seconds, LGI

rPPG method	window AUC %	window ACC %	video AUC %	video ACC	video conf. %
ALL	**88 ±2** %	**80** %	**86 ±3** %	**86** %	89 %
OMIT+GREEN	86 ±2 %	78 %	**85 ±5** %	**85** %	88 %
POS+GREEN	**87 ±3** %	**79** %	84 ±2 %	84 %	88 %
LGI+GREEN	**87 ±2** %	**79** %	83 ±5 %	83 %	88 %
CHROM+GREEN	85 ±3 %	76 %	83 ±4 %	83 %	87 %
GREEN	85 ±3 %	77 %	82 ±4 %	82 %	86 %
OMIT	59 ±3 %	57 %	61 ±4 %	61 %	72 %
POS	62 ±4 %	58 %	63 ±7 %	63 %	70 %
LGI	73 ±4 %	66 %	72 ±4 %	72 %	77 %
CHROM	58 ±4 %	56 %	62 ±7 %	62 %	71 %

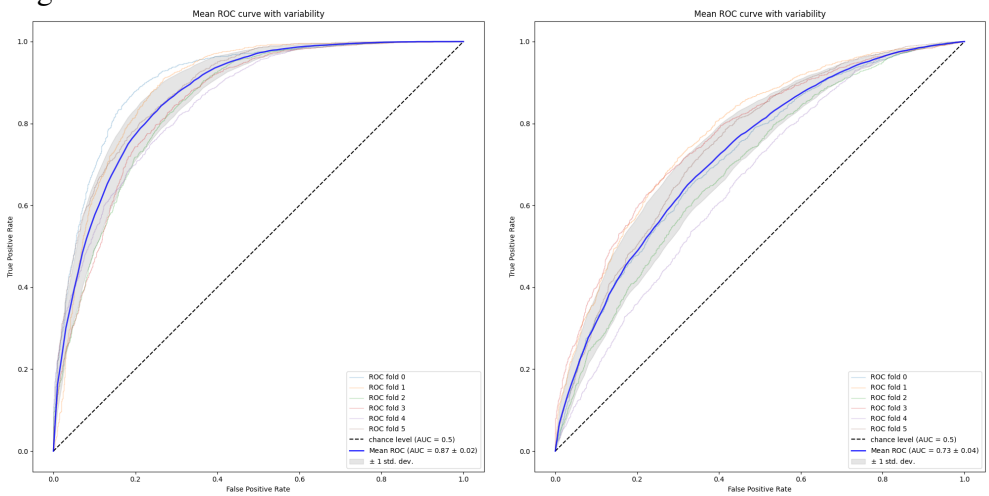
Table 4. Results for 2 second windows cross validation.

Figure 12 shows the ROC curves for the LGI + GREEN and LGI methods using 2 second windowing.

4.2. Analysis

From the results we can see that the classification accuracy is fairly similar for all the methods that include GREEN, especially OMIT, POS and LGI. Notable, the GREEN method alone is able to classify the videos with decent accuracy and the addition of a more sophisticated method does not make much of a difference. In contrast, the rPPG methods without GREEN perform significantly worse than the GREEN signal. This would suggest that the rPPG conversion methods remove relevant features from the signals that can be used by the classification. It can be noted that the LGI method performs better than the other conversion methods, hinting that it preserves the classifiable features of the signal better.

Figure 12. 2 second window ROC curve for LGI + GREEN and LGI



(a) 2 seconds, LGI + GREEN (b) 2 seconds, LGI

It can also be noted that as the heart related features are not extracted from the GREEN signal, the classifier is not able to use them efficiently for the classification. This is an unexpected resolution, as the heart rate and other heart related features are expected to contain valuable information for classification. By observing the feature importances given by the XGBoost classifier, we find that the features with the highest importances often contain the Hjorth mobility and complexity, various entropies (approximate, SVD, permutation, sample) and the fractal dimensions (Petrosian, Higuchi, Katz). The most important features are often intra-patch features, with ROI difference features rarely having high effect in the classification. These results are in consensus with the results obtained by Boccignone et. al. [109], who also report intra-patch complexity parameters to have high significance.

Remarkably, tests with two second windows have almost as good performance as with four or eight seconds. This can be assumed to be the cause of a larger training dataset due to the larger number of windows. However, the benefit of a shorter window length and especially shorter video length can be substantial in many use cases, making the results interesting. It should be noted that the average file classification confidence decreases with the window length.

One large shortcoming of this work is the small dataset and the usage of only one deepfake method, making the classifier unlikely to be able to generalize to other methods. Testing with larger datasets and other methods is a clear future development path of this work.

It might also be beneficial to prune out unnecessary features that have less or no effect on the classification result, to improve the performance and speed of the model. Additionally, the computational performance of the system and other rPPG-based deepfake detection methods could be an interesting and beneficial area of research.

4.2.1. Comparison with Other Research

As shown by Table 5, the resulting classifier performs worse compared to Boccignone et. al. [109], on the FaceForensics++ "DeepFakes" dataset. The result is expected, as the amount of training data is less than the full 1000 videos used by [109]. It should be noted that the AUC comparison is only for the K-fold cross-validation of the "DeepFakes" set, not the whole dataset. The classifier has adequate performance for the small training dataset. The worse performance of the can potentially be because of the small number of "patches", as the difference of patch features is diminished by the already largely averaged patch of half a face, bringing the averages closer to the average of the whole face. Inversely, the Face2PPG is a more sophisticated system for extracting the rPPG signals, and the quality of the actual signal is more likely closer to the actual heart rate.

Method	AUC %
Ours	85 ±4 %
[109]	**90.68 ±0.61** %

Table 5. Comparison of the classifier with another purely rPPG signal features-based work by Boccignone et. al. [109].

5. SUMMARY

The objective of this thesis was to design a classifier for deepfake videos using remote photoplethysmography as the main classification metric, and to study the effect of different RGB to BVP methods and of different feature window lengths on the classification performance. The existing literature of remote photoplethysmography, deepfakes and deepfake detection was studied and an implementation based on *Face2PPG* and *XGBoost* was developed. The resulting classifier was tested with different RGB to BVP methods including GREEN, OMIT, POS, LGI and CHROM, and with window lengths of 2, 4 and 8 seconds.

The results showed that the implementation is capable of classifying deepfake videos with 85 % accuracy, somewhat comparable to another similar method trained on a larger dataset. Of the tested RGB to BVP methods, OMIT, POS and LGI were found to be mostly identical in performance. However, the GREEN signal was found to be important for classification. LGI was also found to be slightly better than the other methods when the GREEN signal was not present. For window length, no minimal difference was found in the classification accuracy even with a window length of only two seconds. All results point out that the actual heart rate extracted from the signal is not significant in the performance of the developed classifier, instead it focuses on other features of the extracted signal.

The developed classifier was trained with a small dataset of only one deepfake method, making it likely not very good at generalization. The classifier works as a proof-of-concept for later development of similar, better models.

6. REFERENCES

[1] Masood M., Nawaz M., Malik K.M., Javed A. & Irtaza A. (2021), Deepfakes generation and detection: State-of-the-art, open challenges, countermeasures, and way forward. URL: https://arxiv.org/abs/2103.00484.

[2] Oltermann P. (2022) European politicians duped into deepfake video calls with mayor of kyiv. The Guardian URL: https://www.theguardian.com/world/2022/jun/25/european-leaders-deepfake-video-calls-mayor-of-kyiv-vitali-klitschko.

[3] Allen J. (2007) Photoplethysmography and its application in clinical physiological measurement. Physiological measurement 28, p. R1.

[4] Casado C.Á. & López M.B. (2022), Face2ppg: An unsupervised pipeline for blood volume pulse extraction from faces. URL: https://arxiv.org/abs/2202.04101.

[5] Chen T. & Guestrin C. (2016) XGBoost. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM. URL: https://doi.org/10.1145%2F2939672.2939785.

[6] Kamal A., Harness J., Irving G. & Mearns A. (1989) Skin photoplethysmography — a review. Computer Methods and Programs in Biomedicine 28, pp. 257–269. URL: https://www.sciencedirect.com/science/article/pii/0169260789901594.

[7] Sinhal R., Singh K. & Raghuwanshi M. (2020) An overview of remote photoplethysmography methods for vital sign monitoring. In: Computer Vision and Machine Intelligence in Medical Image Analysis: International Symposium, ISCMM 2019, Springer, pp. 21–31.

[8] Ni A., Azarang A. & Kehtarnavaz N. (2021) A review of deep learning-based contactless heart rate measurement methods. Sensors 21, p. 3719.

