Deepfake Representation with Multilinear Regression

Sara Abdali
sabda005@ucr.edu
University of California, Riverside

M. Alex O. Vasilescu
maov@cs.ucla.edu
University of California, Los Angeles
Tensor Vision, Los Angeles

Evangelos E. Papalexakis
epapalex@cs.ucr.edu
University of California, Riverside

ABSTRACT

Generative neural network architectures such as GANs, may be used to generate synthetic instances to compensate for the lack of real data. However, they may be employed to create media that may cause social, political or economical upheaval. One emerging media is "Deepfake". Techniques that can discriminate between such media is indispensable. In this paper, we propose a modified multilinear (tensor) method, a combination of linear and multilinear regressions for representing fake and real data. We test our approach by representing Deepfakes with our modified multilinear (tensor) approach and perform SVM classification with encouraging results.

KEYWORDS

Deepfake Detection, Multilinear Projection, multilinear regression, M-mode SVD, Tucker

ACM Reference Format:
Sara Abdali, M. Alex O. Vasilescu, and Evangelos E. Papalexakis. 2021. Deepfake Representation with Multilinear Regression. In *The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'21): Misinformation and Misbehavior Mining on the Web (MIS2) Workshop, August 15, 2021, Virtual.* ACM, New York, NY, USA, 6 pages.

1 INTRODUCTION

Recent advances in Generative Adversarial Networks (GANs) and Convolutional Neural Networks (CNNs) embedded in applications like Zao[1], DeepFakes web β[2], Face Swap by Microsoft[3], DeepFaceLab[4] etc. have led to a broad usage of AI-synthesized media a.k.a. "Deepfake" [5]. Other automated manipulation techniques are Face2Face, FaceSwap, NeuralTextures, and FaceShifter [20].

Due to the potential misuse of Deepfakes e.g., fake pornography, fake news, and financial or political fraud, they have become a major public concern. Thus, different techniques have been introduced to discriminate Deepfakes from pristine videos.

Prior Deepfakes detection can be categorized as [22] approaches that classify based on (a) physical or physiological causal factors which are not well presented in Deepfakes e.g., eye blinking [13] and heart rate[10], or (b) artifacts in imaging factors e.g., relative head pose to the camera position[31], and (c) data-driven techniques that do not leverage specific cues and directly train a deep learning model on a large set of real and Deepfake videos [1, 6].

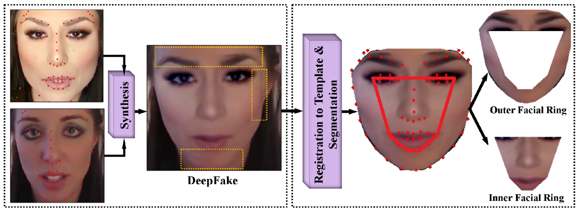


Figure 1: Deepfake technique replaces a person's appearance in an existing image or video with someone else's appearance [20]. This process introduces artifacts specially around the cropping boundaries estimated by facial landmarks. We propose segmenting the output face into inner and outer facial rings. The artifacts are mainly concentrated in outer facial ring.

From the first category, we can mention [31], where Yeng et al. propose using the inconsistencies in head poses to detect the Deepfakes. More precisely, 3D head poses cue is leveraged to estimate errors introduced by splicing process which synthesizes source face region into the target one. The eye blinking cue is anther physiological signal which is not well presented in Deepfakes and Li et al. take advantage of it for discriminating the Deepfakes [13]. More recently, a novel cue has been introduced that considers the heart rate measured by remote photoplethysmography (rPPG) to analyze color changes in the human skin, which is a signal for the presence of blood under the tissues [10].

As an example of the second category, we can refer to the work in [14] where the distinctive feature is the introduced face warping artifacts. In this work, Li et al. discuss limitation of early Deepfake generators which produce images of limited resolutions and transformation of this images leaves certain distinctive artifacts in the Deepfake videos. In addition, in [15], McCloske et al. analyze the structure of the generator network of a GAN and show how the networks treatment of exposure is markedly different from a real camera. They propose leveraging frequency of over-exposed pixels as a feature for this cue to discriminate GAN-generated media from camera imagery.

However, the vast majority of proposed methods for Deepfake detection fall into the third category, i.e., data-driven approaches. For instance, in [4] a hybrid Long Short Term Memory Network (LSTM) and Encoder-Decoder architecture is introduced to detect forgeries in images. In another work [6], a novel CNN network inspired by inception is introduced, where inception modules have been replaced with depth wise separable convolutions. Another example of this category is the work proposed in [1], where two networks are presented, both with a low number of layers to focus on the mesoscopic properties of the images. [9], [17] and [16] are other instances of data-driven approaches which leverage (Recurrent Neural Networks

[1] https://www.zaoapp.net/
[2] https://deepfakesweb.com/
[3] https://www.microsoft.com/en-us/garage/profiles/face-swap/
[4] https://awesomeopensource.com/project/iperov/DeepFaceLab
[5] The term Deepfakes has been widely used for deep learning generated media, but it is also the name of a specific manipulation technique in which face of one person is replaced by another one. To distinguish these, we denote said method by DeepFakes in the entire paper.

