Statistical Analysis of Deepfake Images and Videos in Smart City Applications

ANAND M K

242CS008

1 Introduction

Deepfake technology, which uses advanced machine learning algorithms such as generative adversarial networks (GANs), has made it possible to create highly realistic but entirely synthetic media, including images, videos, and audio. While deepfakes have a range of creative applications, they also present significant risks, particularly in sensitive areas where the authenticity of media is crucial. One such area is the functioning of smart cities, where a wide range of systems rely on media data for decision-making, including surveillance, traffic monitoring, and public safety mechanisms.

In this report, we focus on the statistical analysis of deepfake image and video datasets, specifically examining basic statistical measures such as the mean, median, and mode of pixel intensities. These measures, though simple, are fundamental in exploring the distribution of pixel values and identifying patterns that could indicate synthetic manipulation of media. By applying statistical methods to deepfake datasets, we aim to uncover the intrinsic properties of deepfake images and videos, such as unusual patterns in pixel intensity distributions, that might differentiate them from authentic media.

Additionally, this analysis will explore more advanced statistical techniques to analyze the variations in pixel intensities, including variance, skewness, and kurtosis. These higher-order statistical properties provide deeper insights into the structure and texture of media, potentially highlighting anomalies that are characteristic of deepfake images and videos. The goal is not to detect deepfakes directly but to understand their statistical behavior and identify patterns that may emerge from their generation.

The results of this analysis will provide valuable insights into the statistical characteristics of deepfake media and how these properties may impact data-driven decision-making in smart cities. Understanding these characteristics will help inform future studies and methodologies for dealing with synthetic media in urban environments, ensuring that media used in smart city systems is both reliable and representative of the real world.

By examining deepfake datasets through the lens of statistical analysis, this study contributes to a more nuanced understanding of how statistical measures can be used to analyze and understand synthetic media, laying the foundation for more informed approaches to handling media data in smart cities.

2 Datasets Used

The following datasets were used for the experiments:

- **FaceForensics**: A large-scale dataset containing real and manipulated videos, focusing on deepfake detection.
- **OpenForensics**: A dataset designed for forensic analysis, including real and synthetic images.
- **UADFV**: A dataset containing real and deepfake videos, commonly used for deepfake detection research.

Below are sample images of one real and one fake (deepfake) image from the datasets:

(a) Real Image

(b) Fake (Deepfake) Image

Figure 1: Sample images: Real and Fake (Deepfake) from the datasets.

3 Descriptive Statistics

Descriptive statistics refers to a set of techniques used to summarize and describe the main features of a dataset. It provides simple summaries about the sample and the measures, such as central tendency (mean, median, mode), variability (variance, standard deviation), and distribution (histograms). These methods help in understanding the basic characteristics of the data, making it easier to interpret and analyze.

3.1 Mean, Median, and Mode of Pixel Intensities

The mean, median, and mode are measures of central tendency that describe the average, middle, and most frequent pixel intensity values in an image, respectively.

Image Type	Mean	Median	Mode
Real Image	101.221	95.0	84
Deepfake Image	94.787	91.0	0

Table 1: Mean, Median, and Mode of Pixel Intensities

3.2 Variance and Standard Deviation of Pixel Values

Variance and standard deviation measure the spread of pixel intensity values around the mean. Variance is the average of the squared differences from the mean, while standard deviation is the square root of variance.

Image Type	Variance	Standard Deviation
Real Image	1392.574	37.317
Deepfake Image	1902.004	43.612

Table 2: Variance and Standard Deviation of Pixel Values

3.3 Histograms of Pixel Distributions

A histogram represents the distribution of pixel intensities in an image. It shows the frequency of each intensity value.

