Comprehensive Analysis of Large Language Models on CNN-based Deepfake Detection

1st Toey Lui
Department of Computer Engineering
San José State University
CA, 95192 USA

2nd Quang Duy Tran
Department of Computer Science
San José State University
CA, 95192 USA

3rd Younghee Park
Department of Computer Engineering
San José State University
CA, 95192 USA

***Abstract*—Deepfake techniques are widely used for various purposes, including video augmentation; however, attackers exploit them for malicious goals such as fraud, scams, and phishing attacks. Although Convolutional Neural Network (CNN) is one of the most promising deep learning methods to detect deepfakes, it relies solely on visual feature information and lacks dynamic feedback. This paper proposes an effective multimodal deepfake detection system using Large Language Models (LLMs) and CNNs to identify deepfake video frames and improve detection rates. The proposed system combines visual and language modalities and generates different types of inputs by adding more realistic features and semantic context through prompt engineering with LLMs. Our approach analyzes deepfake videos frame by frame, generates LLM responses using task-specific prompts, transforms these responses into embeddings, and integrates them with CNN-derived embeddings for evaluation through deep learning. Experimental results show that incorporating LLM features into CNN-based models improves overall accuracy by 6.2% to 15.4%. These findings highlight the potential of LLMs in deepfake detection and demonstrate the effectiveness of a multimodal approach for advancing digital forensics.**

***Index Terms*—Deepfake detection, Large language models, Convolutional neural networks, Computer vision, Deep learning.**

I. Introduction

Deepfakes are highly realistic synthetic media that are digitally manipulated using deep learning and computer vision. They come in various forms, including videos, images, audio, and multimodal content. To generate deepfakes, Artificial Intelligence (AI) algorithms, most significantly Generative Adversarial Networks [1], are combined with deep learning and image processing techniques. Several automated manipulation methods have been developed, such as Deepfakes, FaceSwap, Face2Face [2], FaceShifter, and NeuralTextures, each achieving different performance depending on target use case, realism of generated faces, and computational cost. Deepfake techniques involve trade-offs in their application domains, but most of them have been used for malicious purposes that affect our lives. For instance, deepfakes can impersonate influential figures and spread misinformation to deceive vulnerable individuals, thereby threatening public trust, privacy, and security.

Dr. Park is the corresponding author of this work. This work is supported by NSF SaTC Award #2304753, Google CASHI-IRP project, and Silicon Valley Cybersecurity Institute.

Traditional deepfake detection methods have been extensively studied, but recent advances in AI have made deepfakes increasingly realistic and challenging to identify. While Convolutional Neural Network (CNN)-based approaches remain effective [3], [4], relying solely on visual features may be insufficient for advanced deepfake videos, as they do not account for contextual reasoning, coherence, and semantic information beyond pixel-level analysis. Incorporating Large Language Models (LLMs) [5] enables high-level semantic reasoning, helping identify inconsistencies that visual models overlook. To address these challenges, expanding deepfake detection beyond the visual modality by utilizing LLMs can strengthen and diversify detection perspectives.

This paper proposes an effective deepfake detection system that integrates CNNs with prompt engineering techniques in LLMs. The system consists of five main components: (1) generating LLM responses based on image frames extracted from deepfake videos, (2) converting LLM responses into embeddings, (3) extracting convolutional embeddings from CNN layers, (4) concatenating LLM and convolutional embeddings, and (5) evaluating both baseline and combined models using deep learning. This paper presents a multimodal embedding framework that combines visual and language embeddings to enhance deepfake classification, leveraging LLM prompts to capture both low-level features and semantic reasoning from image frames. Furthermore, we evaluate the impact of LLMs on CNN-based models by employing Multilayer Perceptron (MLP) [6] as a classifier and analyzing performance trends across different CNN architectures.