(RNNs), capsule networks and CNN networks for detection of Deepfakes. Lastly, there are works that take advantage of CNNs and RNNs simultaneously to capture both frame level and sequence level information [4, 9, 17].

The first two categories, which mainly leverage feature extraction and image pre-processing techniques to some extent provide interpretability for the classification result which is a key factor for explainable and trustworthy AI. For instance, the predictive model is built upon differences in the nature of pixels [15] or an estimation of regions with high concentration of artifacts [14]. However, the black box methods of the third category, while being highly accurate, do not provide any interpretation for the classification output. Thus, it is not clear if the video is classified as Deepfake due to the difference in the frame by frame movement or because of spatial artifacts or both. Moreover, there is no information about regions of interest and causes of the artifacts e.g., warping artifacts or artifacts in head adjustment, that discriminate Deepfakes from real videos.

We hypothesize that DeepFakes contain artifacts localized either in transition areas between facial images, or contain discrepancies in the overall facial appearance. We concentrate our analysis on the transition areas of the face henceforth referred to as outer facial ring, Figure 1. We segment the outer ring from a facial image that has been registered to a template based on facial landmarks detected by a pretrained model [11]. The outer ring is analyzed with a modified face recognition tensor model [28, 29] that computes real and fake data representations.

We employ a multilinear a.k.a. tensor framework which decomposes basis components of outer facial rings into real and fake class representations. Later on we leverage the derived representation of classes to classify the test frames using a linear SVM. Summarily, our major contributions are as follows:

- **Segmenting face into regions of interest**: We propose Segmenting face into facial parts and leverage parts with high concentration of artifacts to distinguish Deepfakes.
- **Proposing a multilinear representation of Deepfakes for classification:** we employ a multilinear approach to represent Deepfake and real class information and then leverage them for classification.

2 BACKGROUND

In this section, we discuss the relevant tensor algebra [7], [8], [28, 29], [12], [18], [21]. We will follow the notation of Table 1

2.1 Multilinear (tensor) framework

A data tensor $\mathcal{D} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_M}$ is a multi-way array. In fact, when an array has three or more than three dimensions, we call it a tensor. The dimensions of a tensor are usually referred to as modes.

2.2 Singular Value Decomposition (SVD) and Principle Components Analysis (PCA)

In linear algebra, we factorize a matrix $\mathbf{D} \in \mathbb{R}^{I_1 \times I_2}$ using Singular Value Decomposition (SVD) as follows:

$$\mathbf{D} = \mathbf{U}\Sigma\mathbf{V}^{\mathrm{T}} \tag{1}$$

where the columns of $\mathbf{U} \in \mathbb{R}^{I_1 \times r}$ and $\mathbf{V} \in \mathbb{R}^{I_2 \times r}$ are orthonormal and $\Sigma \in \mathbb{R}^{\mathrm{r} \times \mathrm{r}}$ is a diagonal matrix with positive real entries know as singular values. The rank R, SVD decomposition represents a matrix as following equation:

Symbol	Definition
$\mathcal{D}$, $\mathbf{D}$, $\mathbf{d}$	Tensor, Matrix, vector
$\mathcal{D}^{\dagger_m}$	Mode-m tensor pseudo-inverse of $\mathcal{D}$
$\mathbf{D}_{[m]}$	Mode-m tensor matrixizing
$\times_{\mathrm{m}}$	Mode-m product
$\circ$	Outer product

Table 1: Symbols and Definition

$$\mathbf{D} \simeq \sum_{r=1}^{R} \sigma_r \mathbf{u}_r \circ \mathbf{v}_r \tag{2}$$

Rewriting equation 1 in conventional linear algebra, the Principal Components Analysis (PCA) is:

$$\mathbf{D} = \underbrace{\mathbf{U}}_{\text{Basis}} \underbrace{\Sigma\mathbf{V}^{\mathrm{T}}}_{\text{Coefficient}} \tag{3}$$

2.3 Mode-M Matrixizing a Tensor

TMode-m matrixizing of tensor $\mathcal{D} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_M}$ is defined as the matrix $\mathbf{D}_{[m]} \in \mathbb{R}^{I_m \times (I_1 \dots I_{m-1} I_{m+1} \dots I_M)}$ where the parenthetical ordering indicates that column vectors are ordered by sweeping indices of all other modes through their ranges [25]. Therefore:

$$\begin{aligned} &[\mathbf{D}]_{jk} = a_{i_1 \dots i_m \dots i_M} \quad \text{where} \\ &j = i_m \quad \text{and} \quad k = 1 + \sum_{n=0, n \neq m}^{M} (i_n - 1) \prod_{l=0, l \neq m}^{n-1} I_l \end{aligned} \tag{4}$$

A 3-mode tensor may be metricized in three different ways by stacking first, second and third mode slices which are illustrated in Figure.2.

