Visualization of Deepfake detection using forensic methods

Kriti Charudatt Kadolkar
EEMCS
University of Twente
Enschede, Netherlands
k.c.kadolkar@student.utwente.nl

***Abstract*—The increased availability and ease of use of deep learning frameworks have led to a rise in the generation and distribution of deepfake images and videos [11]. This has created a pressing need for algorithms capable of identifying and characterizing these manipulated media [12]. While deep learning frameworks are a popular choice for such tasks, their black-box nature makes it challenging to understand or explain the features leading to a classification decision [13]. This lack of explainability renders these methods unsuitable for scenarios where interpretability is essential [14].**

This study explores two approaches for deepfake detection: forensic methods and deep learning models. The forensic methods focus on visualizing specific artifacts in manipulated media, such as inconsistencies in blending, color, and texture in targeted facial regions like cheeks and eyebrows. These techniques enhance interpretability by highlighting anomalies and facilitating an understanding of manipulations through visual evidence. Conversely, the deep learning approach employs a Convolutional Neural Network (CNN) trained with K-Fold Cross-Validation on the cheek, eyebrow, and full-frame regions, prioritizing classification accuracy over interpretability.

A comparative analysis reveals the strengths and limitations of each method, emphasizing the trade-offs between accuracy and explainability. The findings contribute to the development of hybrid strategies that combine the strengths of forensic visualization with the predictive power of deep learning, advancing the field of deepfake detection. This increases the explanability of deep learning models by not only looking at the larger picture but also at specific parts which contain artifacts that can then be highlighted using forensic method.

***Index Terms*—Visualization, Deepfakes Detection, Forensic Methods, Deep learning**

I. Introduction

With the rise of availability of deep learning frameworks and the advancement of generative models such as Generative Adversarial Networks (GANs), there has been a surge in manipulated images and videos being created. The ease of access to these tools, along with the simplicity of building deep learning models from scratch, has enabled the widespread creation of AI-generated and text-prompt-based images and videos. Deep learning-based frameworks have facilitated not only the generation of fully synthetic images but also manipulations of existing images and videos [11]. While these tools have beneficial applications, their misuse can lead to malicious consequences. As a result, it has become imperative to develop algorithms and processes capable of distinguishing real and fake media.

A wide variety of methods have been proposed for this purpose, ranging from forensic techniques to deep learning-based approaches. Additionally, several hybrid methods combine these approaches for improved decision-making. However, most proposed methods lack visualization-based outputs that help convey the rationale behind their decisions [15]. Decisions are often derived either through opaque, black-box deep learning frameworks or by visual inspection of artifacts present in the media [13]. Explaining these results to audiences not well-versed in the methods used is challenging, making it harder to effectively convey the basis of decisions regarding authenticity.

This study investigates two approaches: a forensic method that uses techniques to visualize specific artifacts and classify them based on these artifacts, and a deep learning method that classifies media based on extracted features or full-frame analysis. The comparative analysis aims to highlight the balance between the explainability of forensic methods and the accuracy of deep learning techniques. This exploration seeks to address the gap between explainability and accuracy, emphasizing the strengths and limitations of each method.

The proposed approach involves forensic techniques such as identifying inconsistencies in blending within specific regions of the face (cheeks, eyebrows) and visualizing these inconsistencies. These results are compared with those produced by a deep learning method that receives the same input data, which can analyze the cheek, eyebrows, or full frames of the video to classify them. The study emphasizes the visualization of artifacts identified by the forensic method while analyzing the accuracy achieved by the deep learning model across these input options. The outcomes provide insights into the trade-offs between visual interpretability and detection accuracy, offering valuable contributions to the development of effective and understandable deepfake detection methods.

II. Deepfakes

Deepfake takes its roots from the words ”Deep learning” and ”fake” [16]. This term was coined by a Reddit user with the same username. As the name indicates DeepFakes are images and videos that are generated using deep learning method such as Generative Adversarial Network (GANs), deep learning, Convolutional Neural Networks (CNN) etc [17]. These images have been manipulated to present them to viewers as a real life image or video. Visual media in these images can be partially or fully generated using deep learning based mechanisms. In the last few years, a large number of generative tools have become available to the public causing a surge in the amount of DeepFake material available online [18] [19]. Most of these are made for fun and artistic creation but some can be harmful as it may be used to propagate misinformation, harassment through non-consensual explicit content, and impersonation for fraud. A popular example is the use of former United States President Barack Obama [23] as a subject for a manipulated video. University of Washington researchers were able to create a video of the former president talking. Since this was not a real video, they used his available photos and images to have him say things he never did. Being a public figure, they were able to gather a large number of his existing photos and videos to train a CNN to replicate and add weight to the misinformation.

Demonstration like these have raised an alarm in our society and made it imperative to create effective detection methods. To identify and remove such media is crucial in maintaining trust as well as safe online environment [20].

Detecting and classifying Deepfakes requires approaches tailored to specific application contexts, as the stakes and priorities vary:

Content Moderation: Platforms such as YouTube and Netflix must prioritize protecting vulnerable audiences, like children, from exposure to manipulated media [21]. In such cases, detection systems must err on the side of caution, favoring high recall (detecting all potential Deepfakes, even with some false positives) over precision.

Legal and Investigative Processes: In criminal investigations or court proceedings, precision and interpretability are paramount [22]. Misclassifying authentic evidence as manipulated—or vice versa—could lead to grave consequences. In these contexts, clear visualizations of detected artifacts are essential for justifying decisions to legal professionals and other stakeholders.

General Public Awareness: Educating the public about the prevalence and risks of Deepfakes also requires detection methods that are interpretable and accessible. Providing clear visual evidence of manipulation helps non-experts understand and trust the detection process.

By trying to build a model that can not only detect but Visualize the artifacts can make it easy for both professionals and laymen to understand the reasoning behind the classification that is gotten as an output from the deep learning model. A hybrid approach combining the advantages of both approaches is a better solution to reach the goals of effective detection and understanding.

III. Literature Review

Deep learning-based approaches have been extensively employed in the detection of deepfakes due to their capability to identify subtle and complex patterns within media that may be imperceptible to human observers. However, a significant limitation of these methods is their inherent opacity—the decision-making process remains hidden within a "black box." This makes it challenging to interpret or explain the rationale behind a model's predictions. Several methods and frameworks have been developed to address this issue, aiming to provide interpretable insights into the decision-making process of deep learning models.

For instance, Bouter et al. [1] introduce ProtoExplorer, a tool specifically designed with forensic analysis in mind. ProtoExplorer not only classifies videos as deepfakes but also enables forensic experts to perform a frame-by-frame analysis of the results. Despite these advantages, ProtoExplorer relies on the black-box nature of deep learning models to make its decisions, limiting its ability to pinpoint specific artifacts. The researchers integrate Prototypical Relevance Propagation (PRP) [2], which incorporates a prototype layer between the final and output layers. This layer highlights the most relevant pixels contributing to the model's decision. While this represents a step toward interpretability, the method primarily identifies the regions significant to the decision rather than the specific artifacts, such as blending errors or color inconsistencies, that signal a deepfake.

Hybrid methods, combining deep learning and forensic approaches, have also gained traction. Salloum et al. [3] propose a methodology that integrates forensic artifact detection with deep learning models. Their approach first employs deep learning to identify suspicious regions and subsequently applies forensic techniques to analyze these regions for artifacts. This combination enhances detection accuracy by leveraging the strengths of both data-driven and evidence-based methodologies.

Verdoliva et al. [4] emphasize the role of explainable AI in deepfake detection, focusing on tools like Grad-CAM [5] and SHAP. These tools generate heatmaps that visually highlight areas within the image or video that the model deems significant for classification. Such methods provide forensic analysts with a data-driven perspective by showcasing the regions of interest. However, similar to PRP, they fall short of identifying specific artifacts, providing only a general visualization of influential areas without detailing the exact features that led to the classification.

