Journal of Information Systems Engineering and Management
2025, 10(30s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Video Manipulation Detection using Sequence Learning and Convolution Networks: A Comparative Study

Renita Kurian[1], Rimjhim Singh[1], Vanshika Goel[1] and Bhaskarjyoti Das[2]
[1]*Department of Computer Science and Engineering, PES University, Bengaluru, Karnataka, India rrenita1206@gmail.com, singhrimjhim2809@gmail.com, goelvanshika1006@gmail.com*
[2]*Department of Computer Science and Engineering in AI & ML, PES University, Bengaluru, Karnataka, India bhaskarjyoti01@gmail.com*

ARTICLE INFO

Received: 31 Dec 2024
Revised: 20 Feb 2025
Accepted: 28 Feb 2025

ABSTRACT

Nowadays, the accessible and technologically advanced edit- ing tools, coupled with the surge in photo and video content pose a great risk to content authenticity. Manipulated content can be used to spread misinformation, cause harassment and infringe human rights. In this article, we compare the effectiveness of two approaches for video manipulation detection using micro and macro information, i.e., a Long Short-Term Memory (LSTM) architecture with frame-level features of videos and their respective ground truths as inputs and a Graph Con- volutional Network (GCN) with frame-level video scene graphs con- catenated using temporal edges. While the LSTM-based model cap- tures frame-level micro-information, the GCN model captures high-level macro-information inside a video.

Keywords: scene graphs, video manipulation detection, LSTM, GCN

1 INTRODUCTION

Video manipulation is the deliberate alteration of video content by humans or AI. Any change that modifies the sequence or intended message of digital content counts as manipulation. While some altered videos might seem harmless, they carry risks in specific contexts. Altered videos can mislead viewers, creating misconceptions and false assumptions. This tactic is used by those aiming to manipulate audiences with misleading or dishonest information.

The exponential increase in the creation of video and image content, as well as the rise of easily accessible and technologically advanced editing tools, poses a threat to video manipulation. The proliferation of manipulated content raises concerns about the integrity of visual content and human rights protection and presents twofold dangers: it accelerates the circulation of misinformation. It serves as a facilitator for various forms of harassment. Thus, accurately detect- ing and identifying manipulated content becomes essential in addressing these widespread issues.

Tampering attacks can be classified into three categories[11] :

– **Spatial tampering:** These alterations are confined within the visual space. Spatial tampering revolves around spatial information, involving adding or removing objects within the frame.

– **Temporal tampering:** Temporal tampering involves manipulations that affect the sequencing or timing of events or frames, including actions such as inserting, duplicating, or reordering frames.

– **Spatio-temporal tampering:** Spatio-temporal tampering is a fusion of spatial and temporal manipulation techniques. This attack combines alter- ations in both spatial information and the sequencing of frames, creating a more complex form of tampering.

One possible way of detecting the video manipulation is to consider the infor- mation contained in the sequence of frames in a sequence model. This approach depends on the low-level information inside the video

frames.

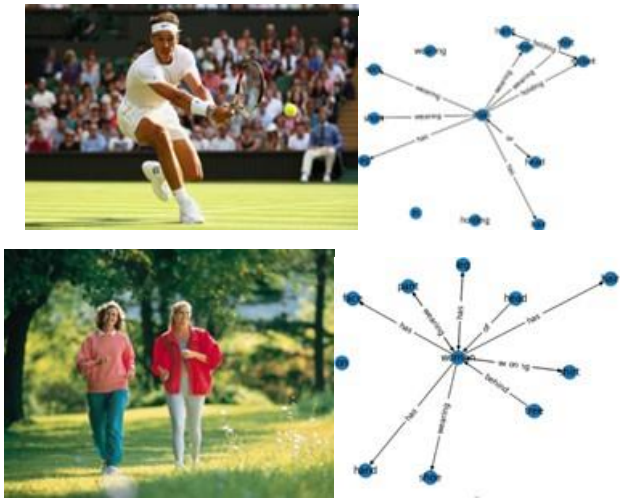

Fig. 1. Scene Graphs generated using pre-trained model RelTR

A scene graph is a structured graphical representation of an image that re- sembles the format used in knowledge base representations, where the nodes are associated with objects, and edges denote pairwise connections between these nodes. As a video is a sequence of frames, video scene graphs can be visual- ized as frame-level scene graphs concatenated by temporal edges. To represent videos, spatio-temporal graphs or 3D scene graphs can be used. This approach provides a macro-level information capturing relationships between objects in a frame. These scene graphs can furnish the macro-level contextual information for detecting manipulations and can be used to detect video manipulation.