2.4 Mode-M product of a matrix and a tensor

The mode-m product [5, 7, 25] of a tensor $\mathcal{D} \in \mathbb{R}^{I_1 \times I_2 \times \dots I_m \times \cdots \times I_M}$ and matrix $\mathbf{A} \in \mathbb{R}^{J_m \times I_m}$ denoted by $\mathcal{D} \times_{\mathrm{m}} \mathbf{A}$ is a tensor of size $\mathbb{R}^{I_1 \times I_2 \times \dots J_m \times \cdots \times I_M}$ where the entries are calculated as

$$[\mathcal{D} \times_{\mathrm{m}} \mathbf{A}]_{i_1 \dots i_{m-1} j_m i_{m+1} \dots i_M} = \sum_{i_m} d_{i_1 i_2 \dots i_{m-1} i_m i_{m+1} \dots i_M} a_{j_m i_m} \tag{5}$$

The mode-M product is interchangeably denoted by matrix multiplication and tensor multiplication as follows:

$$\mathcal{B} = \mathcal{D} \times_{\mathrm{n}} \mathbf{A} \underset{\text{tensorizing}}{\overset{\text{matrixizing}}{\rightleftarrows}} \mathbf{B}_{[m]} = \mathbf{A}\mathbf{D}_{[m]} \tag{6}$$

2.5 M-mode SVD

We can define SVD decomposition in terms of n-mode product as follows:

$$\mathbf{D} = \Sigma \times_1 \mathbf{U} \times_2 \mathbf{V} \tag{7}$$

In multilinear algebra there is a generalization of SVD know as multilinear SVD [7, 8] or M-mode SVD [28, 29] which decomposes

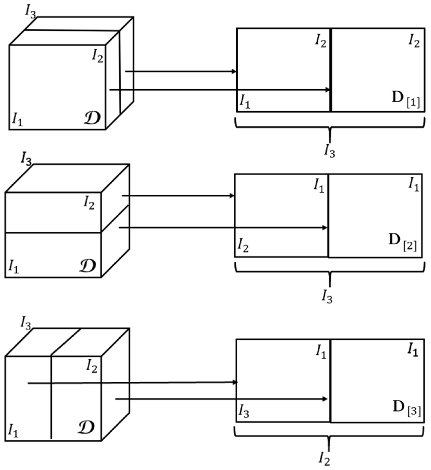


Figure 2: Matrixizing a 3-mode tensor

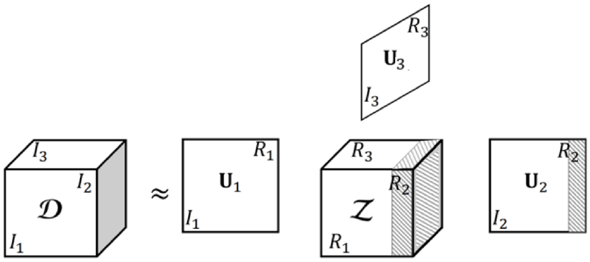


Figure 3: M-mode SVD decomposition of a 3-mode tensor. Some of the singular values corresponding to the components of the second factor matrix i.e., second mode of tensor $\mathcal{D}$ are truncated.

an M-mode tensor $\mathcal{D}$ into the M-mode product of orthonormal spaces:

$$\mathcal{D} \simeq \mathcal{Z} \times_1 \mathrm{U}_1 \times_2 \mathrm{U}_2 \cdots \times_\mathrm{M} \mathrm{U}_\mathrm{M} \tag{8}$$

where $\mathcal{Z}$ is the core tensor that governs the interaction between the orthonormal mode matrices, U_m. The core tensor is analogues to the singular value matrix Σ but unlike the Σ the core tensor is not always diagonal [12, 18, 21].

The M-mode SVD of a 3-mode tensor is demonstrated in Figure 3. $\mathbf{U}_i$ is approximated by left singular vectors of truncated SVD decomposition of $\mathrm{D}_{[i]}$. Meanwhile, since $\mathbf{U}_i$ is orthonoramal, we have $\mathbf{U}_m^{-1} = \mathbf{U}_m^T$ and the core tensor $\mathcal{Z}$ is estimated as follows:

$$\begin{aligned} \mathcal{Z} &= \mathcal{D} \times_1 \mathrm{U}_1^{-1} \times_2 \mathrm{U}_2^{-1} \cdots \times_\mathrm{M} \mathrm{U}_\mathrm{M}^{-1} && (9) \\ &= \mathcal{D} \times_1 \mathrm{U}_1^{T} \times_2 \mathrm{U}_2^{T} \cdots \times_\mathrm{M} \mathrm{U}_\mathrm{M}^{T} && (10) \end{aligned}$$

3 PROPOSED METHOD

A DeepFake is a synthesizing product of two real faces. More precisely, in DeepFake generation process, face of a real person a.k.a. target is synthesized by another face a.k.a., source. This process, usually introduces some artifacts, specially around the cropping edges of source face including eyes and eyebrows Figure 1. Due to the fact that a Deepfake face is a mixture of source and target faces, Sometimes it is not distinguishable from the source and this similarity results in misclassification of the video. In this work, we propose to segment faces into parts henceforth referred to as facial inner and outer rings Figure 1. We define the outer ring as a facial part that comprises the blending boundaries that are mostly the non-facial pixels. We leverage this remaining region i.e., outer ring which has the highest concentration of introduced artifacts as a cue for Deepfakes detection. An example of this process is demonstrated in Figure. 1. This cue is very promising specially when the manipulation masks are not available.