On the other hand, some researchers have explored methods that fully utilize deep learning for deepfake detection without explicitly incorporating forensic techniques. Chaudhary et al. [6] propose a multi-level model combining Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) to address both spatial and temporal inconsistencies in deepfakes. By leveraging CNNs for spatial domain analysis and RNNs for temporal domain analysis, their approach achieves improved detection performance. Similarly, Dang et al. [7] focus on multi-modal deepfake detection by integrating audio and visual streams. Their model identifies inconsistencies between these modalities, enhancing the detection of manipulated media.

Forensic-specific methods have also been developed to provide a more interpretable analysis of deepfakes. For example, Salloum et al.'s framework [3] identifies blending errors, unnatural textures, and lighting inconsistencies, which are common artifacts in deepfake videos. These forensic techniques allow investigators to pinpoint tangible evidence of manipulation, which is particularly useful in legal and evidentiary contexts. By combining forensic methods with machine learning, their approach bridges the gap between data-driven and evidence-based detection, making the results more comprehensible to non-experts.

A notable forensic approach is presented by Matern et al. [8], who focus on detecting biological inconsistencies in facial deepfakes. Their method examines physiological artifacts such as irregular blinking rates, unnatural eye movements, and inconsistencies in skin texture or lighting. This approach is highly interpretable as it relies on measurable deviations from normal biological behavior, making it easier to explain and validate.

Another innovative forensic technique is proposed by Guarnera et al. [9], who leverage the analysis of frequency-domain artifacts to detect deepfakes. By examining the spectral properties of an image, their method can identify inconsistencies introduced during the generative process. This frequency-based analysis provides a robust and interpretable way to detect tampering, as it directly highlights the manipulations in the signal structure.

In a comparative study, Korshunov et al. [10] evaluate the performance of various forensic and deep learning-based methods in detecting deepfakes across multiple datasets. Their findings indicate that while deep learning models excel in overall detection accuracy, forensic methods offer superior interpretability and are particularly effective in identifying specific artifacts. Hybrid approaches that combine these methodologies demonstrate the best balance between accuracy and explainability, highlighting the need for integrated frameworks.

Despite these advancements, the field of explainable deepfake detection remains an ongoing challenge. The integration of forensic techniques with deep learning offers a promising path forward. By providing clear, evidence-based explanations for model decisions, these hybrid methods not only improve detection accuracy but also enhance trust in the results. This is crucial in applications where interpretability and transparency are paramount, such as legal investigations and regulatory compliance.

IV. Explainable detection

Deep learning based approaches have been employed by researchers to detect deepfakes. One of the biggest issues with deep learning based approach for deepfake detection is that the decision process is entire hidden. It is a black-box where data is entered, the system gives an output with a decision. This decision can be visualised using tools such as Grad-CAM [5] or Prototypical Relevance Propagation [2]. This process does indicate which area of the image or video played a major part in the decision process. The visualization is done with a heat-map over the area. However even with this it is hard to see the exact artifacts that indicate that the image or video was a deepfake. It may also not be able to fully explain why an image was classified as a deepfake or not.

To be able to fully show whether or not a image or a video is deepfake forensic methods are needed. They can be used to precisely point out artifacts such as color inconsistencies and blending errors that may be present in the video. Pinpointing certain artifacts will help explain to a novice exactly why the classification as deepfake was made. This can be useful in situations such as legal documents and evidence in court where understanding of machine learning may not be common. Since the forensic methods is evidence based its easier to explain than the inner workings of a neural networks. Deep learning-based approaches have become a cornerstone in the detection of deepfakes due to their ability to identify patterns in data that might be imperceptible to the human eye. However, one of the most significant challenges with these methods is their lack of transparency. The decision-making process in deep learning models is often described as a "black-box," where input data undergoes complex transformations to produce an output without a clear, interpretable explanation of how the decision was reached. Tools such as Grad-CAM [5] and Prototypical Relevance Propagation [2] have been developed to address this issue to some extent. These tools provide visualizations, typically in the form of heatmaps, which highlight the areas of an image or video frame that were most influential in the model's decision.

While such visualizations offer a degree of interpretability, they have notable limitations. The heatmaps can show which regions of the media contributed to the classification but often fail to pinpoint the exact artifacts or features that the model used to determine the presence of a deepfake. For example, heatmaps may highlight a face in a video but may not specify whether the decision was based on blending errors, unnatural lighting, or inconsistencies in texture. Furthermore, these visualizations do not inherently explain why these regions were flagged, making it challenging for non-experts to grasp the rationale behind the model's decision. This limitation becomes particularly problematic in high-stakes scenarios, such as legal proceedings, where decisions need to be fully transparent and understandable to individuals without technical expertise in machine learning.

To overcome these challenges, forensic methods can complement deep learning approaches to provide a more interpretable and evidence-based analysis of deepfakes. Forensic techniques focus on identifying specific artifacts and inconsistencies within the media, such as unnatural color transitions, edge blending issues, and temporal mismatches in video sequences. Unlike deep learning models, which abstractly process data to classify it, forensic methods allow investigators to directly observe and highlight tangible evidence of manipulation.

This approach not only enhances the explainability of deepfake detection but also makes the results more accessible to a broader audience. For instance, forensic evidence can be used to visually demonstrate the presence of artifacts to individuals unfamiliar with the technical complexities of neural networks. This can be particularly valuable in contexts where the explanation needs to be robust and defensible, such as in legal documentation, courtrooms, or compliance audits. Forensic methods offer a level of clarity that aligns with the evidentiary standards often required in such settings, as they rely on observable and measurable features rather than abstract model behavior.

Moreover, forensic techniques can help bridge the gap between expert and novice understanding. By providing concrete, visual evidence, they make it easier for stakeholders without a background in machine learning to comprehend why a particular image or video was classified as a deepfake. This could involve showing blending errors along the edges of a manipulated face, color inconsistencies between the original and altered regions, or unnatural motion artifacts in video frames. Such clear, evidence-based explanations are crucial for building trust in the detection process and ensuring that the findings are accepted and understood by diverse audiences.

V. DeepfakeTIMIT dataset

DeepfakeTIMIT dataset [24] is set of videos that have faces swapped using open-source GAN-based approach which was derived from an autoencoder-based DeepFake algorithm. The original dataset from which this was derived vidTIMIT (developed by C. Sanderson et. al.) [25] comprises of video and audio recordings of 43 individual reciting predetermined sentences. For the creation of DeepfakeTIMIT 16 similar looking individual pairs were taken and their faces swapped. Two quality levels were developed lower quality with 64x64 size and higher quality with 128x128 size. There are 10 videos per person in the original dataset with a corresponding 320 videos that totals to 620 videos in the entire dataset. This comprised the fake part of the dataset. This was available for download from the website. For real videos samples the videos from the original dataset vidTIMIT were taken. They are available and can also be downloaded from the website. However, they are stored as frames and audio and have to be reconstructed before use. There are 43 people with 10 recordings each, so the total media is 430 real videos.

For using the deep learning model there needs to be some pre-processing done. As a first step, screenshots are taken at 2 frames per second from each video. Then using Dlib [26] cheek areas are isolated using points between 3-8 for left and points 10-15 for the right cheek. Adding a padding of 10px so that relevant areas are within the image. After this, resize the image is done to 224x224 and then the image is saved. Images are saved into real and fake images folders based on the source being a real video or the swapped Deepfake.

For the forensic methodology the videos are taken just as they are. Depending on the type of method (single or multiple) either the path to the video or the folder is required. For the deep-learning methods the videos of broken down into frames then every fifth frame is taken and added to the full frame folder. From this a 50x50 pixel of each cheek and each eyebrow is taken and added to their respective frames. These are done for both real and deepfake video where the dataset contains three folders cheek, eyebrows and full-frame which each contain real and fake images.

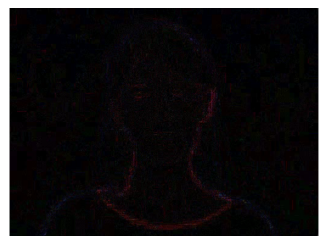
VI. Methodology

There are different approaches for deepfake detection.Its important to first look at approaches that did not fulfill the outcome. Forensic methods provide different starting points from which a method can be developed as well as their being methods that do not reach the conclusion that was hoped for. Here are some of those methods that were looked at and their drawbacks.