Using real deepfake datasets, this paper provides a comprehensive analysis of the role of LLMs in enhancing CNN-based deepfake detection. This work extends ongoing research on multimodal deepfake detection by combining LLM-derived embeddings with CNN visual features through an MLP-based classifier for deepfake detection. Our system introduces several key contributions:

1) We develop a novel multimodal embedding framework that concatenates visual and language embeddings to strengthen deepfake classification.
2) We examine the role of LLMs in deepfake detection by designing prompts that extract both reinforced low-level features and semantic reasoning from image frames.
3) We evaluate the influence of LLMs on CNN-based mod-

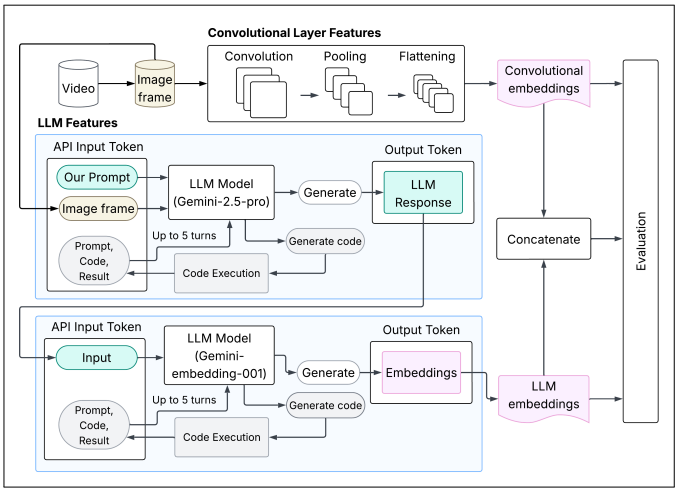

Fig. 1. System Architecture

els, detailing trends and relationships across different CNN architectures.

4) With real deepfake datasets, this paper demonstrates the trade-offs of the proposed system across various performance metrics, including accuracy, F1 score, precision, and recall.

II. Related Work

Traditional deepfake detection methods rely primarily on vision-based deep learning models. As deepfake manipulation advances, recent works have explored multimodal learning by combining visual understanding with audio or language modalities to enhance detection accuracy [7], [8]. While LLMs and Large Vision-Language Models [9] show strong potential for multimodal learning, relatively little research has been dedicated to integrating them into forgery detection [10]–[12]. Some recent approaches have begun to explore this integration. The DD-VQA task [13] employs BLIP [14], BERT [15], and Vision Transformer [16] for deepfake detection based on common sense reasoning, requiring high computational power. Another approach [17] uses dense CNN, multimodal fusion, and InstructBLIP [18] for effective multimodal deepfake detection. A comprehensive summary of related work is omitted due to space constraints.

III. Our Approach

A. Overview

Fig. 1 illustrates the system architecture diagram for our research. The diagram outlines the primary components and subcomponents of our system. The workflow begins with data preprocessing, in which image frames are extracted from videos and stored in a database. To obtain convolutional layer features, the extracted image frames are resized and used as preprocessed input for a convolutional pipeline consisting of convolution, pooling, and flattening operations. The outputs from the flattened layer are extracted as convolutional embeddings. In parallel, to obtain LLM features, each extracted image frame and our prompt are used together as input to an LLM model, generating an LLM response. The LLM response is then used as input to a second LLM model for Natural Language Processing (NLP) [19], which generates corresponding LLM embeddings. The loops indicate that LLM inference is performed on each frame. As a part of the multimodal fusion process, the convolutional embeddings are concatenated with the LLM embeddings during data preprocessing for the next stage. Lastly, the LLM embeddings, convolutional embeddings, and concatenated embeddings are used as input to an MLP classifier for evaluation.