In what follows, we discuss our proposed multilinear pipeline for detecting the Deepfakes.

3.1 Step 1: Vectorizing video frames

Vasilescu [25, Appndix A] argues that in most cases, it is preferable to vectorize an image and treat it as a single observation rather than a collection of independent column/row observations. By vectorizing an image, we treat an image as a point in high dimensional pixel space and calculate all possible combinations of pixel statistics, both near and faraway statistics. On the other hand, when we consider an image as a matrix, every image column (row) is treated as an independent observation, and column (row) covariances are computed. Having this in mind, we also follow the same strategy and vectorize the frames and create a vector for each one of the video frames in the dataset.

3.2 Step 2: Finding eigenfaces of each class

Eigenfaces are eigenvectors when the images are human face. The eigenfaces are derived from the covariance matrix of the pixel distribution over the high dimensional face space. The eigenfaces represent a basis set of all faces used to construct the covariance matrix. So far, the eigenfaces have been successfully leveraged for many facial image related tasks [23]. Leveraging eigenfaces allows for dimensionality reduction such that a smaller set of basis vectors represent the original training faces. Classification could be achieved by comparing how different faces are represented by the basis set of the corresponding class.

Based on principle component terminology, the eigenfaces are equal to basis vectors of PCA decomposition. Therefore, by staking the vectorized frames of each class, we create two separate matrices and decompose them using SVD to capture the eigenfaces of the corresponding class as follows:

$$\begin{aligned} \mathbf{D}_\mathrm{real} &= \mathbf{U}_\mathrm{real}\Sigma_\mathrm{real}\mathbf{V}_\mathrm{real}^\mathrm{T} = \mathbf{B}_\mathrm{real}\mathbf{V}_\mathrm{real}^\mathrm{T} && (11) \\ \mathbf{D}_\mathrm{fake} &= \mathbf{U}_\mathrm{fake}\Sigma_\mathrm{fake}\mathbf{V}_\mathrm{fake}^\mathrm{T} = \mathbf{B}_\mathrm{fake}\mathbf{V}_\mathrm{fake}^\mathrm{T} && (12) \end{aligned}$$

Where $\mathbf{B}_\mathrm{real}$ and $\mathbf{B}_\mathrm{fake}$ are basis matrices and $\mathbf{V}_\mathrm{real}$ and $\mathbf{V}_\mathrm{fake}$ are the normalized coefficient matrices of the corresponding classes.

3.3 Step 3: Leveraging tensor framework to decompose eigenfaces into underlying factors

As seen in previously mentioned tensor is an effective framework for decomposing a set of observation into underlying factors. After reducing the dimentionality of observations using eigenface representation of the classes, we propose leveraging a three-mode tensor where the first mode i.e., measurement mode represents the pixels of an eigenface, the second mode corresponds to the eigenfaces and the third mode is the class mode i.e., DeepFake vs. real. We propose using an M-mode SVD which as we discussed earlier decomposes a tensor into M orthonormal matrices ($M = 3$), and a core tensor which governs the interaction between these spaces. Since the first mode is the measurement mode, We only calculate the M-mode SVD of the tensor by flattening the second and the third modes as follows:

$$\begin{aligned} \mathcal{D} &\simeq \mathcal{Z} \times_1 \mathbf{U}_\mathrm{p} \times_2 \mathbf{U}_\mathrm{f} \times_3 \mathbf{U}_\mathrm{c} && (13) \\ &= \mathcal{T} \times_2 \mathbf{U}_\mathrm{f} \times_3 \mathbf{U}_\mathrm{c} && (14) \end{aligned}$$

where the $\mathbf{U}_\mathrm{c}$ comprises underlying vector representation of original and fake classes.Moreover, the core tensor $\mathcal{T}$ is the signature of this dataset and shows interactions of orthonormal subspaces. Later on, we leverage this signature to project the test frames into the subspaces we derive here.

3.4 Step 4: Embedding the class representations in a higher three dimensional space

Applying M-mode SVD results in a mode matrix $\mathbf{U}_\mathrm{c} \in \mathbb{R}^{2\times 2}$ that spans the class representations. We embed the vector class representations into a higher dimensional space to increase the class separability of the test data. We embed the row vectors of $\mathbf{U}_\mathrm{c}$ into $\mathbb{R}^3$, setting the third coordinate of the real and fake class to +1 and −1 respectively, and normalizing the vector length to 1.

