POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Leveraging Multimodal Large Language Models for Explainable Deepfake Detection

Tesi di Laurea Magistrale in
Computer Science & Engineering - Ingegneria Informatica

Author: **Lorenzo Morelli**

Student ID: 932332
Advisor: Prof. Mark James Carman
Co-advisors: Andrea Sassella, Paolo Bestagini
Academic Year: 2023-24

Abstract

Deepfake synthesis is rapidly advancing, generating hyper-realistic synthetic media that not only challenge digital security but also raise significant ethical concerns. While conventional detection methods based on handcrafted features and standard CNNs often struggle with interpretability and generalization, our research explores a multimodal strategy that integrates both visual and textual cues to enhance detection performance and transparency.

Our approach firstly involves the employ of several state-of-the-art vision-language models under zero-shot and few-shot settings, to assess their ability to identify and explain image manipulations. Following these experiments, we fine-tuned a leading multimodal model, refining its capacity to detect tampering and generate human-understandable explanations.

To comprehensively gauge the system's performance, we adopted a multi-faceted and robust evaluation framework that combines traditional NLP and embedding-based metrics with qualitative assessments from large language models acting as expert judges. Overall, our results show that focused fine-tuning not only enhances the precision of deepfake detection, but also improves the interpretability of the generated explanations. Additionally, our findings support the development of a generalized framework for assessing natural language output.

Keywords: Deepfake Detection, Explainable AI, Multimodal Models, Vision-Language Models, Image Forensics, Natural Language Processing

Abstract in lingua italiana

Negli ultimi anni, la tecnologia deepfake si è evoluta rapidamente, producendo contenuti sintetici estremamente realistici che pongono serie minacce alla sicurezza digitale e sollevano importanti questioni etiche. I metodi tradizionali di rilevamento, basati principalmente su caratteristiche artigianali o reti convoluzionali standard, presentano spesso limiti nella generalizzazione e nella spiegabilità. Questa tesi esplora una strategia multimodale innovativa che combina informazioni visive e testuali per migliorare l'efficacia e la trasparenza del rilevamento.

Inizialmente abbiamo valutato diversi modelli multimodali allo stato dell'arte attraverso esperimenti zero-shot e few-shot, analizzandone la capacità di individuare e descrivere manipolazioni nelle immagini. Successivamente, abbiamo effettuato un fine-tuning su uno dei modelli più promettenti, adattandolo specificamente al compito di rilevamento dei deepfake e generazione di spiegazioni dettagliate.

Per analizzare rigorosamente i risultati ottenuti, abbiamo definito un framework di valutazione robusto e generalizzabile che integra metriche di NLP tradizionali, tecniche basate su embedding e giudizi qualitativi generati tramite modelli linguistici avanzati. Complessivamente, i risultati dimostrano come il fine-tuning mirato migliori significativamente sia la precisione nel rilevamento dei deepfake che la chiarezza e l'interpretabilità delle spiegazioni fornite, aprendo inoltre la strada a una metodologia generalizzata per valutare le predizioni in linguaggio naturale.

Parole chiave: Rilevamento dei Deepfake, Spiegabilità dell'IA, Intelligenza Artificiale Multimodale, Modelli Visione-Linguaggio, Forensica delle Immagini, Elaborazione del Linguaggio Naturale.

Contents

1 | Introduction

Deepfake technology, deep learning-based product of hyper-realistic media, has progressed at a very rapid pace in recent years, challenging traditional definitions of authenticity for digital media. With the advent of generative adversarial networks (GANs), autoencoders, and more recently transformer-based architectures, the ability to manipulate visual and auditory information has reached unprecedented levels today. These advances have posed unimaginable threats to cybersecurity, political trust, and personal privacy, creating a dire need for robust detection methods.

1.1. Background and Motivation of the research

1.1.1. Evolution of Deepfakes

Deepfakes emerged as a side effect of innovations in deep learning, as algorithms that were originally designed for the transformation and generation of images were exploited for the development of artificially created media that can remarkably impersonate authentic. Early implementations often included artifacts visible to the naked eye; however, advances in model architectures (e.g., StyleGAN, DALL-E, Stable Diffusion) have significantly enhanced the realism of these creations. This enhancement not only highlights the technological prowess underlying deepfakes but also increases their potential for misuse, including identity theft, criminal content generation, and manipulation of public discourse.

1.1.2. The Imperative for Explainability

Despite the impressive performance of state-of-the-art deepfake detectors, many rely on black-box models that offer little to no transparency into their decision-making processes. In sensitive applications such as forensic analysis, legal or medical proceedings, and public communication, it is not enough to simply flag a manipulated image: stakeholders and scientists require interpretable, human-understandable explanations to validate detection results and to build trust in the technology. Explainability in artificial intelligence has

therefore become a critical research area, driving efforts to integrate transparent mechanisms into complex models.

1.1.3. Multimodal Approaches and Large Language Models

Traditional deepfake detectors have generally relied on CNN-extracted visual features and manually crafted forensic approaches. However, these struggle to generalize across different types of manipulation techniques. Multimodal approaches, however, which combine visual and text inputs, promise to not only capture slight inconsistencies, but also to generate natural language explanations of the shape of the manipulation. New advances in Vision-Language Models (VLMs) and Large Language Models (LLMs) provide an integrated setting where the two modalities are fused. Such models can understand advanced visual data and translate reasoning into human language, hence providing improved detection accuracy and explainability.

1.2. Thesis Structure

This thesis is structured as follows:

- **Chapter 2** provides a detailed background on deepfakes, covering their evolution, generation techniques, and the ethical implications associated with their proliferation. Along with it, we will also discuss the challenges of achieving explainability in artificial intelligence.
- **Chapter 3** reviews the related work in deep learning, multimodal models, and existing deepfake detection methods, establishing the context for the proposed approach.
- **Chapter 5** describes the creation and annotation of the custom dataset used in this research, with a focus on splicing-based manipulations.
- **Chapter 6** outlines the methodological framework, including the experimental setup, zero-shot and few-shot experiments, and the fine-tuning process using LLaVA OneVision. It also presents and analyzes the experimental results, including performance metrics and qualitative evaluations of the generated explanations.
- **Chapter 7** details the comprehensive evaluation framework used to assess the performance of the model and the quality of the explanation, integrating conventional NLP assessments with embedding-based and LLM-based approaches.
- **Chapter 8** summarizes the findings, discusses the limitations of current work, and

outlines future research directions.

2 | Background

Deepfakes and synthetic media in general have proven to be one of the most exciting breakthroughs in artificial intelligence in recent times. In this chapter, we will see how deepfakes have evolved and then how deepfakes work and can be detected, according to researchers. Next, we will discuss the critical necessity of explainability in AI, to enable the user to comprehend and validate the reasoning behind machine-generated responses.

2.1. Deepfakes and Synthetic Media

Deepfakes have been christened with a name derived from a portmanteau of "fake" and "deep learning". With deep neural networks, humans can generate or manipulate media, such as photos, videos, audio, for example, to make them appear real, when in reality they are not. To grasp both the danger and the use of such technology, it helps to grasp its origin, its processes, and its counter-and regulating strategies.

2.1.1. History of Deepfakes

Long ago, even in the days preceding artificial intelligence, manipulation of the media existed in pop culture. Old methods such as picture manipulation (with programs such as Adobe Photoshop) and computer-generated imagery (CGI) took a lot of time and, most importantly, expertise, and yet displayed everyone with a demonstration of how easy it could be to manipulate public perception through altering one's view of reality. That changed in 2014, when Ian Goodfellow developed Generative Adversarial Networks (GANs) [20]. In a GAN, one network (the generator) creates artificial data (e.g., artificial pictures), and a second, the discriminator, tries to differentiate between real and not real data. Over a series of training rounds, the generator learns to produce increasingly real output.

Although they were initially developed in labs, in 2017, GANs captured everyone's imagination on Reddit, with its community trading faces of famous personalities in video clips, sometimes in inappropriate environments. Software, including FakeApp and DeepFace-

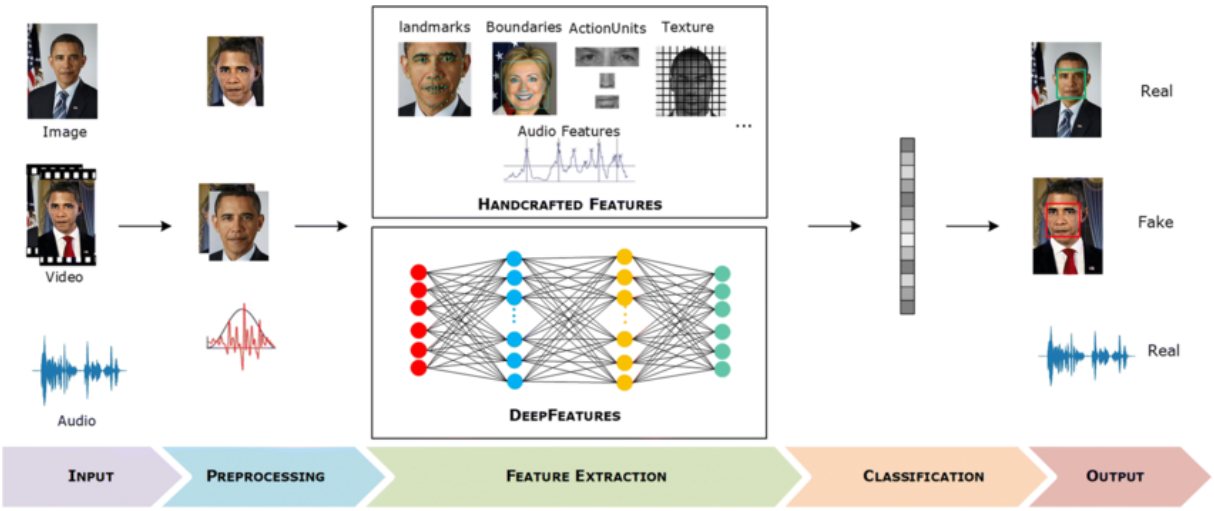


Figure 2.1: A general pipeline for deepfake detection, showing key steps from input preprocessing to classification. Adapted from [44].

Lab [55], quickly lowered the technical barrier, and, in fact, any ordinary individual could generate such manipulated clips without AI expertise. According to a report in Britannica [52], such early samples fueled concerns about disinformation and non-consensual content. Over time, frameworks including TensorFlow and PyTorch, and with advances in hardware, accelerated deepfakes' development. In fact, StyleGAN [28], Stable Diffusion [60], and transformer-based architectures [74] have taken the realism of synthetic media to unprecedented heights.

2.1.2. Generation of Deepfakes

Deepfakes are based most frequently on three categories of deep learning architectures: autoencoders, GANs, and transformer architectures. Autoencoders initially attempted face-swapping, two separate autoencoders, each responsible for encoding and decoding a face. By switching between decoders, coders generated simple face-swaps, but early experiments produced face-swaps with apparent artifacts.

GANs amplified deepfakes' realism: in training a discriminator and a generator in parallel, a generator learns to produce increasingly real output. An example case-in-point is StyleGAN [28], a model that incorporates fine-grain controls of "high level" (for example, face structure) and "low level" (for example, hair, lighting) in its output. With them, deepfake producers could target specific parts, for example, changing a person's age or hair color in a video.

Building on these advancements, transformers, originally a breakthrough in natural lan-

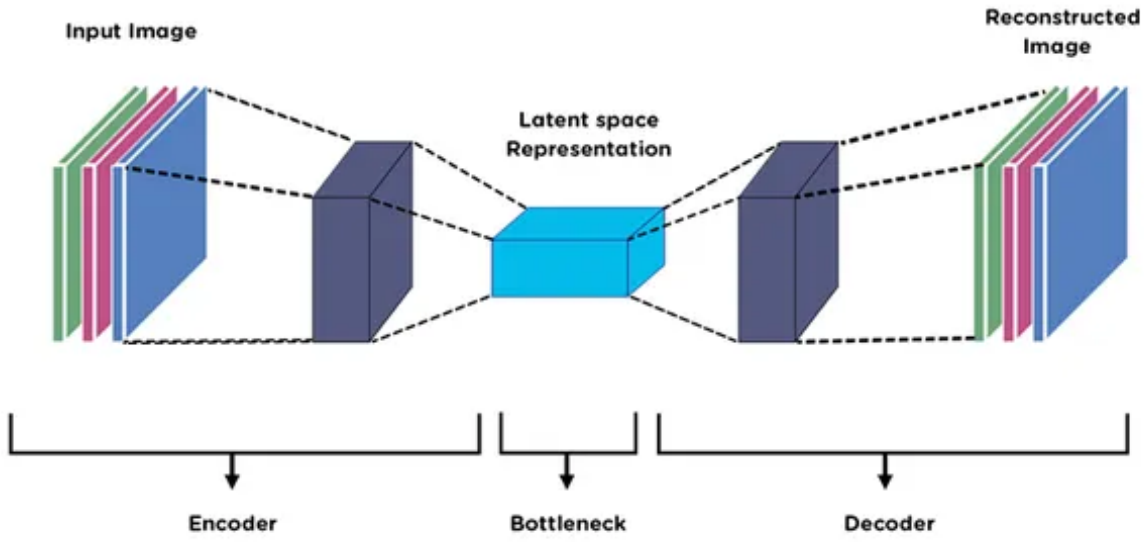


Figure 2.2: A schematic description of an autoencoder, highlighting the encoder, latent space and decoder. Taken from [5]

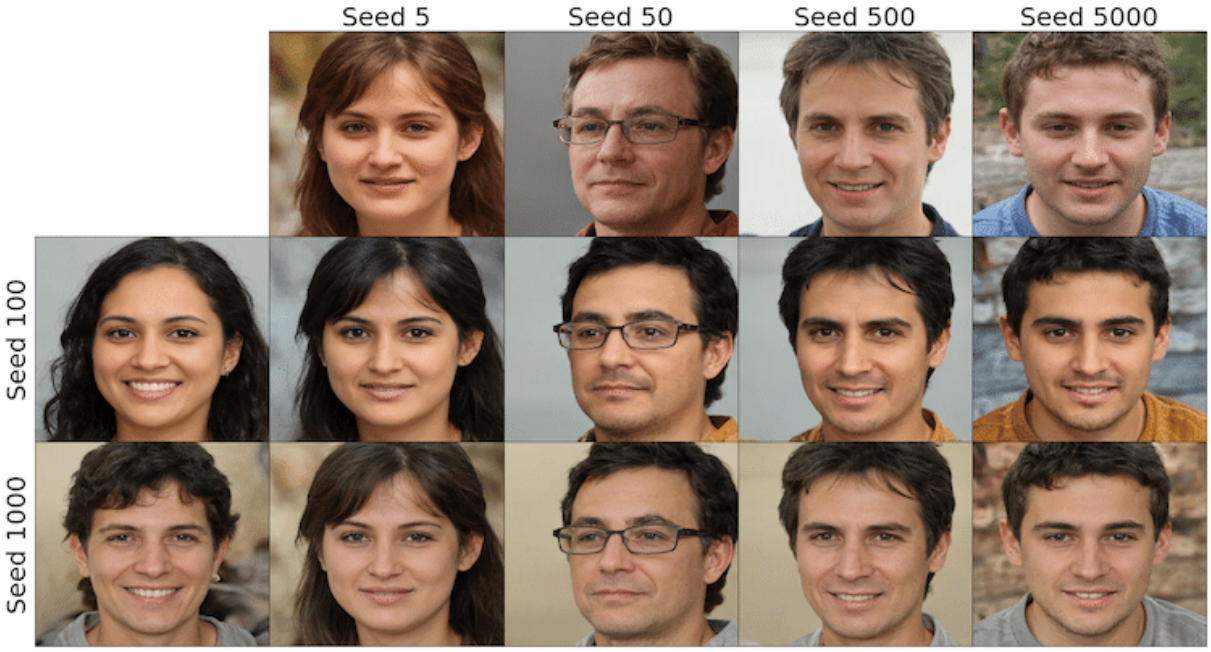


Figure 2.3: StyleGAN2 mixing with a truncation value $\psi = 0.5$. Each column has a different random seed for high-level features, while each row uses a different seed for finer details. Combining these seeds creates blended facial attributes. Adapted from [71].

guage processing, have also made their mark on deepfake technology. Using self-attention mechanisms to handle complex information, transformers enable models such as DALL-E [57], an OpenAI model, to convert text prompts into remarkably detailed images, while Stable Diffusion [60] combines diffusion algorithms with attention layers to generate sharp, high-fidelity, and hyperrealistic photos. Behind the scenes, creating convincing deepfakes is not simply a case of having a powerful AI at its core, but entails processes such as face detection and face alignment in preprocessing, and then polishing touches such as blending edits in post-processing in an attempt to evade detection. Breakthroughs like those detailed in the Facia AI blog [48] reveal that these innovations are not only technically sound, but also bring great capabilities with insightful applications ranging from satires to fraudulent activities, such as identity theft.

After deepfake technology development, academicians and industry leaders have emphasized having effective detection tools, together with stricter legislation, and awareness programs for public awareness. Today, balancing such creation capabilities with abuse and disinformation is a challenge, but we will have a discussion about it in later sections.

2.1.3. Deepfakes Specialized on Humans

Human deepfake technology uses advanced algorithms to replicate faces, voices, and even body language with a high degree of realism. Some of its early achievements include face expression swapping between two individuals in real time through techniques such as Face2Face [70], a starting point for future platforms such as the aforementioned DeepFaceLab and FaceSwap, and increasingly being adopted in Hollywood productions, social networks, and advertisements for use in work such as stand-ins and creating creative content.

Voice synthesis has gone down a similar route. Breakthroughs including WaveNet [73] and Tacotron [76], initially designed to generate speech, soon found a new application in voice cloning. Present-day voice reproduction technology can replicate a speaker's inflections, including tone and pace, with ease. Commercial options such as Respeecher and Play.ht have even democratized voice reproduction technology, with near-immediate voice reproduction at little to no technical expertise.

Another boundary is full-body deepfakes, in numerous instances combining motion capture data to animate computer avatars in motion. Software tools such as Runway ML's Act One simplify such an offering, generating real-life motion for use in animation, gaming, and virtual production. Social-media face filters in use at platforms such as Snapchat and Instagram, seen daily use by many millions, represent public acceptance of real-time

Figure 2.4: AI-generated images of Pope Francis wearing a designer coat—something that never happened in real life. Adapted from [8].

augmented reality, but less sophisticated in its form. Overlays of virtual information onto a face, such as face filters, represent lighter precursors to deepfakes and normalize synthetic media in general.