B. Deep Learning-based Convolutional Features

The extracted image frames vary in resolution and pixel dimensions depending on the original video source. Therefore, as part of standardizing the input data, all frames are resized to 224x224 pixels while preserving the three RGB (Red, Green, Blue) color channels. To assess the impact of integrating LLM features with convolutional features, we evaluated multiple CNN architectures, including DenseNet-169 [20], ResNet-50 [3], and ResNet-101 [21]. In each of these models, after the convolution, pooling, and flattening operations, convolutional vector embeddings V_i are extracted from the flattened layer without proceeding to their fully connected layers in the neural network. C is defined as a collection of all convolutional embeddings extracted from images in our experimental sample sets.

$$\mathcal{C} = \{V_1, V_2, \ldots, V_N\} \tag{1}$$

C. LLM-derived features

In this paper, we define IM_i as the i^{th} image frame extracted from the dataset and P as our fixed LLM prompt. The LLM processes each pair (IM_i, P) to produce a response R_i, which is subsequently turned into NLP vector embeddings V_i through a second LLM model. Both LLM models are used without fine-tuning or additional training. O is defined as the final collection of all LLM embeddings extracted from images in our experimental sample sets.

Gemini is chosen as the LLM for our system due to its advanced capabilities in multimodal understanding. Two Gemini LLM models are used in a two-staged process: Gemini-2.5-pro [22] for generating responses based on each image and prompt pair, and Gemini-embedding-001 [23] for turning LLM text responses into NLP vector embeddings.

In the first stage, the LLM input is a pair consisting of an image frame and a fixed text prompt. To maximize image interpretability, the image frame is preserved at the highest possible resolution.

$$\mathcal{IM} = \{IM_1, IM_2, \ldots, IM_N\} \tag{2}$$

Each pair (IM_i, P) is passed into the Gemini-2.5-pro model via Google Cloud API to generate an LLM text response.

$$R_i = \text{LLM}(IM_i, P) \tag{3}$$

$$\mathcal{R} = \{R_1, R_2, \ldots, R_N\} \tag{4}$$

In the second stage, each R_i is passed into the Gemini-embedding-001 model via Google Cloud API. Each output includes a 3072-dimension NLP vector embedding V_i, calculated internally based on semantic search, classification, and clustering for accurate context-aware results [23].

$$V_i = \text{NLP}(R_i) \tag{5}$$

$$\mathcal{O} = \{V_1, V_2, \ldots, V_N\} \tag{6}$$

This stage efficiently produces LLM-derived embeddings. Before passing the embeddings into the MLP for training and evaluation, the convolutional embeddings are concatenated with the LLM embeddings. Algorithm 1 recaps the LLM feature extraction and concatenation pipeline.

Algorithm 1 LLM Feature Extraction and Concatenation Pipeline

Input: extracted images IM, text prompt P, convolutional embeddings $\mathcal{C}$
Output: concatenated embeddings $\mathcal{F}$
for each img in IM **do**
  $R_i \leftarrow \text{LLM}(IM_i, P)$ // Generate LLM response
end for
for each resp in R **do**
  $V_i \leftarrow \text{NLP}(R_i)$ // Generate NLP vector embedding
end for
$\mathcal{O} \leftarrow \{V_1, \ldots, V_N\}$
$\mathcal{F} \leftarrow \text{Concatenate}(\mathcal{O}, \mathcal{C})$ // Concatenate embeddings
return $\mathcal{F}$

D. Deep Learning: Multi-Layer Perceptron (MLP)

The concatenated feature embeddings consist of both convolutional and LLM-derived embeddings. They are used as input to an MLP for binary classification. The MLP architecture consists of three fully connected layers, each followed by a dropout layer to prevent overfitting. Rectified Linear Unit (ReLU) [24] activation functions are used as the non-linear part for all hidden layers. The final output layer consists of the sigmoid activation function. The features are classified into one of two possible classes: original or altered. The model is optimized using the Adam optimizer and trained using binary cross-entropy loss. An MLP is used as it provides a lightweight and consistent classifier for evaluation.

IV. Evaluation

A. Experimental Setup

FaceForensics++ Dataset. Within the FaceForensics++ dataset, we extracted 2,310 images from 228 videos to construct our training and testing sets. We employed a balanced 50/50 split of original and altered images, including 1,155 original and 1,155 altered samples. The altered samples include two manipulation types: Face2Face and DeepFakes. DeepFakes replace the entire facial identity using generative models, whereas Face2Face performs expression reenactment by altering localized facial regions such as the mouth, eyes, and nose. We applied an 80/20 train-test split, using 1,848 images for training and 462 images for testing.