3.5 Step 5: Multilinear projection of an incoming frame into the orthonormal vector spaces

As mentioned above, the core tensor of each decomposition is the signature of the decomposed space which governs the interaction of constituent factors. we leverage the core tensor and perform a multilinear projection of the incoming frame into the subspaces we derived in the previous step. Let say we have the vectorized frame $\mathbf{d}$. If $\mathbf{d}$ is supposed to be in the same subspaces we derived, then

$$\mathbf{d} = \mathcal{T} \times_2 \mathbf{f}^\mathrm{T} \times_3 \mathbf{c}^\mathrm{T} \qquad (15)$$

where the vectors $\mathbf{f}$ and $\mathbf{c}$ are the coefficient vector representations of a video frame $\mathbf{d}$ in the orthonormal subspaces that are governed by the extended core tensor $\mathcal{T}$. The goal is to find out weather the class coefficient vector $\mathbf{c}$ is more similar to the vector representation of real class or Deepfake class. To this end, we estimate $\mathbf{c}$ representation vector by employing the multilinear projection algorithm [26, 30] that decomposes a vectorized observation, $\mathbf{d}$ into a set of latent vector representation, $\mathbf{r}_\mathrm{n}$ that corresponds to the constituent factors of data formation. The basic multilinear projection is the M-mode SVD/CP decomposition of $\mathcal{T}^{+_1} \times_1 \mathbf{d}^\mathrm{T}$ which can be expressed mathematically as

$$\underbrace{M\text{-mode SVD/CP}\left(\mathcal{T}^{+_1} \times_1 \mathbf{d}^\mathrm{T}\right)}_{\text{Multilinear Projection}} \simeq \mathbf{r}_\mathrm{f} \circ \mathbf{r}_\mathrm{c} \quad \Rightarrow \quad \mathbf{d} \simeq \left(\mathcal{T} \times_2 \mathbf{r}_\mathrm{f}^\mathrm{T} \times_3 \mathbf{r}_\mathrm{c}^\mathrm{T}\right)$$

where $\mathcal{T}^{+_1}$ is mode-1 pseudo-inverse of $\mathcal{T}$ that in matrix notation is expressed as $\mathbf{T}^{+}_{[1]}$, and $\mathbf{r}_\mathrm{c}, \mathbf{r}_\mathrm{f}$ are estimates of vectors $\mathbf{c}$ and $\mathbf{f}$ from eq.(15), respectively.

3.6 Step 6: Classifying an incoming frame

Up to this step, we have the vector representation of each classes in addition to class coefficients of the incoming frame. We use a linear Support Vector Machine (SVM) and estimate the decision boundaries using validation frames and then leverage the defined boundaries for classification of test frames. An overview of the proposed approach is demonstrated in Algorithm 1.

Algorithm 1 DeepFake Detection Algorithm

```
Input : $\mathbf{D}_\mathrm{real}, \mathbf{D}_\mathrm{fake}$ were centered by subtracting the mean of the real
        training data,
  (1) Preprocessing and data tensor organization:
      $[\mathbf{U}_\mathrm{real}, \mathbf{S}_\mathrm{real}, \mathbf{V}_\mathrm{real}] \Leftarrow \mathrm{svd}(\mathbf{D}_\mathrm{real})$
      $[\mathbf{U}_\mathrm{fake}, \mathbf{S}_\mathrm{fake}, \mathbf{V}_\mathrm{fake}] \Leftarrow \mathrm{svd}(\mathbf{D}_\mathrm{fake})$
      $\mathcal{D}(:,:,1) = [\mathbf{U}_\mathrm{real}\mathbf{S}_\mathrm{real}]$
      $\mathcal{D}(:,:,2) = [\mathbf{U}_\mathrm{fake}\mathbf{S}_\mathrm{fake}]$
  (2) Training data decomposition:
      $\mathcal{T} \times_2 \mathbf{U}_\mathrm{f} \times_3 \mathbf{U}_\mathrm{c} \Leftarrow M\text{-mode SVD}(\mathcal{D})$
  (3) Embed the class representations in the higher three dimensional space and set the third coordinate of the real and fake
      class to +1 and −1 respectively. Hence, $\mathbf{U}_\mathrm{c} \in \mathbb{R}^{2\times 2}$ now has
      dimensionality $\mathbb{R}^{2\times 3}$. Normalize the rows of $\mathbf{U}_\mathrm{c}$ to have length
      1.
  (4) Computer the extended core
      $\mathcal{T} := \mathcal{D} \times_2 \mathbf{U}_\mathrm{f}^\mathrm{T} \times_3 \mathbf{U}_\mathrm{c}^{+}$      (16)
  (5) Centering: validation and test data is centered by subtracting
      the mean of the real training data.
  (6) Test data decomposition of a centered $\mathbf{d}_\mathrm{test}$ :
      $\mathbf{d}_\mathrm{test} \simeq \mathcal{T} \times_2 \mathbf{r}_\mathrm{f}^\mathrm{T} \times_3 \mathbf{r}_\mathrm{c}^\mathrm{T} \Leftarrow \text{Multilinear Projection}(\mathcal{T}, \mathbf{d}_\mathrm{test})$
  (7) Finding linear SVM decision boundaries using validation set
  (8) classifying all $\mathbf{d}_\mathrm{test} \in$ test set
```

3.7 Dimensionality reduction in step 3

As mentioned earlier, factor matrix $\mathbf{U}_f$ comprises underlying structures of basis vector continent. Despite the fact that we construct our predictive model by approximating discriminating regions, still there are many shared components which getting rid of them make the model more distinguishable. Since we are interested in noisy regions i.e., artifacts, we propose truncating components of the core tensor $\mathcal{T}$ which correspond to top values of $\mathbf{U}_f$ and keeping lower value components as representatives of noisy parts. In the next section, we will show how this truncation boosts the classification performance of the proposed framework. An example of truncating components corresponding to the second mode of a 3-mode tensor is depicted in Figure. 3.