[9] Huang R.Y. & Dung L.R. (2015) A motion-robust contactless photoplethysmography using chrominance and adaptive filtering. In: 2015 IEEE Biomedical Circuits and Systems Conference (BioCAS), pp. 1–4.

[10] McDuff D., Gontarek S. & Picard R. (2014) Remote measurement of cognitive stress via heart rate variability. Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual International Conference 2014, p. 2957—2960. URL: https://doi.org/10.1109/EMBC.2014.6944243.

[11] Blöcher T., Schneider J., Schinle M. & Stork W. (2017) An online ppgi approach for camera based heart rate monitoring using beat-to-beat detection. In: 2017 IEEE Sensors Applications Symposium (SAS), pp. 1–6.

[12] Wu B.F., Chu Y.W., Huang P.W., Chung M.L. & Lin T.M. (2016) A motion robust remote-ppg approach to driver's health state monitoring. In: ACCV Workshops.

[13] Zhou K., Krause S., Blöcher T. & Stork W. (2020) Enhancing remote-ppg pulse extraction in disturbance scenarios utilizing spectral characteristics. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 1130–1138.

[14] Chandrasekhar A., Yavarimanesh M., Natarajan K., Hahn J.O. & Mukkamala R. (2020) Ppg sensor contact pressure should be taken into account for cuff-less blood pressure measurement. IEEE Transactions on Biomedical Engineering 67, pp. 3134–3140.

[15] de Haan G. & Jeanne V. (2013) Robust pulse rate from chrominance-based rppg. IEEE Transactions on Biomedical Engineering 60, pp. 2878–2886.

[16] Kumar A., Kaur A. & Kumar M. (2019) Face detection techniques: a review. Artificial Intelligence Review 52, pp. 927–948.

[17] Zou Z., Chen K., Shi Z., Guo Y. & Ye J. (2023) Object detection in 20 years: A survey. Proceedings of the IEEE .

[18] Ahonen T., Hadid A. & Pietikäinen M. (2004) Face recognition with local binary patterns. In: Computer Vision-ECCV 2004: 8th European Conference on Computer Vision, Prague, Czech Republic, May 11-14, 2004. Proceedings, Part I 8, Springer, pp. 469–481.

[19] Dalal N. & Triggs B. (2005) Histograms of oriented gradients for human detection. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), vol. 1, vol. 1, pp. 886–893 vol. 1.

[20] Krizhevsky A., Sutskever I. & Hinton G.E. (2017) Imagenet classification with deep convolutional neural networks. Communications of the ACM 60, pp. 84–90.

[21] Bradski G. (2000) The OpenCV Library. Dr. Dobb's Journal of Software Tools .

[22] Rouast P.V., Adam M.T., Chiong R., Cornforth D. & Lux E. (2018) Remote heart rate measurement using low-cost rgb face video: a technical literature review. Frontiers of Computer Science 12, pp. 858–872.

[23] Lee K.Z., Hung P.C. & Tsai L.W. (2012) Contact-free heart rate measurement using a camera. In: 2012 Ninth Conference on Computer and Robot Vision, pp. 147–152.

[24] Liu W., Anguelov D., Erhan D., Szegedy C., Reed S., Fu C.Y. & Berg A.C. (2016) SSD: Single shot MultiBox detector. In: Computer Vision – ECCV 2016, Springer International Publishing, pp. 21–37. URL: https://doi.org/10.1007%2F978-3-319-46448-0_2.

[25] Jin X. & Tan X. (2017) Face alignment in-the-wild: A survey. Computer Vision and Image Understanding 162, pp. 1–22. URL: https://www.sciencedirect.com/science/article/pii/S1077314217301455.

[26] Alvarez Casado C. & Bordallo Lopez M. (2021) Real-time face alignment: evaluation methods, training strategies and implementation optimization. Journal of Real-Time Image Processing 18, pp. 2239–2267.

[27] Kazemi V. & Sullivan J. (2014) One millisecond face alignment with an ensemble of regression trees. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1867–1874.

[28] Ren S., Cao X., Wei Y. & Sun J. (2014) Face alignment at 3000 fps via regressing local binary features. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1685–1692.

[29] Cootes T.F., Edwards G.J. & Taylor C.J. (2001) Active appearance models. IEEE Transactions on pattern analysis and machine intelligence 23, pp. 681–685.

[30] Asthana A., Zafeiriou S., Cheng S. & Pantic M. (2013) Robust discriminative response map fitting with constrained local models. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3444–3451.

[31] Kowalski M., Naruniec J. & Trzcinski T. (2017), Deep alignment network: A convolutional neural network for robust face alignment.

[32] Kim D.Y., Lee K. & Sohn C.B. (2021) Assessment of roi selection for facial video-based rppg. Sensors 21. URL: https://www.mdpi.com/1424-8220/21/23/7923.

[33] Pirnar, Žan F., Miha P. & Primož (2021) Performance evaluation of rppg approaches with and without the region-of-interest localization step. Applied Sciences 11. URL: https://www.mdpi.com/2076-3417/11/8/3467.

[34] Poh M.Z., McDuff D.J. & Picard R.W. (2010) Non-contact, automated cardiac pulse measurements using video imaging and blind source separation. Optics express 18, pp. 10762–10774.

[35] Poh M.Z., McDuff D.J. & Picard R.W. (2011) Advancements in noncontact, multiparameter physiological measurements using a webcam. IEEE Transactions on Biomedical Engineering 58, pp. 7–11.

[36] Verkruysse W., Svaasand L.O. & Nelson J.S. (2008) Remote plethysmographic imaging using ambient light. Opt. Express 16, pp. 21434–21445. URL: https://opg.optica.org/oe/abstract.cfm?URI=oe-16-26-21434.

[37] Lempe G., Zaunseder S., Wirthgen T., Zipser S. & Malberg H. (2013) Roi selection for remote photoplethysmography. In: Bildverarbeitung für die Medizin 2013: Algorithmen-Systeme-Anwendungen. Proceedings des Workshops vom 3. bis 5. März 2013 in Heidelberg, Springer, pp. 99–103.

[38] Po L.M., Feng L., Li Y., Xu X., Cheung T.C.H. & Cheung K.W. (2018) Block-based adaptive roi for remote photoplethysmography. Multimedia Tools and Applications 77, pp. 6503–6529.

[39] Bousefsaf F., Maaoui C. & Pruski A. (2013) Continuous wavelet filtering on webcam photoplethysmographic signals to remotely assess the instantaneous heart rate. Biomedical Signal Processing and Control 8, pp. 568–574. URL: https://www.sciencedirect.com/science/article/pii/S1746809413000840.

[40] Wang G. (2021) Influence of roi selection for remote photoplethysmography with singular spectrum analysis. In: 2021 IEEE International Conference on Artificial Intelligence and Industrial Design (AIID), pp. 416–420.

[41] Kumar M., Veeraraghavan A. & Sabharwal A. (2015) Distanceppg: Robust non-contact vital signs monitoring using a camera. Biomedical optics express 6, pp. 1565–1588.

[42] Unakafov A.M. (2018) Pulse rate estimation using imaging photoplethysmography: generic framework and comparison of methods on a publicly available dataset. Biomedical Physics & Engineering Express 4, p. 045001. URL: https://dx.doi.org/10.1088/2057-1976/aabd09.

[43] Tarassenko L., Villarroel M., Guazzi A., Jorge J., Clifton D. & Pugh C. (2014) Non-contact video-based vital sign monitoring using ambient light and auto-regressive models. Physiological measurement 35, p. 807.

[44] Hülsbusch M. (2008) An image-based functional method for opto-electronic detection of skin-perfusion. RWTH Aachen (in German) .

[45] Feng L., Po L.M., Xu X., Li Y. & Ma R. (2014) Motion-resistant remote imaging photoplethysmography based on the optical properties of skin. IEEE Transactions on Circuits and Systems for Video Technology 25, pp. 879–891.

[46] Abdi H. & Williams L.J. (2010) Principal component analysis. Wiley interdisciplinary reviews: computational statistics 2, pp. 433–459.

[47] Lewandowska M., Rumiński J., Kocejko T. & Nowak J. (2011) Measuring pulse rate with a webcam—a non-contact method for evaluating cardiac activity. In: 2011 federated conference on computer science and information systems (FedCSIS), IEEE, pp. 405–410.

[48] Comon P. (1994) Independent component analysis, a new concept? Signal Processing 36, pp. 287–314. URL: https://www.sciencedirect.

com/science/article/pii/0165168494900299, higher Order Statistics.

[49] Cardoso J.F. (1999) High-order contrasts for independent component analysis. Neural Computation 11, pp. 157–192.