The rapid development of this technology is best illustrated by viral events like AI-created picture of Pope Francis donning a high-end overcoat (Figure 2.4), which misled many observers [8]. These events highlight the potential of current generative tools such as Midjourney and Stable Diffusion [60], generating output ever less discernible from reality. Midjourney is a sophisticated text-to-image generator that translates user prompts into visually compelling and detailed images, which exploits deep learning techniques to refine both artistic style and composition. Meanwhile, Stable Diffusion uses a latent diffusion process to convert textual descriptions into high-resolution images.

2.1.4. Detection of Deepfakes

Now, with deepfakes becoming increasingly realistic and hard to detect, a competition to develop effective tools for detection is intensifying. In its infancy, a deepfake could

be detected with ease—models consistently failed because of obvious mistakes. Eyes did not blink naturally, smiles appeared unnatural, and lights clashed with the background [37]. Today, such giveaways rarely occur, having been eradicated by rapid advancement in AI-driven content creation. Detecting deepfakes today entails much more sophisticated approaches.

Modern tools most closely depend on deep learning, such as deep neural networks (DNNs) and transformers, respectively. These tools have been trained with gargantuan sets of real and artificial material, enabling them to learn the subtle, nearly invisible cues that reveal a deepfake [16]. For instance, DNNs can pinpoint frame-by-frame mismatches in a video, ones an eye won't even perceive. Transformers, in contrast, can inspect sequences (e.g., a face moving through a sequence of video shots) for abnormalities in motion or timing. And when working with faked speech, analysis of a spectrogram can reveal unnatural harmonies or computer glitches that reveal corrupted audio [53].

One of the most effective techniques for discovering deepfakes is looking for "GAN fingerprints". GANs leave small glitches in their wake when creating counterfeit work [43]. It is a virtual watermark, but one that isn't intentional and less apparent. GAN fingerprints can appear in small intensity discrepancies, for instance, or in discrepancies in a photograph's distribution of colors. By training a detection system to scan for such signatures, researchers have been successful in tracking deepfakes to their origin of creation. However, it is not simple, as not only are deepfakes becoming easier to evade detection, but forgeries are becoming sophisticated, too: with a little manipulation, added noise, and smoothing, fakers can make them untraceable.

And then, naturally, there is multimodal detection, a notch higher in technology that examines both audio and visual information at the same time. Imagine a face swap in a video with a near-perfect convincing face swap. It may deceive your eyes at first, but when audio doesn't perfectly coordinate with the mouth and takes an unnatural voice, it can still be detected [45]. By combining information from a variety of sources of information, such techniques attempt to detect discrepancies that will go under less sophisticated tools for detection. But then comes *generalizability*. Most deepfake detection algorithms work best at distinguishing between specific categories of deepfakes that have been encountered during training. Present them with a new one, and they can sink altogether. To counteract this, researchers have been striving to develop smarter and more flexible algorithms. Techniques such as zero-shot and few-shot learning, in which algorithms learn to detect deepfakes with fewer, even zero, examples, have become increasingly prevalent [80]. Adversarial training, in which algorithms are deliberately trained with sneaky fakes in an effort to make them robust, is also under investigation.

2.1.5. Threats and Ethical Implications

Deepfakes are not just a marvel of technology; they also introduce a set of profound risks that touch on politics, cybersecurity, personal safety, and the broader fabric of society. Fascinating and deeply unsettling, they can manipulate reality with an unprecedented level of precision.

Politics One of the most alarming threats of deepfakes is their potential to destabilize trust in democratic institutions. Deepfakes with political motivations can trigger riots by manipulating speeches, interviews, or actions of any public figure, including politicians. One can imagine how fast a fabricated video of a world leader announcing war or support for a harmful policies would spread over social media to create chaos before fact-checkers get an opportunity to act. Following Chemerys et al. [10], such a possibility makes deepfakes exceptionally potent in spreading misinformation, leading to the erosion of trust in public discourse and institutions. And in a world where "seeing is believing", the risk is even higher as people will always tend to believe what they see over written claims.

Cybersecurity In the field of cybersecurity, deepfakes have already proven their potential for harm. Fraudsters have used voice cloning to impersonate the voice of a chief executive or high-ranking officer to deceive an employee into sanctioning fraudulent wire transfers or revealing sensitive information. According to a notable incident reported by De Rancourt-Raymond et al. [15], some criminals stole millions using AI-generated voices. Such attacks, also referred to sometimes as "vishing" or voice phishing, represent an increasingly important overlap of deepfakes with social engineering techniques, opening up novel vulnerabilities both in personal and corporate worlds.

Cyberbullying The impact of deepfake technology on private lives is already catastrophic. Victims of non-consensual deepfake pornography suffer extreme psychological trauma that inflicts permanent damage on their reputations and characters. Furthermore, as deepfake tools become more widely available, the possibilities for creating malicious content will also swell. This trend puts not only celebrities in danger, but also affects private individuals, thereby adding to instances of bullying, harassment, and character assassination by the weaponization of deepfakes.

An interesting yet troublesome aspect concerning deepfakes is a phenomenon described as the "liar's dividend" [63]. Basically, the concept defines that a person can discredit an actual evidence because they say it is faked, using the presence of deepfakes as a plausible deniability. This would make holding somebody accountable in some high-stake cases, be

it in investigations or public scandals, a little more difficult, since the case would have an incriminating video or recording which will be said to be fabricated.

Deepfakes fall into a gray area from both a legal and ethical standpoint. Current laws often fail to address the nuances of this technology. For example, if a deepfake causes harm, who is responsible: the creator, the distributor, or the platform hosting it? On issues of privacy, this makes the question of consent very pressing. Should individuals have a legal right to control the use of their likeness in synthetic media? How do we balance that with the legitimate uses of the technology, such as filmmaking or education? Most of these questions are still open, leaving space for people to exploit them.

Fighting the threat through education and regulation Public awareness will also help combat these risks. Definitely, it would teach people the skill of detecting manipulated content to reduce scams and misinformation from spreading uninhibited. These technical solutions would include blockchain authentication mechanisms that test the authenticity of digital media [62], such as embedding metadata in images or videos so that a user can find where a certain image comes from, as in Adobe's Content Authenticity Initiative. On the regulation front, authorities and organizations have begun to address this issue. Enacting a benchmark for the use of AI within ethical boundaries-such as Deepfakes-the proposed legislation at least aspires to be the AI Act of the European Union, but such enforcement will remain key and also requires international coordination since deep fake content knows no borders.

2.2. Explainability of Artificial Intelligence

Today, AI systems tick every box to be invited into potential practical applications such as healthcare diagnostics and financial decisions; therefore, it becomes very important to perceive how exactly AI comes to formulate its decisions [22, 40, 46]. "Explainability" concerns the very *act* of simplifying AI's decision-making to rationalize human trust, auditability, and ethical conformance [17].

An important challenge is posed by the very nature of modern AI systems in their being black boxes, such as deep neural networks, CNNs, and RNNs. With possibly thousands or millions of parameters maintaining such an obscured system, examining their internal workings remains too complex for interested humans. Therefore, questions of fairness and safety regarding consequences of these systems become apprehensive.[61].

2.2.1. Applications of Explainability

From the perspective of a doctor, banker and regulator, a rationale is always required to accept a decision of an AI system [27]. If diagnostic AI tells a doctor to perform a specific treatment without telling why, it could lead to distrust in the AI or, conversely, lead the doctor to ignore important symptoms. Concerning banking, a loan denial may require a legally acceptable justification [75].

Biases commit and mistakes get made far more easily when the reasoning of a system is clear. In the absence of transparency on such a black-box model, bias and errors might slide under the radar with little indication of who may be responsible.

Explainability is useful for just about all AI-related fields. Educators should know how the AI tutors or grading systems provide feedback. In case of any queries on their driving behavior, self-driving cars will log information on the rationale behind their decisions for safety measures. AI can work to analyze data in the courtroom, but judges and attorneys need more transparency to understand how these conclusions were reached.

An AI that flags network threats should provide evidence as to why it classifies something as malicious. Transparency and accountability need to permeate the defense and military domains for AI-enhanced decisions to be ethical and strategically sound.

2.2.2. Challenges and Technical Barriers to Explainability

The transparency of deep learning models is difficult to achieve because these systems often operate on data in ways that are not recognizable by any scheme of human perception [47]. For example, CNNs detect features of images such as edges, textures, and shapes, but the heatmap generated from important pixels does not necessarily describe how the prediction has arrived at that conclusion [64].

RNNs deal with sequential data like text or time series. The hidden states of an RNN are meant to store information over time, but but the contribution of each state to the final output remains unclear, particularly with long inputs [26].

There are also trade-offs between model accuracy and interpretability, with simpler models such as decision trees being much easier to explain, but overall, less accurate than state-of-the-art deep neural networks [61].

2.2.3. Approaches to Explainability

To address these issues, researchers have adopted various approaches. Some create inherently interpretable models: these include rule-based systems or simple decision trees. The others use so-called "post-hoc" methods, which are methods to explain an already-trained black-box model. For instance, LIME [58] and SHAP [42] study how changes in input affect predictions, thus revealing model focus.

In deep neural networks, saliency maps and feature attribution are typically used or, in combination, layer-wise relevance propagation (LRP) [3]. Another line of thought is disentangled representations, which are designed to separate latent factors such as color, shape, or style, which can then be manipulated independently.

But, importantly, the means of arriving at these explanations should really be comprehensible, and *honest*: if an AI system gives dishonest explanations, it defeats the purpose of transparency. Striking this balance will be one of the major challenges for future AI research.

3 | Related Work

In this chapter, we will cast an eye over the most important features incorporated in definitions of deep learning and machine learning, and lay down the fundamentals for our deepfake work and explainability. First, a brief background in machine learning will occur, and then a review of important deep learning architectures, including convolutional neural networks (CNNs) and recurrent neural networks (RNNs). Next, a review of transformers, and then a new field in multimodal models will occur. Finally, a review of how these methodologies can be leveraged for deepfake detectability and explainability will occur.

3.1. A Brief History of Machine Learning

Machine learning (ML) is a computational field of study concerned with algorithms and statistical models that can enable computers to learn from examples, and not through direct programming [6]. In its early years, during the 1950s and 1960s, activity in ML concentrated almost exclusively in symbolic techniques and simple classifiers (e.g., Perceptron), but during the 1990s, with increased computational powers and availability of big datasets, activity in ML gained momentum. From Regression to Classifiers

From Regression to Classifiers One of the first techniques in ML is linear regression, utilized for predicting continuous values through fitting a linear model to a seen observation. Classification techniques soon developed, with a view to predicting discrete labels for new inputs. Decision trees, k-nearest neighbors, and support vector machines (SVMs) [13] are well-known classifiers. These algorithms performed well for structured problem sets, but most of them took a lot of feature engineering and performed poorly with high unstructure such as images or raw text.

Rise of Data-Driven Approaches With increased volumes of information, researchers have increasingly embraced information-dependent techniques that acquire representations directly and explicitly through examples. That transformation opened the stage for deep learning, in which neural networks with many layers can learn to extract high-level

features from unprocessed inputs [33].

3.2. Deep Learning and Neural Networks

Deep learning changed the fortunes of machine learning by utilizing the power of heavy neural network architecture with equally high-order training algorithms [31]. Unlike earlier methods that relied on careful feature engineering, deep networks learn features directly from raw data. The basic building block of a neural network is a collection of neurons arranged in layers, each of which transforms its input in some way. It is not until they mixed efficient back-propagation together with GPU acceleration and vast, labeled datasets, even though a multilayer variant of this does have existed, really, in most respects. That led to phenomenal improvements in tasks such as image recognition.

Early work by LeCun et al. on LeNet [32], which recognized handwritten digits, showed how convolutional structures can extract useful patterns from images. However, the restricted computing resources limited progress for some time. As GPUs became widely available, training deep networks became significantly faster, fuelling a series of breakthroughs in computer vision and beyond.

Many neural architectures have come and gone over the years, and some have really resonated. EfficientNet [69], for example, offered a systematic way to scale model depth, width, and resolution to minimize trial-and-error networking design. ResNet [23] allowed skip (residual) connections which enabled training of networks hundreds of layers deep without descending into the abyss of vanishing gradients. Inception [67] operates by processing multiple filter sizes in parallel at each layer to capture features from different scales. With this, both architectures have helped change the essence of computer vision and set new benchmarks for classification, detection, and segmentation.

3.2.1. Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNNs) are used for processing two-dimensional data such as images by sliding some learn-able kernels (or filters) over the input data. This architecture dramatically reduces the number of parameters in comparison with fully connected layers. At the initial layers of a CNN, the filters would usually learn to detect simple features such as edges or corners, while the filters from the final layers would detect more complex structures and patterns. The same kernel is reused across different image regions, imparting translation invariance on the model, which is an essential property for application in object recognition.

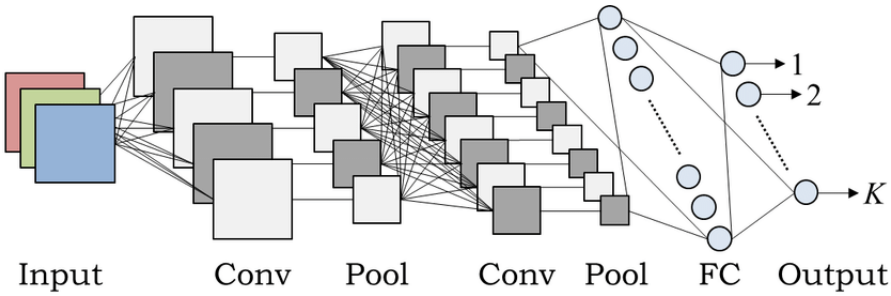


Figure 3.1: A generic CNN architecture, illustrating the convolution, pooling, and fully connected layers that form the foundation of many vision-based deep learning models. Adapted from [25].

CNNs are not just for classification; they are also used for semantic segmentation, instance segmentation, and object detection. Architectures U-Net and Mask R-CNN are two good examples where convolutions allow precise localization of objects or the delineation of boundaries at a pixel-wise level. However, the downside of having such prowess is that CNNs are often immensely resource demanding, especially when talking about very deep architectures. Cooperation among methods such as batch normalization, parameter pruning, and optimizing GPU kernels counters those costs.

Furthermore, CNNs are also primarily local receptive field-based. A shallower or narrower network may face the challenge of capturing the global context across the entirety of an image. Irrespective, they remain irreplaceable in applications ranging from, but not limited to, face recognition and image synthesis-all performers in any pipeline for deepfakes. They remain the benchmarks against which any computer vision task proves its flexibility and robustness.

3.2.2. Recurrent Neural Networks (RNNs)

CNNs have performed well in the processing of static data including images. Whereas RNNs are the ones that take care of sequences wherein the dependence of each element is on what comes prior. Standard RNNs accomplish this through the updating of a hidden state with respect to time so they can efficiently process speech, text, or any other sequential input. They, however, face challenges with very long sequences, as vanishing or exploding of gradients can occur.

The second generation of RNNs, such as LSTM (Long Short-Term Memory) [26] and GRU (Gated Recurrent Unit) [12] have been designed to overcome the aforementioned problems. Gating mechanisms are designed to govern the amount of information flowing through

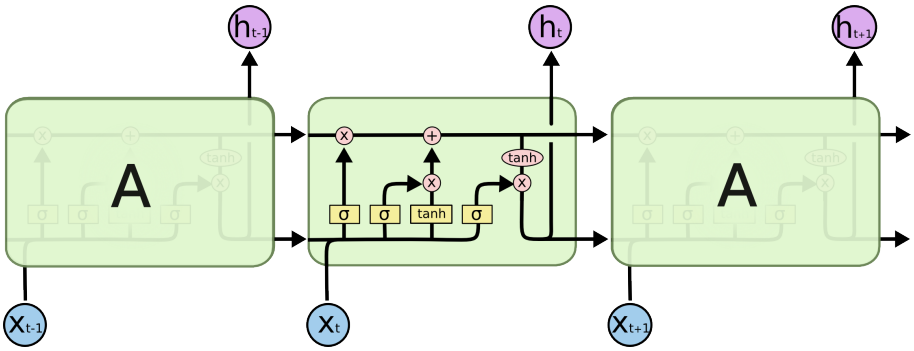


Figure 3.2: An unrolled LSTM cell demonstrating how the hidden state $\mathbf{h}_t$ evolves over time. The input $\mathbf{x}_t$ is processed through gating mechanisms (input, forget, and output gates), which decide how much new information to incorporate and how much past information to retain. The tanh activation helps regulate the cell state, enabling LSTM networks to capture long-range dependencies in sequential data. Taken from [50].

the network, enabling it to find long-range dependencies with more efficiency. Once, RNNs ruled the tasks like machine translation and language modeling, but nowadays, in many applications, they have been taken over by transformer models that can handle long sequences well and parallelize the computations. Nonetheless, RNNs can still work reasonably well with moderately long sequences applied to problem domains where such sequential processing matches the nature of the data.

Towards Transformers Historically, RNNs were the leading choice for sequence modeling, but attention-based mechanisms now take center stage. The next section examines how transformers use attention to parallelize sequence processing and capture long-range dependencies more effectively than RNNs could. We will also see how these newer models became central to modern deep learning.

3.3. Transformers

Transformers shifted the landscape of natural language processing (NLP) by using attention mechanisms instead of the usual step-by-step operations in recurrent neural networks (RNNs). This change, introduced by Vaswani et al. [74], allows transformers to look at every part of an input sequence in parallel, making them much better at capturing long-range relationships. They also train faster, which has led to a surge of new applications—ranging from machine translation and text summarization to question answering and beyond.

3.3.1. Tokenization and Word Vectorization

Before a transformer can process any text, the input must be converted into a numerical form. This process involves two main steps: tokenization and word vectorization.

Tokenization

Tokenization is the process of segmenting raw text into smaller but understandable units called tokens. In contrast, early NLP systems commonly performed tokenization based on words by treating spaces and punctuation marks as delimiters for the segmentation of sentences. For example, the sentence

"The student put the book on the table."

might simply be split into tokens such as `["The", "student", "put", "the", "book", "on", "the", "table", "."]`.