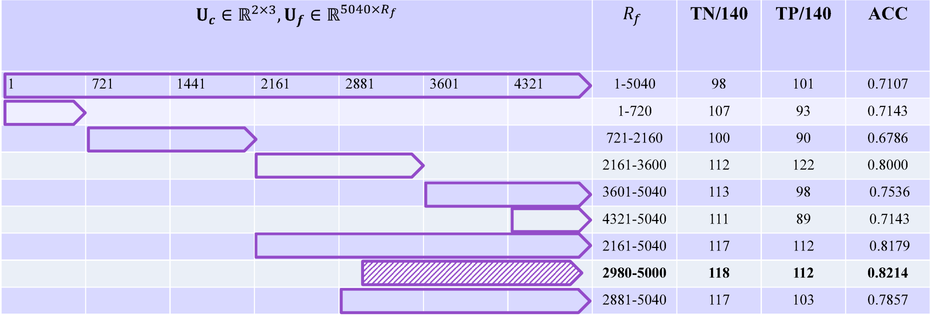
4 EXPERIMENTAL EVALUATION

In this section, we first introduce the dataset and benchmark on this dataset and then we discuss the implementation details and the experimental evaluation.

4.1 Dataset description

One of the most popular and widely used databases for image or video forgeries detection is FaceForensics++[6] which first was introduced in 2018 [19]. FaceForensics++ comprises more than 500,000 frames from 1000 youtube videos that contain mostly frontal

[6] https://github.com/ondyari/FaceForensics


$\mathbf{U}_c \in \mathbb{R}^{2\times3}, \mathbf{U}_f \in \mathbb{R}^{5040\times R_f}$
1
721
1441
2161
2881
3601
4321

R_f	TN/140	TP/140	ACC
1-5040	98	101	0.7107
1-720	107	93	0.7143
721-2160	100	90	0.6786
2161-3600	112	122	0.8000
3601-5040	113	98	0.7536
4321-5040	111	89	0.7143
2161-5040	117	112	0.8179
2980-5000	**118**	**112**	**0.8214**
2881-5040	117	103	0.7857

Figure 4: Dimensionality reduction experiments video frames compressed with quantization 23. Truncating top 2979 and bottom 40 components of the core tensor corresponding to factor matrix U_c increases the classification performance. Significant component mostly represent high level structures while insignificant ones may represent noise e.g., artifacts which we want to leverage as a discriminating feature.

Method	Accuracy
ZAntiFakeBio	**1.000**
Leo	**1.000**
Aquarius	**1.000**
RobustForensics	0.991
NoSenseAtAll	0.982
PredictFake	0.973
Cancer	0.964
Balance	0.918
unet+res	0.882
HRC	0.827
GAEL-Net	0.718

Table 2: Summary of Benchmark on FaceForensics++, DeepFakes method [20]

.

faces [20]. This dataset also includes 1000 videos which are the manipulated version of the original onesand have been manipulated by four automated face manipulation methods: Deepfakes, Face2Face, FaceSwap and NeuralTextures. All original and manipulated videos have constant frame rate of 30 fps and have been compressed lossless with H.264. Moreover, the videos are split up into train set of size 720, validation set of size 140 and test of 140 videos. binary classification scenario on this dataset. A summarized benchmark of existing techniques on videos manipulated by DeepFakes method is demonstrated in table 2. The state-of-the-art benchmark on FaceForensics++ is available in GitHub[7]. In this work, we experiment on images manipulated by DeepFake technique.

4.2 Implementation

Our work was implemented in MATLAB partially using Tensor Toolbox version 2.6. [2, 3]. Since all videos have constant frame rate 30 fps, we extracted up to 7 frames for each video by snapping almost one frame per each 30 seconds using OpenCV library in Python. Moreover, for detecting facial landmarks, we used pretrained dlib face detector[8] which is created using the classic Histogram of Oriented Gradients (HOG) feature combined with a linear classifier, an image pyramid, and sliding window detection scheme [11]. For the second step, we calculated the SVD rank r where the r is equal to "number of train videos × 7 = 720 × 7 = 5040" for all experiments. The intuition behind this estimation is to have an individual component for each frame. Moreover, in contrast to many deep learning approaches for Deepfake detection, our approach does not require GPU base configuration and both train and test steps can be executed on an ordinary CPU based configuration. The description of the CPU based configuration we experimented on is as follows: Intel(R) Core (TM) i5-8600K CPU @3.60GHz,CentOS Linux 7 (Core) operating system and 40GB RAM memory.