[50] Wei L., Tian Y., Wang Y., Ebrahimi T. & Huang T. (2013) Automatic webcam-based human heart rate measurements using laplacian eigenmap. In: Computer Vision–ACCV 2012: 11th Asian Conference on Computer Vision, Daejeon, Korea, November 5-9, 2012, Revised Selected Papers, Part II 11, Springer, pp. 281–292.

[51] Agrafiotis D.K. (2003) Stochastic proximity embedding. Journal of computational chemistry 24, pp. 1215–1221.

[52] de Haan G. & van Leest A. (2014) Improved motion robustness of remote-ppg by using the blood volume pulse signature. Physiological Measurement 35, pp. 1913 – 1926.

[53] Wang W., Stuijk S. & de Haan G. (2016) A novel algorithm for remote photoplethysmography: Spatial subspace rotation. IEEE Transactions on Biomedical Engineering 63, pp. 1974–1984.

[54] Wang W., den Brinker A.C., Stuijk S. & de Haan G. (2017) Algorithmic principles of remote ppg. IEEE Transactions on Biomedical Engineering 64, pp. 1479–1491.

[55] Pilz C.S., Zaunseder S., Krajewski J. & Blazek V. (2018) Local group invariance for heart rate estimation from face videos in the wild. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 1335–13358.

[56] van Es V.A.A., Lopata R.G.P., Scilingo E.P. & Nardelli M. (2023) Contactless cardiovascular assessment by imaging photoplethysmography: A comparison with wearable monitoring. Sensors 23. URL: https://www.mdpi.com/1424-8220/23/3/1505.

[57] Haugg F., Elgendi M. & Menon C. (2022) Effectiveness of remote ppg construction methods: A preliminary analysis. Bioengineering 9, p. 485.

[58] Qiu Y., Liu Y., Arteaga-Falconi J., Dong H. & Saddik A.E. (2019) Evm-cnn: Real-time contactless heart rate estimation from facial video. IEEE Transactions on Multimedia 21, pp. 1778–1787.

[59] Luguev T., Seuß D. & Garbas J.U. (2020) Deep learning based affective sensing with remote photoplethysmography. In: 2020 54th Annual Conference on Information Sciences and Systems (CISS), pp. 1–4.

[60] Zhan Q., Wang W. & de Haan G. (2020) Analysis of cnn-based remote-ppg to understand limitations and sensitivities. Biomed. Opt. Express 11, pp. 1268–1283. URL: https://opg.optica.org/boe/abstract.cfm?URI=boe-11-3-1268.

[61] Chen W. & McDuff D. (2018), Deepphys: Video-based physiological measurement using convolutional attention networks.

[62] Reiss A., Indlekofer I., Schmidt P. & Van Laerhoven K. (2019) Deep ppg: Large-scale heart rate estimation with convolutional neural networks. Sensors 19. URL: https://www.mdpi.com/1424-8220/19/14/3079.

[63] Lee E., Chen E. & Lee C.Y. (2020), Meta-rppg: Remote heart rate estimation using a transductive meta-learner.

[64] Špetlík R., Franc V. & Matas J. (2018) Visual heart rate estimation with convolutional neural network. In: Proceedings of the british machine vision conference, Newcastle, UK, pp. 3–6.

[65] Yu Z., Peng W., Li X., Hong X. & Zhao G. (2019), Remote heart rate measurement from highly compressed facial videos: an end-to-end deep learning solution with video enhancement.

[66] Yu Z., Li X. & Zhao G. (2019), Remote photoplethysmograph signal measurement from facial videos using spatio-temporal networks.

[67] Perepelkina O., Artemyev M., Churikova M. & Grinenko M. (2020) Hearttrack: Convolutional neural network for remote video-based heart rate monitoring. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 1163–1171.

[68] Bousefsaf F., Pruski A. & Maaoui C. (2019) 3d convolutional neural networks for remote pulse rate measurement and mapping from facial video. Applied Sciences 9. URL: https://www.mdpi.com/2076-3417/9/20/4364.

[69] Liu S.Q. & Yuen P.C. (2020) A general remote photoplethysmography estimator with spatiotemporal convolutional network. In: 2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020), pp. 481–488.

[70] Verdoliva L. (2020) Media forensics and deepfakes: An overview. IEEE Journal of Selected Topics in Signal Processing 14, pp. 910–932.

[71] Rossler A., Cozzolino D., Verdoliva L., Riess C., Thies J. & Nießner M. (2019) Faceforensics++: Learning to detect manipulated facial images. In: Proceedings of the IEEE/CVF international conference on computer vision, pp. 1–11.

[72] Moran L. Ye olde photoshoppe: The first ever altered images. Daily Mail URL: https://www.dailymail.co.uk/news/article-2107109/Iconic-Abraham-Lincoln-portrait-revealed-TWO-pictures-stitched-together.html.

[73] Fakeapp. https://www.faceapp.com/. Accessed: 2023-05-18.

[74] Kowalski M., Faceswap. https://github.com/MarekKowalski/FaceSwap/. Accessed: 2023-05-18.

[75] Zao. https://zaodownload.com/. Accessed: 2023-05-18.

[76] Seow J.W., Lim M.K., Phan R.C.W. & Liu J.K. (2022) A comprehensive overview of deepfake: Generation, detection, datasets, and opportunities. Neurocomputing .

[77] Karras T., Laine S., Aittala M., Hellsten J., Lehtinen J. & Aila T. (2020) Analyzing and improving the image quality of stylegan. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 8110–8119.

[78] Choi Y., Choi M., Kim M., Ha J.W., Kim S. & Choo J. (2018) Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8789–8797.

[79] Li T., Qian R., Dong C., Liu S., Yan Q., Zhu W. & Lin L. (2018) Beautygan: Instance-level facial makeup transfer with deep generative adversarial network. In: Proceedings of the 26th ACM international conference on Multimedia, pp. 645–653.

[80] Jo Y. & Park J. (2019) Sc-fegan: Face editing generative adversarial network with user's sketch and color. In: Proceedings of the IEEE/CVF international conference on computer vision, pp. 1745–1753.

[81] Natsume R., Yatagawa T. & Morishima S. (2019) Fsnet: An identity-aware generative model for image-based face swapping. In: Computer Vision–ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers, Part VI 14, Springer, pp. 117–132.

[82] Li L., Bao J., Yang H., Chen D. & Wen F. (2019) Faceshifter: Towards high fidelity and occlusion aware face swapping. arXiv preprint arXiv:1912.13457 .

[83] Perov I., Gao D., Chervoniy N., Liu K., Marangonda S., Umé C., Dpfks M., Facenheim C.S., RP L., Jiang J., Zhang S., Wu P., Zhou B. & Zhang W. (2021), Deepfacelab: Integrated, flexible and extensible face-swapping framework.

[84] Zhang J., Zeng X., Pan Y., Liu Y., Ding Y. & Fan C. (2019) Faceswapnet: Landmark guided many-to-many face reenactment. arXiv preprint arXiv:1905.11805 2, p. 3.

[85] Rana M.S., Nobi M.N., Murali B. & Sung A.H. (2022) Deepfake detection: A systematic literature review. IEEE Access .

[86] Dolhansky B., Howes R., Pflaum B., Baram N. & Ferrer C.C. (2019), The deepfake detection challenge (dfdc) preview dataset.

[87] Dolhansky B., Bitton J., Pflaum B., Lu J., Howes R., Wang M. & Ferrer C.C. (2020), The deepfake detection challenge dataset.

[88] Korshunov P. & Marcel S. (2020) Deepfake detection: humans vs. machines. arXiv preprint arXiv:2009.03155 .

[89] Yu P., Xia Z., Fei J. & Lu Y. (2021) A survey on deepfake video detection. Iet Biometrics 10, pp. 607–624.

[90] Yang X., Li Y. & Lyu S. (2018), Exposing deep fakes using inconsistent head poses.

[91] Korshunov P. & Marcel S. (2018), Deepfakes: a new threat to face recognition? assessment and detection.

[92] Khodabakhsh A., Ramachandra R., Raja K., Wasnik P. & Busch C. (2018) Fake face detection methods: Can they be generalized? In: 2018 International Conference of the Biometrics Special Interest Group (BIOSIG), pp. 1–6.

[93] Rössler A., Cozzolino D., Verdoliva L., Riess C., Thies J. & Nießner M. (2018), Faceforensics: A large-scale video dataset for forgery detection in human faces.

[94] Contributing data to deepfake detection research. `https://ai.googleblog.com/2019/09/contributing-data-to-deepfake-detection.html`. Accessed: 2023-05-11.