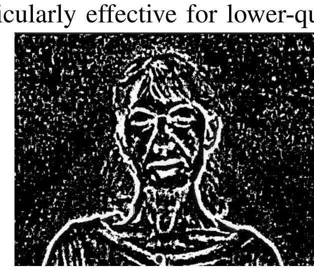
A. previous methods

The goal of this experiment was to highlight specific artifacts present in deepfake videos and visualize them for easier interpretation of classification decisions. Initially, the focus was on analyzing the entire frame to isolate and highlight potential artifacts using various filters. Filters are useful for exposing inconsistencies often present in deepfake images, such as texture mismatches, unnatural edges, or compression irregularities. To begin, five methods were considered:

Error Level Analysis(ELA) Used to detect inconsistencies in compression levels across manipulated and untouched areas.



JPEG CompressionArtifact Analysis Highlighted differences in compression patterns caused by tampered regions, particularly effective for lower-quality videos.



Fast Fourier Transform(FFT) Applied to identify anomalies in frequency patterns, which could indicate tampering.

High-Pass Filtering Enhanced high-frequency details, such as edges, to reveal manipulations that are otherwise difficult to detect.

SIFT PointsCombined with Filters SIFT was used to identify key points in the image where feature descriptors were significantly different. These key points were then analyzed in combination with filters to highlight areas of inconsistencies. The approach aimed to isolate regions where manipulations were most likely to occur, such as facial features or textured areas.

Each of these methods was applied to identify and visualize areas of inconsistencies in images and videos. While these techniques successfully highlighted some manipulated areas—such as the forehead and cheeks—they also highlighted regions that were not manipulated. For instance, areas with greater variations in texture or color, such as the hair or nose-mouth region, were often falsely marked as tampered.

For lower-quality videos, JPEG compression artifact analysis yielded better results, as it was able to emphasize manipulated areas more distinctly. Combining SIFT points with filters also provided additional focus on specific regions of interest, helping to narrow down areas for analysis. However, a significant challenge remained: differentiating genuine manipulations from background noise or natural variations in texture and color.

To better understand the discrepancies between real and manipulated content, side-by-side comparisons were used. These comparisons helped identify why certain inconsistencies were indicative of deepfake artifacts while others were natural variations in authentic images. However, in real-world scenarios, such comparisons are not feasible, as we rarely have access to both the real and manipulated versions of an image or video. This limitation highlights a critical challenge: while these methods can pinpoint areas of interest, they may struggle to provide clear classification or explainability without a reference point.

The primary aim of this experiment was to make deepfake detection more interpretable by visualizing specific artifacts. However, focusing on the entire frame introduced too much noise, making it difficult to distinguish between genuine manipulations and false positives. Similarly, while SIFT-based methods showed promise in narrowing down areas of potential manipulation, they shared the same downfall: without a reference or additional constraints, distinguishing deepfake artifacts from normal image features proved challenging. As a result, it became necessary to narrow the scope to specific regions of the image and focus on distinct artifacts, such as inconsistencies along borders, unnatural textures, or variations in underlying redness. This approach aimed to improve interpretability by reducing noise and honing in on the most relevant features indicative of deepfake manipulation.

B. Dataset and Video Criteria

To ensure reliable analysis and artifact detection, the videos used in the study needed to meet specific criteria:

Participant Orientation Videos required participants to face forward to maintain consistency.

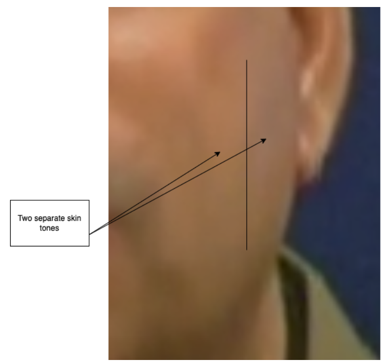
Background A neutral background was chosen to minimize distractions and avoid confounding factors.

Face Alignment Faces were centered and aligned to simplify artifact highlighting and avoid complications from tilts or rotations. Face centering and tilt correction is performed however large deviations can cause issues.

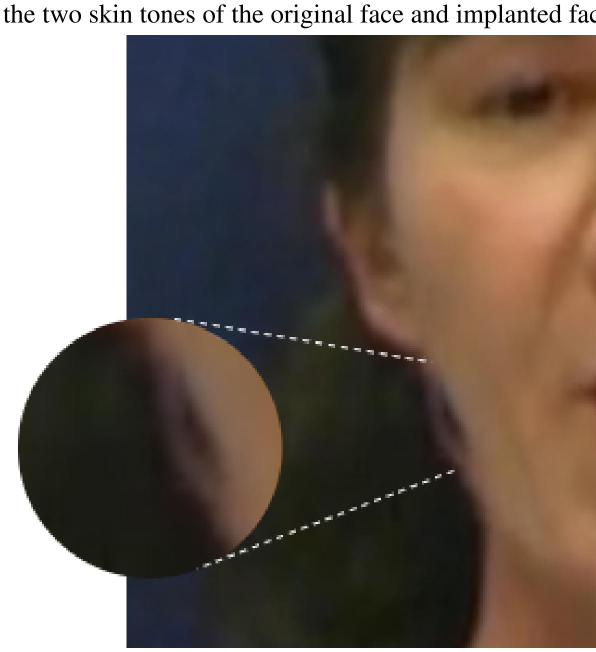
C. Forensic Methods

Building on the insights from the previous section, the next step involved identifying specific regions to highlight and the artifacts those regions presented. Through detailed visual inspection of the videos, the following patterns were observed:

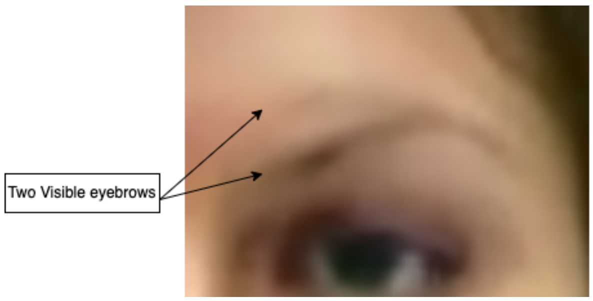
Cheek Area – Color Inconsistency In many videos, the cheek area exhibited noticeable color inconsistencies. This often appeared as a distinct second skin tone overlaid on the background face. These artifacts, likely resulting from poor blending during face generation, can be subtle and easily mistaken for natural shadows in the video if not closely examined.



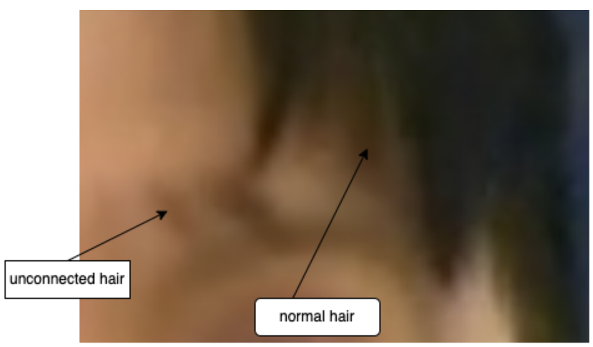
Cheek Area – Background Visibility In certain frames, parts of the background color were visible through the cheeks. This issue, often caused by blending errors or mismatches in the size of the two faces, was especially apparent during head movements. Videos where participants spoke lines exacerbated the issue, as jaw movement made these artifacts more pronounced in some frames. The image above shows a solid line that is that separates the two skin tones of the original face and implanted face.



Eyebrow Area – Double Eyebrows In many frames, discrepancies in eyebrow shapes between the original and manipulated faces were evident. Some frames showed two distinct eyebrows pointing in different directions, while others revealed clearly overlapping eyebrows. These artifacts became particularly noticeable when one participant moved their eyebrows while the other did not, exposing synchronization issues.