The work described in this paper compares the effectiveness of micro and macro-level information for video manipulation detection.

The subsequent sections of this article are structured as follows. Section 2 provides an overview of relevant research related to the problem statement or the technology used. Section 3 explains the methodology followed. Section 4 de- scribes the datasets employed in the study and pre-processing performed. Section 5 describes the models used. Section 6 presents the results obtained from testing the model, along with a discussion of its limitations. Finally, section 7 outlines potential avenues for further research and concludes the article.

2 RELATED WORK

Many different techniques may achieve video manipulation. As mentioned ear- lier[11], the manipulation techniques may be broadly categorized into three types, i.e., Spatial Tampering, Temporal Tampering, and Spatio-temporal tam- pering. An example of a temporal attack would be frame insertion, duplication, or shuffling. Spatial attacks have to do with the spatial information. Attacks such as object addition or removal are spatial attacks. Spatio-temporal attacks refer to attacks that combine both spatial and temporal tampering techniques. The various techniques implemented are of two categories:

– **Active manipulation detection:**Active tampering detection relies on hid- den data, such as pre-embedded digital signatures or watermarks. This in- cludes detection using digital signatures, intelligent techniques employing algorithms like SVM, and watermarking.

– **Passive manipulation detection:** In contrast, passive tampering detec- tion depends solely on the statistical features of the video and does not require pre-embedded data. Passive tampering detection methods involve a range of techniques, including recognizing edits through camera analysis, de- tecting changes based on coding traces, inconsistency-driven detection, and identifying instances of copy-move manipulation.

2.1 Video Classification Methods

Video manipulation can also be seen as an application of video classification where videos are classified as

manipulated or not. Different methods of classifi- cation have been used in the past to classify videos, but the prominent methods are :

– Using CNN: This method refers to a 3D CNN architecture as present in [21], [15]. Some papers also propose an alternate approach using a multi-resolution CNN [31].

– Using RNN: This method generally relies on a CNN and RNN model. In this method, a CNN architecture, such as Xception [3] or ResNet[12], is uti- lized for extracting features from videos, and these obtained features are subsequently input into a recurrent neural network to capture temporal de- pendencies.

2.2 Video Manipulation Detection

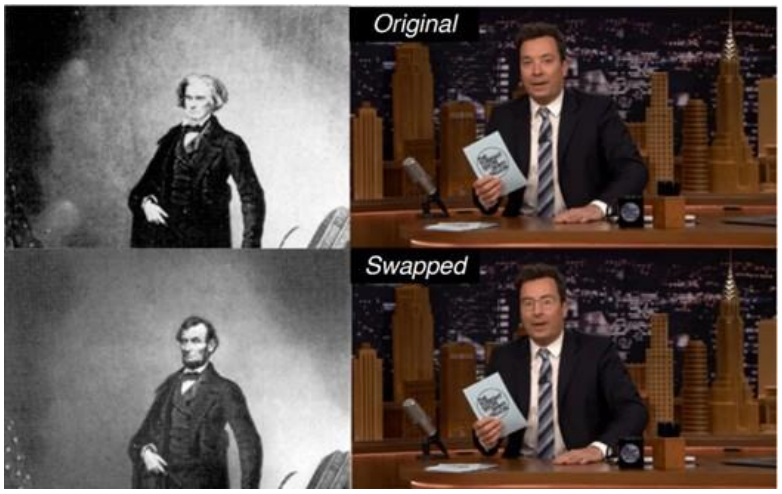

Fig. 2. Example of Deepfakes[10]

The existing models of video manipulation detection are mostly limited to de- tecting Deepfakes. Deepfakes refer to a type of manipulation where the subject's facial features in the video are altered to make it look like someone else. The ex- ample shown by Figure 2 [10] is an example of a Deepfakes. Significant work has been completed in the domain of Deepfakes and many commendable approaches have been published, having achieved accuracies above 97%. The SOTA Deep- fakes models could detect facial manipulations with an accuracy above 95%. The popular methods for Deepfakes detection were using CNNs[1, 23, 17, 5] and RNN [25, 18, 10]. RNNs can capture the dependencies from the sequence and use it to predict the next output. Hochreiter and Schmidhuber[13] introduced Long Short Term Memory (LSTM) Networks. LSTM is a variant of RNN that can learn long and short-term dependencies in a data sequence. LSTMs have proven to be a popular choice for various sequential data tasks. LSTMs are also used with video data[10].