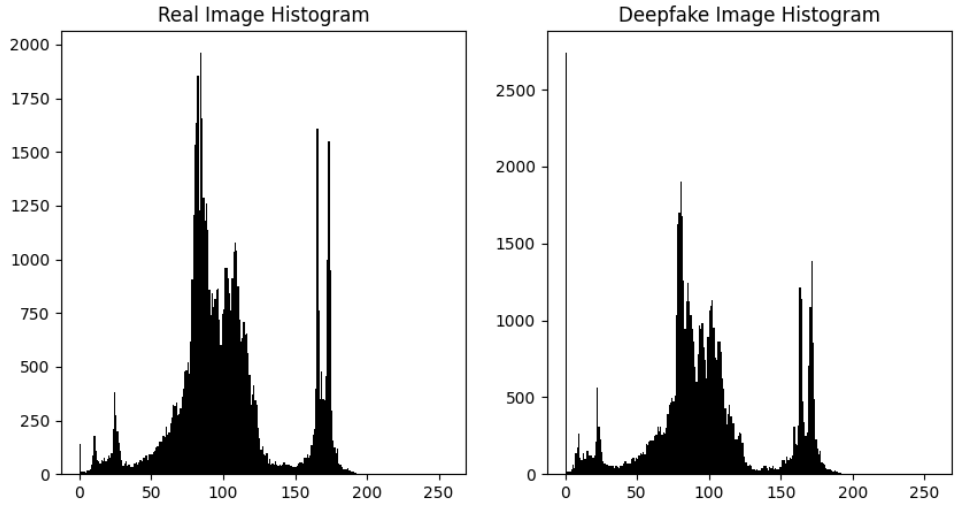

Figure 2: Histograms of pixel distributions for real and deepfake images.

3.4 Correlation Between Pixel Values

Correlation measures the relationship between pixel values in different regions of an image. A value close to 1 indicates a strong positive correlation, while a value close to -1 indicates a strong negative correlation.

Image Type	Correlation
Real Image	-0.038
Deepfake Image	0.171

Table 3: Correlation Between Pixel Values

3.5 Temporal Statistics for Videos

Temporal statistics analyze frame-by-frame changes in pixel intensity for videos. This helps in understanding the dynamics of pixel intensity over time.

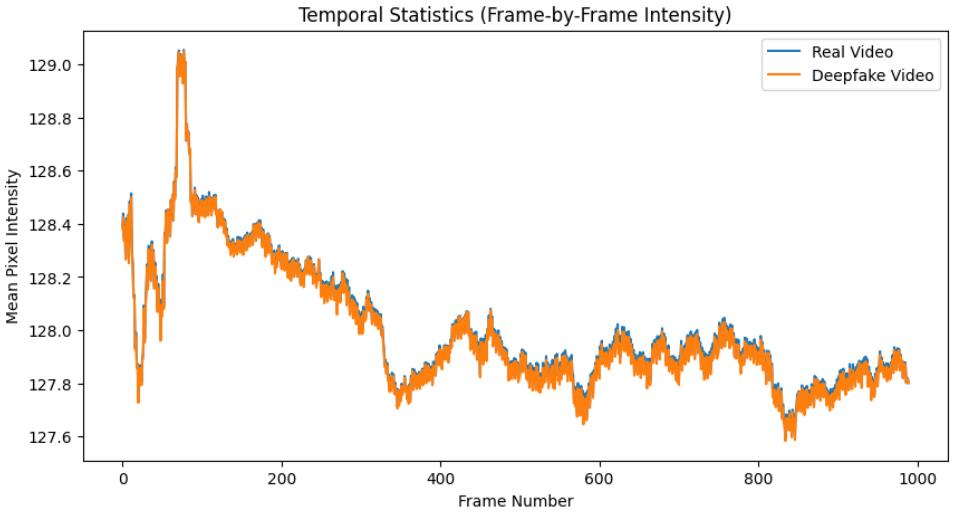

Figure 3: Temporal statistics (frame-by-frame intensity changes) for real and deepfake videos.

4 Feature Extraction Statistics

Feature extraction is a crucial step in image and video analysis, allowing us to obtain meaningful patterns and characteristics from the data. Various methods are applied to analyze texture, edges, frequency components, and motion to distinguish between real and deepfake media. This section presents the results of applying multiple feature extraction techniques to both real and deepfake images and videos.

4.1 Canny Edge Detection

Canny edge detection is a multi-stage algorithm used to detect edges in an image. It applies:

1. A Gaussian filter to smooth the image and reduce noise.
2. Gradient computation to identify intensity changes.
3. Non-maximum suppression to thin out edges.
4. Hysteresis thresholding to detect strong and weak edges.

This method is effective for highlighting boundaries and structural features in an image.