Forehead Area – Hair Inconsistencies The dynamic nature of hair made it a common source of artifacts. Issues such as color inconsistencies and differing haircuts were visible clues to potential deepfake manipulations. However, hair-related artifacts also posed challenges similar to those encountered in the previous methods, as they covered a large portion of the frame and could result in false positives, particularly when participants had dyed hair or unconventional styles.



As there are a few artifacts identified for this experiment it was crucial to narrow down which ones were going to be used. As these is a restricted amount of time to implement these as well as capacity to add various overlapping features enhancement, cheek area was selected. This allowed more region to derive artifacts from as well as more area to add features such as lines to highlight artifacts(This is important as the different methods might add details that may affect the subsequent steps). As movement could allow possible enhancements of these artifacts and the dataset contained participants whom were asked to read specific lines, the cheek area was prioritized as expressions such as eyebrow movement may not occur. The color variation(method 2) was noted in other dataset as well so it was another reason to priorities this region. This led to giving priority to cheek area with two artifacts identified as well as third method to combine the methods. The following highlights each of these in detail.

D. Proposed Method 1: Line Derivative for Background Transition

To highlight artifacts, present in the cheek area, specifically where the background appears within the face, it is essential to capture the transition from the background to the facial region. This transition is visualized by drawing a line from the background to the center of the cheek. Ideally, on the intensity graph of the line, only one significant ”jump” should be observed, representing the transition from the background to the face. In deepfake videos, however, additional bumps are expected due to blending inconsistencies.

To enhance this visualization:

1) The derivative of the intensity graph is calculated, where transitions appear as peaks.
2) Deepfake videos typically show extra peaks corresponding to these blending inconsistencies.

For a single video, the user can manually select a point on the cheek, and the transition is visualized dynamically.

For classification, 10 equidistant points are selected automatically along the cheek region. Peaks in the derivative graph are calculated for each point, and frames are classified based on the number of significant peaks.

Algorithm for Multiple Videos:

1) Retrieve Video:Load the video and extract frames using CV2. Convert frames to grayscale for processing.
2) Align Face:Use Dlib points 36 and 45 (eye edges) to calculate the face angle. Use OpenCV functions getRotationMatrix2D and warpAffine to center the face.
3) Define Cheek Points:Use Dlib points 3 and 4 (left cheek) and 11 and 12 (right cheek) to define cheek regions.
4) Generate Lines: Calculate 10 equidistant points along the defined cheek lines. For each point, draw a 50-pixel line (25 pixels on either side) using Bresenham’s algorithm.
5) Normalize Line Pixels: Normalize pixel values along the line to minimize outliers.
6) Calculate Derivative: Compute the derivative of pixel intensities. Invert the derivative for the right side (multiply by -1).
7) Set Threshold: Determine the peak threshold as 50 percent of the maximum derivative value (calculated using np.max).
8) Count Peaks: Count the peaks exceeding the threshold and subtract 1 to identify anomalies.
9) Store Results: Save results for each frame (sum of peaks for left and right sides) in a CSV file.
10) Classification: Label frames as ”Deepfake” if the sum of peaks on either side exceeds a preset threshold(either via stats function or feedback loop), otherwise ”Not Deepfake.”
11) Repeat: Repeat process till all videos are completed.
12) video summary: If 5% number of frames are labeled ”Deepfake” then the video is concluded as Deepfake and summary of each video is stored in a separate csv file.

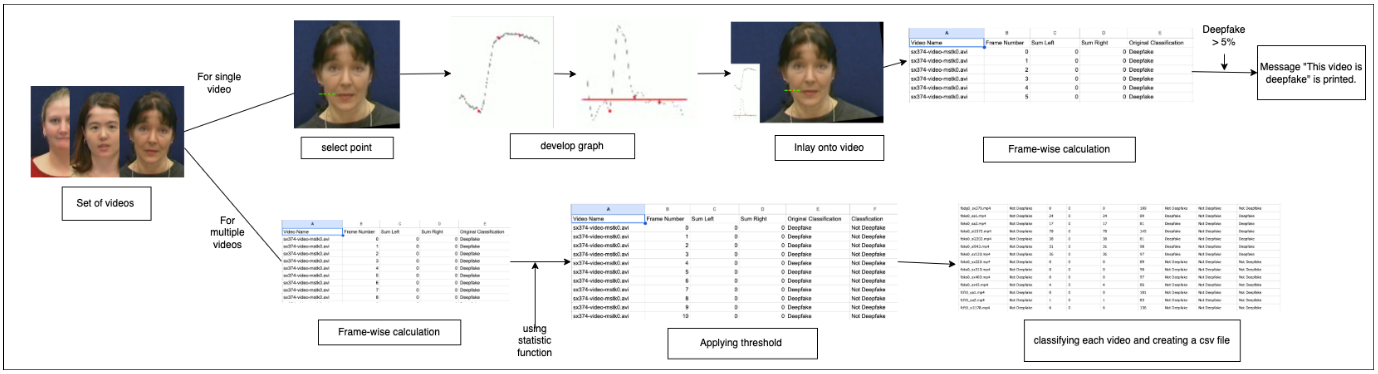


Figure 1: Proposed method 1: flowchart

This produces a CSV file per video that contains frame by frame breakdown of the video. This also generates CSV file called video statistic which contains classification assigned to all videos.

Algorithm for Single Video:

1) Retrieve Video: Load the video and display the first frame for 5 seconds to allow the user to select a point on the cheek manually.
2) Align Face: Align the face using Dlib points 36 and 45 as described for multiple videos.
3) Draw Line: Draw a 50-pixel line from the selected point. Extract pixel values along the line using Bresenham's algorithm.
4) Plot Intensities: Display intensity and derivative graphs using Matplotlib.
5) Set Threshold: Determine the peak threshold as 50 percent of the maximum derivative value (calculated using np.max).
6) Count Peaks: Count the peaks exceeding the threshold and subtract 1 to identify anomalies.
7) Store Results: Save results for each frame (sum of peaks for left and right sides) in an array.
8) Classification: Label frames as "Deepfake" if the sum of peaks on either side exceeds a preset threshold, otherwise "Not Deepfake."
9) print classification: if 5% number of frames are labeled "Deepfake" then the video is concluded as Deepfake and printed.

This produces a video that contains a frame by frame display of intensity graph and derivative graph for simultaneous viewing of changes in value with each frame.

E. Proposed Method 2: Redness-Based Artifact Detection

Artifacts caused by blending issues often result in different skin tones in the cheek area. By isolating and quantifying redness values near the cheek borders, these inconsistencies can be highlighted.

1) Items are numbered automatically.
2) The numbers start at 1 with each use of the `enumerate` environment.
3) Another entry in the list

The redness is calculated as the average pixel value in the red channel for two regions:

1) Near the border (region 1).
2) 30 pixels inside the border (region 2).

The absolute difference between the two averages (termed abs_diff) indicates the level of inconsistency.
Frames are classified as "Deepfake" if abs_diff exceeds a preset threshold.

Algorithm for Multiple Videos:

1) Retrieve Video: Load the video and extract frames using CV2. Convert frames to grayscale for alignment and processing.
2) Align Face: Align the face using Dlib points 36 and 45, as described in Method 1.
3) Define Regions: Use Dlib points 3 and 11 for cheek borders. Offset coordinates by ±5 pixels to define region 1 and by ±30 pixels to define region 2.
4) Calculate Redness: Compute the average redness (average values red channel values within the region) for both regions.
5) Calculate Difference: Compute abs_diff as the absolute difference between the two averages.
6) Store Results: Save left and right abs_diff values in a CSV file.
7) Classification: Label frames as "Deepfake" if either abs_diff value exceeds a preset threshold, otherwise "Not Deepfake."
8) Repeat: Repeat process till all videos are completed.
9) video summary: If 5% number of frames are labeled "Deepfake" then the video is concluded as Deepfake and summary of each video is stored in a separate CSV file.