Video manipulation detection has also been attempted using metadata [9]. However, the paper's authors acknowledged that although the model performed very well, it was limited by meta-data availability. This approach may also be vulnerable to video re-encoding attacks. Ehsan Nowroozi et al.[22] proposed an approach for identifying manipulations in the background of video conferenc- ing calls. The paper aimed to prevent deception and ensure that users present themselves honestly during calls. This model achieved an accuracy of over 99%. However, this model was limited to one attack, and its performance varied sig- nificantly over different platforms and devices.

Although the models mentioned above performed well in specific domains and contexts, they failed to give satisfactory results in a more generalized context.

2.3 Graph Convolution Networks (GCNs)

In recent years, GCNs have also gained popularity for computer vision tasks. GCNs have been used in the past for human pose prediction [6], [27], image captioning and understanding [7], [16], [8], video anomaly detection [2], [19], [30]

and visual question-answering [14], [29], [33], [32]. GCNs have also been proposed for Deepfakes detection and

were able to obtain benchmark results. Zhihua Shang et al. [26] used a graph network to detect spatio-temporal inconsistencies. In the aforementioned paper, videos were represented as a spatio-temporal graph, and the obtained graph was then passed into the proposed model. This approach was able to increase the benchmark for Deepfakes detection.

3 METHODOLOGY

Most existing models for detecting video manipulation focus on facial changes, like those in Deepfakes. While these models excel at recognizing altered faces, they struggle with broader manipulation detection. The SOTA Deepfakes models could detect facial manipulations with an accuracy above 95%. The research paper released by Adobe [20], who created the VideoSHAM dataset, mentions an accuracy lower than 50% for all Deepfakes models run on their dataset.

Four types of attacks are investigated:

1. **Addition of objects:** In this attack, an object or entity is added to the current video from another source.

2. **Removal of objects:** This attack involves removing a particular object/entity from all frames of a video and filling the gaps using background settings.

3. **Background/Color Change:** This type of attack deals with a change in the background or the colour of a small entity in the video.

4. **Text Replaced/Added:** Some text is added or removed from the video, or some existing text is replaced.

This paper aims to tackle video manipulation detection with dual-level data, both at the micro and macro levels. We propose a novel way of generating dataset-related information at micro-level (frame-level) features, extracted using the pre-trained CLIP model [24] for all relevant videos from our dataset, as well as macro-level (frame-level scene concatenated graphs using temporal edges), generated using the pre-trained RelTR model [4].

These two approaches are then compared to see how well LSTM and GCN models perform in spotting video manipulation at different levels of detail.:

– **LSTM model:** This model uses detailed frame-level features to detect ma- nipulation at the frame level

– **GCN model:** It utilizes graph-based data to detect manipulation by un- derstanding relationships within the video.

4 DATASET

In this work, the VideoSHAM dataset[20] has been used. This dataset is a com- prehensive repository focused on a wide spectrum of video manipulations rather than being confined solely to facial alterations. This dataset encompasses over 500 videos, exhibiting varying lengths and quality, offering a diverse and contex- tually rich assortment of human-centric manipulated videos. Comprising six dis- tinct categories of spatial and temporal manipulations, the VideoSHAM dataset delves into various alterations applied to videos. Sourced from the online plat- form Vimeo, this dataset features professionally edited videos spanning durations from 1 to 31 seconds. Accompanying the dataset is a .CSV file furnishing pre- cise details regarding the specific attack executed on each video, along with the corresponding timestamp of manipulation.

The different types of manipulation attacks covered in this dataset are :

– Addition of object/entity, where an object is added from some other source into the current video.