[95] Li Y., Sun P., Qi H. & Lyu S. (2020) Celeb-DF: A Large-scale Challenging Dataset for DeepFake Forensics. In: IEEE Conference on Computer Vision and Patten Recognition (CVPR), Seattle, WA, United States.

[96] Gupta P., Chugh K., Dhall A. & Subramanian R. (2020), The eyes know it: Fakeet – an eye-tracking database to understand deepfake perception.

[97] Ciftci U., Demir I. & Yin L. (202) Fakecatcher: Detection of synthetic portrait videos using biological signals. IEEE Transactions on Pattern Analysis & Machine Intelligence .

[98] Jiang L., Li R., Wu W., Qian C. & Loy C.C. (2020), Deeperforensics-1.0: A large-scale dataset for real-world face forgery detection. URL: `https://arxiv.org/abs/2001.03024`.

[99] Zi B., Chang M., Chen J., Ma X. & Jiang Y.G. (2021), Wilddeepfake: A challenging real-world dataset for deepfake detection. URL: `https://arxiv.org/abs/2101.01456`.

[100] He Y., Gan B., Chen S., Zhou Y., Yin G., Song L., Sheng L., Shao J. & Liu Z. (2021) Forgerynet: A versatile benchmark for comprehensive forgery analysis. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 4360–4369.

[101] Kwon P., You J., Nam G., Park S. & Chae G. (2021), Kodf: A large-scale korean deepfake detection dataset.

[102] Khalid H., Tariq S., Kim M. & Woo S.S. (2022), Fakeavceleb: A novel audio-video multimodal deepfake dataset.

[103] Fernandes S., Raj S., Ortiz E., Vintila I., Salter M., Urosevic G. & Jha S. (2019) Predicting heart rate variations of deepfake videos using neural ode. In: 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), pp. 1721–1729.

[104] Chen R.T.Q., Rubanova Y., Bettencourt J. & Duvenaud D. (2019), Neural ordinary differential equations.

[105] Qi H., Guo Q., Juefei-Xu F., Xie X., Ma L., Feng W., Liu Y. & Zhao J. (2020), Deeprhythm: Exposing deepfakes with attentional visual heartbeat rhythms. URL: https://arxiv.org/abs/2006.07634.

[106] Afchar D., Nozick V., Yamagishi J. & Echizen I. (2018) MesoNet: a compact facial video forgery detection network. In: 2018 IEEE International Workshop on Information Forensics and Security (WIFS), IEEE. URL: https://doi.org/10.1109%2Fwifs.2018.8630761.

[107] He K., Zhang X., Ren S. & Sun J. (2015), Deep residual learning for image recognition.

[108] Hernandez-Ortega J., Tolosana R., Fierrez J. & Morales A. (2020), Deepfakeson-phys: Deepfakes detection based on heart rate estimation.

[109] Boccignone G., Bursic S., Cuculo V., D'Amelio A., Grossi G., Lanzarotti R. & Patania S. (2022) Deepfakes have no heart: A simple rppg-based method to reveal fake videos. In: Image Analysis and Processing – ICIAP 2022: 21st International Conference, Lecce, Italy, May 23–27, 2022, Proceedings, Part II, Springer-Verlag, Berlin, Heidelberg, p. 186–195. URL: https://doi.org/10.1007/978-3-031-06430-2_16.

[110] Pudil P., Novovičová J. & Kittler J. (1994) Floating search methods in feature selection. Pattern Recognition Letters 15, pp. 1119–1125. URL: https://www.sciencedirect.com/science/article/pii/0167865594901279.

[111] Liang J. & Deng W. (2021) Identifying rhythmic patterns for face forgery detection and categorization. In: 2021 IEEE International Joint Conference on Biometrics (IJCB), pp. 1–8.

[112] Wu J., Zhu Y., Jiang X., Liu Y. & Lin J. (2023) Local attention and long-distance interaction of rppg for deepfake detection. The Visual Computer , pp. 1–12.

[113] Jeon S.M., Seong H.A. & Lee E.C. (2023) Deepfake video detection using the frequency characteristic of remote photoplethysmography. In: Intelligent Human Computer Interaction: 14th International Conference, IHCI 2022, Tashkent, Uzbekistan, October 20–22, 2022, Revised Selected Papers, Springer, pp. 1–6.

[114] Chen M., Liao X. & Wu M. (2022) PulseEdit: Editing Physiological Signals in Facial Videos for Privacy Protection URL: https://www.techrxiv.org/articles/preprint/PulseEdit_

Editing_Physiological_Signal_in_Facial_Videos_for_Privacy_Protection/14647377.

[115] Sun Z. & Li X. (2022) Privacy-phys: Facial video-based physiological modification for privacy protection. IEEE Signal Processing Letters 29, pp. 1–5.

[116] Wang Z., Ba Y., Chari P., Bozkurt O.D., Brown G., Patwa P., Vaddi N., Jalilian L. & Kadambi A. (2022) Synthetic generation of face videos with plethysmograph physiology. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.

[117] Heusch G., Marcel S. & Anjos A. (2016), Cohface. URL: https://doi.org/10.34777/ff3f-ba56.

[118] Thies J., Zollhöfer M., Stamminger M., Theobalt C. & Nießner M. (2020), Face2face: Real-time face capture and reenactment of rgb videos.

[119] K. Torzdf A., Faceswap. https://github.com/deepfakes/faceswap. Accessed: 2023-05-18.

[120] Thies J., Zollhöfer M. & Nießner M. (2019), Deferred neural rendering: Image synthesis using neural textures.

[121] Álvarez Casado C. & Bordallo López M. (2021) Real-time face alignment: evaluation methods, training strategies and implementation optimization. Journal of Real-Time Image Processing , pp. 1–29.

[122] Harris C.R., Millman K.J., van der Walt S.J., Gommers R., Virtanen P., Cournapeau D., Wieser E., Taylor J., Berg S., Smith N.J., Kern R., Picus M., Hoyer S., van Kerkwijk M.H., Brett M., Haldane A., del Río J.F., Wiebe M., Peterson P., Gérard-Marchant P., Sheppard K., Reddy T., Weckesser W., Abbasi H., Gohlke C. & Oliphant T.E. (2020) Array programming with NumPy. Nature 585, pp. 357–362. URL: https://doi.org/10.1038/s41586-020-2649-2.

[123] Virtanen P., Gommers R., Oliphant T.E., Haberland M., Reddy T., Cournapeau D., Burovski E., Peterson P., Weckesser W., Bright J., van der Walt S.J., Brett M., Wilson J., Millman K.J., Mayorov N., Nelson A.R.J., Jones E., Kern R., Larson E., Carey C.J., Polat İ., Feng Y., Moore E.W., VanderPlas J., Laxalde D., Perktold J., Cimrman R., Henriksen I., Quintero E.A., Harris C.R., Archibald A.M., Ribeiro A.H., Pedregosa F., van Mulbregt P. & SciPy 1.0 Contributors (2020) SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. Nature Methods 17, pp. 261–272.

[124] Vallat R., Antropy. https://github.com/raphaelvallat/antropy. Accessed: 2023-05-18.

[125] Makowski D., Pham T., Lau Z.J., Brammer J.C., Lespinasse F., Pham H., Schölzel C. & Chen S.H.A. (2021) NeuroKit2: A python toolbox for neurophysiological signal processing. Behavior Research Methods 53,

pp. 1689–1696. URL: https://doi.org/10.3758%2Fs13428-020-01516-y.

[126] van Gent P., Farah H., van Nes N. & van Arem B. (2019) Heartpy: A novel heart rate algorithm for the analysis of noisy signals. Transportation Research Part F: Traffic Psychology and Behaviour 66, pp. 368–378. URL: https://www.sciencedirect.com/science/article/pii/S1369847818306740.

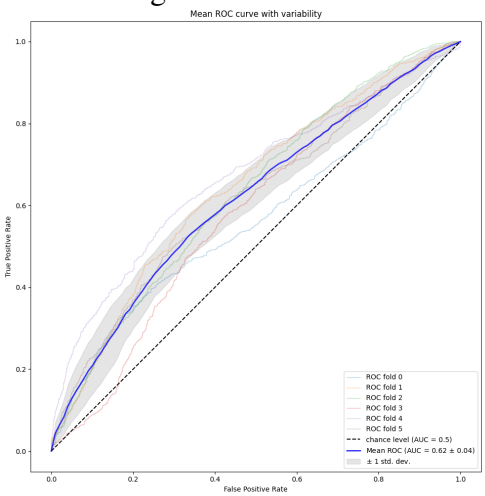
[127] Casado C.Á., Cañellas M.L. & López M.B. (2022), Depression recognition using remote photoplethysmography from facial videos. URL: https://arxiv.org/abs/2206.04399.