This produces a CSV file per video that contains frame by frame breakdown of the video. This also generates CSV file called video statistic which contains classification assigned to all videos.

Algorithm for Single Video:

1) Retrieve Video: Load the video and extract frames using CV2. Convert frames to grayscale for alignment and processing.

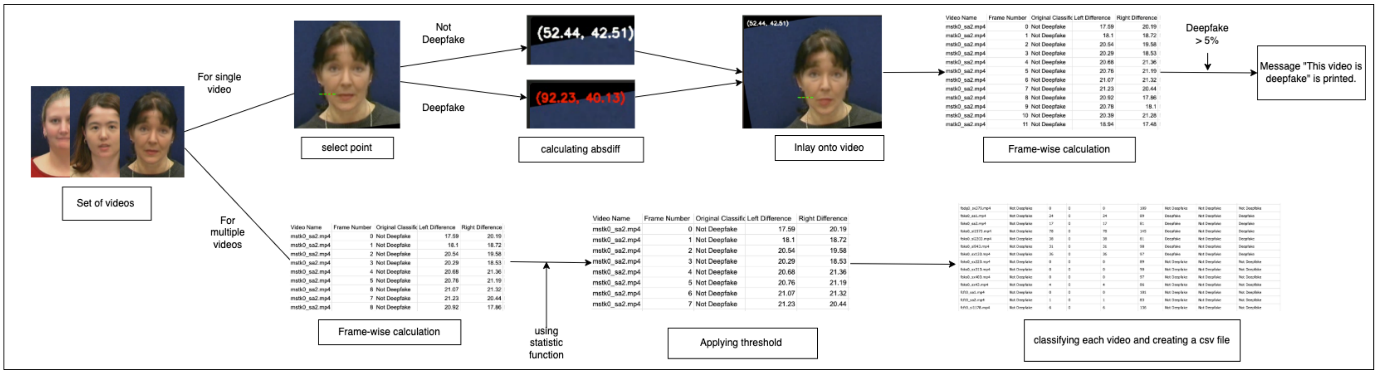


Figure 2: Proposed method 2: flowchart

2) Align Face: Align the face using Dlib points 36 and 45, as described in Method 1.
3) Define Regions: Use Dlib points 3 and 11 for cheek borders. Offset coordinates by ±5 pixels to define region 1 and by ±30 pixels to define region 2.
4) Calculate Redness: Compute the average redness (average values red channel values within the region) for both regions.
5) Calculate Difference: Compute abs_diff as the absolute difference between the two averages. Repeat for both sides and store as left_difference and right_difference.
6) Display Results: If either left_difference or right_difference is greater than preset threshold display the results in the top left corner with color red else color is white.
7) Store Results: Save results for each frame (left_difference and right_difference) in an array.
8) Classification: Label frames as "Deepfake" if the left_difference or right_difference exceeds a preset threshold, otherwise "Not Deepfake."
9) print classification: if 5% number of frames are labeled "Deepfake" then the video is concluded as Deepfake and printed.

This saves a video which contain values of left_difference and right_difference at the top left corner which is colored red or white for ease of identification.
While forensic methods emphasize interpretability, deep learning approaches offer superior classification accuracy.

F. Deep learning Method

To show the same using Deep learning models a Convolutional Neural Network (CNN) architecture optimized with K-Fold Cross-Validation (K-Fold CV). This was done to both the specific areas of the face and the full face.

1) Dataset Preparation: The original dataset from which this was derived vidTIMIT (developed by C. Sanderson et. al.) [25] comprises of video and audio recordings of 43 individual reciting predetermined sentences. For the creation of DeepfakeTIMIT 16 similar looking individual pairs were taken and their faces swapped. Two quality levels were developed lower quality with 64x64 size and higher quality with 128x128 size. There are 10 videos per person in the original dataset with a corresponding 320 videos that totals to 620 videos in the entire dataset. This comprised the fake part of the dataset. This was available for download from the website. For real videos samples the videos from the original dataset vidTIMIT were taken. They are available and can also be downloaded from the website. However, they are stored as frames and audio and have to be reconstructed before use. There are 43 people with 10 recordings each, so the total media is 430 real videos.

Before use in the deep learning method this was altered as following. The new dataset consists of three folders for each of the regions: cheek, eyebrow and full frame, each of which is further divided into train, test and validation (val). Within the dataset there were 10 videos per individual in the deepfake and real folders. To ensure that each individual was not only present in training and validation but also testing, one video per person was kept aside for testing in both deepfake and real categories(They were combined without labels).The K-fold cross validation will split the remaining dataset as needed for training and validation. Image augmentation was applied to the training set using ImageDataGenerator function from TensorFlow. This is done to reduce the amount of overfitting and generalizing the dataset. For validation and testing only rescaling is applied to normalize pixel values between 0 and 1. This is then further split using K-Fold cross-validation to evaluate the model without manually splitting the dataset.

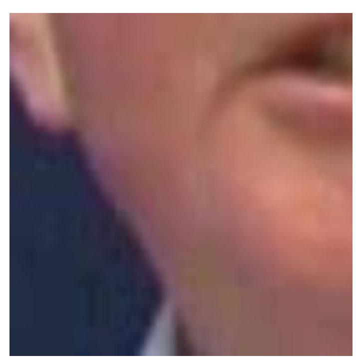


Figure 3: Cheek left

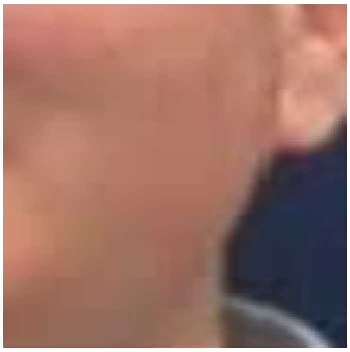


Figure 4: cheek right

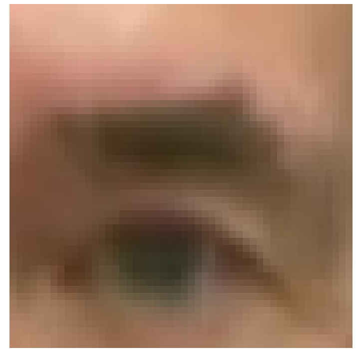


Figure 7: right eyebrow

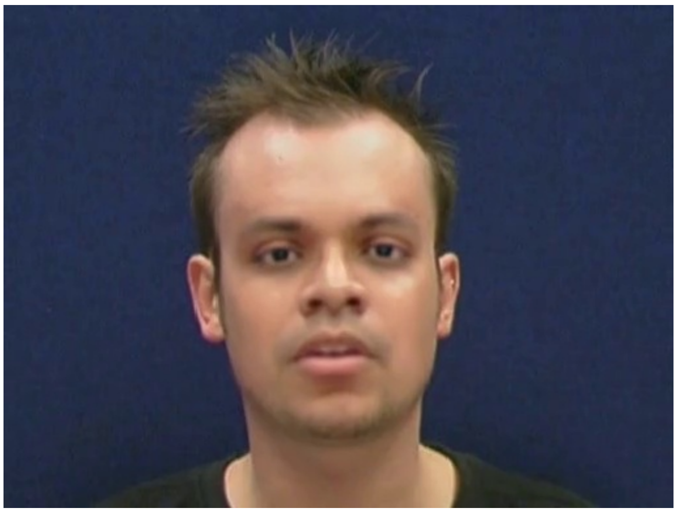


Figure 5: Full frame

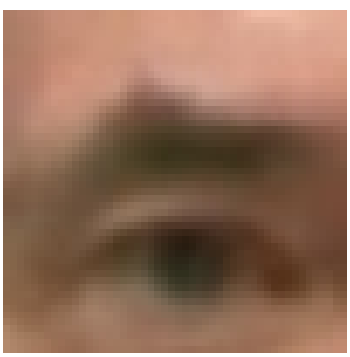


Figure 6: left eyebrow

2) Model architecture: The model architecture consists of the following layers:

- Convolutional Layers: Three convolutional blocks with 32, 64, and 128 filters, each followed by Batch Normalization and MaxPooling to reduce dimensionality while retaining critical features.
- Dropout Layers: Dropout was applied after each convolutional block (20%) and the fully connected layer (30%) to prevent overfitting.
- Fully Connected Layers: A dense layer with 512 neurons (ReLU activation) followed by a final output layer with a sigmoid activation function to classify images as real or fake.
- Regularization: L2 regularization was employed to further mitigate overfitting.