However, tokenization at word level often can't handle issues like rare words or unknown words. Some new approaches are solving these problems with a technique called *subword tokenization*, which sub-divides a word into small components. Two commonly used methods are:

- **Byte Pair Encoding (BPE):** BPE merges pairs of characters or character sequences in an iterative fashion, which leads to the formation of subword units. And the word `"unbelievable"` can be tokenized to `"un"`, `"believ"`, and `"able"` [65].
- **WordPiece:** Similar in spirit to BPE, WordPiece starts with individual characters and builds a whole vocabulary of subwords. For instance, a word like `"unbelievable"` might be tokenized as `"un"`, `"##believ"`, and `"##able"`, where the prefix `"##"` indicates that the token is a continuation of a word [78].

The subword methods allow models to manage unusual or out-of-vocabulary words by splitting them into known components, which implies reducing the number of unseen tokens at inference.

Word Vectorization (Embeddings)

After tokenization, every token must be mapped to its numerical representation, more commonly referred to as an *embedding*. This mapping from tokens to vectors allows the models to perform mathematical operations on text. Earlier NLP systems generally produced embeddings using unsupervised approaches on large corpora.

Early Embedding Techniques

- **GloVe:** The Global Vectors (GloVe) model operates by treating word embeddings as global statistics of co-occurrence. For example, word embeddings for `"king"` and `"queen"` will not only express high similarity for these two words, but also relational distinctions such as gender [54].
- **fastText:** In contrast to GloVe in representing only words, fastText, with its characteristic of being able to take an arbitrary length of character n-grams, finds a way to infer embeddings of words not seen during training. This n-grams-based subword treatment is thus advantageous for morphologically rich languages [7].

Transformer-Based Embeddings Modern transformer architectures learn the embeddings in the course of training. Such an approach gives rise to contextualized embeddings in which the representation of the token varies with the context in which it occurs. For example, the word `"bank"` is assigned different embeddings in:

- *"I sat by the bank of the river,"* where the vector reflects a riverside context.
- *"I need to deposit money at the bank,"* where the vector represents a financial institution.

Transformer models have the unique ability of creating representations far more rich in their nuances by combining tokenization and embedding method in a complete end-to-end training. Each token, as a vector, carries a wealth of semantics and syntax that the model can utilize in various downstream applications, such as translation, summarization, or question answering.

3.3.2. The Attention Mechanism

Traditional RNNs process data one step at a time, which can make it tricky to keep track of distant parts of a sequence. Transformers introduced an attention system (3.1) that looks at every position at once, so any token can quickly refer to any other token. This parallel approach was a real turning of the tables on the old sequential mindset, letting models train faster and handle longer texts.

Self-Attention At the core of a transformer is *self-attention*, which helps each token figure out how much it should pay attention to other tokens in the sequence. Concretely, tokens are turned into query, key, and value vectors, and the model uses them to decide which tokens influence each other. For example, in the sentence *"The student put the*

$$\text{Attention}(Q, K, V) = \text{softmax}\Big(\frac{QK^\top}{\sqrt{d_k}}\Big)V \tag{3.1}$$

Figure 3.3: The scaled dot-product attention, where Q, K, and V are query, key, and value matrices, and d_k is the dimensionality of the query/key vectors.

book on the table", the word "book" might strongly attend to *"table"* because they go together.

Multi-Head Attention Transformers don't just do this once; they use multiple *attention heads* in parallel, each focusing on a different aspect of the data. These heads then come together to form a richer view of the sequence. One head might track grammar, while another tracks meaning, and yet another might pick up on a specific style.

The original Transformer design has two main parts: an encoder and a decoder [74]. The encoder reads the entire input (like a sentence in French) and turns it into a set of context-aware vectors. Then the decoder uses both self-attention (to check what it's generated so far) and encoder-decoder attention (to look at the encoder's output) to produce the translated sentence (say, in English) one token at a time. Compared to older RNN-based systems, transformers handle long sentences more smoothly and can process them in fewer steps. As a result, they tend to be faster and less prone to forgetting earlier parts of the text.

3.3.3. BERT, GPT, and Beyond

BERT: Bidirectional Encoder Representations from Transformers The first paper on BERT by Devlin et al. stated that it is basically the encoder of a transformer modified for the specific tasks. BERT's predictions of masked words are bidirectional, based on the context it receives from both left and right positions. This allows BERT to learn embeddings rich in context. The task of BERT became quickly very important for almost any task: classification, NER, or question answering.

GPT and ChatGPT The earliest approach started with an emphasis on the decoder with lesser knowledge, this Radford et al. set the stage for GPT-2 and -3 [9], asserting that when large amounts of texts have been used to train the model, it can generate sentences in fluent human language. In that regard, OpenAI's ChatGPT adds a twist to such chitchat by fine-tuning GPT for dialogue-much back-and-forth hacking discussions

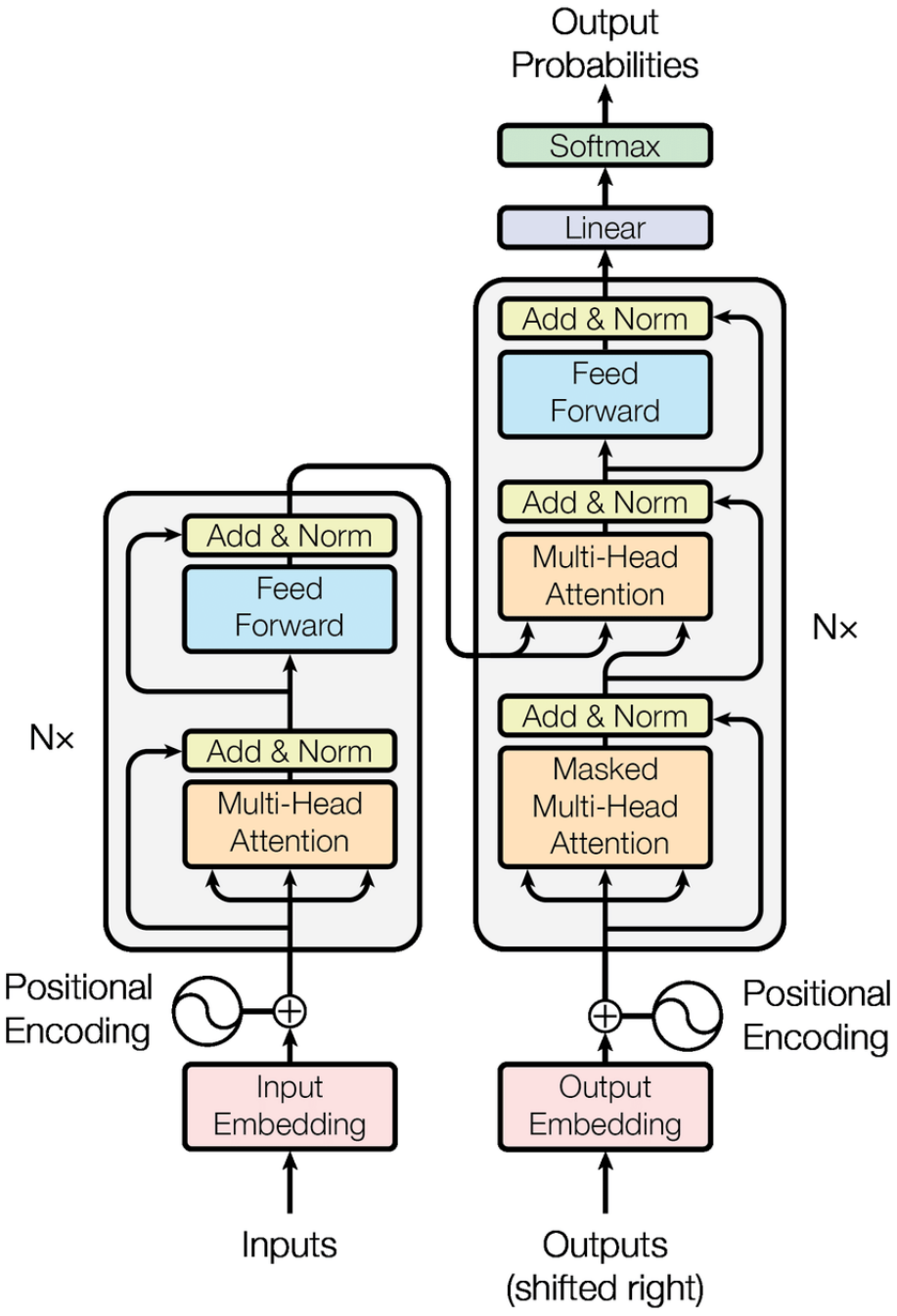


Figure 3.4: A schematic representation of the Transformer's encoder-decoder structure. Each block uses self-attention, while the decoder also employs encoder-decoder attention to integrate information from the encoded input. Taken from [74].

or maybe even creative writing.

LLaMA, Qwen, Gemma, and Other Large Language Models From the perspective of the last one year, numerous teams and organizations have ramped up the development and release of large language models (LLMs) with very diverse optimizations and areas of improvement. Meta's LLaMA series, for instance, is geared to efficiently scale with fewer parameters while still maintaining competitive performance with OpenAI's GPT models. The other major focus of LLaMA is on enterprise readiness and thus LLaMA provides a very compelling option for organizations in need of deployable, open-weight AI solutions that are highly adaptable and cost-effective. Likewise, Qwen from Alibaba sets itself apart with robust multilingual capabilities and enterprise-ready features that allow it to present a diverse array of solutions to companies across different linguistic and cultural ecosystems. This multilingual capability is matched by training that has also been optimized for application-level generalization. Meanwhile, Gemma, the Google DeepMind creation, embodies the thrust for new training mechanisms and domain-specific fine-tuning; indeed, many applications from research institutions and industries alike explore state-of-the-art approaches in reinforcement learning from human feedback (RLHF), retrieval-augmented generation (RAG), and so on, as their highly specialized datasets are oriented toward application areas such as medicine, law, or finance.

3.3.4. Implications for Explainability and Large Contexts

Above all, transformers manage to somewhat demystify the workings of a model focusing on certain portions of text. Attention maps reveal during prediction-making processes which tokens matter most for given predictions, providing a valuable insight for explainability [59]. This is especially important in fields such as medicine or law, where decisions can carry deleterious repercussions.

The other advantage that transformers have in this respect is in being able to handle larger text pieces without hidden states, which suits them well for tasks like summarizing long documents. As larger and more capable versions of such models are still built upon the same transformer architecture, the next section will deal with how these concepts extend into other modalities, enabling systems that couple images or audio with language to confront harder problems.

3.4. Multimodal Models

Deep learning has always focused on a single input type, such as text, images, or audio. However, many real-life applications require an understanding of multiple data sources concurrently, which is where the *multimodal* models come into play. These models accept two or more modalities in parallel-e.g. language and vision-to produce richer representations that perform better on tasks that may otherwise be limited by a single data type.

Most human communication is inherently multimodal. We do not only depend on words; we use facial expressions, gestures, and tone of voice. Similarly, many of the practical applications of AI combine text, images, audio, and sometimes video to gain more subtle insights. For example, when reporting on a global event, an article often presents photos or videos that offer context text alone cannot adequately supply [4].

Beyond human communication, certain domains naturally create data in multiple formats. Medical records consist of text-based patient histories, lab results, and diagnostic images such as X-rays or magnetic resonance images. E-commerce platforms commonly combine product descriptions (text) with images or videos of the item for sale, whereas social media posts frequently juxtapose text captions with pictures or short clips. How these multimodal models derive the answers combines these disparate inputs together to answer more complex questions-like what kind of product it is, along with whether or not it matches the user's aesthetic sense.

3.4.1. Multimodal Fusion Approaches

Researchers have always trained separate models for each modality, say a CNN for images and LSTM for text, then fused the model outputs at a later time. Although effective, such late-fusion strategies could not capture deeper document reading cross-modal interactions. Another way would be to use join architectures, often based on transformers [72], where attention layers are at work evaluating tokens of both text and image representations at the same time.

In a typical vision-language model, the image is broken down into a sequence of patches or region features (extracted by a CNN or vision transformer), which then interact with text tokens in the same attention mechanism [18]. This *cross-attention* ensures that the words in the text can "attend" to specific parts of the image, and vice versa, leading to a richer alignment between the modalities. In addition, a common architecture in vision–language models employs a vision encoder to project images into the same embedding space as a pre-trained language model. By aligning image embeddings with text embeddings, this

design enables robust cross-modal interactions that facilitate downstream tasks such as image captioning and visual question answering.

3.4.2. Recent Architectures

CLIP (Contrastive Language–Image Pre-training) Developed by OpenAI, CLIP [56] takes an image and a text description, embedding them into the same latent space. Through contrastive learning, CLIP aligns image–text pairs that belong together while distancing unrelated ones. The result is a model that can handle zero-shot classification by matching an unseen image to the text label that best fits it.

Flamingo Flamingo [1], from DeepMind, blends a large, pre-trained language model with adapter layers trained on image–text data. The core feature is that the main language model remains mostly frozen, while the new layers learn to integrate visual features. This approach allows tasks like image captioning or visual Question Answering with fewer additional parameters than building a multimodal system from scratch.

LLaVA (Large Language and Vision Assistant) Notably, LLaVA extends a pre-trained Large Language Model to which a vision encoder has been added, facilitating complex functions such as image-driven dialogue, interpretation, and instruction-following. In the integration of text and visual input, LLaVA shows how multimodal data could increase the understanding and generative ability of AI. The flexible design points to the promise of general-purpose assistants that seamlessly blend language and vision.

3.4.3. Beyond Vision and Language

Although vision–language is by far the most common focus in multimodal research, other combinations exist. Audio–text systems already power speech recognition and synthesis, and recent tri-modal setups combine vision, text, and audio [72]. This might involve analyzing a video that features spoken dialogue and text on screen, providing a holistic understanding of the content. In robotics or augmented reality, sensor streams (like LiDAR) can also join the mix, offering real-time data that goes beyond simple visuals or language.

3.4.4. Training and Datasets

Multimodal models often rely on large datasets covering multiple modalities. For vision–language, popular choices include MS COCO [39], Flickr30k [81], and Conceptual

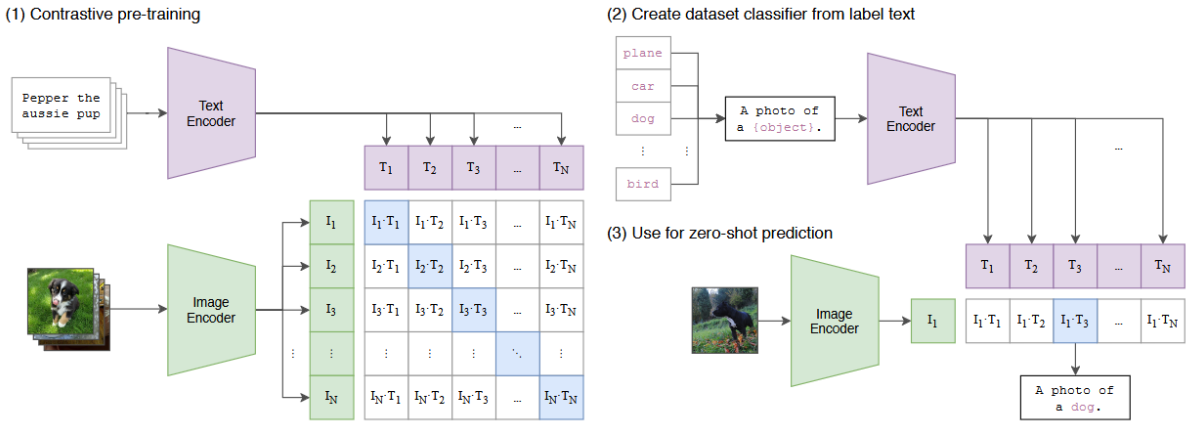


Figure 3.5: A schematic of the architecture of CLIP. In the course of contrastive pre-training (1), text and image encoders learn to represent with alignment across extensive collections of pairwise text-image data. Afterward, inference uses label text prompts (2) to generate a classifier with no training for a particular task, thus allowing for zero-shot predictions (3): image embeddings are then compared to text embeddings. Taken from [56]

Captions [66]. Audio–visual tasks may use AVSpeech [19] or VoxCeleb [49]. Training these models can be computationally heavy, since each modality might require substantial processing—images need high-resolution transforms, and text can run into thousands of tokens.

3.4.5. Practical Applications and Challenges

Multimodal models combine different data streams to accomplish tasks that are nearly impossible for single-modality methods:

- **Visual Question Answering (VQA):** A visual question-and-answer scenario where a system sees an image and answers a natural language question about it [2].
- **Image Captioning:** The task of generating descriptive text from an image or frame.
- **Cross-Modal Retrieval:** The task of retrieving images based on text prompts (and, sometimes, vice-versa).
- **Assistive Tech:** The provision of textual/audio description to users with sight or hearing impairment.

Nevertheless, integrating multiple modalities can create noise or inconsistencies if the data is not perfectly aligned (e.g., when a caption does not match an image). Large multimodal models may also inherit or amplify any bias present in their training sets [24], emphasizing the need for careful curation of the datasets and mitigation of bias.

3.4.6. Explainability in Multimodal Settings

In diverse environments, multimodal models can assist explainability by providing information concerning system focus such as which regions of an image or segments of text. Take, for example, AI used for scene descriptions; attention maps can show every object or area that contributes to any specific sentence. Text-audio alignments work under the same principle, showing the segments of an audio wave that influenced the textual output. This kind of explainability gains meaningfulness in domains such as healthcare and legal analysis.

3.4.7. Future Directions

In light of these promising developments, multimodal models present an exciting direction for deepfake detection technologies. By being capable of leveraging complementary information from visual, textual, and even audio modalities, such models are well-placed to capture subtle inconsistencies that may be the signature of manipulated content. Such integration not only enhances detection performance, but also explainability by providing insights into which features drive the decision-making. For the remainder of this thesis, we will rely on these observations to explore how multimodal methods, combined with large language models, can be leveraged to address the deepfake detection issue.

4 | Research Questions

This thesis explores the potential of combining multimodal language models into deep-fake detection systems. After discussing the background of deepfake detection and explainability, and talking about the relevant literature in deep learning, we may finally introduce the research questions posed for our work. Despite the traditional limitations of unimodal detectors based solely on images, the study aims to overcome these limitations. This study is guided by the following research questions:

1. **What role do multimodal language models play in generating natural language explanations by integrating both textual and visual context for deepfake detection?**
   In this work, a multimodal approach is primarily adopted to improve the explainability of deepfake detection. Instead of using classical methods, which only highlight tampered regions by showing visual cues, our approach generates natural language explanations for what and how much the manipulation takes place. In this manner, it is possible for the user not only to understand that tampering has been inserted, but also to understand it better, ensuring more trust and less suspicion. The experimental evaluation focuses on the quality, clarity, and informativeness of these explanations compared to traditional visual indicators.