– Removal of object/entity, where an object or entity is removed from all the frames of a video and the gaps are filled using background settings

– Background/color change, where the background or the color of a small entity of the video is

changed.

– Text replacement or addition, where some text is added or removed from the video, or existing text is replaced.

– Frames duplication/deletion, where some frames are randomly duplicated, removed or dropped from a video, to make the video inconsistent.

– audio replacement, where the existing audio of a video is replaced with some other audio.

Although the dataset consisted of six attacks, this paper is primarily concen- trated on videos pertaining to four specific attacks: addition of object, removal of object, background change, and text replacement or addition. Following post- processing procedures, a total of 209 manipulated videos were successfully ob- tained, alongside approximately 250 unedited videos. Some videos were excluded due to inaccuracies in annotations, making them unsuitable for the proposed ar- chitecture.

4.1 Data Pre-processing

Splitting the videos into frame The Adobe VideoSHAM dataset is cate- gorised into two core segments:

1. Unedited: This section comprises the original, unaltered videos.

2. Processed: Within the processed segment, videos housed in the unedited directory undergo manipulation via six distinct attack types and are subse- quently stored in the processed directory.

For frame-by-frame extraction, the videos undergo splitting into individual frames or images leveraging the OpenCV library's support. This process allows for meticulous analysis and handling of each frame as discrete visual units, en- abling comprehensive study and assessment of the video content at the frame level.

Extracting Ground Truth Labels The dataset includes a metadata file con- taining information about the specific type of manipulation and timestamps denoting when these alterations occurred within the videos. From a total of 448 videos, our analysis involved 167 unedited videos and 281 processed or ma- nipulated videos. The available metadata for the remaining videos lacked the necessary accuracy to generate reliable ground truth labels.

The ground truth representation for each video is represented as a vector. This vector's length aligns with the number of frames or images constituting the video. Within this vector, a value of 1 indicates tampering within the cor- responding frame, creating a sequence of frames delineating tampered content. Deriving this ground truth involves a calculation considering crucial video pa- rameters such as the frame rate, the total count of frames in the videos, and the specific time intervals during which manipulations were executed within the video. By utilizing this data, we generate a comprehensive ground truth vec- tor that accurately pinpoints tampered frames, offering a detailed and granular depiction of manipulations within the video content.

5 MODEL

5.1 Feature Extraction

This methodology intricately addresses both micro and macro levels of informa- tion present within the video content.

Image Encoder for Micro-Level Feature Extraction: At the micro level, we study the intricate details using Contrastive Language Image Pre-Training (CLIP) [24], an advanced neural network framework recognized for its proficiency in learning visual concepts through natural language supervision. Leveraging CLIP involves an adaptable approach applicable to diverse visual classification benchmarks. This model's 'zero-shot' capabilities, akin to those found in GPT-2 and GPT-3, allow it to recognize visual categories simply by supplying their names.

CLIP employs both an Image Encoder and a Text Encoder for zero-shot prediction, both of which are viable for transfer learning. In this paper, we specifically utilized the Image Encoder component of CLIP to generate features for each frame within the video. Notably, CLIP is lighter than conventional neural

networks like ResNet [12] and InceptionNet [28] and has proven to give good results for zero-shot transfer and other image classification problems.

The CLIP Image Encoder operates by producing a CLIP embedding of 512 dimensions for an individual image, as shown in Figure 3. Consequently, when dealing with a video, which is essentially a sequence of frames, this translates into representing the video as a sequential arrangement of CLIP embeddings. This sequence of CLIP embeddings enables the encapsulation of pixel-level or micro-level information intrinsic to each frame within the video.

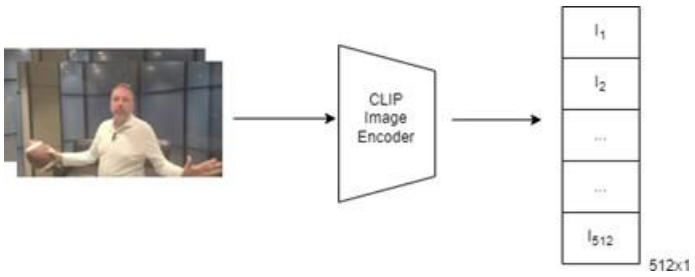

Fig. 3. Architecture of CLIP Image Encoder for Micro-level Feature Extraction

Scene Graphs for Macro Level Features Extraction: For capturing video- level or macro-level information, we employ scene graphs. To extract and con- struct these scene graphs, we utilize a pre-trained model specifically designed for Scene Graph Generation, known as the Relation Transformer (RelTR) [4]. Some Scene Graphs generated by RelTR are shown in Figure 1.