Layer (type)	**Output Shape**	**Param #**
conv2d (Conv2D)	(None, 222, 222, 32)	896
batch_normalization (BatchNorm)	(None, 222, 222, 32)	128
max_pooling2d (MaxPooling2D)	(None, 111, 111, 32)	0
dropout (Dropout)	(None, 111, 111, 32)	0
conv2d_1 (Conv2D)	(None, 109, 109, 64)	18,496
batch_normalization_1 (BatchNorm)	(None, 109, 109, 64)	256
max_pooling2d_1 (MaxPooling2D)	(None, 54, 54, 64)	0
dropout_1 (Dropout)	(None, 54, 54, 64)	0
conv2d_2 (Conv2D)	(None, 52, 52, 128)	73,856
batch_normalization_2 (BatchNorm)	(None, 52, 52, 128)	512
max_pooling2d_2 (MaxPooling2D)	(None, 26, 26, 128)	0
dropout_2 (Dropout)	(None, 26, 26, 128)	0
flatten (Flatten)	(None, 86528)	0
dense (Dense)	(None, 512)	44,302,848
dropout_3 (Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 1)	513

Table I: Layer details of the CNN model

The model was compiled using the Adam optimizer with a learning rate of 0.001 and evaluated with the binary cross-entropy loss function. Performance metrics included accuracy and the Area Under the Curve (AUC).

To prevent manually splitting or having and ensure that the model was robust K-Fold cross validation is used. For each fold (in this case 3) the dataset was divided into training and validation. The model is trained on this training set and validation is done on this validation set. This is then used to tune hyperparameters and monitor overfitting.

3) Training: Training was conducted over 50 epoch with a batch size of 32. Early stopping model checkpoint and learning rate scheduler was used to ensure that the model was trained, and the best model was saved. This is to also reduce the amount of time required if the model was not improving as well as adjusting the learning rate in case of a plateau.

Training was performed for each facial region and saved individually so that they could be evaluated separately and can be used for comparison basis for within Deep Learning model as well as forensic methods mentioned above

After training each fold, the model's performance was evaluated on the independent test set. Metrics such as test loss, accuracy, and AUC were recorded for each fold and region. The best-performing model for each region was saved for further analysis.

4) Parsing Analysis: For ease of understanding the output log was parsed to retrieve information and structure then to be easier to understand. The logs contained detailed information about the training process for each fold and facial region, including validation loss (val_loss) and validation accuracy (val_accuracy) recorded after every epoch. To reduce the time required to go through the log first the log was checked for region using the phrase "Training fold () for region ()". This separated out region as well as fold such that it can be sorted per fold per region. Each epoch contains val_loss and val_accuracy and at the end of each fold there is Test_loss, Test_accuracy and Test_auc. These are stored in individual arrays.
Then best validation loss and best validation accuracy are calculated as well as averages per fold per region are calculated. This is then printed as fold and region along with Fold, region and epoch stored into a CSV file. The test results are stored in a separate CSV file.
The visualization was generated using matplotlib, where the validation loss was plotted against the number of epochs for each combination of fold and region. This allowed for a quick comparison of performance across models trained on different facial regions.

VII. Results

A. Forensic methods

1) Single video: For a single video, the output is a video file which contains frame-wise intensity graph, derivative graph and both the difference. This can be then used to further analyze as required. The absolute difference is placed in the top left corner of the frame and changes color based on whether the threshold is crossed or not. It's white if both differences are under the threshold but if either is over the threshold then the color is turned to red. For the intensity and derivative graphs, those are placed at the bottom left side of the frame. The frames for showing Deepfake and Not Deepfake are showing in following images 8 and 9.

2) Multiple video: A total of 1070 videos were processed using this method. For each of those videos all three methods were applied. To ensure that the threshold was applied correctly a feedback loop was made which ran through all the videos using different sets of thresholds.

Figure 8: Frame showing deepfake

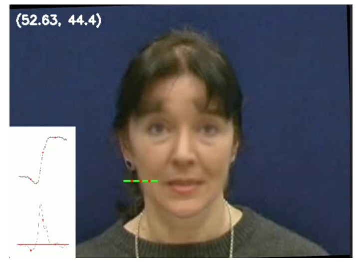


Figure 9: Frame showing Not Deepfake

These thresholds were for peaks from 3 to 7 i.e. if there are there are more peaks present than the set threshold and for abs_diff the range was set from 22 to 49 i.e. if the difference of redness was higher than set threshold. For video level classification, if the classification for frame-wise classification was more than 5 percent then the video would be classified as deepfake.

Both the methods redness and peaks were applied at the same time to avoid running the same set twice. As the result is generated in the feedback loop the third method, the third method (combined classification) is done at the end. The output present as the following:

Method of calculation is mentioned followed by the accuracy precision and f1 score. This is followed by a further breakdown of Deepfake vs Not Deepfake showing precision, recall, and F1-score for both categories. This highlights the performance of the method.

For each method the following thresholds generated the best results to simplify the overall understanding target was set to have the highest Not Deepfake recall.

Method 1: Number of peaks
For calculation of peaks the best threshold was 6 peaks or higher. The highest deepfake accuracy was at 0.72 which means that it can classify deepfakes as predicted for 72% which is estimation of deepfakes with recall as 46% means that it will its missing a significant portion of deepfakes. For non-deepfake videos, the model achieves recall (73%) but at the cost of a lower precision (48%), meaning it misclassified some deepfakes as non-deepfakes.

Discard the absolute diff value as it is not applied in this case

Method 2: Absolute Difference
For calculation of absolute difference, the best overall result was at abs_diff at 35. The model shows precision for deepfake

Class	Precision	Recall	F1-Score	Support
Deepfake	0.72	0.46	0.56	640
Not Deepfake	0.48	0.73	0.58	430
Accuracy			0.57	1070
Macro Avg	0.60	0.60	0.57	1070
Weighted Avg	0.62	0.57	0.57	1070

Table II: Metrics for Total Classification (Video-Level)

detection at (68%), but its recall is low at 36%, meaning it misses a substantial portion of deepfake videos with redness.

For non-deepfake videos, the model has recall (75%), but the precision is relatively low (44%), meaning it misclassified some non-deepfakes as deepfakes. This indicated the least imbalance between sets when looking at the macro averages. The Not Deepfake recall did keep increasing after this measurement but the model accuracy declines for below 0.50.

Class	Precision	Recall	F1-Score	Support
Deepfake	0.68	0.36	0.47	640
Not Deepfake	0.44	0.75	0.56	430
Accuracy			0.52	1070
Macro Avg	0.56	0.56	0.51	1070
Weighted Avg	0.59	0.52	0.50	1070

Table III: Metrics for Redness Classification (Video-Level)

Discard the number peaks value as it is not applied in this case

Method 3: Combined method

For the combined method, both the above methods were applied to all videos and then a 'OR' loop was run on the results and saved as combined classification.

Class	Precision	Recall	F1-Score	Support
Deepfake	0.71	0.48	0.57	640
Not Deepfake	0.48	0.71	0.57	430
Accuracy			057	1070
Macro Avg	0.59	0.59	0.57	1070
Weighted Avg	0.62	0.57	0.57	1070

Table IV: Metrics for Combines Classification (Video-Level)

The Combined Classification model performs deepfake detection at the video level, with an accuracy of 57% and an F1-score of 57%.

The model shows precision for deepfakes at 71% but lower recall 48%, meaning it misses a significant portion of deepfake videos.

For non-deepfake videos, the model achieves recall of 71% but relatively low precision at 48%, meaning significant portion of non-deepfakes are misclassified as deepfakes.