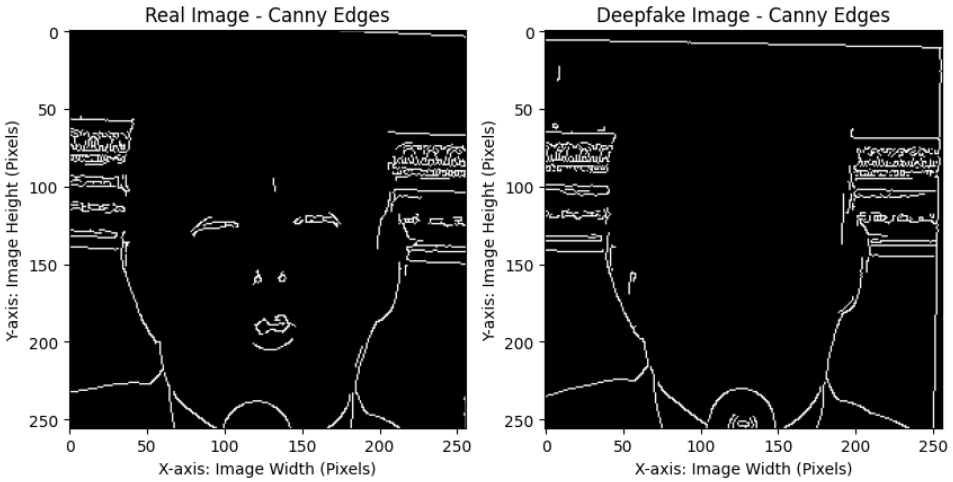

Figure 4: Real Image - Canny Edge Detection

4.2 Haralick Features

Haralick features are texture features derived from the Gray-Level Co-occurrence Matrix (GLCM). These features capture statistical properties of texture, including:

1. **Contrast:** Measures the intensity difference between adjacent pixels.
2. **Correlation:** Describes the degree of similarity between neighboring pixels.
3. **Energy:** Indicates uniformity in texture patterns.
4. **Homogeneity:** Measures how uniform the intensity distribution is.

Haralick features are widely used in texture classification and image analysis.

Feature	Real Image	Deepfake Image
Contrast	57.7027	104.8085
Correlation	0.9793	0.9723
Energy	0.0540	0.0693
Homogeneity	0.4950	0.5452

Table 4: Haralick Features for Real and Deepfake Images

4.3 Sobel Edge Detection

The Sobel operator detects edges by computing intensity gradients at each pixel. It enhances regions with high spatial frequency, making edges more pronounced.

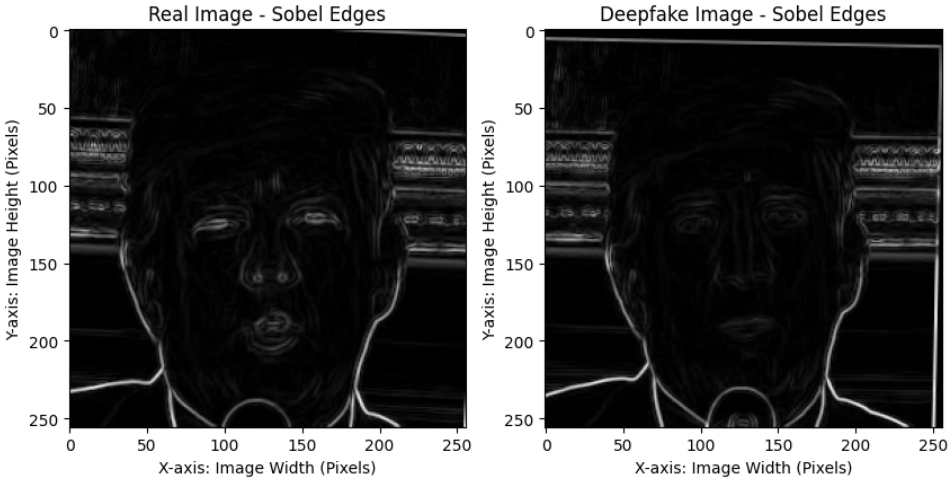

Figure 5: Real Image - Sobel Edge Detection

4.4 Local Binary Patterns (LBP)

Local Binary Patterns (LBP) encode texture by comparing each pixel with its neighbors:

1. Taking a pixel and comparing it to its neighbors.
2. Assigning binary values based on whether a neighboring pixel is brighter or darker.
3. Converting the binary pattern into a decimal value for classification.