2. **How does the zero-shot performance vary for different tampering techniques?**
   This question investigates how many generalizations can be expected from multimodal models when they are given tasks that they have not been explicitly trained on, such as deepfake detection. In this thesis, we will study the model's performance on different tampering techniques (e.g., splicing, content-aware fill, and copy-move) in a zero-shot setting.

3. **How do zero-shot and few-shot strategies influence detection outcomes, and what insights do they provide regarding the quality and diversity of the custom deepfake dataset?**
   In developing a customized dataset of manipulated images, this question explores

whether applying zero-shot and few-shot testing can not only yield competitive performance but also serve as indicators of the dataset's adequacy in terms of quality, diversity, and real-life scenario coverage.

4. **To what extent does fine-tuning a multimodal language model enhance the interpretability of deepfake detection?**
   Besides mere performance gains, the interpretability of model decisions plays an important role in the establishment and maintenance of trustworthiness and transparency. In this research, an analysis is conducted to determine whether fine-tuning a multimodal language model enhances the quality of a prediction. The investigation also includes the aspects of whether the model can mark important regions in images and key phrases in the text that highlight manipulations in deepfake.

5. **How can the performance and interpretability of the multimodal approach be systematically analyzed both quantitatively and qualitatively?**
   This question focuses on establishing robust evaluation methods. Traditional detection metrics (accuracy, precision, recall, and F1 score) will be employed, alongside assessments of natural language explanations through sentence similarity measures with ground truth captions, to provide a comprehensive evaluation of both performance and explainability.

5 | Dataset creation

Before detailing the dataset creation process, it is important to understand the rationale behind it. As described with more details in Appendix A, our preliminary experiments with LLaVA OneVision explored several deepfake manipulation techniques: splicing, copy-move, and in-painting. These early investigations revealed that splicing produced the most distinct and predictable alterations. Motivated by this insight, we decided to develop a custom dataset that focuses on splicing manipulations, while also incorporating genuine images to mitigate bias, to eventually test an array of detection models in zero-shot, few-shots and fine-tuned settings.

This dataset was created to provide a consistent, well-annotated source specifically tailored for deepfake detection research. Its primary purpose is to support the development and benchmarking of advanced detection methods by offering rich captions that detail the type of manipulation performed on each image. The dataset preparation involved three key stages:

- Generating detailed textual descriptions for manipulated images.
- Integrating genuine, non-manipulated images to ensure a balanced dataset.
- Finalizing the dataset composition to create a robust foundation for both zero-shot and few-shot model evaluations, and for potential fine-tuning.

Again, by concentrating on splicing, our dataset offers a controlled and structured environment that facilitates the reliable training and assessment of deepfake detection systems in practical, real-world scenarios.

5.1. Overview of the Dataset Composition

The initial dataset derives from a pre-existing collection named DIS100K [68], which contains images of the amount of well over 100,000, that have been manipulated through splicing. Each image is paired with a binary mask, that labels the regions where the splicing occurs, hence the ground truth. The variability in proportions and dimensions

Figure 5.1: Example from DIS100K: A woman's figure has been spliced into the original image. The related mask clearly delineates the tampered region, highlighting the area where the manipulation occurred.

among the images tells the deepfake detector not to correlate manipulation with any fixed size or fixed aspect ratio.

Beyond the manipulated images, genuine images are also crucial to balance the dataset. In this case, the unmodified images were obtained from the dataset RAISE [14] provided by the University of Trento, and also made from cropping tampered regions from selected DIS100K images. To avoid bias, we must ensure that the dataset contains an equal number of genuine and altered images. The final dataset contains 20,000 images in the following division:

- 10,000 manipulated images (with splicing) from DIS100k.
- 5,000 genuine images from the RAISE dataset.
- 5,000 genuine images obtained by removing tampered regions from DIS100K.

5.2. Generation of Descriptive Captions

The description generation process for tampered images has two closely intertwined phases: first, identifying the tampered region; and secondly, elaborating a dynamic caption that explains the manipulation.

5.2.1. Isolating the tampered regions and generating captions

Each image in the DIS100K dataset has been paired with a binary mask that pinpoints the location of the splicing. By multiplying the pixels of the modified image with those in the corresponding mask, we are able to extract the inserted object or region. This focused extraction ensures that subsequent captions describe the manipulated segment as precisely as possible. To generate the caption for the isolated object, we leveraged *BLIP Large* by Salesforce [36] model. BLIP is a state-of-the-art image captioning system known for its high reliability and accuracy, providing a robust baseline description for tampered content.

5.2.2. Creating diverse captions

Once the *BLIP* baseline caption of the object was produced, the next stage was to create an enriched and diverse description that contextualized the manipulation. Instead of following a fixed template, manipulating various linguistic components results in a more diverse dataset.

- **Define Caption Patterns:** A predefined set of sentence templates is prepared, each explicitly describing evidence of image manipulation and containing a placeholder for the manipulated object. Examples include:
  - `"Yes, the image has been tampered because {object} was inserted into it."`
  - `"Yes, it's clear that {object} has been added to the scene."`
  - `"Yes, manipulation is evident as {object} appears in the image."`
- **Random Selection and Caption Generation:** For each manipulated image, the manipulated object is first identified using a binary mask, and an object description is generated with BLIP. Then, a caption pattern is randomly chosen from the predefined set, and the object description is inserted into this pattern to generate the final caption.
- **Example Outputs:** This method produces captions such as:
  - `"Yes, the image has been tampered because a vase was inserted into it."`
  - `"Yes, it's clear that a car has been added to the scene."`
  - `"Yes, manipulation is evident as a flower appears in the image."`

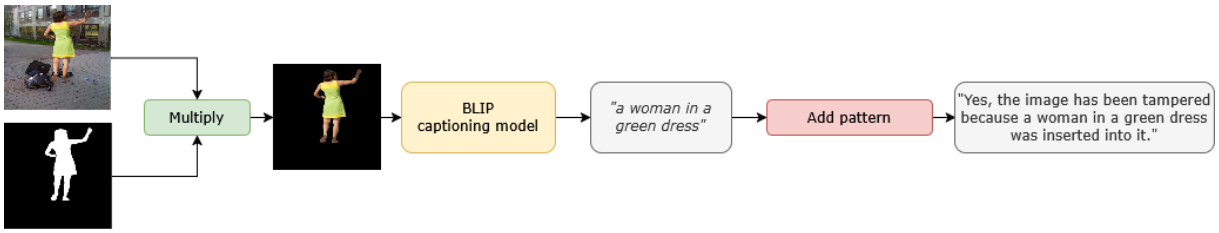


Figure 5.2: General pipeline of our custom dataset creation.

The diversity of captions prevents the model from overfitting to a single descriptive style and would add reinforcement to the correlation between the visual manipulation and its contextual explanation. In describing the dataset, the superficial isolation of tampered regions and an enriched initial description given by BLIP through custom-built linguistic components would provide precise visual annotation and a variety of descriptive language.

Following the assembly and annotation of the 20,000-image dataset, after randomly shuffling, we partitioned it into three distinct subsets to train and evaluate the deepfake detection model robustly. Specifically, 90% of all dataset images, or 18,000 of them, went toward training, while the remaining images were split equally into 1,000 images for validation and 1,000 images for testing (5% in either case). With this strategic split, the model can learn from a wide variety of examples while performance can still be objectively assessed on data not previously seen by the model. In the next section, we will cover the fine-tuning process and the model architecture.

6 | Methodology and Experimental Results

This chapter outlines the methodology used in our research and reports the experimental results obtained during the process of conducting our deepfake detection experiment. We evaluated the performance of several state-of-the-art vision language models in few-shot and zero-shot settings to identify their ability to detect and describe image forgery. Finally, we will provide the evaluation metrics with the results obtained.

6.1. Zero-Shot and Few-Shot Experiments

6.1.1. Models Employed

Our study uses three distinct vision language models, each designed with different architectures and capacities.

LLaVA OneVision 0.5B *LLaVA OneVision* [35] integrates a robust vision encoder (SigLIP), a two-layer MLP multimodal projector, and the Qwen2 language model. The overall architecture comprises 899 million parameters designed to efficiently merge advanced visual processing with sophisticated language understanding.

Llama 3.2 Vision Instruct 11B Built on the robust Llama framework, *Llama 3.2 Vision Instruct* [41] is a cutting-edge multimodal model that seamlessly integrates visual and text processing. With 11 billion parameters, the model is intended to perform image recognition, visual reasoning, captioning, and visual question-answering tasks. The "Instruct" module specifies that the model was trained on instruction-based data, via supervised fine-tuning and human feedback-guided RL (Reinforcement Learning) to make its responses more aligned with user-provided prompts. This training enables the model to carry out extremely detailed instructions and perform advanced, context-sensitive visual analysis, and is well suited for application in deepfake detection, where correct interpre-

tation of visual manipulation is critical.

DeepSeek VL2 Tiny *DeepSeek VL2 Tiny* [79] is a highly optimized small vision language model for rapid inference and robust visual processing. Built using a Mixture-of-Experts (MoE) framework, it only activates the most relevant subnetworks during inference time, thereby reducing computational expense without compromising on competitive performance. With 1.0B activated parameters, *DeepSeek VL2 Tiny* is engineered to balance resource constraints with the capability to detect subtle deepfake manipulations. Its design allows quick processing of tasks such as visual question answering, optical character recognition, document understanding, and visual grounding, making it a valuable resource in forensic applications where quick and precise analysis is critical. Furthermore, the resource usage efficiency of the model and support for BF16 tensor types enable deployment in real-world scenarios with limited computational capacity, so it becomes possible to conduct deepfake detection effectively and timely even in the setting of limited resources.

6.1.2. Zero-Shot Experiments

To comprehensively assess model performance, we designed two types of experiments: zero-shot and few-shot. Both protocols were applied to our dataset's test set comprising 1,000 images, each labeled as either tampered or non-tampered. In the zero-shot setting, no prior task-specific training was performed. Each model was simply presented with a test image and prompted the question:

> `"Has this image been manipulated? If so, what has been added to the image?"`.

The models' outputs were then classified into one of four categories: true positives, false positives, true negatives, or false negatives, based on their ability to correctly determine whether the image had been tampered with, but we will talk about it later in Section 6.1.4.

6.1.3. Few-Shot Experiments

The few-shot experiments aimed to provide contextual examples to enhance the model's performance on the detection task. For each test image, three random images from the training set (each with its ground truth indicating tampering or non-tampering) were presented sequentially to the model. The dialogue was structured as follows:

1. The session began with the prompt: `"Has this image been manipulated? If so, what has been added to the image?"` followed by the first training image and its caption.
2. The conversation continued with: `"Awesome! What about this new image, has this image been manipulated? If so, what has been added to the image?"` accompanied by the second training image and its caption.
3. A similar interaction was repeated with a third training image.
4. Finally, the test image was introduced with the same query.

The expectation was that by exposing the model to a few task-specific examples, it would better internalize the nuances of image manipulation and thus provide more accurate responses.

6.1.4. Evaluation Metrics

At this stage, it becomes crucial to analyze the performance of the models. Since the generated captions are in natural language, a straightforward classification approach is not feasible. Instead, we performed a manual evaluation of the responses, in which each caption was assigned one of the following labels:

- **True Positive (TP)**: the image was tampered and the model correctly identified it as modified.
- **True Negative (TN)**: the image was genuine and the model correctly recognized it as such.
- **False Positive (FP)**: the image was genuine but the model erroneously detected tampering.
- **False Negative (FN)**: the image was tampered but the model failed to detect the modification.

The models' performances were evaluated using standard classification metrics: Accuracy, Precision, Recall, and F1 Score. These metrics offer a well-rounded view of each model's strengths and weaknesses in correctly classifying manipulated images.

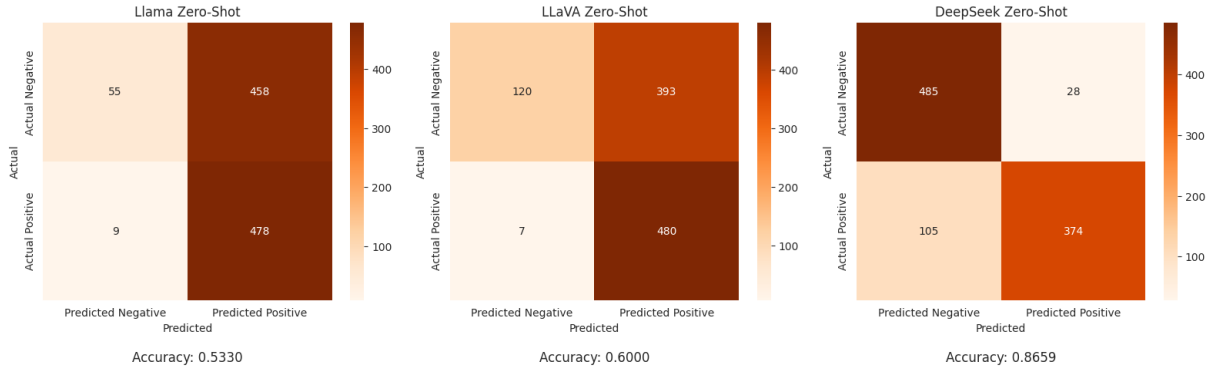


Figure 6.1: Confusion matrices for zero-shot evaluation.

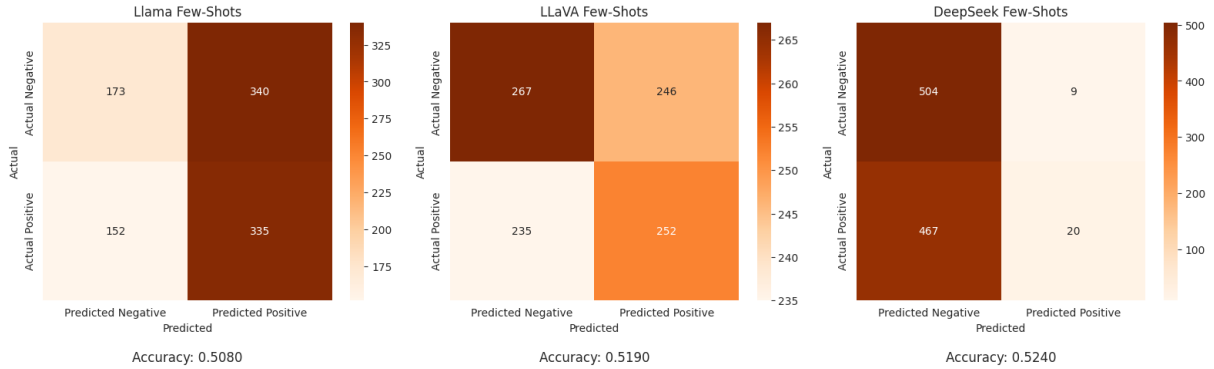


Figure 6.2: Confusion matrices for few-shot evaluation.

Model	Accuracy	Precision	Recall	F1 Score
Llama Zero-Shot	0.5330	0.5107	0.9815	0.6718
LLaVA Zero-Shot	0.6000	0.5498	0.9856	0.7059
DeepSeek Zero-Shot	0.8659	0.9303	0.7808	0.8490

Table 6.1: Zero-Shot Evaluation Metrics

Model	Accuracy	Precision	Recall	F1 Score
Llama Few-Shot	0.5080	0.4963	0.6879	0.5766
LLaVA Few-Shot	0.5190	0.5060	0.5175	0.5117
DeepSeek Few-Shot	0.5240	0.6897	0.0411	0.0775

Table 6.2: Few-Shot Evaluation Metrics

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \qquad \text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN} \qquad \text{F1 Score} = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$$

Figure 6.3: Evaluation metrics formulae.

6.1.5. Analysis of True Positives

A deeper analysis within the true positive category revealed further complexities. In several cases, even when a model correctly recognized that an image had been tampered with, it sometimes failed to correctly identify the specific object inserted during the manipulation. To capture these nuances, we subdivided true positives into two sub-categories:

1. **True Positives with Correct Object:** Instances where the model both detected the tampering and accurately identified the inserted object.
2. **True Positives with Incorrect Object:** Instances where the model acknowledged the tampering but misidentified the inserted object.

This subdivision was necessary because ambiguities often arose, for example, differences in recognizing items held by a person, variations in clothing, or cases with multiple objects where only the primary modification was expected. Even if the description was slightly overgeneralized, or more specific than expected, as long as the model captured the general area of modification and the overall content, it was considered acceptable.

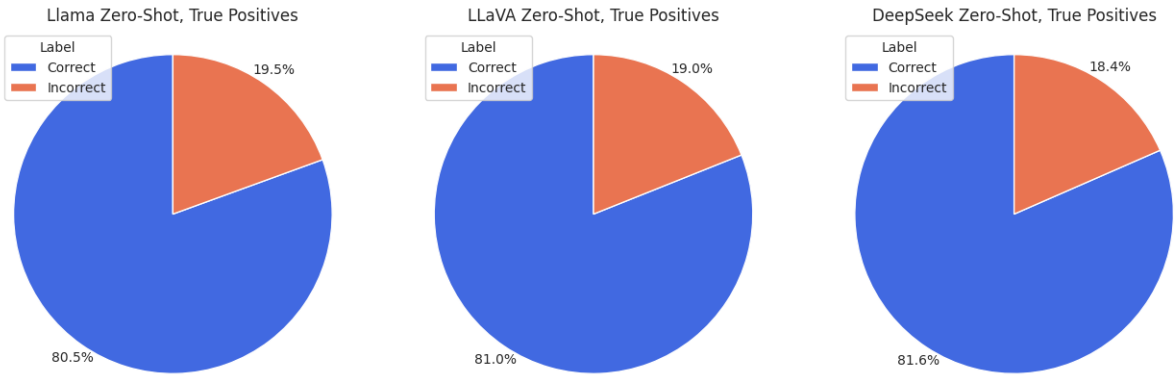


Figure 6.4: Pie charts illustrating the distribution of correct versus incorrect identification of the spliced object within the True Positive predictions, across all models.

6.2. Fine-tuning the Model

Despite promising initial results, the performance observed in both zero-shot and few-shot settings indicated that further improvements were required, particularly in the precise identification of manipulated content. This motivated the fine-tuning phase of our research. In this section, we detail the fine-tuning process applied to our selected model *LLaVA OneVision 0.5B* [35].