Scene graphs are systematically extracted for all frames within the videos, and subsequently, these graphs are concatenated by incorporating temporal edges. Temporal edges, representing weighted edges, are used to represent tem- poral connection between objects. The weights of these edges are determined by the difference in timestamps between various frames.This concatenated rep- resentation combines individual scene graphs into a unified structure termed the Video Scene Graph. This comprehensive structure encapsulates the video's macro information, elucidating high-level relationships and contextual associa- tions embedded within the video content. This distinct approach facilitates the consolidation of high-level relationships and overarching contextual information into a singular graph-based structure for the comprehensive macroscopic analysis of the video content.

5.2 Seqential and Graph Neural Models

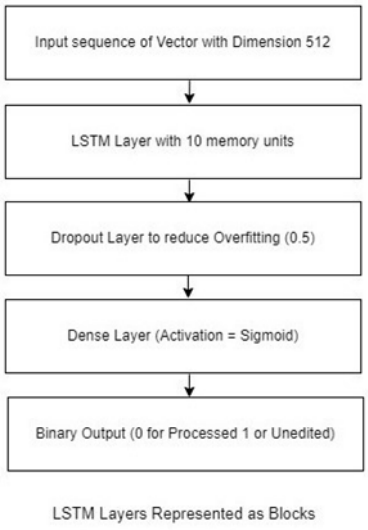

Fig. 4. Recurrent Neural Network (RNN) with Long Short-Term Memory (LSTM) Architecture

Long Short Term Memory (LSTM) architecture: The architecture em- ployed for processing micro-level information involves a Recurrent Neural Net- work (RNN) with a Long Short-Term Memory (LSTM) architecture, meticu- lously trained on videos represented as sequences of CLIP embeddings. To

ac- commodate varying frame counts across videos, the sequences undergo padding using constant padding techniques, ensuring uniformity in sequence length. Com- prising two crucial layers, the LSTM architecture, represented in Figure 4, encompasses an LSTM layer followed by a Dense Layer. This design aims to produce outputs for every frame within the video, essentially identifying and flagging tampered frames within the sequence. The LSTM model employs the Adam optimizer and is trained through back-propagation using Binary Cross Entropy Loss, a pivotal method for optimizing the model's performance.

For partitioning the dataset into training and testing subsets, a standard 80-20 Test-Train split is employed. Within the architecture, the Dense Layer integrates the sigmoid activation function, crucial for facilitating binary classifi- cation, enabling the identification of tampered frames within the video sequences.

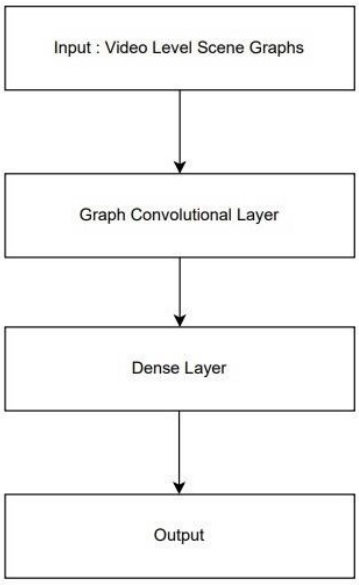

Fig. 5. Graph Convolutional Network (GCN)

Graph Convolution Network (GCN) architecture: The GCN (Graph Convolution Network) model is designed to take video scene graphs as input, generating predictions regarding their manipulation flag. Developed utilizing the Spektral library, this model architecture, represented in Figure 5, com- prises a singular convolutional layer and a subsequent dense layer. Notably, it- erations with models incorporating 2 and 3 convolutional layers were trained and assessed. However, diminished performance was observed in these models, prompting the selection of the architecture featuring a single convolutional layer and a subsequent dense layer for optimal predictive accuracy.