LBP is useful for identifying patterns in textures, such as in facial recognition and material classification.

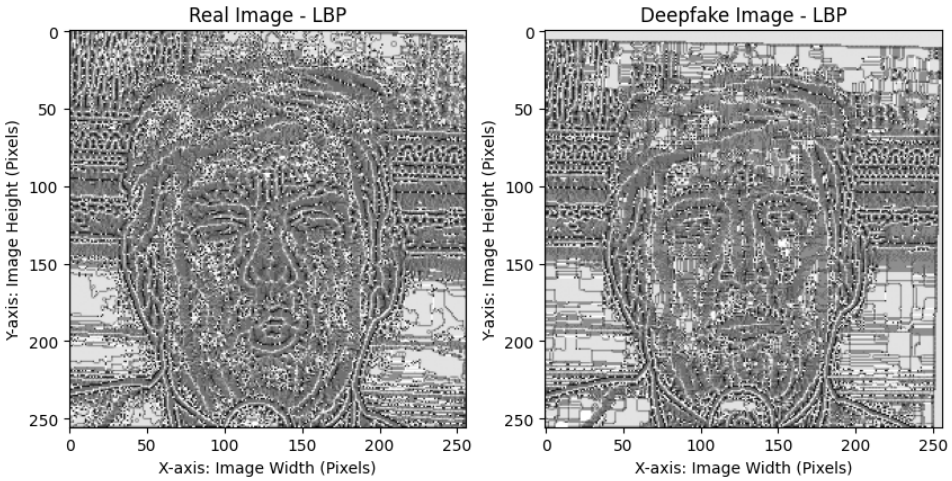

Figure 6: Real Image - Local Binary Patterns

4.5 Fourier Transform

The Fourier Transform decomposes an image into its frequency components. It is useful for:

1. Identifying periodic patterns in textures.
2. Filtering noise and unwanted frequency components.
3. Analyzing frequency-based features in images.

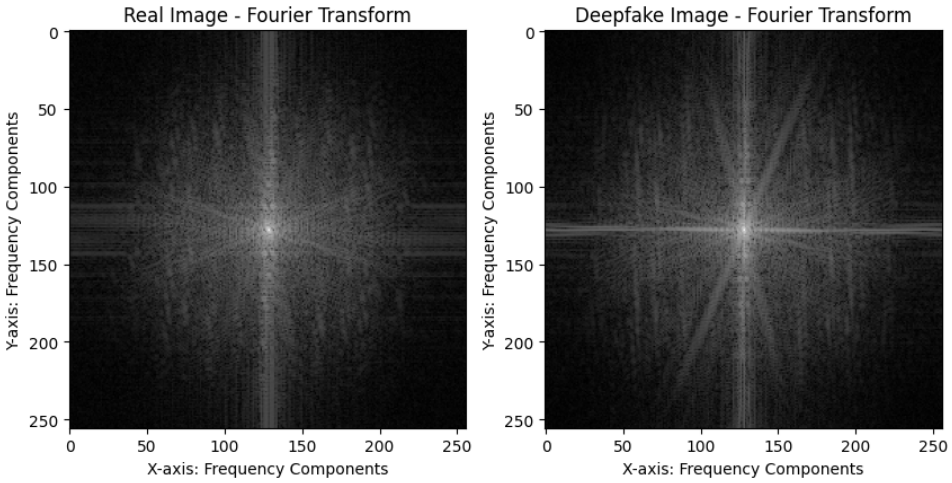

Figure 7: Real Image - Fourier Transform

4.6 Wavelet Transform

The Wavelet Transform decomposes an image into different frequency sub-bands, providing both spatial and frequency information. It breaks an image into:

1. **Approximation (cA):** Represents low-frequency components.
2. **Horizontal Detail (cH):** Highlights horizontal edges.
3. **Vertical Detail (cV):** Highlights vertical edges.
4. **Diagonal Detail (cD):** Captures diagonal edge information.