The macro and weighted averages indicate balanced performance, but the model's recall for deepfakes could be significantly improved.

As anticipated this model had the best performance of all three methods as it can eliminate the issues with the above methods.

B. Deep learning methods

For Deep learning model, a Convolutional Neural Network (CNN) architecture optimized with K-Fold Cross-Validation (K-Fold CV) was used.

- **Best Performance:**
  - *Region Eyebrow* consistently has the lowest best loss (0.1357–0.1502) and highest best accuracy (up to 0.9827).
  - *Region Cheek* in Fold 2 achieves the highest best accuracy of 0.9919.
- **Average Performance:**
  - *Region Eyebrow* maintains better average loss and accuracy compared to the other regions, demonstrating stability.
  - *Region Full Frame* shows large average losses (7.45–11.22), likely due to more complex input features.
- **Variability Across Folds:**
  - *Cheek region* shows significant fold-to-fold variability in average metrics. Fold 2 demonstrates significantly better results than Fold 1 and Fold 3.
- **Full Frame Region:**
  - Performs well in terms of best accuracy but suffers from much higher average losses. Likely indicating difficulty in optimization due to complex input data(more features to analyze).

Key Observations: The Eyebrow region stands out as the most reliable and consistently high-performing region across all folds. The Full Frame region, despite its high potential accuracy, struggles with optimization (high average loss). Cheek region shows fold-dependent performance, suggesting sensitivity to dataset variations.

For the testing a set of video which was not present within either the test or the validation split was used the table V shows the breakdown of test loss, test accuracy and test AUC metric.

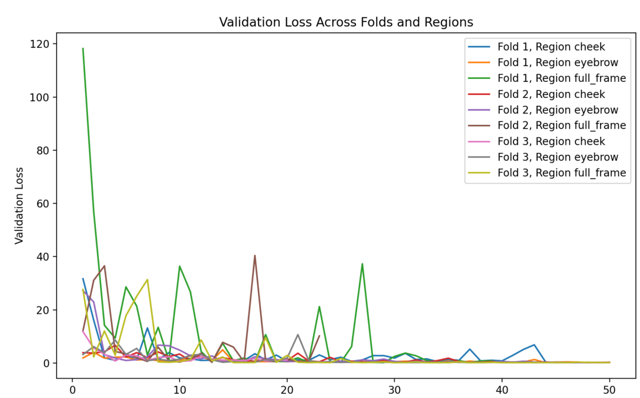


Figure 10: learning curve

The figure 10 shows the validation loss over epoch for multiple folds. Validation sharply declines at the beginning indicating that there is rapid learning in the early stages. Loss stabilizes (or fluctuates minimally for full_frame) after certain number of epochs. To stop training once the loss has has reached minimal, function such as Early stopper and model checkpoint are used after 10 epoch when the loss is stable. This causes model to not complete all 50 epochs for all folds. The initial high validation loss for full frame is mostly due to more features present in the full frame verses eyebrow or cheek region. Given this cheek and eyebrows converges fast and are stable this could be due to less complexity. The spikes indicate that the model struggles to generalize either due to overfitting or the split of validation set. Within the time constraints it wasn't possible to look further into resolving these spikes in validation loss especially for full frame.

Fold	Region	Test Loss	Test Accuracy	Test AUC
1	Cheek	0.6699	0.8636	0.9224
1	Eyebrow	0.2125	0.9658	0.9911
1	Full Frame	0.3162	0.9535	0.9877
2	Cheek	0.2439	0.9602	0.9893
2	Eyebrow	0.2613	0.9547	0.9854
2	Full Frame	0.2866	0.9532	0.9962
3	Cheek	0.6671	0.7994	0.8687
3	Eyebrow	0.1983	0.9718	0.9911
3	Full Frame	0.1739	0.9763	0.9956

Table V: Performance Metrics Across Folds and Regions

The eyebrow region consistently demonstrated the lowest test loss and highest test accuracy and AUC across all folds. This indicates that features from the eyebrow region are easiest to identify and use for classification. Full frame classification performed consistently across all folds, with high test accuracy (above 0.953) and excellent AUC values. The cheek area showed variability in performance, with Fold 1 and Fold 3 reporting relatively high test loss (0.6699 and 0.6671) and lower test accuracy (0.8636 and 0.7994).

Region	Test Loss	Test Accuracy	Test AUC
Cheek	0.5270 ± 0.2431	0.8744 ± 0.0833	0.9268 ± 0.0611
Eyebrow	0.2240 ± 0.0313	0.9641 ± 0.0075	0.9892 ± 0.0029
Full Frame	0.2589 ± 0.0667	0.9609 ± 0.0103	0.9932 ± 0.0040

Table VI: Condensed Performance Metrics Across Regions

VIII. Discussion

The primary objective of this study was to visualize certain artifacts to highlight if a video or image is a deepfake or not.

Three methods were proposed: Method 1 or number of peaks evaluated if there was smoothness in the transition from the face to the background. It was plotted as two graphs intensity graph and derivative graph. It was easy enough to understand how it works as it explains that more than a certain number of peaks indicates deepfake. Method 2 or Absolute difference also evaluates similar region to see the amount of redness at two points of the cheek. Method 3 or combined method just combined the previous methods.

Some of this to consider when using these methods is that it was developed on one dataset, these artifacts may or may not be present in other datasets. This is not only an issue with this method but also many deep learning methods. This causes new methods to be developed for both forensic and deep learning methods of detection. However, once an artifact is identified it's easy enough to use preexisting libraries in python to highlight these and can be compounded into a simple program similarly to any deep learning model generation. Another thing to consider is that the participants in this study were most light skinned against a darker background, so the methods were not adapted to other alternatives such as lighter background with light skin or darker background with dark skin. As there were specific requirements for the video analysis such as forward facing and even lighting it was harder to find other datasets to use for this case. Specifically in the method 2 as it looks as the underlying redness present in the cheek are which could be affected by skin tone or presence of additional redness in that area such as sunburn.

As for the results method one had lower overall accuracy meaning that the method was moderately performing at 57% with non-deepfake recall around 73% but was consistent through all iterations. Method 2 has significantly better non-deepfake recall as high as 88% but with poor overall accuracy sometimes below even 50%. For the purposes of this paper the aim was to have a non-deepfake recall of at least 85% so that there is a high classification for labelling. To overcome this Method 3 was used so that both the stable accuracy of method 1 and high non-deepfake recall of method 2 can be combined. Method 3 it showed stable accuracy at 57% with a higher recall of method 2 with non-deepfake recall being as high as 80%. This also had the most balanced performance of the three methods.

However, these are not enough to be used for in all cases such as proving authenticity in a criminal investigation or in case of proving beyond a reasonable doubt in a criminal proceeding. Some potential applications can be used for detecting deepfakes in children's programs where it's essential to not expose them to harmful content even if blocking some non-deepfake videos occurs. Overall, its use as a detection method is not as sufficient as needed but it's possible to use it for the purpose of visualization as well as in use to show how other artifacts can be detected.

For the deep learning model, the same dataset was used to create a model previously without any additional filters or changes. The model was given frames like above along with only a 50 px by 50 px area of cheek and eyebrows. These were done as a comparison to the forensic method to test the accuracy that was available as deep leaning method and to set a standard for detection. To ensure that the model itself was performing well a study by Haliem et al. [**?**], was used as a reference. They achieved high accuracy in detecting deepfakes on the DeepfakeTIMIT dataset, with 99% accuracy on low-quality videos and 91.88% on high-quality videos. They also feed videos as images to several deep learning models. For the deep learning model described above the model was able to reach best accuracy for full frame at 0.9935 but had higher losses (7.45-11.22) this is likely due to more information present within each frame. For eyebrows the accuracy was the most consistent, being between 0.96 and 0.98 across all three folds as well as the lowest loss across all three folds. This presents an interesting opportunity as a second area that can be used in the forensic process initially considered as the second potential artifact to highlight. The cheek area has high accuracy in fold 2 at 0.9919 which could mean that the model was able to classify extremely well with that given split as the other two folds have significantly lower accuracy at 0.8837 and 0.8214. Potentially there can be issues with how the frames are split which could be causing a larger portion of non-deepfake video to be used in the training process or in the second fold split such that its overfitting the data in that fold. For the testing both eyebrows and full frame had high test accuracy whereas the cheek area had to high test loss (0.6699 and 0.6671) and lower test accuracy (0.8636 and 0.7994). This could be indicative of two things, first being that there is good split for eyebrow and full frame and an issue with the distribution of cheek region is not split evenly. Secondly it could be indicative that the presence of the artifacts is so prevalent that the model confuses them as normal. There is also a potential chance that cheek area may not have as distinct features for extraction as expected from visual inspection.