6.2.1. Rationale

The rationale for fine-tuning is grounded in the need to enhance the model's sensitivity to forensic cues inherent in manipulated images. By selectively fine-tuning 5.8 million parameters (while keeping the remaining 893 million parameters frozen) we aimed to preserve the robust language understanding of the pre-trained Qwen2 component while adapting the visual and multimodal modules to the specific challenges of deepfake detection.

The upcoming sections will elaborate on the fine-tuning strategy, including the choice of loss functions, hyperparameter tuning, and the overall training regimen. This process is critical to achieving a balance between maintaining the general language capabilities and boosting the forensic precision required for reliable deepfake detection.

6.2.2. The Model's Architecture

`llava-onevision-qwen2-0.5b-ov-hf` is a complex multi-modal architecture consisting of a state-of-the-art language model with specialized components to process visual information. In terms of architecture, it can be conceived of as three main parts:

- **Vision Tower (SigLIP):** This module processes the raw visual input. The feature extraction from the images takes place using a network architecture tailored for visual tasks.
- **2-Layer MLP Multimodal Projector:** This projector maps visual features into a representation space compatible with the language model and thus acts as a bridge between modalities. The use of a two-layer MLP is justified because it helps in learning a compact yet enriched embedding that aligns with the textual features.
- **Language Model (Qwen2):** At the core, there lies the Qwen2 language model, with great capabilities in textual understanding and generation. For fine-tuning, therefore, the language model is kept intact so as not to compromise its forensic and linguistic skills trained in the pre-training phase.

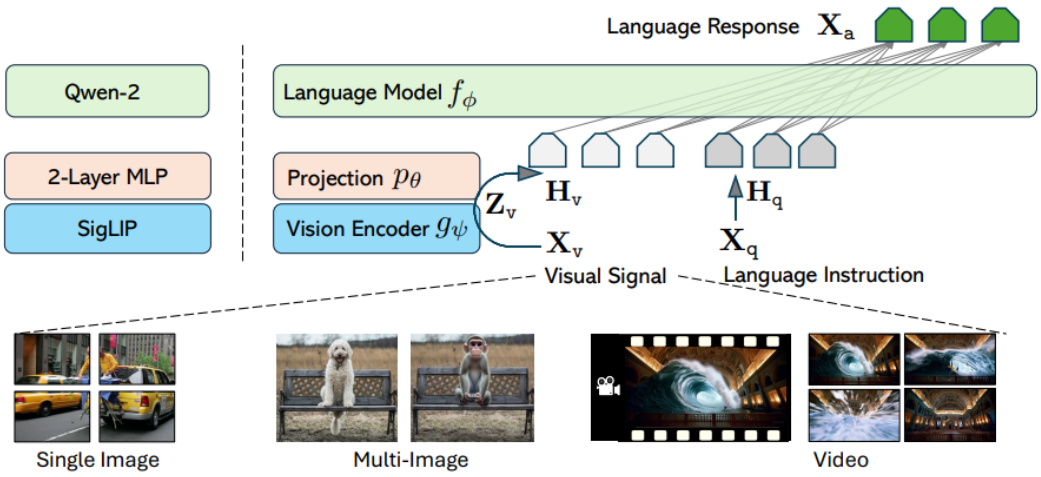


Figure 6.5: Architecture of LLaVA OneVision. It integrates a vision encoder (SigLIP), projection module, and language model (Qwen-2) to process single images, multi-image inputs, and videos for generating language responses. Taken from [35].

The entire model is composed of 899 million parameters. However, only a small fraction of those parameters (5.8M) were being fine-tuned, meaning that the rest (893M) were kept frozen. This method is an example of parameter efficiency, since it aims at training the visual and multimodal aspects toward the forensic domain without hurting the perfectly valid language model capabilities.

6.2.3. Parameter-Efficient Fine-Tuning with LoRA

The model was adapted to our forensic dataset by Parameter-Efficient Fine-Tuning (PEFT) using Low-Rank Adaptation (LoRA). LoRA inserts additional low-rank matrices into the layers of the model, allowing efficient adaptation of the model with the smallest increment in the number of trainable parameters. The LoRA approach was applied only to the vision tower and the MLP multimodal projector, while the language model (Qwen2) was kept intact to preserve its cherished forensic features and linguistic strength. The rationale behind using LoRA can be broken down in two main reasons:

1. **Efficiency:** Compared to full-model fine-tuning, this approach drastically reduces the computational and memory overhead by updating only a small fraction of the parameters. In our case, we decided to set both LoRA rank and alpha equal to 8).
2. **Stability:** Freezing the language model guarantees the stability of the core textual representations and forensic reasoning, while allowing for the adaptation of visual

components to the usability particulars presented by our dataset. This is the desired "forensic vibe" instilled into LLAVA, ensuring that the model adaptation occurs only in the locales that warrant such domain adaptation.

The following are further training configurations for LoRA:

- **Dropout:** A dropout rate of 0.1 was used for the LoRA layers to help prevent over-fitting.
- **Initialization:** LoRA weights were initialized from a Gaussian distribution for smoother blending of LoRA parameters with those given in the existing network.
- **Precision:** The training was done in mixed precision. This was done to save GPU memory on the expense of numerical stability.
- **Determinism:** The training was made deterministic, thus enabling reproducibility.

6.2.4. Training Setup and Hyperparameters

The fine-tuning process was orchestrated using the Adam optimizer [30] enhanced with weight decay, which is particularly effective in training large-scale models. The key hyperparameters and training setup were as follows:

- **Learning Rate:** A learning rate of 1×10^{-4} was chosen to ensure gradual adaptation of the trainable parameters.
- **Epochs:** The model was trained for 44 epochs on the complete forensic dataset.
- **Hardware:** Training was performed on 4 NVIDIA A40 GPUs. Each GPU, featuring 10752 CUDA cores clocked at 1305 MHz and equipped with 45 GiB of memory, provided the necessary computational power.
- **Batch Size and Preprocessing:** A batch size of 2 was used due to memory constraints. Additionally, the image dimensions were reduced by 50% to ensure that the entire dataset could be accommodated within the GPU memory during training.

6.2.5. Chat-Based Training Example and Deployment

To simulate a realistic interaction scenario, the training proceeded using a chat-based prompt. In this setup, the input prompt was comprised of the text:

```
"Has this image been manipulated?  Why?"
```

to which the corresponding image was added. The target answer was the caption created in the previous stage of dataset generation. After successful training, the fine-tuned model has been published on the Hugging Face Hub for further research and application in the forensic domain.

6.2.6. Results of the Fine-tuned Model

Following the fine-tuning process, the adapted model was evaluated on the same test set to assess its forensic precision after targeted adaptation. The captions generated by the model were manually reviewed and categorized using the same evaluation framework as described in the earlier sections.

Figure 6.6 displays the confusion matrix, where the model correctly classified 513 genuine images (true negatives) and 371 tampered images (true positives). No genuine images were mistakenly flagged as manipulated (false positives), and 116 tampered images were overlooked (false negatives). These numbers highlight the substantial improvements achieved through the fine-tuning process, particularly in minimizing false positive errors.

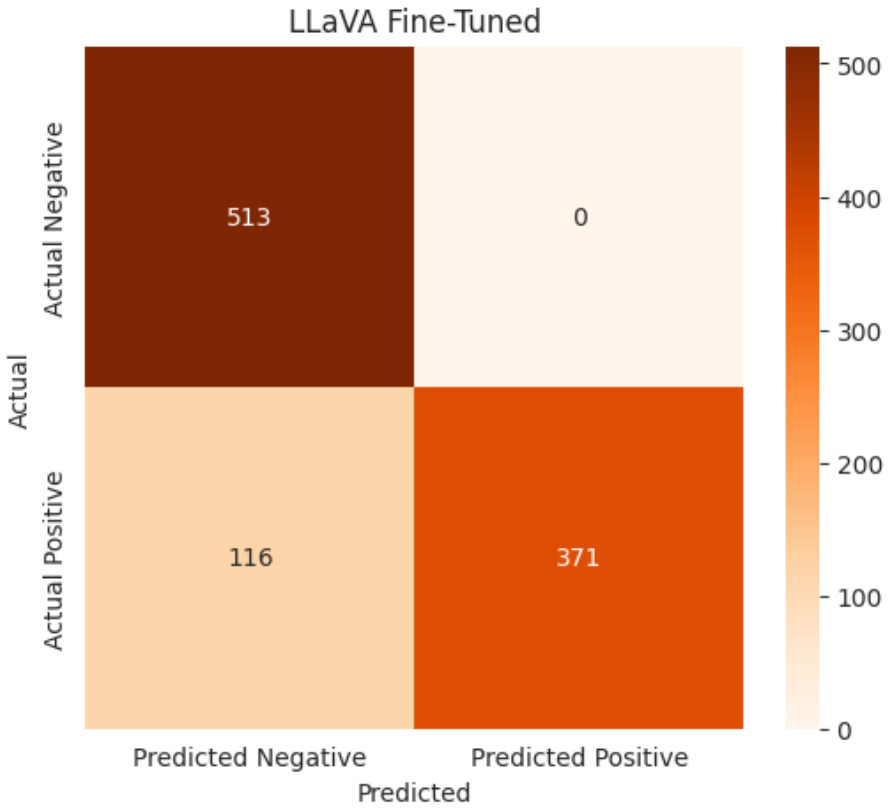


Figure 6.6: The confusion matrix assessing the quality of the fine-tuned model's responses.

The evaluation metrics further give proof of the model's enhanced performance (see Table

6.3). With an accuracy of 88.39% and an exceptionally high precision of 100%, the fine-tuned model demonstrates a robust ability to correctly identify manipulated images while completely avoiding false alarms. The recall of 76.18% indicates that a substantial majority of tampered images were successfully detected, and the F1 score of 86.49% reflects a balanced performance between precision and recall.

Metric	Value
Accuracy	0.8840
Precision	1.0000
Recall	0.7618
F1 Score	0.8648

Table 6.3: Evaluation metrics for the fine-tuned model.

The pie chart in Figure 6.7 elucidates the distribution of true positives by distinguishing between cases where the inserted object was accurately identified versus those where it was misidentified. This nuanced breakdown underscores the model's capability to not only detect tampering but also, in most instances, provide a reliable description of the modifications. Such granularity is especially valuable in forensic applications, where precise identification of alterations can be crucial.

In summary, the results of all our experiments, shown in Table 6.4, are quite diverse across various experimental settings. *DeepSeek* achieved the best accuracy and F1 measure in the zero-shot experiment but performed significantly badly in the few-shot setting, suggesting that it lacks strong abilities to leverage contextual examples. Both *Llama* and *LLaVA* reported comparatively middle performance for both zero-shot and few-shot experiments, and not many improvements were observed from the few-shot environment. In contrast, the fine-tuned LLaVA model achieved the highest F1 and accuracy with perfect precision, demonstrating its highest ability to correctly detect manipulated images with no false positives. The findings strongly indicate that in forensic deepfake detection, task-specific fine-tuning yields the best results.

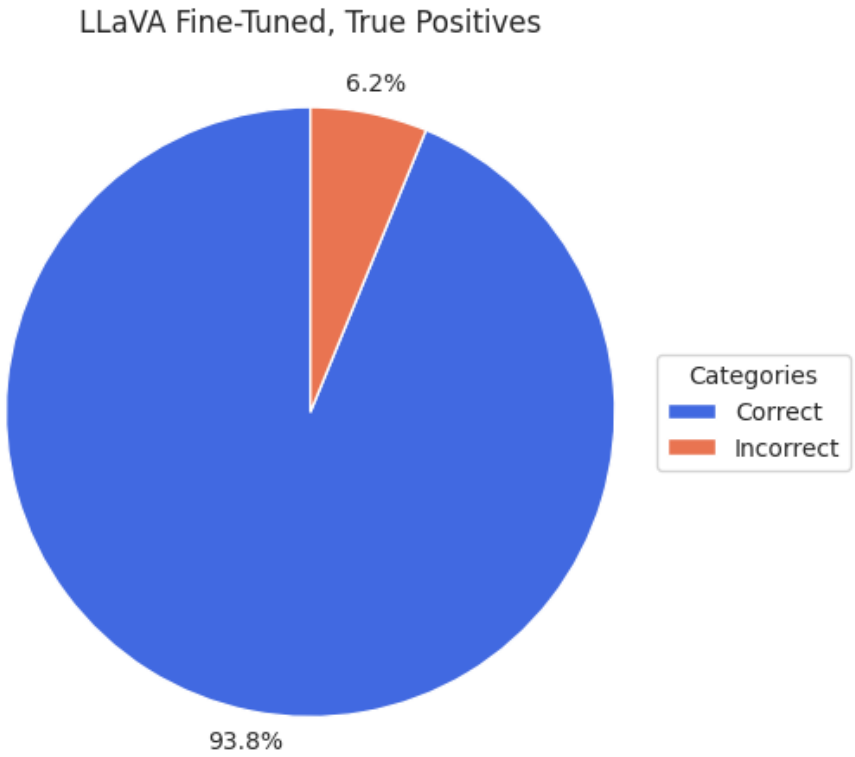


Figure 6.7: Pie chart showing the distribution of True Positives in the fine-tuned model.

Model	Setting	Accuracy	Precision	Recall	F1 Score
Llama	Zero-Shot	0.5330	0.5107	0.9815	0.6718
LLaVA	Zero-Shot	0.6000	0.5498	0.9856	0.7059
DeepSeek	Zero-Shot	0.8659	0.9303	0.7808	0.8490
Llama	Few-Shot	0.5080	0.4963	0.6879	0.5766
LLaVA	Few-Shot	0.5190	0.5060	0.5175	0.5117
DeepSeek	Few-Shot	0.5240	0.6897	0.0411	0.0775
LLaVA	Fine-Tuned	0.8840	1.0000	0.7618	0.8648

Table 6.4: Combined Evaluation Metrics for Zero-Shot, Few-Shot, and Fine-Tuned Settings.

7 | Methods for the Analysis of the Results

In real-world forensic use, determining whether an image has been manipulated—and understanding exactly how—matters. Consider, for instance, an image of a street scene where an automated system generates the caption: "A manipulated street view where a car is present with altered headlights". In contrast, the ground-truth caption might read: "An image showing digital manipulation where the car's headlights have been replaced with a brighter, non-original variant". Such subtle differences are not always captured by metrics based on simple word overlap. Manual parsing of these captions to evaluate semantic equivalence is not only time-consuming, but also prone to subjectivity

Our array of methods seeks to overcome these problems by providing a strong, automated alternative that scores captions with satisfactory semantic fidelity. To this end, we propose a comprehensive evaluation framework with three complementary families of evaluation approaches: conventional NLP metrics, embedding-based evaluations, and LLM as judges. In addition, we introduce the concept of *meta-classification*: a process to assess the evaluation methods themselves by automatically classifying a caption produced as good or bad with optimized thresholds. This hybrid solution not only streamlines the process of evaluation but also lays the groundwork for additional improvements in automated caption analysis and deepfake detection.

Since our fine-tuned model produced better results compared to other models, its generated captions are taken as the primary subject of our evaluation, compared directly with our own dataset of ground-truth captions.

7.1. Conventional NLP Assessments

Traditional metrics provide a baseline to quantify the overlap between generated and reference texts:

- **BLEU**. The BLEU score (BiLingual Evaluation Understudy) [51] is one of the most

commonly used metrics in machine translation. It determines the n-gram overlap between a generated caption and one or more reference captions, while applying a brevity penalty to discourage very short outputs. The n-gram regularity that BLEU provides offers a word-limit means for assessing fluency and adequacy.

- **ROUGE**. The ROUGE metric (Recall-Oriented Understudy for Gisting Evaluation) [38] emphasizes recall, thus assessing how much overlap exists between the generated caption and the reference caption. It has gained prominence in various applications, particularly summarization and translation.

While these metrics capture surface-level similarity, they rely primarily on n-gram overlap, which can fall short in assessing deeper semantic correspondence.

7.2. Embedding-Based Evaluation

Embedding-based testing is an approach that leverages high-dimensional, fixed-length vector representations of words to compute semantic similarity. Unlike the legacy n-gram overlap measures, embedding-based approaches are able to capture deeper linguistic nuances by embedding sentences into a continuous vector space, in which semantically similar texts are positioned close to each other.

Methodologically, this process involves two key steps:

1. **Vectorization**: Each sentence or caption is transformed into a dense vector using a pre-trained embedding model. This vector encapsulates semantic features of the text, such as context, syntax, and meaning.

2. **Similarity Computation**: Once the text is embedded, the semantic similarity between two sentences is quantified by computing the cosine similarity between their respective vectors. Cosine similarity is defined as:

$$\text{sim}(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \, \|\mathbf{v}\|} \tag{7.1}$$

   where $\mathbf{u}$ and $\mathbf{v}$ are the embedding vectors for the two sentences, $\mathbf{u} \cdot \mathbf{v}$ denotes their dot product, and $\|\mathbf{u}\|$ and $\|\mathbf{v}\|$ represent the Euclidean norms of $\mathbf{u}$ and $\mathbf{v}$ respectively. This measure provides a normalized score between -1 and 1, with 1 indicating perfect semantic equivalence. The results are then bounded between 0 and 1.

Before describing the specific models adopted, it is helpful to highlight some general

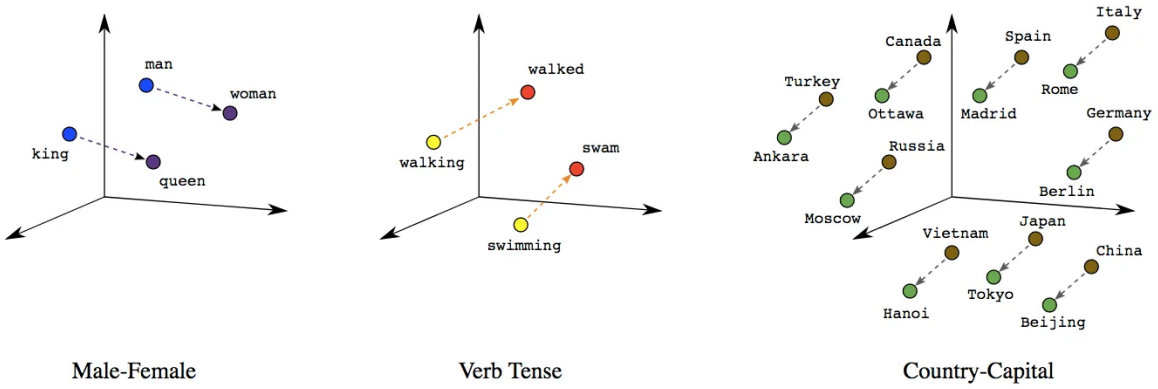


Figure 7.1: Word embeddings capturing semantic relationships. These examples demonstrate how word vectorization encodes meaningful linguistic structure in a shared vector space.

characteristics of embedding-based methods:

- **Dimensionality**: Text embeddings are fixed-length vectors whose dimensionality varies by model. Higher dimensions may capture richer semantic details at the cost of increased computational resources.
- **Preprocessing**: The quality of embeddings can be sensitive to preprocessing steps such as stop-word removal, lowercasing, and token normalization, which help reduce noise.
- **Similarity Computation**: As detailed above, the cosine similarity (or a normalized dot product) is typically used to quantify the semantic closeness between two text embeddings.