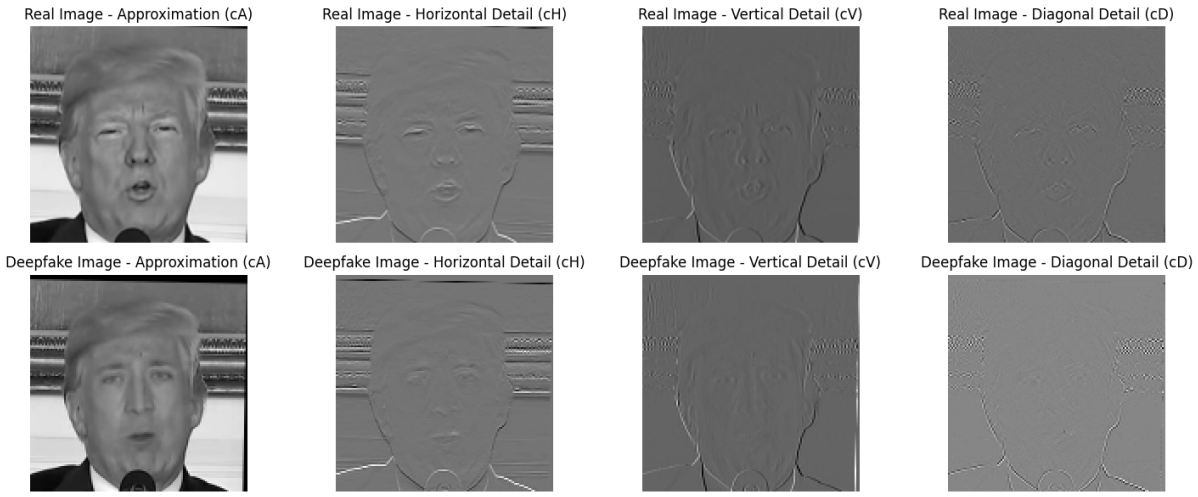

Figure 8: Real Image - Wavelet Transform Approximation (cA)

4.7 Average Frame Difference

The average frame difference method calculates pixel intensity changes between consecutive frames in a video. It helps detect motion and temporal inconsistencies, useful in:

1. Identifying scene changes and motion intensity.
2. Analyzing temporal consistency in real vs. deepfake videos.
3. Spotting unusual changes that could indicate manipulation.

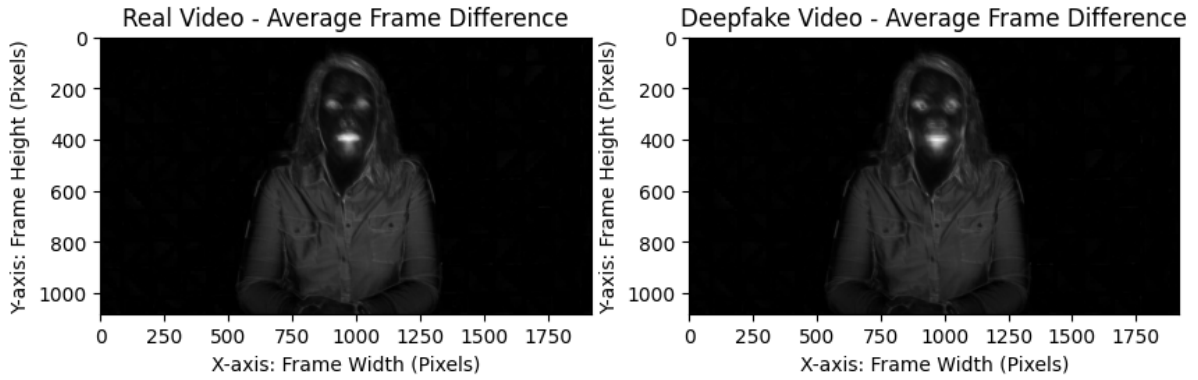

Figure 9: Real Video - Average Frame Difference

5 Statistical Modeling Methods

This section explores the application of statistical modeling techniques to analyze real and deepfake images and videos. The methods used include Gaussian Mixture Models (GMM) for image analysis and Hidden Markov Models (HMM) for video analysis. These models help in understanding the underlying statistical properties and temporal patterns of real and deepfake data.

5.1 Gaussian Mixture Models (GMM)

Gaussian Mixture Models (GMM) are probabilistic models used to represent the distribution of pixel intensities in images. By fitting a mixture of Gaussian distributions to the image pixels, we can analyze their statistical properties and detect patterns that distinguish real and deepfake images.