[128] Shaffer F. & Ginsberg J.P., An overview of heart rate variability metrics and norms. https://www.frontiersin.org/articles/10.3389/fpubh.2017.00258/full. Accessed: 2023-05-18.

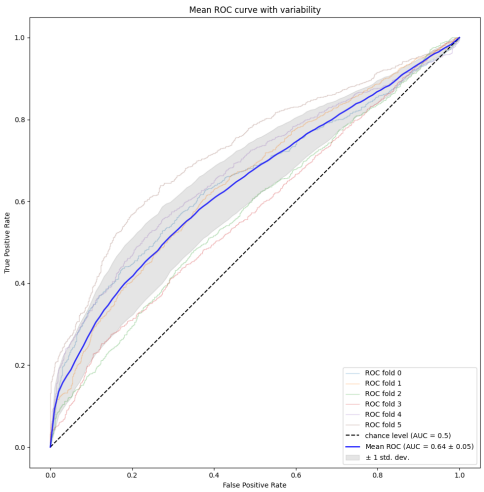
[129] Pedregosa F., Varoquaux G., Gramfort A., Michel V., Thirion B., Grisel O., Blondel M., Prettenhofer P., Weiss R., Dubourg V., Vanderplas J., Passos A., Cournapeau D., Brucher M., Perrot M. & Duchesnay E. (2011) Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 12, pp. 2825–2830.

7. APPENDICES

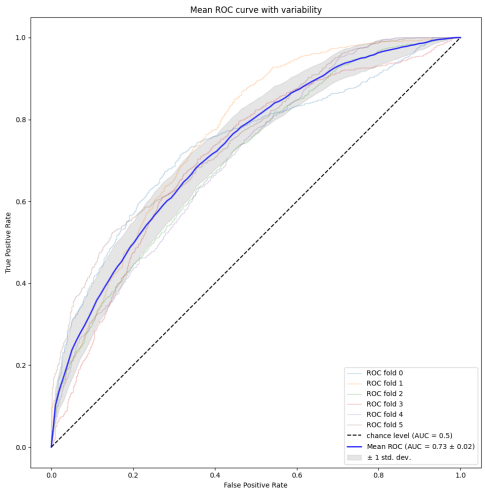
Figure 13. 8 second window ROC curves with no GREEN