4.3 Evaluation

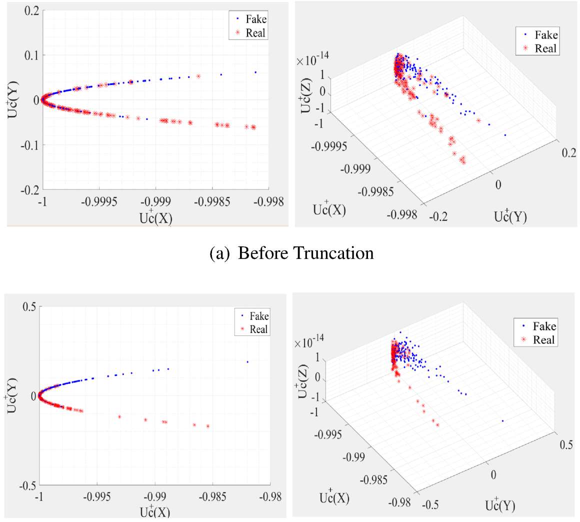
4.3.1 Classification performance. Classification performance of our proposed multilinear framework when we keep all of the components as well as when we truncate different ranges of components, is illustrated in Figure. 4. In this Figure, TN, TP, and ACC. denote true negative, true positive, and accuracy respectively.

As demonstrated, truncating top 2980 and bottom 40 components, significantly improves the classification accuracy. In this work, we aim to find discriminating representations for outer ring of real vs Deepfake videos introduced by synthesizing artifacts. Thus, we hypothesize the noisy components i.e., components with insignificant values may represent those artifacts. So, by truncating the top components, we avoid high level facial structures and only keep those that correspond to what we aim to capture i.e., artifacts, for the classification. Moreover, the last 40 components are the most insignificant ones that might be introduced by noises other than synthesizing artifacts. Anyhow, keeping components in range $2980 - 5000$ results in around 0.82% accuracy.

4.3.2 Effects of truncation on class representations. To clarify the efficacy of truncation, we depict the PCA coefficients of column vectors of $\mathbf{U}_\mathbf{c}^+$ for test frames before and after applying truncation. The distribution of the PCA coefficients is demonstrated in Figure.5. As shown, truncating the undiscriminating components, makes coefficients of each class more similar and as a result put them closer to each other. Specially in case of samples that are located in outer parts of the semicircle i.e., outliers. In other words, the representations after truncation are more linearly separable than those before applying the truncation.

[7] http://kaldir.vc.in.tum.de/faceforensics-benchmark/

[8] http://dlib.net/face_landmark_detection.py.html



(a) Before Truncation

(b) After truncation (2980-5000)

Figure 5: Distribution of class coefficients before and after truncation. As illustrated, after truncation, data points of each class get closer to each other and as a result, the number of outliers decreases significantly and and the classes are more linearly separable. In these plots, there are scale differences in the axes but in reality the distributions are nearly straight lines.

5 CONCLUSION AND FUTURE WORK

In this work, we leverage the region that we hypothesize has highest concentration of artifacts, the face outer ring, for classification of Deepfakes using our proposed multilinear framework. Our preliminary results show that using only the outer facial ring we achieve 82% accuracy. In future work, we will learn class representations by subdividing an image into parts [24] and treating them as either items in a part-based hierarchy or as items in a "bag of parts" whose representations may learned bottom-up [27]. Another direction for future work, is to use binary masks released by [20]. The binary mask can be leveraged for precise segmentation of the frames into regions of interest.

6 ACKNOWLEDGEMENTS

The authors would like to thank Ghazal Mazaheri and Amit Roy-Chowdhury for initial help with the dataset. Research was supported by the National Science Foundation CDS&E Grant no. OAC-1808591 and a UCR Regents Faculty Fellowship. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the funding parties.

REFERENCES

[1] Darius Afchar, Vincent Nozick, Junichi Yamagishi, and I. Echizen. Mesonet: a compact facial video forgery detection network. 09 2018.

[2] Brett W. Bader and Tamara G. Kolda. Efficient MATLAB computations with sparse and factored tensors. SIAM Journal on Scientific Computing, 30(1):205–231, December 2007.

[3] Brett W. Bader, Tamara G. Kolda, et al. Matlab tensor toolbox version 2.6. Available online, February 2015.

[4] Md Jawadul Bappy, Cody Simons, Lakshmanan Nataraj, B. Manjunath, and Amit Roy-Chowdhury. Hybrid lstm and encoder-decoder architecture for detection of image forgeries. IEEE Transactions on Image Processing, PP, 01 2019.

[5] J. D. Carroll and J. J. Chang. Analysis of individual differences in multidimensional scaling via an N-way generalization of 'Eckart-Young' decomposition. Psychometrika, 35:283–319, 1970.

[6] François Chollet. Xception: Deep learning with depthwise separable convolutions. pages 1251–1258, 2017.