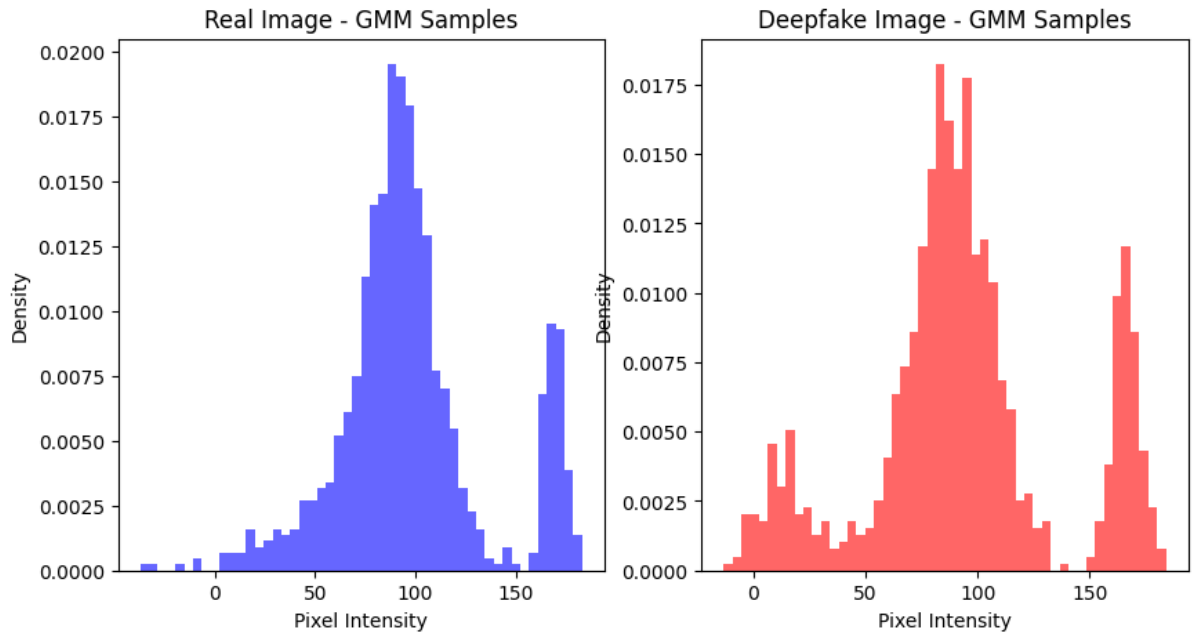

Figure 10: GMM Pixel Intensity Distributions for Real and Deepfake Images

5.2 Hidden Markov Models (HMM)

Hidden Markov Models (HMM) are used to analyze temporal sequences, such as frame-by-frame changes in videos. By modeling the progression of pixel intensities across frames, HMM helps in identifying patterns that differentiate real and deepfake videos.