6 RESULTS AND DISCUSSION

The Recurrent Neural Network (RNN) with Long Short-Term Memory (LSTM) architecture demonstrated a notable achievement, attaining an accuracy of 77.09%

Table 1. Result metrics for all models

Model	Accuracy	Precision
Individual Frame CNN	0.50	0.3125
Video Scene Graph GCN	0.41	0.24
Frame Sequence RNN using LSTM	0.7709	0.9552

when trained and tested on a dataset comprising all videos. As shown in Table 1, the LSTM model's

performance surpasses baseline values (<50%) while excelling in frame-level manipulation detection. Additionally, it displayed a precision rate of 31.25%. These metrics indicate the model's ability to detect manipulation at the frame level, showcasing good accuracy in flagging manipulated frames within the video content. The Sequential RNN with LSTM architecture, uti- lizing CLIP-based feature extraction, demonstrates highly promising outcomes compared to the video-level Graph Convolution Network. This substantiates the premise that countering video tampering attacks often necessitates sequential learning paradigms, treating a video as a continuous flow of information dis- rupted by manipulation attacks.

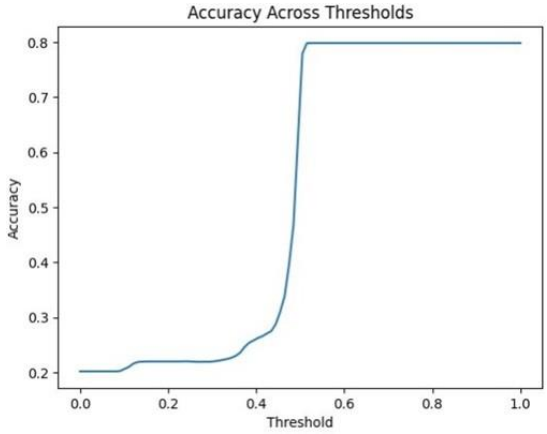

Fig. 6. Accuracy across thresholds for RNN with LSTM

In contrast, the Graph Convolutional Network (GCN) exhibited an accuracy of 41% when trained and tested on the video dataset, specifically focusing on attacks involving object/entity addition and removal. This lower accuracy signi- fies the model's struggle in effectively detecting video manipulation using only the high-level relationships embedded within the videos.

The observed disparity underscores the need for improved high-level feature extraction methodologies to support the GCN model's effectiveness in detect- ing macro-level manipulations for video content. The macro-level information represented through a concatenated video scene graph falls short in detecting manipulation attacks within videos. The representation of macro information as a single video scene graph renders attributes of edges and nodes obscured. Consequently, to enhance the detection capabilities, macro information should also be delineated as a sequence of scene graphs.

7 CONCLUSION AND FUTURE WORK

In this work, we have evaluated two possible approaches to detect video anomaly detection. While the micro-level information offers sufficient performance when wrapped in a sequence model, the macro-level information, in the form of scene graphs, proves ineffective. To better encapsulate macro-level details, a paradigm shift is crucial — representing macro information as a sequential arrangement of scene graphs for individual frames within videos, rather than combining indi- vidual frame scene graphs into a concatenated form.

As a next step, we plan to capture the macro-level scene-graph-based in- formation in a sequence model to better preserve the temporal aspect of the macro-level information. In addition, we plan to ensemble the micro and macro models to surpass the efficiency offered by each.

REFERENCES

[1] Bonettini, N., Cannas, E.D., Mandelli, S., Bondi, L., Bestagini, P., Tubaro, S.: Video face manipulation detection through ensemble of cnns. In: 2020 25th inter- national conference on pattern recognition (ICPR). pp. 5012–5019. IEEE (2021)
[2] Chen, H., Mei, X., Ma, Z., Wu, X., Wei, Y.: Spatial–temporal graph attention network for video anomaly detection. Image and Vision Computing **131**, 104629 (2023)
[3] Chollet, F.: Xception: Deep learning with depthwise separable convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition.
[4] pp. 1251–1258 (2017)