Computer Specifications. We conducted all experiments on a Lenovo ThinkPad X1 Carbon Gen 9 (Intel i7-1185G7, 16 GB RAM, Intel Xe Graphics, 1 TB SSD) with Ubuntu 22.04.5 LTS.

Baseline. To evaluate test results consistently across different models, we established a baseline using an MLP architecture. The model consists of three fully connected layers with 1024, 512, and 256 ReLU units, each followed by a dropout layer of 0.5, and a final dense layer with a single sigmoid unit for binary classification.

B. Experimental Results

To effectively evaluate the influence of LLMs on CNN-based deepfake detection, we designed two prompt variations and conducted experiments on two distinct feature sets. The first, a reinforced feature set, uses Prompt 1 (P1) to guide the LLM to generate features aligned with CNN features. The second, an additional feature set, uses Prompt 2 (P2) to direct the LLM to provide binary classification with semantic reasoning. During prompt design, we tested multiple candidates and selected the two that produced the most meaningful results. We further optimized them with the assistance of GPT-4 [25] and role prompting to improve response relevance.

Impact of Reinforced Feature Set Driven by CNN Features: In a similar way that CNNs extract visual features and identify patterns, Prompt 1 guided the LLM to analyze low-level visual features strictly without subjective reasoning. As demonstrated in Table I, based on an altered frame in Fig. 2, the LLM response followed a structured and non-subjective format, providing a consistent forensic-focused input for the subsequent NLP processing. The results in Table II indicate that concatenating LLM and convolutional embeddings yielded accuracy improvements ranging from 6.2% to 14.1%. DenseNet-169 showed the most significant improvement (+14.1%), highlighting the LLM's positive contribution to CNN models with lower baseline accuracy. While ResNet-50 and ResNet-101 exhibited more minor improvements, the results suggest that LLM reinforcement can still contribute positively to CNN models with a stronger baseline. Because the dataset is balanced, F1 scores closely follow accuracy. Based on this experiment, the combination of visual and language features enriches aligned feature sets, compensates for blind spots, and strengthens classification reliability in deepfake detection tasks.

> **Prompt 1 (P1).** You are an image analysis assistant. Describe all visible low-level visual features in the image without making any assumptions about its meaning or authenticity. Focus strictly on objective characteristics such as:
> - Lighting direction, intensity, and shadow distribution
> - Edge definition and sharpness
> - Textures of skin, hair, or surfaces
> - Color gradients and transitions

- Presence of blur, noise, or compression artifacts
- Symmetry or geometric alignment of facial features and objects

Avoid any subjective or interpretive language. Report only what is visually present.

Fig. 2. Example of original and altered frame from the FaceForensics++ dataset.

TABLE I
SAMPLE LLM RESPONSES TO AN ALTERED FRAME UNDER TWO PROMPT DESIGNS (P1–P2).

Prompt	Sample Response (truncated)
P1	"Lighting from front-left; soft shadows on right cheek and jawline. Face outline appears blurry; skin texture smooth with minimal pores. Hair lacks detail and blends into background. Significant blur and compression artifacts visible..."
P2	"Altered. Visual cues: unnatural edges along jawline, inconsistent lighting between face and hair, abrupt hairline transition, texture mismatch between face and background, and color cast differences..."