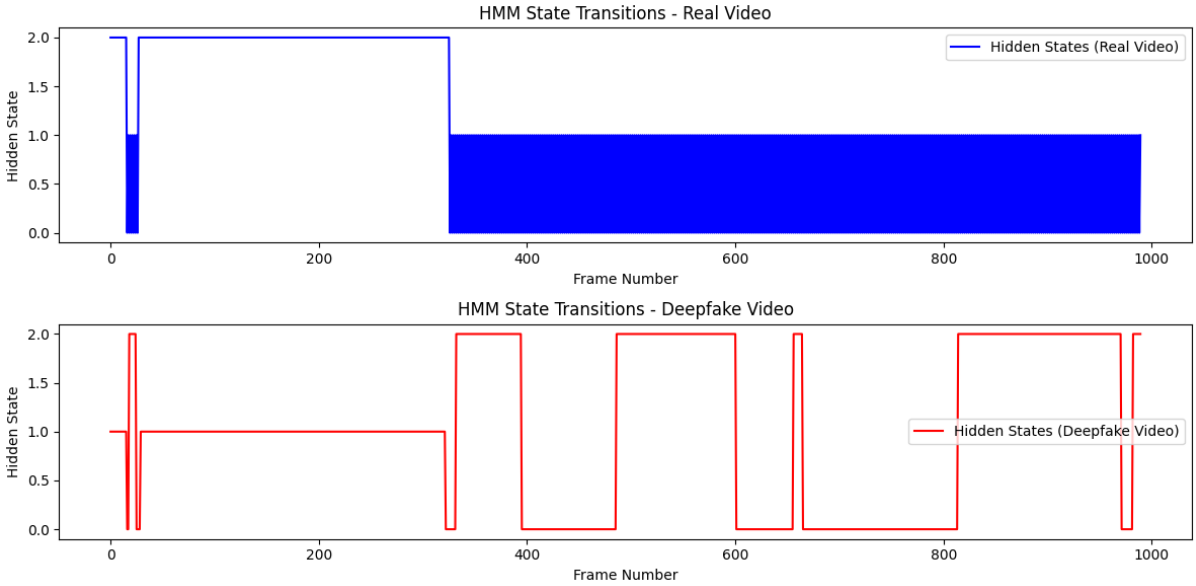

Figure 11: HMM State Transitions for Real and Deepfake Videos

6 Statistical Tools for Analyzing Deepfake Images and Videos

6.1 OpenCV (Open Source Computer Vision Library)

OpenCV is a widely-used open-source computer vision library. It provides several statistical methods that can be applied to deepfake image and video analysis:

- **Histogram Analysis**: Statistical comparison of color histograms to find discrepancies in pixel distribution between real and deepfake images.
- **Optical Flow Analysis**: Optical flow algorithms track the movement of objects across video frames, which can help identify inconsistencies in motion patterns typical of deepfakes.
- **Edge Detection**: OpenCV provides tools like the Canny edge detector that can reveal subtle inconsistencies in deepfake images by analyzing edge features.

6.2 Scikit-Image

Scikit-Image is a Python library for image processing, built on top of Scikit-learn. It offers a variety of statistical tools for image analysis:

- **Texture Analysis**: Local Binary Patterns (LBP) and other texture features can be used to detect statistical differences between real and fake textures.
- **Statistical Image Measures**: Tools to calculate entropy, skewness, and kurtosis of pixel distributions, which can expose unnatural patterns that deepfakes often introduce.

6.3 Matlab and Image Processing Toolbox

Matlab is a high-performance language for technical computing, and its Image Processing Toolbox offers several statistical techniques:

- **Fourier Transform**: Fourier analysis of the frequency domain to reveal inconsistencies in the deepfake's pixel composition that wouldn't occur in natural images.
- **Fractal Dimension Analysis**: This statistical method measures the complexity of textures in images and can be used to detect anomalies in deepfakes.
- **Principal Component Analysis (PCA)**: PCA reduces the dimensionality of image data and reveals the key features, helping identify statistical differences between real and deepfake media.

6.4 PyTorch and TensorFlow

PyTorch and **TensorFlow** are deep learning frameworks that can be used for custom models to perform statistical analysis on deepfake media:

- **Residual Analysis**: Statistical analysis of the difference between predicted (realistic) and actual deepfake images/videos.
- **Discrepancy in Feature Distribution**: Models can be trained to focus on features like textures, motion, or edges in videos, and statistical tests can highlight differences between real and manipulated data.

6.5 NumPy and SciPy

NumPy and **SciPy** are fundamental libraries for scientific computing in Python. They offer functions for:

- **Mean, Standard Deviation, and Variance**: Basic statistical operations to analyze pixel intensities across images or video frames.
- **Correlation Coefficients**: Compute pixel correlations across regions to find inconsistencies in deepfake images or videos.
- **Statistical Clustering**: Methods like k-means clustering can be used to group image pixels and analyze statistical differences between real and fake images.

Conclusion

In this study, various statistical methods were applied to analyze real and deepfake images and videos. Descriptive statistics, including mean, median, variance, and histograms, provided insights into pixel intensity distributions. Feature extraction techniques, such as edge detection, texture analysis, and frequency domain transformations (Fourier and Wavelet), highlighted structural differences between real and manipulated content. Statistical modeling using Gaussian Mixture Models (GMM) and Hidden Markov Models (HMM) further captured the underlying distributions and temporal dynamics of the data. Tools like OpenCV, Scikit-Learn, and HMMLearn, along with datasets such as FaceForensics++ and DFDC, were instrumental in implementing these methods. The results demonstrated significant differences in pixel intensity, texture, and temporal patterns, offering a foundation for robust deepfake detection systems. Future work could integrate deep learning models and explore larger datasets to enhance detection accuracy and real-time applicability. Overall, this study underscores the effectiveness of statistical methods in understanding and combating deepfake media.