Two embedding models were employed in our evaluation:

- **all-MiniLM-L6-v2**: Developed by the Sentence-Transformers team, this model produces 384-dimensional embeddings. It is well-suited for fast inference and robust sentence-level semantic capture, especially in resource-constrained environments.
- **NV-Embed-v2**: A larger, more sophisticated model [34] that not only produces embeddings but also incorporates a query-based mechanism. Configured with a specific prompt, NV-Embed-v2 outputs 4096-dimensional embeddings, offering a richer representation critical for fine-grained similarity evaluation.

7.2.1. Prompt Engineering

In our experiments with NV-Embed-v2, extensive prompt engineering was performed to identify the best prompt for evaluating semantic similarity between captions. We computed an average similarity score for each prompt configuration by considering only unambiguous cases, that means samples where the generated caption and the reference caption were unequivocally correct. For each of these pairs, we obtained their embeddings using the respective prompt and then calculated the similarity score via the normalized dot product. The final average score for a given prompt was determined by taking the mean of these similarity scores. Since we expected the similarity score to approach 1 in cases of perfect semantic alignment, a higher average score indicates that the prompt is more effective. Table 7.1 summarizes these averages in descending order, and the prompts with the highest average scores were chosen for further experiments.

Prompt	Average Score
`Assess the extent to which the two sentences convey the same meaning`	0.7378
`Evaluate the degree to which the two sentences share the same meaning`	0.7366
`Given two sentences, determine how semantically similar they are`	0.7310
`Given two image captions, output a semantic similarity score between 0 and 1, where 0 means no similarity and 1 means complete semantic equivalence`	0.7137
`Given two image captions, evaluate their semantic similarity with a score between 0 and 1. Consider whether both captions capture the same key elements--such as the presence of tampering, the correct identification of inserted objects, and the overall descriptive details. A score of 1 indicates that the captions are semantically identical, while a score of 0 indicates no semantic overlap`	0.6851
`Given two sentences about an image's authenticity and modifications, determine whether they agree on the genuinity of the image and whether they refer to the same inserted object`	0.6676

Table 7.1: Average similarity scores for various NV-Embed-v2 prompts. Based on these results, the top three prompts (Prompt A, Prompt B, and Prompt C) were selected for further experiments.

Based on the results in Table 7.1, the three best-performing prompts are:

1. **Prompt A**: `"Assess the extent to which the two sentences convey the same meaning"`.

2. **Prompt B**: `"Evaluate the degree to which the two sentences share the same meaning"`.
3. **Prompt C**: `"Given two sentences, determine how semantically similar they are"`.

Figure 7.2 displays histograms of similarity scores for correctly classified cases (i.e. tampered images accurately predicted as tampered, and genuine images accurately predicted as genuine). These visualizations demonstrate that NV-Embed-v2 yields consistently better scores, as its average score is higher than all the other methods.

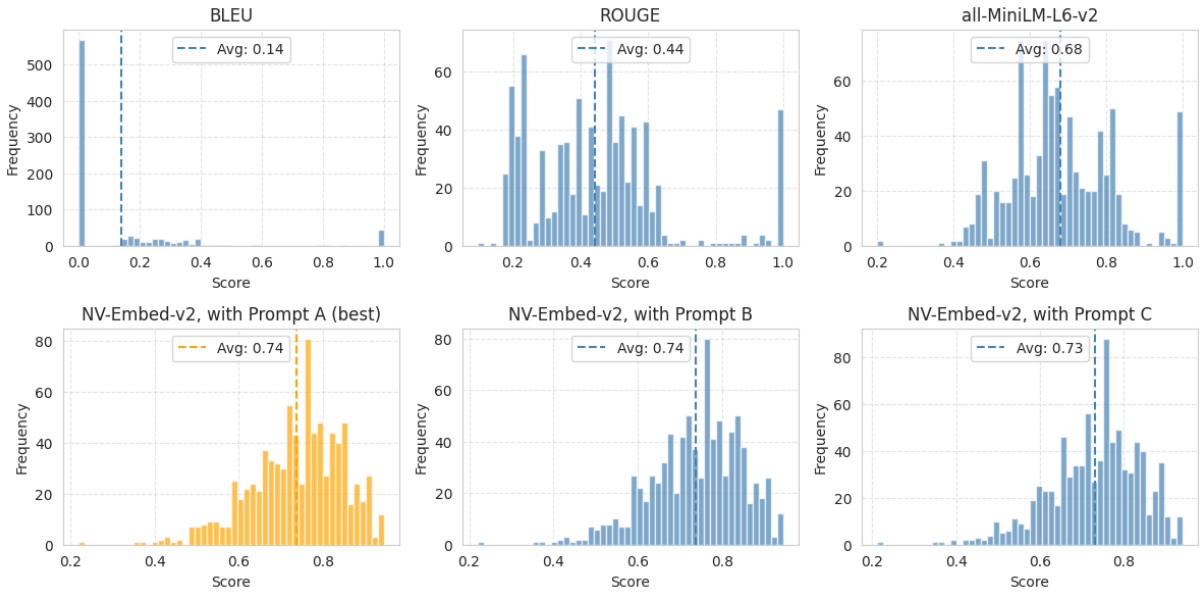


Figure 7.2: Histograms showing the score distribution for each method, considering only True Positives and True Negatives.

7.2.2. Threshold Optimization

Having established that NV-Embed-v2 outperforms the other evaluation methods, we proceeded to compare the different versions of the embedding model with the three different prompt configurations (denoted as Prompt A, Prompt B, and Prompt C) we previously used. For each prompt, we conducted a greedy search over thresholds ranging from 0 to 1 (with an increment of 0.01) to determine the one that maximized the F1 score when classifying the model's outputs. This threshold represents the similarity score at which the embedding-based classifier best distinguishes between correct and incorrect predictions.

Once the optimal threshold was determined for each prompt configuration, we computed the corresponding confusion matrices. In these matrices:

- **True Positives (TP)** are the cases where the generated caption is correct and the embedding model correctly categorizes it as such (i.e., the similarity score exceeds the threshold).
- **True Negatives (TN)** correspond to the cases where the generated caption is incorrect and the model correctly categorizes it as incorrect (i.e., the similarity score falls below the threshold).
- **False Positives (FP)** occur when an incorrect caption is mistakenly classified as correct by the model.
- **False Negatives (FN)** are instances where a correct caption is misclassified as incorrect.

Figure 7.3 displays these confusion matrices for the three prompt configurations. By analyzing these matrices, we can clearly observe how effectively the chosen threshold separates the two classes, thereby providing a quantitative validation of our meta-classification approach. The evaluation metrics derived from these confusion matrices are summarized in Table 7.2:

Metric	Prompt A	Prompt B	Prompt C
Accuracy	0.9630	0.9620	0.9630
Precision	0.9714	0.9757	0.9714
Recall	0.9861	0.9803	0.9861
F1 Score	0.9787	0.9780	0.9787

Table 7.2: Evaluation metrics for NV-Embed-v2 using the three selected prompts.

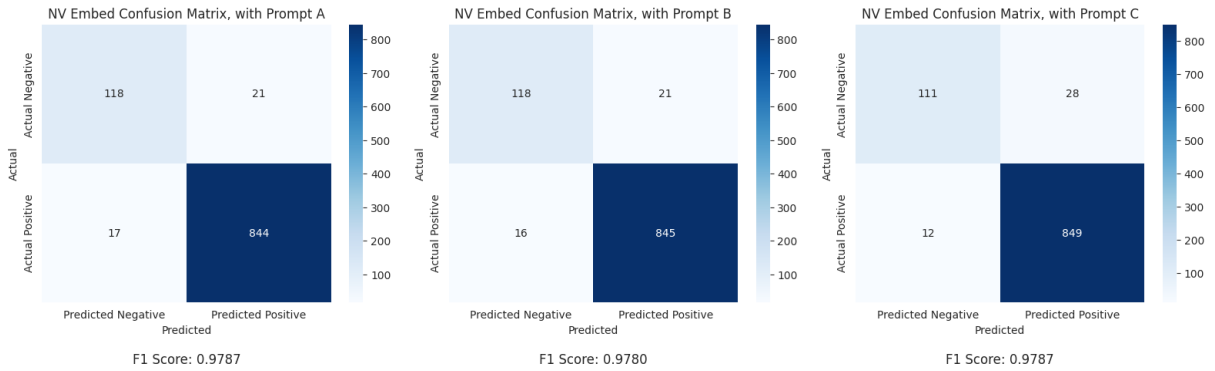


Figure 7.3: Confusion matrices for NV-Embed-v2 using different prompt configurations.

The results indicate that, while all three prompt configurations achieve excellent perfor-

mance, their metrics are very similar. To further validate our threshold optimization, we present the scatter plot for the best-performing prompt configuration in Figure 7.4. In this scatter plot, data points are color-coded as follows:

- **Green**, for correct predictions (true positives with correct identification).
- **Orange**, for correct predictions (true positives), but misinterpreted inserted object.
- **Red**, for incorrect predictions (false positives and false negatives).

The clear separation observed in the scatter plot corroborates the reliability of our chosen threshold and the overall meta-classification approach.

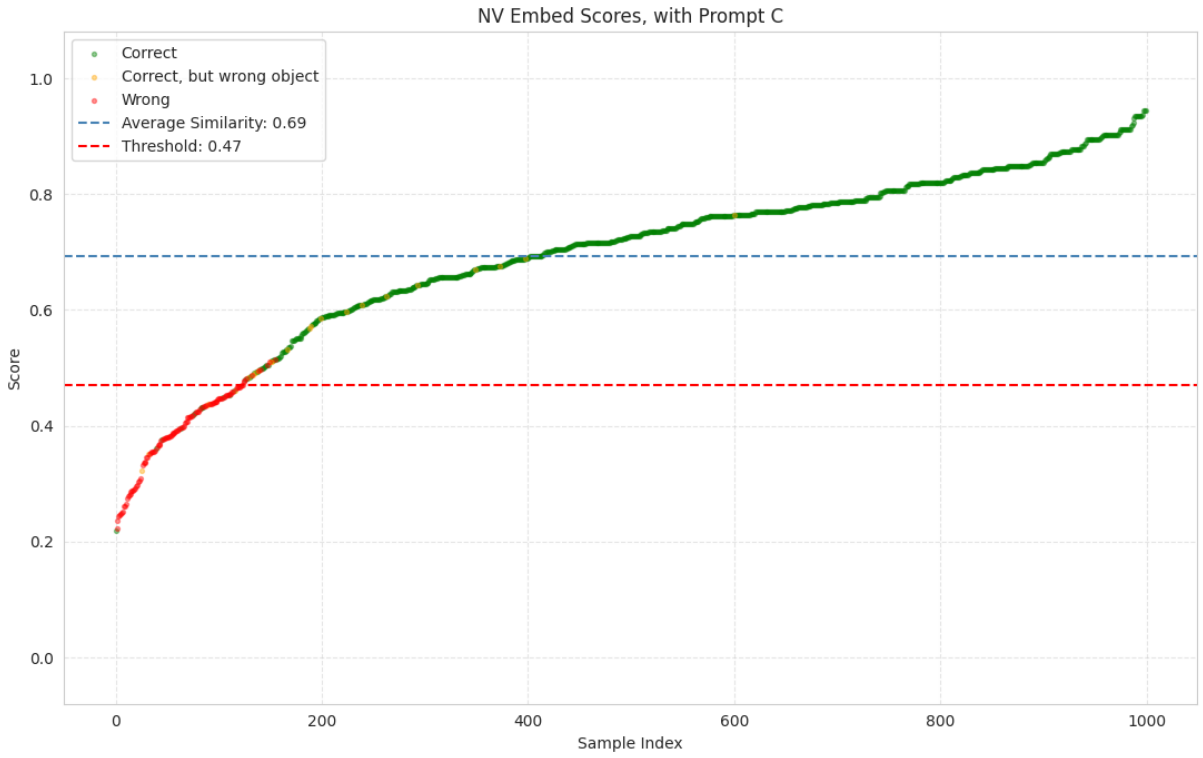


Figure 7.4: Scatter plot visualizing NV-Embed-v2 classification predictions. Points above the red threshold line are classified as valid, while those below are marked as invalid.

7.3. Text Preprocessing Considerations

In Natural Language Processing, stop-words are common words (e.g. "the", "is", "at", etc.) that are generally considered to give little to no semantic contribution to a phrase. As a result, stop-word removal is a widely used pre-processing step, intended to reduce noise and help algorithms focus on the more meaningful components of the text.

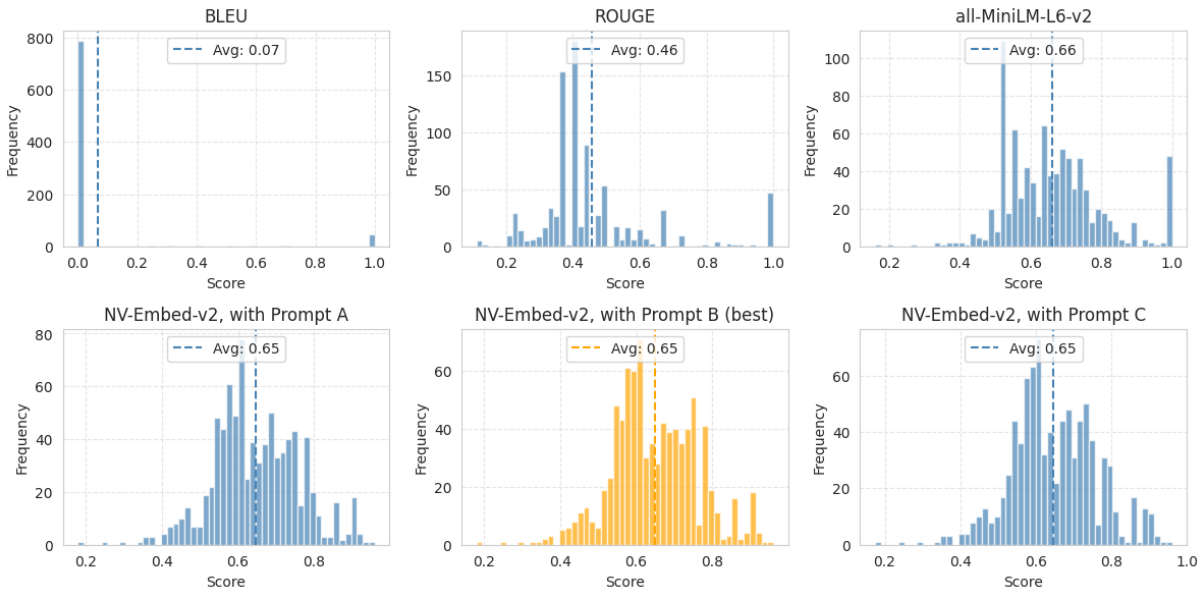


Figure 7.5: Histograms showing the score distribution for each method, considering only True Positives and True Negatives, after stop-word removal.

Typically, it is expected that removing stopwords will help the model to focus on meaningful words that contribute to the core meaning of the text. Yet, in our experiments, with stop-word removal we found that the performance was actually adversely affected. What we found out is that each method performed worse than the baseline using the full unaltered text. The histogram provided indicates this fact, suggesting that for our specific site, leaving the stop-words inside our texts could benefit the overall performance of our methods.

7.4. LLMs as Judges

To complement our quantitative metrics, we employed an LLM-based evaluation to obtain a human-like judgment of semantic similarity between generated captions. For this purpose, we used *Llama-3.1-8B-Instruct* for text generation, configuring it as an expert judge. The model was provided with two sentences—the generated caption and the corresponding reference—and was prompted to assign a similarity score. The prompt used was as follows:

```
"You are an expert judge tasked with assessing the similarity between
two sentences that describe a potentially manipulated image.  Your job
is to assign a similarity score from 0 to 2 based on how closely both
sentences describe the same manipulated element in the image.  Please
```

```
note that a score of 0 should only be given if one sentence indicates
that the image is manipulated while the other does not.  If both sentences
indicate manipulation but describe the manipulated element differently,
assign a score of 1.  If both sentences indicate manipulation (or both
indicate no manipulation) and they refer to the same manipulated element
despite minor wording differences, assign a score of 2.

Return only your answer in the format 'Score:  X/2' with no additional
commentary."
```

In each evaluation round, the LLM was given a pair of sentences to compare, and it returned a discrete score of 0, 1, or 2. We set the threshold for a correct prediction at a score of 1. This threshold is designed to count cases where both sentences indicate manipulation—even if the inserted object is misinterpreted—as correct (i.e., a true positive with an incorrect object identification).

To integrate these discrete scores with our other evaluation methods, we normalized the outputs to a scale of 0 to 1 by dividing the score by 2. This normalization allowed us to directly compare the LLM judgments with the continuous similarity scores obtained from embedding-based methods. The normalized scores were then used to generate confusion matrices and scatter plots, analogous to those produced for NV-Embed-v2, providing visual insight into the distribution of judgments. Figure 7.6 illustrates a scatter plot of the normalized LLM scores, with data points color-coded as described in Section 7.2.2.

The performance metrics (accuracy, precision, recall, and F1 score) computed from the LLM-based evaluation align well with our embedding-based and conventional NLP metrics, reinforcing the reliability of our meta-classification framework, as can be noted in Table 7.3. This LLM approach not only provides a coarse yet valuable qualitative assessment but also bridges the gap between automated metrics and human-like evaluation, enhancing our overall analysis of the caption generation system.

Metric	**Value**
Accuracy	0.9530
Precision	0.9963
Recall	0.9489
F1 Score	0.9720

Table 7.3: Evaluation metrics for Llama-as-a-Judge.