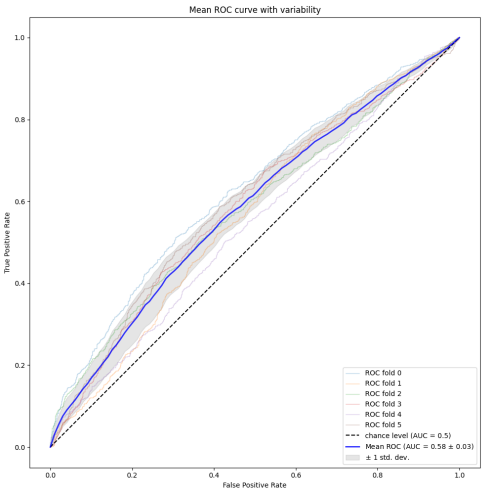
(a) 8 seconds, OMIT



(b) 8 seconds, POS

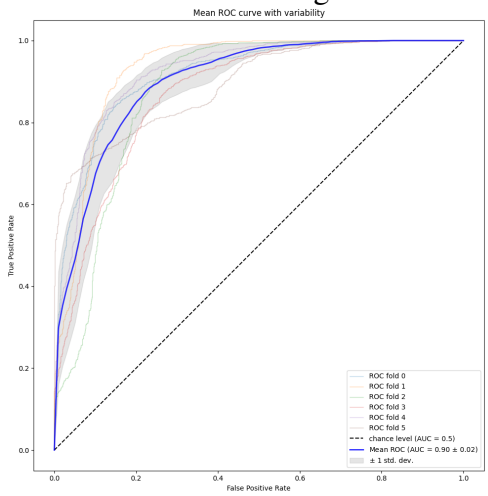


(c) 8 seconds, LGI

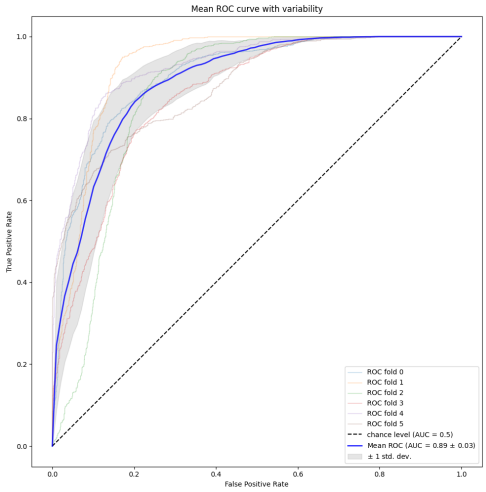


(d) 8 seconds, CHROM

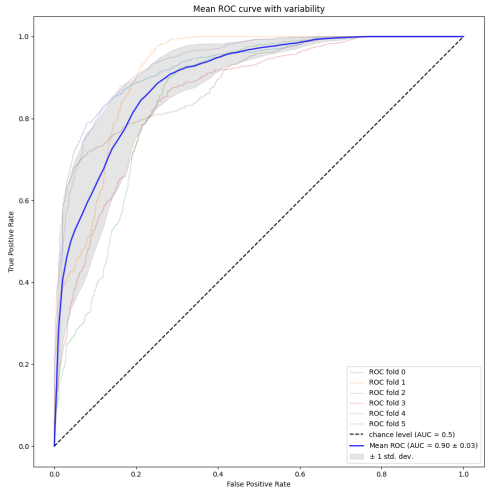
Figure 14. 8 second window ROC curves



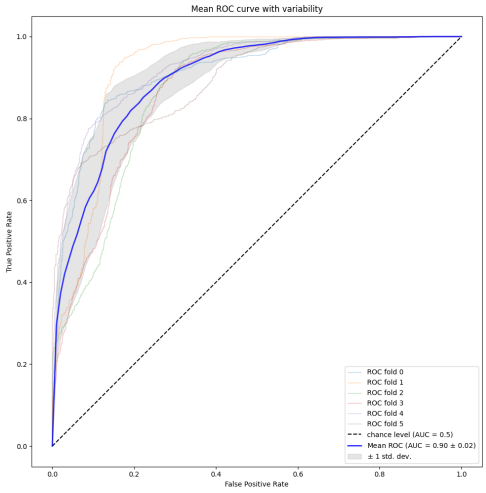
(a) 8 seconds, All methods



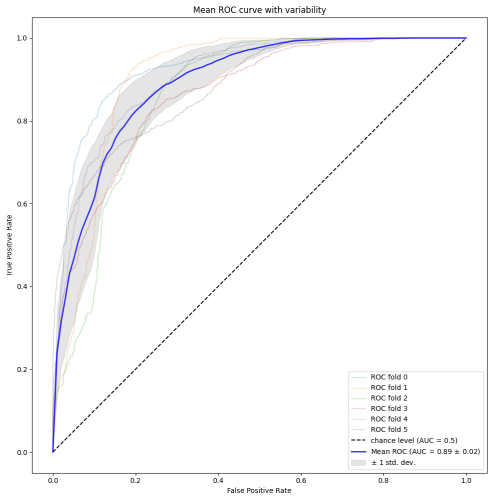
(b) 8 seconds, OMIT+GREEN



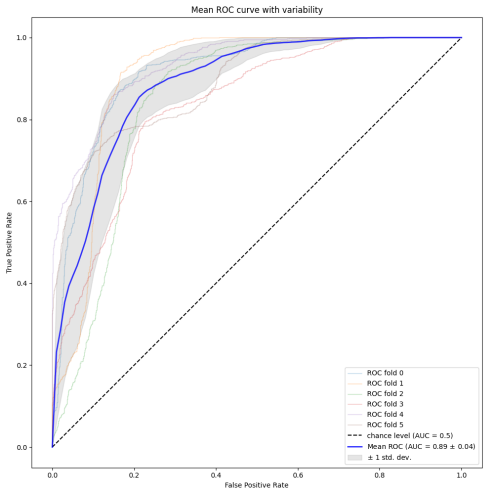
(c) 8 seconds, POS+GREEN



(d) 8 seconds, LGI+GREEN

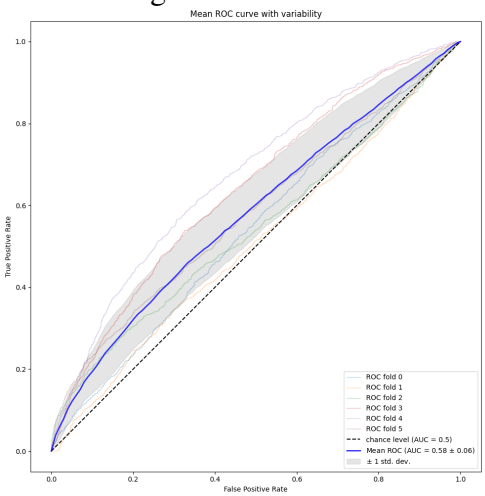


(e) 8 seconds, CHROM+GREEN

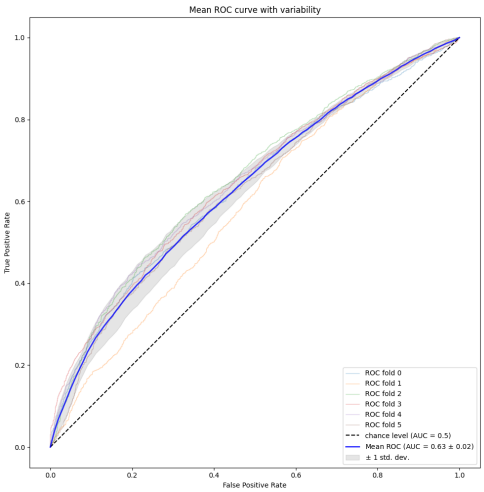


(f) 8 seconds, GREEN

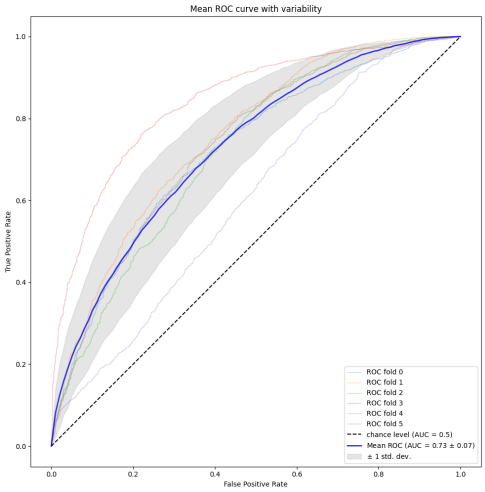
Figure 15. 4 second window ROC curves with no GREEN