TABLE II
ACCURACY, F1 SCORE, PRECISION, AND RECALL OF MODELS ON REINFORCED FEATURE SET

Model		Metric	Baseline	CNN+LLM	Δ
LLM (Prompt 1)		**Accuracy**	**0.831**	**N/A**	N/A
		F1 Score	0.837	N/A	N/A
		Precision	0.810	N/A	N/A
		Recall	0.866	N/A	N/A
CNN	DenseNet-169	**Accuracy**	**0.790**	**0.931**	**+0.141**
		F1 Score	0.789	0.933	+0.144
		Precision	0.801	0.903	+0.102
		Recall	0.777	0.965	+0.188
	ResNet-50	**Accuracy**	**0.818**	**0.880**	**+0.062**
		F1 Score	0.823	0.880	+0.057
		Precision	0.809	0.860	+0.051
		Recall	0.837	0.900	+0.063
	ResNet-101	**Accuracy**	**0.803**	**0.872**	**+0.069**
		F1 Score	0.805	0.877	+0.072
		Precision	0.803	0.844	+0.041
		Recall	0.807	0.913	+0.106

Impact of Additional Feature Set Driven by LLM Semantics: Building upon the characteristics of LLMs, Prompt 2 assigned the LLM the role of a digital-image forensics expert, followed by a targeted question with examples and constraints. Prompt 2 instructed the LLM to respond with a prediction and semantic reasoning, allowing subjective judgments. As demonstrated in Table I, based on the same altered frame shown in Fig. 2, the Prompt 2 response began with its classification prediction. It continued with reasoning that used subjective terms such as *inconsistent*, *abrupt*, and *mismatch* to simulate forensic expert analysis. The results in Table III show that the LLM baseline in Prompt 2 achieved a subtle 1.3% improvement compared to Prompt 1. However, when the LLM and convolutional embeddings were combined, accuracy improvements ranging from 6.2% to 15.4% were observed, which are comparatively higher than those of the reinforced feature set. Prompt 2, combined with DenseNet-169, achieved the most significant improvement (+15.4%) and the highest overall accuracy among all experiments. Similar to the reinforced feature set experiment, ResNet-50 and ResNet-101 again exhibited minor improvements, and F1 scores again showed consistent improvements with accuracy. These findings suggest that the semantic reasoning perspective introduced by the LLM, distinct from CNN-derived visual features, contributes to advancing overall model accuracy. In comparison to the reinforced feature set, the additional feature set demonstrates that LLMs can extend CNN-based detection by introducing fresh perspectives and classification cues.

> **Prompt 2 (P2).** You are a digital-image forensics expert. Is this image an unaltered video frame of a real person, or has it been manipulated (e.g., face-swap, AI-synthesis, Poisson blend)? Answer "original" or "altered" and list visual cues.

TABLE III
ACCURACY, F1 SCORE, PRECISION, AND RECALL OF MODELS ON ADDITIONAL FEATURE SET

Model		Metric	Baseline	CNN+LLM	Δ
LLM (Prompt 2)		**Accuracy**	**0.844**	**N/A**	N/A
		F1 Score	0.844	N/A	N/A
		Precision	0.844	N/A	N/A
		Recall	0.844	N/A	N/A
CNN	DenseNet-169	**Accuracy**	**0.790**	**0.944**	**+0.154**
		F1 Score	0.789	0.946	+0.157
		Precision	0.801	0.915	+0.114
		Recall	0.777	0.980	+0.203
	ResNet-50	**Accuracy**	**0.818**	**0.880**	**+0.062**
		F1 Score	0.823	0.879	+0.056
		Precision	0.809	0.863	+0.054
		Recall	0.837	0.896	+0.059
	ResNet-101	**Accuracy**	**0.803**	**0.881**	**+0.078**
		F1 Score	0.805	0.886	+0.081
		Precision	0.803	0.852	+0.049
		Recall	0.807	0.922	+0.115

Summary of Experimental Results: Based on the experiments of the two feature sets, we can conclude that LLMs have a positive influence on CNN-based models by improving deepfake detection accuracy. As illustrated in Fig. 3, CNN-LLM combined models consistently outperformed both CNN and LLM alone. The degree of improvement varied by CNN architectures; the lower the baseline accuracy of the CNN model, the higher the improvement when combined with the LLM. These experimental results reveal that LLMs can complement CNN-based models in diverse ways with high reliability in deepfake detection.