[7] L. de Lathauwer, B. de Moor, and J. Vandewalle. A multilinear singular value decomposition. SIAM J. of Matrix Analysis and Applications, 21(4):1253–78, 2000.

[8] L. de Lathauwer, B. de Moor, and J. Vandewalle. On the best rank-1 and rank-$(R_1, R_2, \ldots, R_n)$ approximation of higher-order tensors. SIAM J. of Matrix Analysis and Applications, 21(4):1324–42, 2000.

[9] David Guera and Edward Delp. Deepfake video detection using recurrent neural networks. pages 1–6, 11 2018.

[10] Javier Hernandez-Ortega, Ruben Tolosana, Julian Fierrez, and Aythami Morales. Deepfakeson-phys: Deepfakes detection based on heart rate estimation, 2020.

[11] Vahid Kazemi and Josephine Sullivan. One millisecond face alignment with an ensemble of regression trees. In 2014 IEEE Conference on Computer Vision and Pattern Recognition, pages 1867–1874, 2014.

[12] Tamara G. Kolda and Brett W. Bader. Tensor decompositions and applications. SIAM Review, 51(3):455–500, September 2009.

[13] Yuezun Li, Ming-Ching Chang, Hany Farid, and Siwei Lyu. In ictu oculi: Exposing ai generated fake face videos by detecting eye blinking. 06 2018.

[14] Yuezun Li and Siwei Lyu. Exposing deepfake videos by detecting face warping artifacts. CoRR, abs/1811.00656, 2018.

[15] S. McCloskey and M. Albright. Detecting gan-generated imagery using saturation cues. In 2019 IEEE International Conference on Image Processing (ICIP), 2019.

[16] Huy Nguyen, Fuming Fang, Junichi Yamagishi, and I. Echizen. Multi-task learning for detecting and segmenting manipulated facial images and videos. 06 2019.

[17] Huy Nguyen, Junichi Yamagishi, and I. Echizen. Use of a capsule network to detect fake images and videos. 10 2019.

[18] Evangelos E. Papalexakis, Christos Faloutsos, and Nicholas D. Sidiropoulos. Tensors for data mining and data fusion: Models, applications, and scalable algorithms. ACM Trans. Intell. Syst. Technol., 8:16:1–16:44, 2016.

[19] Andreas Rössler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. Faceforensics: A large-scale video dataset for forgery detection in human faces. arXiv, 2018.

[20] Andreas Rössler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. FaceForensics++: Learning to detect manipulated facial images. In International Conference on Computer Vision (ICCV), 2019.

[21] N.D. Sidiropoulos, Lieven De Lathauwer, Xiao Fu, Kejun Huang, Evangelos Papalexakis, and Christos Faloutsos. Tensor decomposition for signal processing and machine learning. IEEE Transactions on Signal Processing, PP, 07 2016.

[22] Ruben Tolosana, Ruben Vera-Rodriguez, Julian Fierrez, Aythami Morales, and Javier Ortega-Garcia. Deepfakes and beyond: A survey of face manipulation and fake detection. arXiv preprint arXiv:2001.00179, 2020.

[23] Mathew A. Turk and Alex P. Pentland. Eigenfaces for recognition. Journal of Cognitive Neuroscience, 3(1):71–86, 1991.

[24] M. Vasilescu and D. Terzopoulos. Adaptive meshes and shells: Irregular triangulation, discontinuities, and hierarchical subdivision. In Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'92), page 829832, Champaign, IL, Jun 1992.

[25] M. Alex O. Vasilescu. A Multilinear (Tensor) Algebraic Framework for Computer Graphics, Computer Vision, and Machine Learning. PhD thesis, University of Toronto, 2009.

[26] M. A. O. Vasilescu. Multilinear projection for face recognition via canonical decomposition. In Proc. IEEE Inter. Conf. on Automatic Face Gesture Recognition (FG 2011), pages 476–483, Mar 2011.

[27] M. Alex O. Vasilescu, Eric Kim, and Xiao S. Zeng. Causalx: Causal explanations and block multilinear factor analysis. In 2020 25th International Conference of Pattern Recognition (ICPR 2020), pages 10736–10743, Jan 2021.

[28] M. A. O. Vasilescu and D. Terzopoulos. Multilinear analysis of image ensembles: TensorFaces. In Proc. European Conf. on Computer Vision (ECCV 2002), pages 447–460, Copenhagen, Denmark, May 2002.

[29] M. A. O. Vasilescu and D. Terzopoulos. Multilinear independent components analysis. In Proc. IEEE Conf. on Computer Vision and Pattern Recognition, volume I, pages 547–553, San Diego, CA, 2005.

[30] M. Alex O. Vasilescu and Demetri Terzopoulos. Multilinear projection for appearance-based recognition in the tensor framework. In IEEE 11th International Conference on Computer Vision, ICCV 2007, Rio de Janeiro, Brazil, October 14-20, 2007, pages 1–8. IEEE Computer Society, 2007.

[31] X. Yang, Y. Li, and S. Lyu. Exposing deep fakes using inconsistent head poses. pages 8261–8265, 2019.