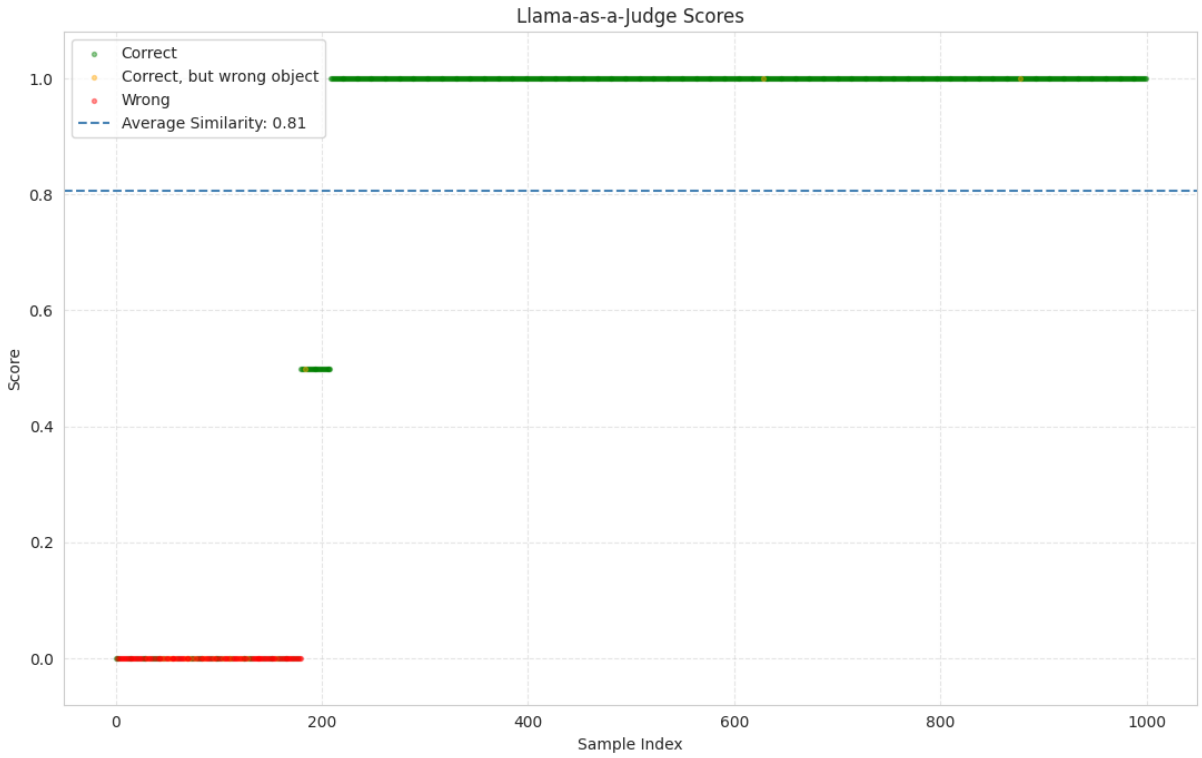


Figure 7.6: Scatter plot of normalized LLM judge scores. Each point represents a caption pair evaluated by Llama, normalized. Like in the scatter plot of NV Embed, above the red threshold line points are classified as valid, while those below are marked as invalid.

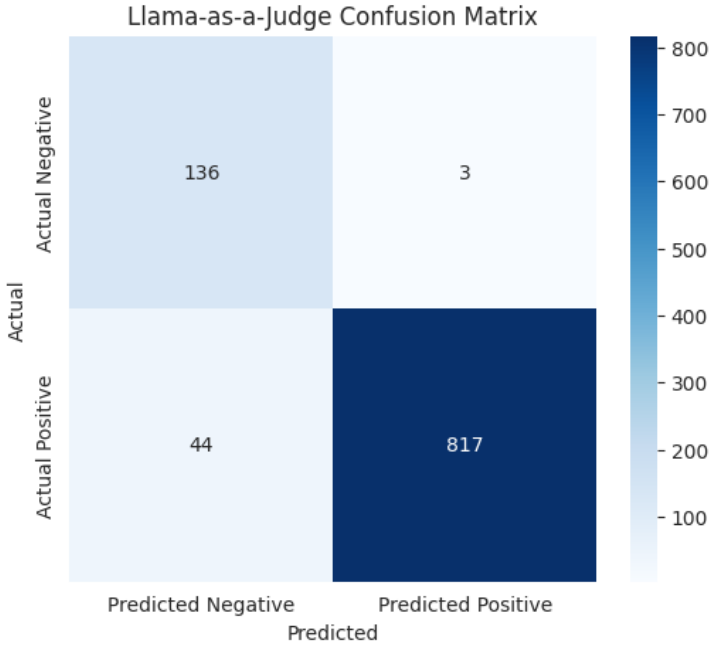


Figure 7.7: Confusion matrix for Llama-as-a-Judge.

7.5. Final Evaluation

The final evaluation metrics, summarized in Table 7.4, demonstrate the robust performance of our methods. NV-Embed-v2 with the selected prompts achieves high accuracy, precision, recall, and F1 scores, while the LLM-as-a-Judge approach offers valuable qualitative insights that complement the embedding-based evaluations. These findings validate our hybrid evaluation framework and provide a solid foundation for further advancements in automated caption analysis and forensic deepfake detection.

Approach	**Accuracy**	**Precision**	**Recall**	**F1 Score**
NV-Embed-v2, Prompt A	0.9630	0.9714	0.9861	0.9787
NV-Embed-v2, Prompt B	0.9620	0.9757	0.9803	0.9780
NV-Embed-v2, Prompt C	0.9630	0.9714	0.9861	0.9787
Llama-as-a-Judge	0.9530	0.9963	0.9489	0.9720

Table 7.4: Evaluation metrics for NV-Embed-v2 using the three selected prompts compared to Llama-as-a-Judge.

8 | Conclusions

The rapid advancement of fake technologies represents a critical challenge in the authenticity, cybersecurity, and ethical governance of digital media. In this thesis, we explored a novel multimodal approach leveraging vision-language models (specifically LLaVA OneVision) to enhance the detection and interpretability of manipulated images through a vision-language approach. Our research concentrated mainly on the splicing method, illustrating how fine-tuning multimodal large language models greatly improves both the detection accuracy and the interpretability of explanations produced by the model.

Our research also highlighted the significance of creating a well-annotated and diverse dataset since our custom dataset proved to be highly effective in enhancing the model's accuracy and robustness in the face of splicing-based tampering attacks. Through the careful annotation and diversification of the dataset with dynamic captions, we have exhibited that such organized data greatly contributes to model training, guaranteeing greater reliability and accuracy in real-world applications.

Despite these encouraging results, the work presented in this thesis is subject to several limitations. Primarily, our evaluation was restricted to splicing manipulations within static images. Expanding the scope of the dataset to include additional tampering techniques such as copy-move, in-painting, and adversarial manipulations would substantially improve the generalizability of detection algorithms. Additionally, the incorporation of multi-image reasoning capabilities into the model architecture would address current limitations and significantly enhance forensic analysis.

Looking ahead, there are many promising directions in which this work might be extended. Firstly, examining **other types of digital tampering** besides splicing (e.g., copy-move and more advanced in-painting techniques) might enhance the robustness of multimodal detection models. Secondly, expanding our framework to cover **video deepfakes** is an important and timely direction for investigation, as video-based manipulations become increasingly more common and more sophisticated. Finally, the utilization of progress in multi-image reasoning and video processing in multimodal models may set the stage for the development of more advanced forensic systems that can **detect and explain deepfakes**

in real time, greatly reducing the threat of maliciously manipulated multimedia content.

In summary, this thesis has proved the effectiveness and potential of multimodal large language models in supporting deepfake detection and explainability. It paves the way for future research in the direction of enhancing robustness, generalizing capabilities to a variety of tampering methods, and eventually deploying these advanced systems in real-world forensic applications.

Bibliography

[1] J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds, R. Ring, E. Rutherford, S. Cabi, T. Han, Z. Gong, S. Samangooei, M. Monteiro, J. Menick, S. Borgeaud, A. Brock, A. Nematzadeh, S. Sharifzadeh, M. Binkowski, R. Barreira, O. Vinyals, A. Zisserman, and K. Simonyan. Flamingo: a visual language model for few-shot learning, 2022. URL https://arxiv.org/abs/2204.14198.

[2] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[3] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLOS ONE*, 10(7):1–46, 07 2015. doi: 10.1371/journal.pone.0130140. URL https://doi.org/10.1371/journal.pone.0130140.

[4] T. Baltrušaitis, C. Ahuja, and L.-P. Morency. Multimodal machine learning: A survey and taxonomy, 2017. URL https://arxiv.org/abs/1705.09406.

[5] D. Birla. Autoencoders. https://medium.com/@birla.deepak26/autoencoders-76bb49ae6a8f, 2017. Accessed: 2025-01-31.

[6] C. M. Bishop and N. M. Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

[7] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. Enriching word vectors with subword information, 2017. URL https://arxiv.org/abs/1607.04606.

[8] C. Brito. Pope francis' puffer jacket: The story behind the ai-generated viral image. CBS News, Online article, Mar. 2023. URL https://www.cbsnews.com/news/pope-francis-puffer-jacket-fake-photos-deepfake-power-peril-of-ai/.

[9] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger,

T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners, 2020. URL https://arxiv.org/abs/2005.14165.

[10] H. Chemerys. Deepfakes as a problem of modernity, 10 2023.

[11] X. Chen, X. Wang, S. Changpinyo, A. Piergiovanni, P. Padlewski, D. Salz, S. Goodman, A. Grycner, B. Mustafa, L. Beyer, A. Kolesnikov, J. Puigcerver, N. Ding, K. Rong, H. Akbari, G. Mishra, L. Xue, A. Thapliyal, J. Bradbury, W. Kuo, M. Seyedhosseini, C. Jia, B. K. Ayan, C. Riquelme, A. Steiner, A. Angelova, X. Zhai, N. Houlsby, and R. Soricut. Pali: A jointly-scaled multilingual language-image model, 2023. URL https://arxiv.org/abs/2209.06794.

[12] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation, 2014. URL https://arxiv.org/abs/1406.1078.

[13] C. Cortes. Support-vector networks. *Machine Learning*, 1995.

[14] D.-T. Dang-Nguyen, C. Pasquini, V. Conotter, and G. Boato. Raise: a raw images dataset for digital image forensics. In *Proceedings of the 6th ACM Multimedia Systems Conference*, MMSys '15, page 219–224, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450333511. doi: 10.1145/2713168.2713194. URL https://doi.org/10.1145/2713168.2713194.

[15] A. de Rancourt-Raymond and N. Smaili. The unethical use of deepfakes. *J. Financ. Crime*, 30(4):1066–1077, May 2023.

[16] B. Dolhansky, J. Bitton, B. Pflaum, J. Lu, R. Howes, M. Wang, and C. C. Ferrer. The deepfake detection challenge (dfdc) dataset, 2020. URL https://arxiv.org/abs/2006.07397.

[17] F. Doshi-Velez and B. Kim. Towards a rigorous science of interpretable machine learning, 2017. URL https://arxiv.org/abs/1702.08608.

[18] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021. URL https://arxiv.org/abs/2010.11929.

[19] A. Ephrat, I. Mosseri, O. Lang, T. Dekel, K. Wilson, A. Hassidim, W. T. Freeman, and M. Rubinstein. Looking to listen at the cocktail party: a speaker-independent

audio-visual model for speech separation. *ACM Transactions on Graphics*, 37(4): 1–11, July 2018. ISSN 1557-7368. doi: 10.1145/3197517.3201357. URL http://dx.doi.org/10.1145/3197517.3201357.

[20] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks, 2014. URL https://arxiv.org/abs/1406.2661.

[21] F. Guillaro, D. Cozzolino, A. Sud, N. Dufour, and L. Verdoliva. Trufor: Leveraging all-round clues for trustworthy image forgery detection and localization, 2023. URL https://arxiv.org/abs/2212.10957.

[22] D. Gunning, E. Vorm, J. Y. Wang, and M. Turek. Darpa's explainable ai (xai) program: A retrospective. *Applied AI Letters*, 2(4):e61, 2021. doi: https://doi.org/10.1002/ail2.61. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/ail2.61.

[23] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[24] L. A. Hendricks, K. Burns, K. Saenko, T. Darrell, and A. Rohrbach. Women also snowboard: Overcoming bias in captioning models (extended abstract), 2018. URL https://arxiv.org/abs/1807.00517.

[25] A. Hidaka and T. Kurita. Consecutive dimensionality reduction by canonical correlation analysis for visualization of convolutional neural networks. In *Proceedings of the ISCIE International Symposium on Stochastic Systems Theory and its Applications*, volume 2017, pages 160–167, 12 2017. doi: 10.5687/sss.2017.160.

[26] S. Hochreiter. Long short-term memory. *Neural Computation MIT-Press*, 1997.

[27] A. Holzinger, G. Langs, H. Denk, K. Zatloukal, and H. Müller. Causability and explainability of artificial intelligence in medicine. *WIREs Data Mining and Knowledge Discovery*, 9(4):e1312, 2019. doi: https://doi.org/10.1002/widm.1312. URL https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/widm.1312.

[28] T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks, 2019. URL https://arxiv.org/abs/1812.04948.

[29] M. Kazemi, N. Dikkala, A. Anand, P. Devic, I. Dasgupta, F. Liu, B. Fatemi, P. Awasthi, D. Guo, S. Gollapudi, and A. Qureshi. Remi: A dataset for reasoning with multiple images, 2024. URL https://arxiv.org/abs/2406.09175.

[30] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization, 2017. URL https://arxiv.org/abs/1412.6980.

[31] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. Burges, L. Bottou, and K. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. URL https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

[32] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791.

[33] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *nature*, 521(7553):436, 2015.

[34] C. Lee, R. Roy, M. Xu, J. Raiman, M. Shoeybi, B. Catanzaro, and W. Ping. Nv-embed: Improved techniques for training llms as generalist embedding models, 2025. URL https://arxiv.org/abs/2405.17428.

[35] B. Li, Y. Zhang, D. Guo, R. Zhang, F. Li, H. Zhang, K. Zhang, P. Zhang, Y. Li, Z. Liu, and C. Li. Llava-onevision: Easy visual task transfer, 2024. URL https://arxiv.org/abs/2408.03326.

[36] J. Li, D. Li, C. Xiong, and S. Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation, 2022. URL https://arxiv.org/abs/2201.12086.

[37] Y. Li, M.-C. Chang, and S. Lyu. In ictu oculi: Exposing ai generated fake face videos by detecting eye blinking, 2018. URL https://arxiv.org/abs/1806.02877.

[38] C.-Y. Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://aclanthology.org/W04-1013/.

[39] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár. Microsoft coco: Common objects in context, 2015. URL https://arxiv.org/abs/1405.0312.

[40] Z. C. Lipton. The mythos of model interpretability, 2017. URL https://arxiv.org/abs/1606.03490.

[41] M. Llama. Llama 3.2-11b vision instruct. https://huggingface.co/meta-llama/Llama-3.2-11B-Vision-Instruct, 2024. Accessed: March 7, 2025.

[42] S. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions, 2017. URL https://arxiv.org/abs/1705.07874.

[43] F. Marra, D. Gragnaniello, L. Verdoliva, and G. Poggi. Do gans leave artificial fingerprints?, 2018. URL https://arxiv.org/abs/1812.11842.

[44] M. Masood, M. Nawaz, K. Malik, A. Javed, A. Irtaza, and H. Malik. Deepfakes generation and detection: state-of-the-art, open challenges, countermeasures, and way forward. *Applied Intelligence*, 53:1–53, 06 2022. doi: 10.1007/s10489-022-03766-z.

[45] T. Mittal, U. Bhattacharya, R. Chandra, A. Bera, and D. Manocha. Emotions don't lie: An audio-visual deepfake detection method using affective cues, 2020. URL https://arxiv.org/abs/2003.06711.

[46] C. Molnar. Interpretable machine learning-a guide for making black box models explain-443 able, 2019.

[47] G. Montavon, W. Samek, and K.-R. Müller. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, 73:1–15, Feb. 2018. ISSN 1051-2004. doi: 10.1016/j.dsp.2017.10.011. URL http://dx.doi.org/10.1016/j.dsp.2017.10.011.

[48] T. Myers. Ai is changing reality: Revealing the history of deepfakes. Facia AI Blog, 2024. URL https://facia.ai/blog/ai-is-changing-the-reality-revealing-the-history-of-deepfakes/. Accessed 28 January 2025.

[49] A. Nagrani, J. S. Chung, and A. Zisserman. Voxceleb: A large-scale speaker identification dataset. In *Interspeech 2017*, interspeech_2017. ISCA, Aug. 2017. doi: 10.21437/interspeech.2017-950. URL http://dx.doi.org/10.21437/Interspeech.2017-950.

[50] C. Olah. Understanding lstm networks. colah's blog, 2015. URL https://colah.github.io/posts/2015-08-Understanding-LSTMs/. Accessed: 29 January 2025.

[51] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311–318, USA, 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073135. URL https://doi.org/10.3115/1073083.1073135.

[52] L. Payne. deepfake. Encyclopedia Britannica, Dec. 2024. URL https://www.britannica.com/technology/deepfake. Accessed 28 January 2025.

[53] G. Pei, J. Zhang, M. Hu, Z. Zhang, C. Wang, Y. Wu, G. Zhai, J. Yang, C. Shen, and D. Tao. Deepfake generation and detection: A benchmark and survey, 2024. URL https://arxiv.org/abs/2403.17881.

[54] J. Pennington, R. Socher, and C. Manning. GloVe: Global vectors for word representation. In A. Moschitti, B. Pang, and W. Daelemans, editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, Oct. 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1162. URL https://aclanthology.org/D14-1162/.

[55] I. Perov, D. Gao, N. Chervoniy, K. Liu, S. Marangonda, C. Umé, Dpfks, C. S. Facenheim, L. RP, J. Jiang, S. Zhang, P. Wu, B. Zhou, and W. Zhang. Deepfacelab: Integrated, flexible and extensible face-swapping framework, 2021. URL https://arxiv.org/abs/2005.05535.

[56] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning transferable visual models from natural language supervision, 2021. URL https://arxiv.org/abs/2103.00020.

[57] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever. Zero-shot text-to-image generation, 2021. URL https://arxiv.org/abs/2102.12092.