Comparing the forensic method with the deep learning method, in terms of visualization of certain artifacts it's easier to highlight certain areas using forensic methods rather than using tools such as Grad-cam which gives an overall area of manipulation. In terms of accuracy the deep learning outperformed the forensic method by a significant amount. However, only taking into account the cheek areas in both cases. The test accuracy and the overall accuracy of the forensic method is not as significant. This could be an indication of the artifact identification issues within this specific region.

IX. Future Work

A. Forensic Method

- Highlight other areas of the face, such as eyebrows, to help improve accuracy.
- Perform a frame-by-frame analysis to ensure the method classifies correctly rather than relying on other factors.
- Modify the methodology to include various backgrounds and skin tones.
- Expand the dataset to include more diverse backgrounds and skin tones to identify potential issues.
- Incorporate multi-frame classification or analyze the continuous presence of artifacts rather than relying on a single frame.

B. Deep Learning Methods

- Test whether the model functions well when the test set is shuffled.
- Use a more combined area, rather than analyzing individual regions like the cheek and eyebrows separately.
- Address data splitting issues to reduce overfitting or imbalance in specific folds.
- Explore hybrid methods that integrate forensic artifact visualization as features in deep learning models.

Priority should be given to the deep learning model as there more accuracy present. The main priority is shuffling for test set to check model function and creating a potential hybrid method. Looking for a dataset that expands the initial dataset can also be done so that there can be less bias present within both methods.

X. Conclusion

The objective of this study was to explore forensic and deep learning methods for detecting deepfake artifacts, with an emphasis on visualization and classification. The forensic methods provided a straightforward approach to identifying specific artifacts, such as redness and smoothness in transitions, through frame-by-frame analysis. Among these, the combined method (Method 3) performed the best, achieving stable accuracy and higher non-deepfake recall compared to the individual methods.

The deep learning models, on the other hand, achieved significantly higher accuracy and robustness, particularly in the eyebrow and full-frame regions. The eyebrow region consistently showed high performance across folds, with accuracies ranging from 96% to 98%, and maintained low test loss. In contrast, the cheek region exhibited variability in accuracy, highlighting potential issues with data splitting and preprocessing. The higher losses for the full-frame region suggest challenges related to the complexity of features in this area.

While the forensic methods are less accurate, they provide greater interpretability, enabling visualization of specific artifacts that are harder to identify using deep learning techniques. This suggests the potential for integrating forensic methods into deep learning pipelines to enhance both interpretability and detection accuracy.

In conclusion, this study demonstrates that combining forensic insights with deep learning methods can create a more effective framework for deepfake detection, balancing artifact-specific visualization with high classification accuracy.

References

[1] M. Bouter et al., "ProtoExplorer: Interpretable Forensic Analysis," 2023.

[2] S. Gautam et al., "Prototypical Relevance Propagation," in Proceedings of the 2021 AI Conference, 2021.

[3] H. Salloum et al., "Hybrid Deepfake Detection: Forensic and Machine Learning Integration," Journal of Digital Forensics, vol. 15, no. 3, pp. 45-58, 2022.

[4] L. Verdoliva et al., "Explainable AI for Deepfake Detection," IEEE Transactions on Information Forensics and Security, vol. 17, pp. 1124-1137, 2022.

[5] R. R. Selvaraju et al., "Grad-CAM: Visual Explanations from Deep Networks," in IEEE Conference on Computer Vision and Pattern Recognition, 2019.

[6] P. Chaudhary et al., "Multi-Level Deep Learning for Deepfake Detection," in International Conference on Multimedia Analysis, 2022.

[7] H. Dang et al., "Multi-modal Deepfake Detection via Audio-Visual Streams," IEEE Multimedia Magazine, vol. 29, no. 2, pp. 24-35, 2022.

[8] V. Matern et al., "Detecting Biological Artifacts in Deepfakes," Forensic Science International, vol. 314, pp. 109-120, 2021.

[9] L. Guarnera et al., "Frequency Domain Analysis for Deepfake Detection," Pattern Recognition Letters, vol. 145, pp. 84-91, 2023.

[10] P. Korshunov et al., "Comparative Study of Deepfake Detection Methods," Journal of Machine Learning Research, vol. 23, pp. 1-18, 2022.

[11] T. Goodfellow et al., "Deep Learning for Media Manipulation," IEEE Transactions on Neural Networks and Learning Systems, vol. 29, no. 2, pp. 24-35, 2022.

[12] X. Zhang et al., "Detection of Fake Media Using Advanced AI Techniques," Pattern Recognition Letters, vol. 125, pp. 84-91, 2021.

[13] D. M. Brown et al., ”Understanding the Black-Box Nature of Neural Networks,” Journal of AI Research, vol. 45, pp. 456-469, 2020.
[14] J. Lee et al., ”Challenges in Explainable AI for High-Stakes Scenarios,” IEEE Intelligent Systems, vol. 34, no. 5, pp. 25-30, 2021.
[15] R. James et al., ”Artifact-Based Forensic Techniques for Media Integrity,” Forensic Science Review, vol. 12, no. 1, pp. 33-47, 2023.
[16] T. Chesney, ”Deepfakes: A game changer in cyber security,” Computers & Security, vol. 97, pp. 101941, 2020.
[17] I. Goodfellow et al., ”Generative Adversarial Networks,” in Advances in Neural Information Processing Systems (NIPS), 2014, pp. 2672-2680.
[18] A. K. Jain and B. Klare, ”Face recognition: Beyond the visible spectrum,” IEEE Transactions on Information Forensics and Security, vol. 14, no. 12, pp. 3325–3343, Dec. 2019.
[19] H. Dang, F. Liu, J. Stehouwer, X. Liu, and A. Jain, ”On the Detection of Digital Face Manipulation,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 5781-5790.
[20] S. Rossler et al., ”FaceForensics++: Learning to detect manipulated facial images,” in Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 1-11.
[21] Z. Wang and S. Wang, ”Deep learning-based human face manipulation and its detection: A survey,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 43, no. 11, pp. 3838-3858, Nov. 2021.
[22] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb, ”Learning from Simulated and Unsupervised Images through Adversarial Training,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 2107-2116.
[23] University of Washington, ”Synthesizing Obama: Learning lip sync from audio,” [Online]. Available: https://grail.cs.washington.edu/projects/AudioToObama/. Accessed: Jan. 1, 2025.
[24] C. Sanderson and B.C. Lovell Multi-Region Probabilistic Histograms for Robust and Scalable Identity Inference. Lecture Notes in Computer Science (LNCS), Vol. 5558, pp. 199-208, 2009.
[25] C. Sanderson. Biometric Person Recognition: Face, Speech and Fusion. VDM-Verlag, 2008. ISBN 978-3-639-02769-3.
[26] D. E. King, “Dlib: A toolkit for machine learning and computer vision,” 2017, arXiv:1706.08016. [Online]. Available: https://arxiv.org/abs/1706.08016. [Accessed: Jan. 1, 2025].
[27] A. Haliem, M. Abdelhamid, and A. M. Badawy, ”Deepfake detection using fusion of deep CNN models,” arXiv preprint arXiv:2102.05950, Feb. 2021. [Online]. Available: https://arxiv.org/abs/2102.05950.