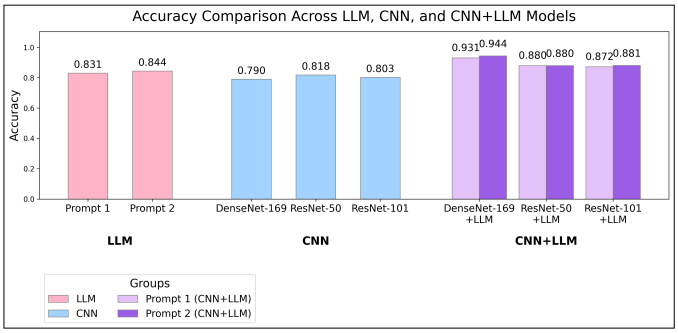

Fig. 3. Accuracy comparison chart across LLM, CNN, and CNN+LLM models

Computational Time: The end-to-end computational time required approximately 87 hours, dominated by preprocessing (85 hours). The LLM pipeline dominated preprocessing due to API rate limits (71.73 hours), while the image-processing pipeline accounted for the remaining time. Feature concatenation was negligible. Model training was lightweight, taking under a minute for LLM baseline models, approximately 102 minutes for CNN baseline models, and approximately 3 minutes for combined models.

V. Conclusion

This paper proposes a new method for measuring the influence of LLMs in CNN-based deepfake detection, utilizing two LLM models and three CNN-based models, and evaluating their performance with an MLP classifier. Our samples are evaluated using standard classification metrics, including accuracy, F1 score, precision, and recall. Across two experiments, combining LLM and convolutional embeddings improved overall accuracy, ranging from 6.2% to 15.4%. The results of the two feature sets demonstrate that LLMs can contribute positively from different perspectives, both from reinforcing CNN-derived visual features and from providing semantic reasoning. Ultimately, these experimental results signify the positive potential of LLMs in enhancing CNN-based deepfake detection. For CNN models with a weaker baseline, LLMs can significantly enhance deepfake detection. As for models with a stronger baseline, LLMs can also slightly improve detection accuracy. Future work includes extending frame-level analysis to video-level analysis, exploring broader prompt designs, and evaluating generalization across additional datasets to further strengthen multimodal deepfake detection.

References

[1] I. Goodfellow et al., "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, Oct. 2020, doi: 10.1145/3422622.

[2] J. Thies, M. Zollhöfer, M. Stamminger, C. Theobalt and M. Nießner, "Face2Face: Real-time face capture and reenactment of RGB videos," *Communications of the ACM*, vol. 62, no. 1, pp. 96–104, Dec. 2018, doi: 10.1145/3292039.

[3] S. Praveena, R.Kaviya, K.Sheerin Farhana and S.Bhuvanasri, "Deep fake video detection using transfer learning Resnet50", *International Research Journal on Advanced Engineering and Management*, vol. 3, no. 3, pp. 585–590, Mar. 2025, doi: 10.47392/IRJAEM.2025.0094.

[4] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *CVPR*, Honolulu, HI, USA, 2017, pp. 1800–1807, doi: 10.1109/CVPR.2017.195.

[5] A. Vaswani et al., "Attention is all you need," in *31st International Conference on Neural Information Processing Systems*, pp. 6000–6010, Dec. 2017.

[6] F. Murtagh, "Multilayer perceptrons for classification and regression," *Neurocomputing*, vol. 2, no. 5–6, pp. 183–197, July 1991, doi: 10.1016/0925-2312(91)90023-5.

[7] A. Haliassos, R. Mira, S. Petridis and M. Pantic, "Leveraging real talking faces via self-supervision for robust forgery detection," in *CVPR*, New Orleans, LA, USA, 2022, pp. 14930–14942, doi: 10.1109/CVPR52688.2022.01453.

[8] W. Yang et al., "AVoiD-DF: Audio-Visual Joint Learning for Detecting Deepfake," in *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 2015-2029, 2023, doi: 10.1109/TIFS.2023.3262148.