[58] M. T. Ribeiro, S. Singh, and C. Guestrin. "why should i trust you?": Explaining the predictions of any classifier, 2016. URL https://arxiv.org/abs/1602.04938.

[59] A. Rogers, O. Kovaleva, and A. Rumshisky. A primer in bertology: What we know about how bert works, 2020. URL https://arxiv.org/abs/2002.12327.

[60] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models, 2022. URL https://arxiv.org/abs/2112.10752.

[61] C. Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead, 2019. URL https://arxiv.org/abs/1811.10154.

[62] D. Sarkar. Decentralized deepfake detection blockchain network using dynamic algorithm management, 2023. URL https://arxiv.org/abs/2311.18545.

[63] K. J. Schiff, D. S. Schiff, and N. Bueno. The Liar's Dividend: The Impact of Deepfakes and Fake News on Trust in Political Discourse. SocArXiv x43ph, Center for

Open Science, Aug. 2023. URL https://ideas.repec.org/p/osf/socarx/x43ph.html.

[64] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 128(2):336–359, Oct. 2019. ISSN 1573-1405. doi: 10.1007/s11263-019-01228-7. URL http://dx.doi.org/10.1007/s11263-019-01228-7.

[65] R. Sennrich, B. Haddow, and A. Birch. Neural machine translation of rare words with subword units, 2016. URL https://arxiv.org/abs/1508.07909.

[66] P. Sharma, N. Ding, S. Goodman, and R. Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In I. Gurevych and Y. Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2556–2565, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1238. URL https://aclanthology.org/P18-1238/.

[67] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions, 2014. URL https://arxiv.org/abs/1409.4842.

[68] E. Tahir. Dis100k. https://www.kaggle.com/datasets/erentahir/dis100k, 2024. Data set on Kaggle. Retrieved February 6, 2025.

[69] M. Tan and Q. V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks, 2020. URL https://arxiv.org/abs/1905.11946.

[70] J. Thies, M. Zollhöfer, M. Stamminger, C. Theobalt, and M. Nießner. Face2face: Real-time face capture and reenactment of rgb videos, 2020. URL https://arxiv.org/abs/2007.14808.

[71] P. Tinsley, A. Czajka, and P. Flynn. This face does not exist ... but it might be yours! identity leakage in generative models, 2020. URL https://arxiv.org/abs/2101.05084.

[72] Y.-H. H. Tsai, S. Bai, P. P. Liang, J. Z. Kolter, L.-P. Morency, and R. Salakhutdinov. Multimodal transformer for unaligned multimodal language sequences, 2019. URL https://arxiv.org/abs/1906.00295.

[73] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalch-

brenner, A. Senior, and K. Kavukcuoglu. Wavenet: A generative model for raw audio, 2016. URL https://arxiv.org/abs/1609.03499.

[74] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need, 2023. URL https://arxiv.org/abs/1706.03762.

[75] S. Wachter, B. Mittelstadt, and C. Russell. Counterfactual explanations without opening the black box: Automated decisions and the gdpr, 2018. URL https://arxiv.org/abs/1711.00399.

[76] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, Q. Le, Y. Agiomyrgiannakis, R. Clark, and R. A. Saurous. Tacotron: Towards end-to-end speech synthesis, 2017. URL https://arxiv.org/abs/1703.10135.

[77] T.-H. Wu, G. Biamby, , J. Quenum, R. Gupta, J. E. Gonzalez, T. Darrell, and D. M. Chan. Visual haystacks: Answering harder questions about sets of images. *arXiv preprint arXiv:2407.13766*, 2024.

[78] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, Łukasz Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, and J. Dean. Google's neural machine translation system: Bridging the gap between human and machine translation, 2016. URL https://arxiv.org/abs/1609.08144.

[79] Z. Wu, X. Chen, Z. Pan, X. Liu, W. Liu, D. Dai, H. Gao, Y. Ma, C. Wu, B. Wang, Z. Xie, Y. Wu, K. Hu, J. Wang, Y. Sun, Y. Li, Y. Piao, K. Guan, A. Liu, X. Xie, Y. You, K. Dong, X. Yu, H. Zhang, L. Zhao, Y. Wang, and C. Ruan. Deepseek-vl2: Mixture-of-experts vision-language models for advanced multimodal understanding, 2024. URL https://arxiv.org/abs/2412.10302.

[80] J. Yan, Z. Li, Z. He, and Z. Fu. Generalizable deepfake detection via effective local-global feature extraction, 2025. URL https://arxiv.org/abs/2501.15253.

[81] P. Young, A. Lai, M. Hodosh, and J. Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78, 2014. doi: 10.1162/tacl_a_00166. URL https://aclanthology.org/Q14-1006/.

[82] B. Zhao, Y. Zong, L. Zhang, and T. Hospedales. Benchmarking multi-image understanding in vision and language models: Perception, knowledge, reasoning, and multi-hop reasoning, 2024. URL https://arxiv.org/abs/2406.12742.

A | Appendix A

A.1. Preliminary Approach to Multimodal Models

In this chapter, we will describe the preliminary approach used to detect image-based deepfakes using *LLaVA OneVision*, a recent vision-language model that combines large language model capabilities with visual understanding. We begin by explaining why we chose LLaVA and why our focus is on image tampering rather than video. We then discuss our initial zero-shot experiments on splicing, copy-move, and in-painting, detailing how LLaVA reacted to each type of manipulation. Finally, we consider integrating techniques like *Noiseprint++* to enhance detection, reflecting on the insights (and challenges) from our preliminary trials.

A.1.1. LLaVA OneVision

Large Language and Vision Assistant (LLaVA) models, notably the OneVision branch [35], have gained some notoriety for their ability to make sense of textual and image inputs across modalities. Unlike text-only based systems, LLaVA first reads visual features from its built vision backbone, then generates or recognizes outputs with the help of the language model. OneVision (0.5B parameters) was selected for two major reasons:

1. **Scalability:** OneVision is one of the smaller models at a parameter count of 0.5B amongst some mainstream billion-scale ones, thereby allowing it an easy run on our GPU hardware with a reduced effort towards setting up complicated parallelization.

2. **Recent Performance:** Initial benchmarks indicate that LLaVA does not lag behind in image-based question answering and captioning problems. We hypothesized that its robust vision-language alignments could be exploited for the purpose of manipulations detection even in zero-shot conditions.

While other vision-language models (e.g., PaLI [11] or Flamingo [1]) would also do well, we settled on LLaVA OneVision for its ease of integration and the very promising initial results.

A.1.2. Focusing on Images Rather than Video

Deepfake detection often targets videos, as many high-profile cases involve animated face swaps or lip-syncing. However, video-based deepfake analysis requires significantly more computational horsepower, especially if you plan to inspect each frame. Since our GPU resources were limited and our prior expertise lay in image processing, we decided to narrow our experiments to *static* image tampering. This approach streamlined our workflow and let us concentrate on evaluating LLaVA's vision-language abilities without the overhead of video processing.

A.1.3. Zero-Shot Experiments

We ran LLaVA OneVision in zero-shot mode, feeding it tampered images and prompting it to describe or assess authenticity. Our dataset covered three core manipulation types:

Splicing Splicing involves cutting a region from one image and pasting it into another. The pasted region can introduce lighting or perspective mismatches, which might help an AI model spot anomalies.

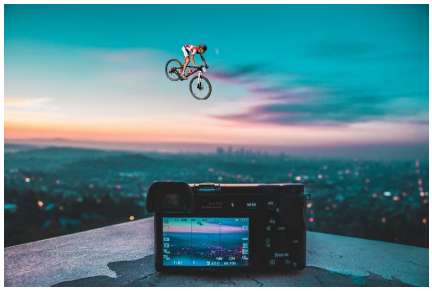
Copy-Move Copy-move duplicates a region within the same image. Because the source and target patches come from the same photo, color and lighting differences are often minimal, making it trickier for simple detectors to notice.

In-Painting In-painting removes or modifies a region, typically filling the gap with AI-generated textures, or context-aware fill. Modern in-painting methods can be highly seamless, leaving few traces of the original boundary.

When asked to describe or judge authenticity, LLaVA OneVision generally handled *splicing* more successfully than *copy-move* or *in-painting*. We suspect that spliced regions can introduce sharper edges or contextual mismatches, which a vision-language model might latch onto. In contrast, copy-move typically maintains the same texture and lighting, making it less obvious. In-painting can also be deceptive if the content was generated or blended smoothly.

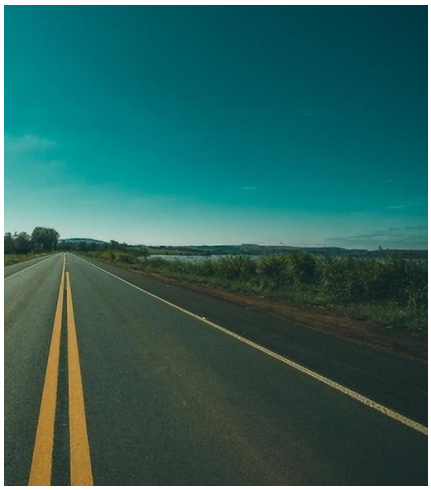
Even so, results were far from perfect in zero-shot mode, indicating a need for additional training or domain-specific cues to refine the model's accuracy.

Tampering Technique: Splicing
Description: *The image appears to be a composite, combining elements of a mountain and a fish. The fish is superimposed onto the mountain, which is not a natural occurrence. This kind of manipulation is often done for artistic or humorous purposes, but it's not a typical occurrence in nature.*
Comments: Correct.



Tampering Technique: Splicing
Description: *Yes, the image has been manipulated. The person in the image appears to be a real person, but the bicycle and the background have been altered to create a surreal and artistic effect. The person is depicted in mid-air, suggesting a moment of action or performance, while the background shows a cityscape.*
Comments: While the image is manipulated, the bike has been spliced in, rather than both bike and background being fully altered.

Tampering Technique: None
Description: *The image appears to be a natural photograph with no obvious signs of manipulation. The colors, lighting, and composition are consistent with a real-life scene. However, the image quality might be enhanced for aesthetic purposes, such as to make it look more vibrant or to emphasize certain elements like the boat or the...*
Comments: Correct.



Tampering Technique: None
Description: *The image appears to be a photograph of a road, not a manipulated image. The absence of any visible alterations or distortions suggests that the image is a straightforward representation of a road scene.*
Comments: Correct.

Tampering Technique: Splicing
Description: *The image appears to be a composite, likely created by a digital editing software. This is suggested by the seamless transition between the beach and the cityscape, as well as the uniformity of the buildings and the ship's appearance. The image does not exhibit any obvious signs of manipulation, such as distort...*
Comments: Incorrect, as the ship has been spliced into the picture.

Tampering Technique: In-painting
Description: *The image appears to be a natural photograph with no apparent signs of manipulation. The objects, such as the Frisbee and the pile of rocks, are in their natural state, and the lighting and shadows suggest it was taken during the day.*
Comments: The model didn't considered the in-painting of a person, which has been removed from the scene, altering the original content despite the natural appearance.

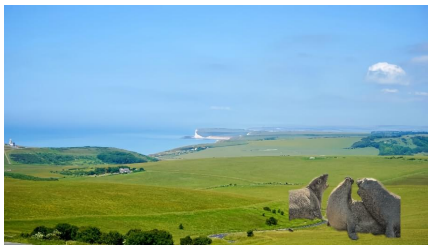
Tampering Technique: Copy-move
Description: *The image appears to be a real photograph, not manipulated. The scene is a typical retail environment with a large number of magazines and newspapers displayed in a way that suggests a busy shopping area. The people are engaged in typical shopping activities, and the layout is consistent with a typical newsstand or a section...*
Comments: The response misses the fact that some shelves have been copied and pasted elsewhere, altering the original layout of the library.

Tampering Technique: Splicing
Description: *The image appears to be a photograph of a real scene, not a manipulated image. The woman's pose and the scattered items suggest a casual, everyday setting, possibly a street or a public area. The absence of any obvious signs of manipulation or alteration to the image's content makes it difficult to determine...*
Comments: The response misses the fact that the woman has been spliced into the scene. The lack of a shadow and visible cut marks make the manipulation noticeable.

Tampering Technique: None
Description: *The image appears to be a natural photograph with no apparent signs of manipulation. It captures a scene from a European town with a classical building and a statue in the foreground. The buildings have a weathered look, and the statue is positioned in a way that suggests it is part of the town's historical...*
Comments: Correct.

Tampering Technique: Splicing
Description: *The image appears to have been manipulated, as it shows a slice of pizza with a distinct triangular shape, which is not a typical pizza slice shape. The pizza in the background seems to have a more irregular shape, possibly due to the angle of the shot or the way the pizza was cut. This...*
Comments: The response is incorrect. The issue isn't the shape of the slice, but the fact that a slice has been inserted into the image. Since pizza is typically cut in triangles, the reasoning behind the response doesn't make sense.

Tampering Technique: Copy-move
Description: *The image appears to be a straightforward photograph without any apparent signs of manipulation. It shows a corridor with a series of framed artworks on the wall, two individuals walking by, and a clear, unaltered view of the environment. The lighting and shadows are consistent with a real-life setting, and there...*
Comments: The response is incorrect—the image has been manipulated by copying and pasting an artwork from the right of the door to the left, altering the original scene.



Tampering Technique: Splicing
Description: *The image appears to have been manipulated, as the two large stone sculptures that are superimposed on the landscape seem to be superimposed onto the scene. This is a common technique used in digital art and photo manipulation to create a surreal or abstract effect. The manipulation does not alter the natural elements...*
Comments: The response is incorrect, as the manipulation involves inserting seals, not stone sculptures.

A.1.4. Exploring Noiseprint++ for Enhanced Detection

Noiseprint++ is a digital image forensics tool proposed by researchers at the University of Naples (UniNA). It extracts camera-specific noise patterns (often referred to as "fingerprints") and can highlight inconsistencies where such patterns are missing or altered. This leads to a "mask" indicating potential tampering regions.

The model Noiseprint++ is part of a more extensive forensics architecture called TruFor, one that has been designed for the purposes of generalized forgery detection [21]. TruFor encompasses various forensic techniques based both in deep learning for the extraction of features and on some classical handcrafted forensic procedures, thus improving image authenticity verification. By being instantiated in TruFor, Noiseprint++ is able to derive a reliable and comprehensible way to classify manipulated content, thus becoming a potent tool for the detection of deepfakes.

Our ultimate goal was to fine-tune Noiseprint++ so that it could more precisely spot visual anomalies, then feed both the tampered image and the Noiseprint++ output into LLaVA OneVision. In doing so, we hoped to guide the model's attention toward suspicious regions of the image, thereby increasing its accuracy in assessing whether a photo had been manipulated. However, in practice, we discovered that LLaVA (in zero-shot mode) struggled to handle two aligned images (the original plus its associated mask). For instance, the model often failed to link the pixels highlighted by Noiseprint++ to the corresponding regions in the original image.

This limitation relates to a broader challenge in vision-language models: *multi-image reasoning*. While current systems excel when given a single image paired with text, handling multiple images in parallel is significantly harder. Recent studies have begun addressing this gap:

- **ReMI: A Dataset for Reasoning with Multiple Images** [29] discusses how transformers trained primarily on single-image tasks can falter when asked to compare or integrate information from multiple pictures.
- **VHs: The Visual Haystacks Benchmark** [77] explores whether models are "ready for multi-image reasoning," highlighting the complexity of tasks that require cross-image comparison and reasoning.
- **Benchmarking Multi-Image Understanding in Vision and Language Models** [82] further examines perception, knowledge, multi-hop reasoning, and other dimensions of multi-image comprehension.

These works suggest that truly robust multi-image reasoning demands either specialized training protocols or architectural innovations that can fuse information from multiple visuals. In our case, adopting such methods would be necessary to ensure that Noiseprint++ masks genuinely inform LLaVA OneVision's understanding of where and how an image is tampered. We concluded that either an extensive fine-tuning or a custom extension to the model's architecture would likely be required to achieve reliable multi-image analysis in this forensic context.

A.1.5. Towards Fine-Tuning the Model

The analysis of multi-image reasoning caused us to consider infusing other forensic masks into the equation; however, we realized that an even more straightforward method would be to fine-tune LLaVA OneVision itself. Instead of making the model juggle separate inputs like the outputs of Noiseprint++ and the original image, we could train the vision encoder to find the "deepfake clues" independently. Such a process would unfold in a couple of steps.

List of Figures

List of Tables

Acknowledgments

Desidero ringraziare in primis il mio relatore, Mark James Carman, per la disponibilità, la grande guida, il supporto e l'entusiasmo mostrato durante la stesura di questa tesi. Ringrazio inoltre sinceramente i miei corelatori, Paolo Bestagini e Andrea Sassella, per la loro pazienza infinita, i consigli puntuali, la professionalità e per avermi guidato con competenza e attenzione durante il mio lavoro. Il vostro aiuto è stato fondamentale per affrontare le difficoltà incontrate lungo il percorso.

Ringrazio gli amici di Milano (anche se nessuno è davvero di Milano) conosciuti durante gli anni universitari, in particolar modo Alessio, Alessandro, Davide, Nikita e Simone. Con voi ho condiviso tante avventure, concerti e bivaccate, e spero che riusciremo sempre a beccarci per una birra al *Cheers* o per una mano a *Bang!*.

Ringrazio con affetto anche i miei amici di Brescia, Andrea e Teresa, ormai espatriati, la cui amicizia perdura dai primi anni di liceo, siete responsabili di diffondere un po' di brescianità nel mondo. Ringrazio anche le Salamandre di Bagnolo, che sono state parte integrante della mia adolescenza e della mia crescita personale. Grazie per essere sempre stati presenti con affetto nonostante la lontananza: dovunque sarò nel mondo, sappiate che ci sarò sempre per un torneo di briscolone.

Ringrazio di cuore anche tutta la mia famiglia, pilastro fondamentale della mia vita. In particolare, ringrazio i miei genitori e mia sorella Giulia per l'affetto costante, l'incoraggiamento incondizionato e il vostro sostegno nei momenti più complicati con comprensione e amore infinito.

Infine, un ringraziamento speciale, a Martina, amica, compagna e famiglia. Sono davvero felice e onorato di avere al mio fianco una persona così dolce ed empatica, genuina e trasparente, sempre disposta ad aiutare il prossimo anche a costo di mettersi da parte. Da te, ho imparato cosa significa amare davvero qualcuno. Spero di poter passare con te ancora tanti di questi giorni.