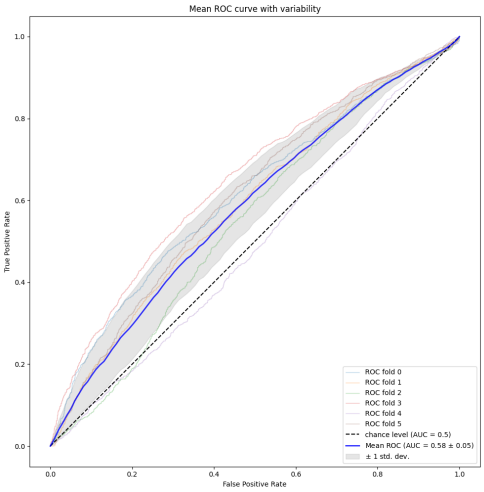
(a) 4 seconds, OMIT



(b) 4 seconds, POS

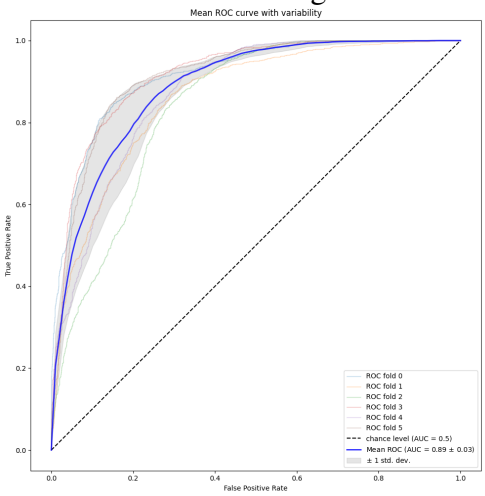


(c) 4 seconds, LGI

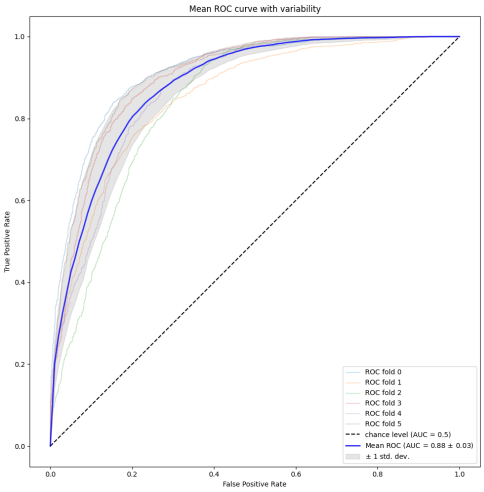


(d) 4 seconds, CHROM

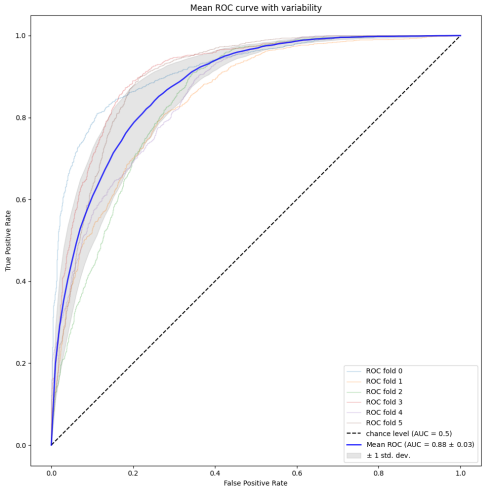
Figure 16. 4 second window ROC curves



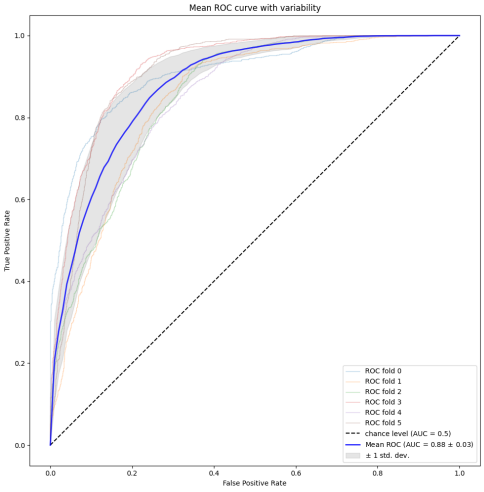
(a) 4 seconds, All methods



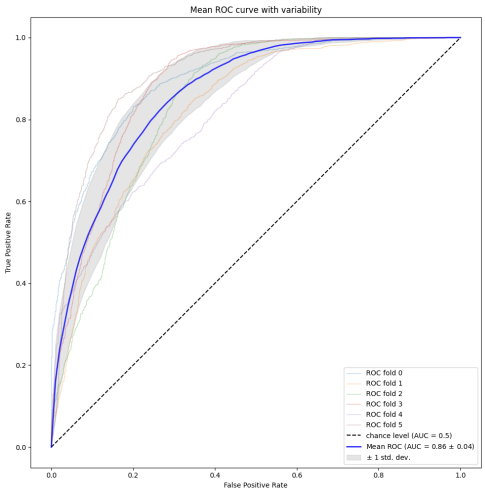
(b) 4 seconds, OMIT+GREEN



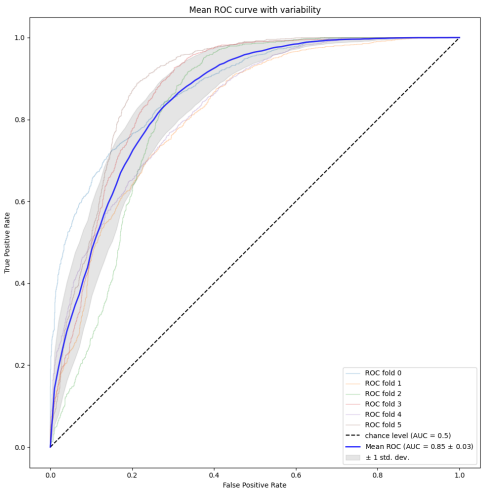
(c) 4 seconds, POS+GREEN



(d) 4 seconds, LGI+GREEN

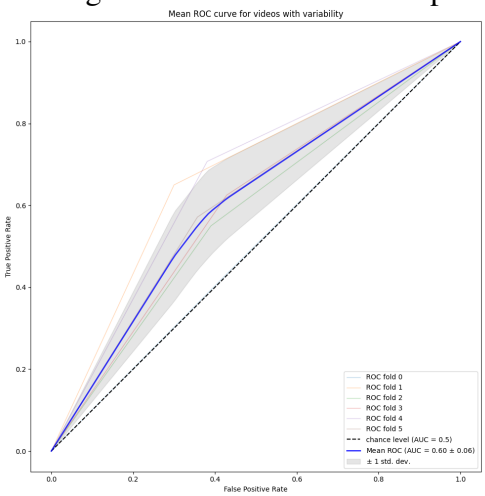


(e) 4 seconds, CHROM+GREEN

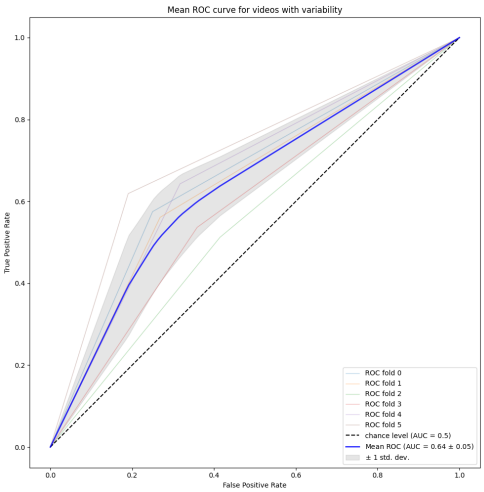


(f) 4 seconds, GREEN

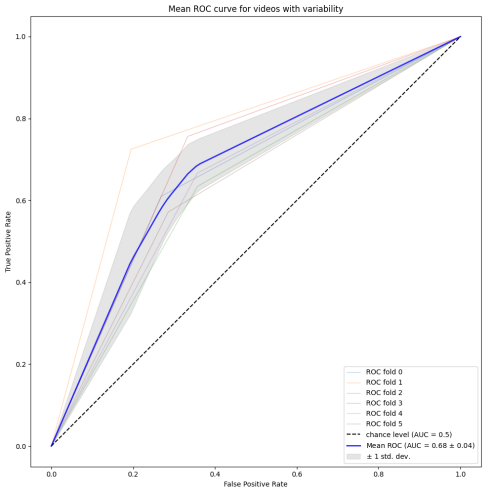
Figure 17. 8 second window per-video ROC curves with no GREEN



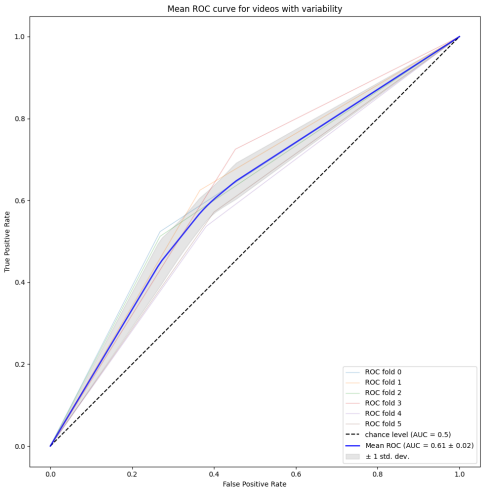
(a) 8 seconds, OMIT



(b) 8 seconds, POS

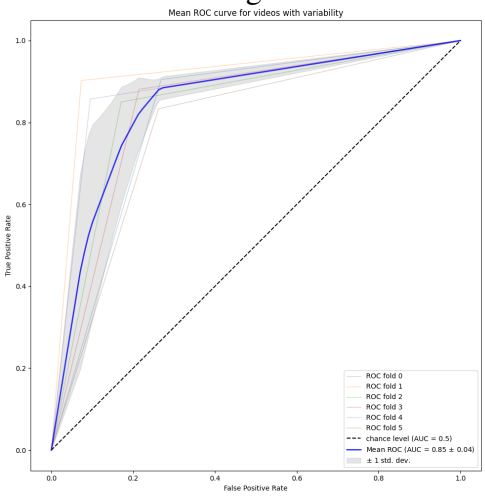


(c) 8 seconds, LGI

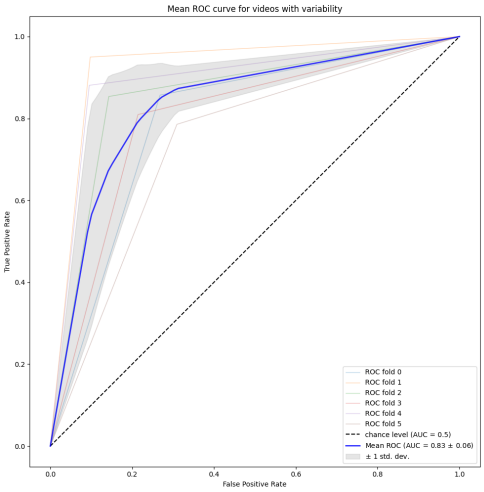


(d) 8 seconds, CHROM

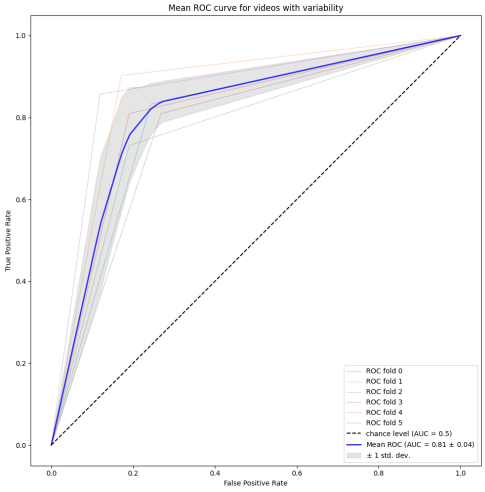
Figure 18. 8 second window per-video ROC curves