[9] J. Zhang, J. Huang, S. Jin and S. Lu,"Vision-Language Models for Vision Tasks: A Survey" in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 8, pp. 5625–5644, Aug. 2024, doi: 10.1109/TPAMI.2024.3369699.

[10] P. Yu et al., "Unlocking the capabilities of Large Vision-Language Models for generalizable and explainable deepfake detection," Poster session, in *ICML Virtual*, 2025. [Online] Available: https://icml.cc/virtual/2025/poster/43687. [Accessed July 28, 2025].

[11] Y. He, Y. Cao, B. Yang and Z. Zhang, "Can GPT tell us why these images are synthesized? Empowering multimodal Large Language Models for forensics," in *ACM Workshop on Information Hiding and Multimedia Security*, pp. 24–34, June 2025, doi: 10.1145/3733102.373313.

[12] G. Wu et al., "Cheap-fake detection with LLM using prompt engineering," in *IEEE International Conference on Multimedia and Expo Workshops (ICMEW)* Brisbane, Australia, 2023, pp. 105–109, doi: 10.1109/ICMEW59549.2023.00025.

[13] Y. Zhang, B. Colman, X. Guo, A. Shahriyari and G. Bharaj, "Common sense reasoning for deepfake detection," in *Computer Vision – ECCV: 18th European Conference*, Milan, Italy, Sept.–Oct. 2024, vol. 15146, pp. 399–415, doi: 10.1007/978-3-031-73223-2_22.

[14] J. Li, D. Li, C. Xiong and S. Hoi, "BLIP: bootstrapping language-image pre-training for unified vision-language understanding and generation," in *ICML*, 2022, pp. 12888–12900.

[15] J. Devlin, M.-W. Chang, K. Lee and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding", *arXiv preprint*, Oct. 2018, doi: 10.48550/arXiv.1810.04805.

[16] A. Dosovitskiy et al., "An image is worth 16x16 words: transformers for image recognition at scale," *arXiv preprint*, June 2021, doi: 10.48550/arXiv.2010.11929.

[17] S. E. VP, C. M. S and R. Dheepthi, "LLM-enhanced deepfake detection: Dense CNN and multi-modal fusion framework for precise multimedia authentication," in *International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS)*, Chennai, India, 2024, pp. 1–6, doi: 10.1109/ADICS58448.2024.10533511.

[18] W. Dai et al., "InstructBLIP: towards general-purpose vision-language models with instruction tuning", in *37th International Conference on Neural Information Processing Systems*, 2023, no. 2142, pp. 49250–49267.

[19] D. Khurana, A. Koli, K. Khatter and S. Singh. "Natural language processing: state of the art, current trends and challenges," *Multimed Tools and Applications*, vol. 82, pp. 3713–3744, 2023, doi:10.1007/s11042-022-13428-4.

[20] G. Huang, Z. Liu, L. Van Der Maaten and K. Q. Weinberger, "Densely Connected Convolutional Networks," in *CVPR*, Honolulu, HI, USA, 2017, pp. 2261–2269, doi: 10.1109/CVPR.2017.243.

[21] Q. Zhang, "A novel ResNet101 model based on dense dilated convolution for image classification," *Discover Applied Sciences*, vol. 4, 2022, doi:10.1007/s42452-021-04897-7.

[22] Google DeepMind, "Gemini 2.5 Pro," [Online]. Available: https://deepmind.google/models/gemini/pro/. [Accessed: July 28, 2025].

[23] Google AI for Developers, "Embeddings | Gemini API," [Online]. Available: https://ai.google.dev/gemini-api/docs/embeddings. [Accessed: July 28, 2025].

[24] A. F. Agarap, "Deep learning using Rectified Linear Units (ReLU)," *arXiv preprint*, Feb. 2019, doi: 10.48550/arXiv.1803.08375.

[25] OpenAI, "GPT-4," [Online]. Available: https://openai.com/index/gpt-4-research/. [Accessed: Sept. 1, 2025].