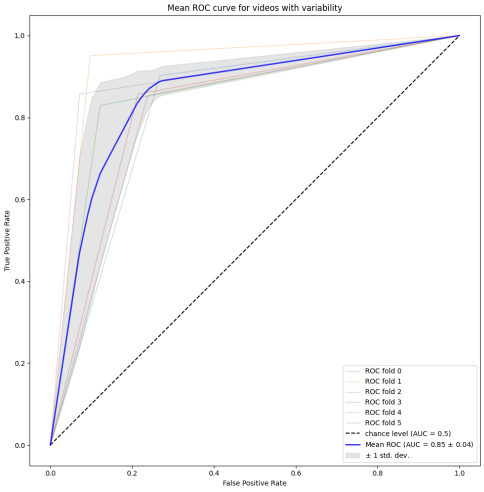
(a) 8 seconds, All methods



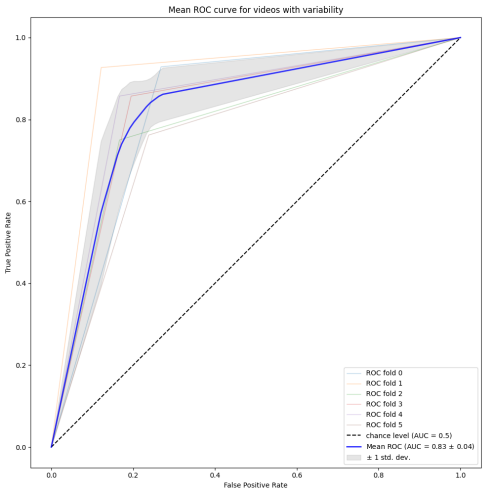
(b) 8 seconds, OMIT+GREEN



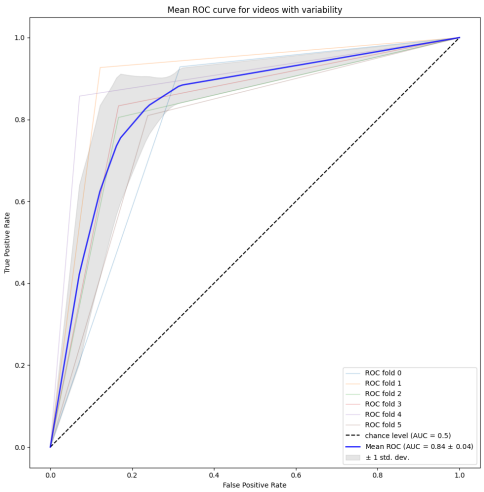
(c) 8 seconds, POS+GREEN



(d) 8 seconds, LGI+GREEN

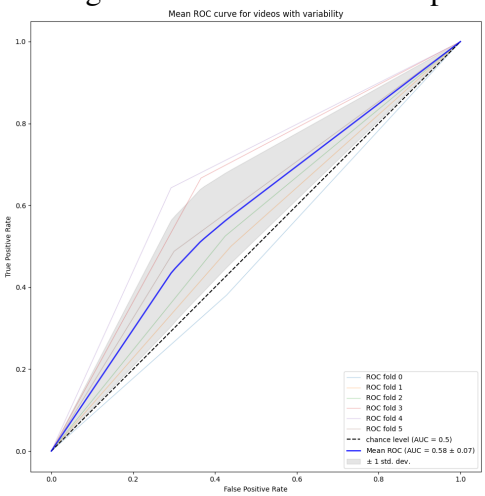


(e) 8 seconds, CHROM+GREEN

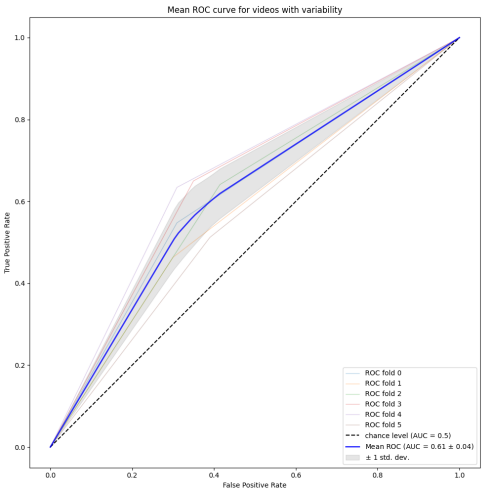


(f) 8 seconds, GREEN

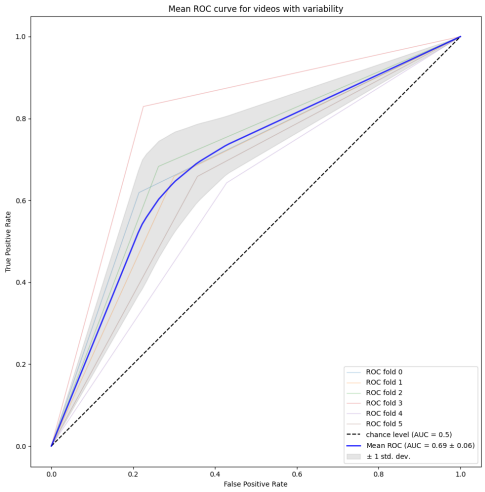
Figure 19. 4 second window per-video ROC curves with no GREEN



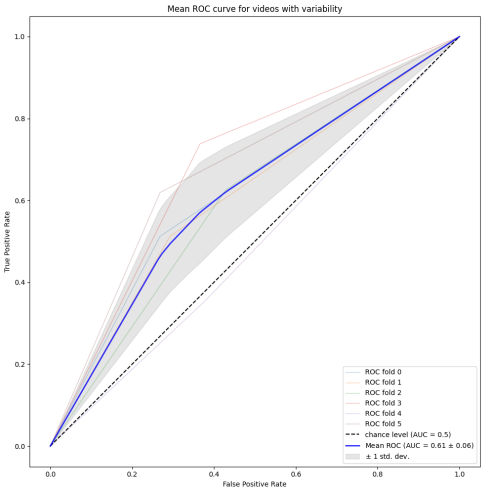
(a) 4 seconds, OMIT



(b) 4 seconds, POS

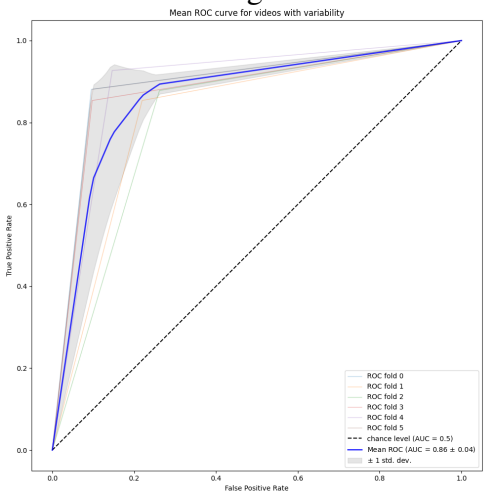


(c) 4 seconds, LGI

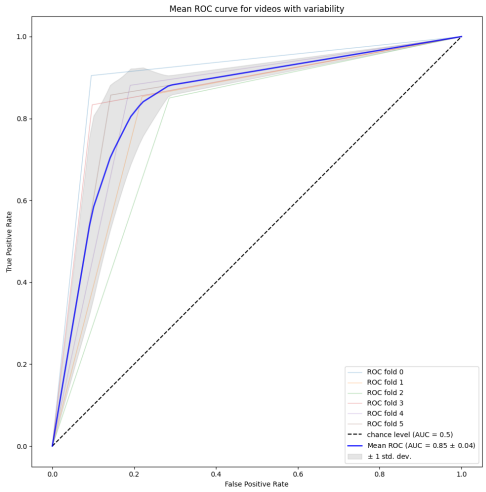


(d) 4 seconds, CHROM

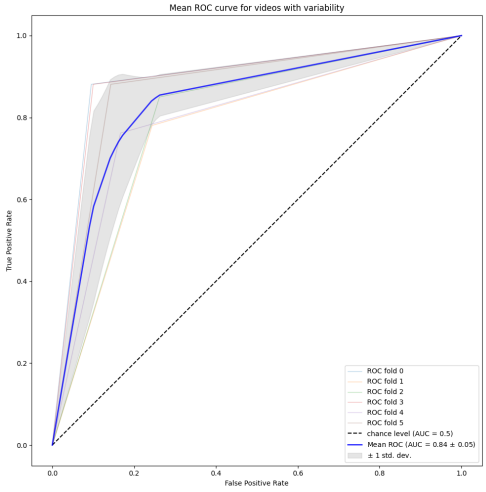
Figure 20. 4 second window per-video ROC curves



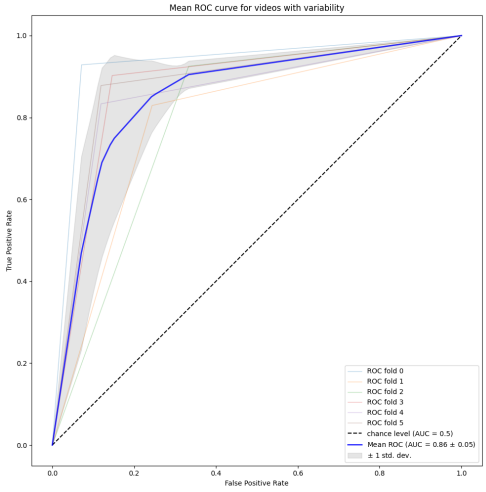
(a) 4 seconds, All methods



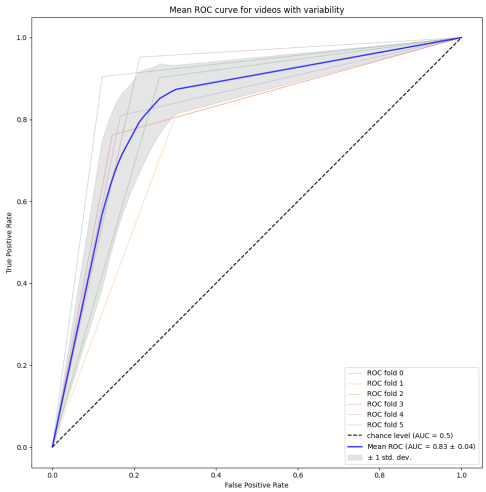
(b) 4 seconds, OMIT+GREEN



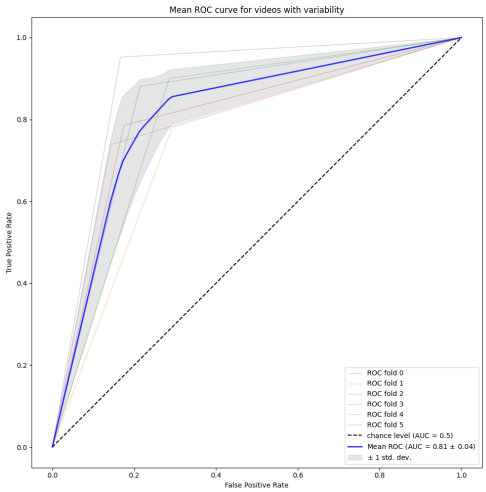
(c) 4 seconds, POS+GREEN



(d) 4 seconds, LGI+GREEN



(e) 4 seconds, CHROM+GREEN



(f) 4 seconds, GREEN