[5] Cong, Y., Yang, M.Y., Rosenhahn, B.: Reltr: Relation transformer for scene graph generation. IEEE Transactions on Pattern Analysis and Machine Intelli- gence (2023)
[6] Dang, H., Liu, F., Stehouwer, J., Liu, X., Jain, A.K.: On the detection of digital face manipulation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern recognition. pp. 5781–5790 (2020)
[7] Dang, L., Nie, Y., Long, C., Zhang, Q., Li, G.: Msr-gcn: Multi-scale residual graph convolution networks for human motion prediction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 11467–11476 (2021)
[8] Dong, X., Long, C., Xu, W., Xiao, C.: Dual graph convolutional networks with transformer and curriculum learning for image captioning. In: Proceedings of the 29th ACM International Conference on Multimedia. pp. 2615–2624 (2021)
[9] Geng, Y., Mei, H., Xue, X., Zhang, X.: Image-caption model based on fusion fea- ture. Applied Sciences **12**(19), 9861 (2022)
[10] Gü era, D., Baireddy, S., Bestagini, P., Tubaro, S., Delp, E.J.: We need no pixels: Video manipulation detection using stream descriptors. arXiv preprint arXiv:1906.08743 (2019)
[11] Gü era, D., Delp, E.J.: Deepfake video detection using recurrent neural networks. In: 2018 15th IEEE international conference on advanced video and signal based surveillance (AVSS). pp. 1–6. IEEE (2018)
[12] Habeeb, R., Manikandan, L.: A review: video tampering attacks and detection techniques. Int J Sci Res Comput Sci Eng Inform Technol **5**(5), 2456–3307 (2019)
[13] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
[14] Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
[15] Hu, Z., Wei, J., Huang, Q., Liang, H., Zhang, X., Liu, Q.: Graph convolutional net- work for visual question answering based on fine-grained question representation. In: 2020 IEEE fifth international conference on data science in cyberspace (DSC).
[16] pp. 218–224. IEEE (2020)
[17] Kanagaraj, K., Priya, G.L.: A new 3d convolutional neural network (3d-cnn) frame- work for multimedia event detection. Signal, Image and Video Processing **15**, 779– 787 (2021)
[18] Kumar, A., Agrawal, A., Shanly, K.A., Das, S., Harilal, N.: Image caption generator using siamese graph convolutional networks and lstm. In: 5th Joint International Conference on Data Science & Management of Data (9th ACM IKDD CODS and 27th COMAD). pp. 306–307 (2022)
[19] Li, L., Bao, J., Zhang, T., Yang, H., Chen, D., Wen, F., Guo, B.: Face x-ray for more general face forgery detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5001–5010 (2020)
[20] Li, Y., Lyu, S.: Exposing deepfake videos by detecting face warping artifacts. arXiv preprint arXiv:1811.00656 (2018)
[21] Luo, W., Liu, W., Gao, S.: Normal graph: Spatial temporal graph convolutional networks based prediction network for skeleton based video anomaly detection. Neurocomputing **444**, 332–337 (2021)
[22] Mittal, T., Sinha, R., Swaminathan, V., Collomosse, J., Manocha, D.: Video ma- nipulations beyond faces: A dataset with human-machine analysis. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 643–652 (2023)
[23] Naik, K.J., Soni, A.: Video classification using 3d convolutional neural network. In: Advancements in Security and Privacy Initiatives for Multimedia Images, pp. 1–18. IGI Global (2021)
[24] Nowroozi, E., Mekdad, Y., Conti, M., Milani, S., Uluagac, S., Yanikoglu, B.: Real or virtual: A video conferencing background manipulation-detection system. arXiv preprint arXiv:2204.11853 (2022)
[25] Qian, Y., Yin, G., Sheng, L., Chen, Z., Shao, J.: Thinking in frequency: Face forgery detection by mining frequency-aware clues. In: European conference on computer vision. pp. 86–103. Springer (2020)

[26] Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PMLR (2021)
[27] Sabir, E., Cheng, J., Jaiswal, A., AbdAlmageed, W., Masi, I., Natarajan, P.: Recur- rent convolutional strategies for face manipulation detection in videos. Interfaces (GUI) **3**(1), 80–87 (2019)
[28] Shang, Z., Xie, H., Yu, L., Zha, Z., Zhang, Y.: Constructing spatio-temporal graphs for face forgery detection. ACM Transactions on the Web **17**(3), 1–25 (2023)
[29] Sofianos, T., Sampieri, A., Franco, L., Galasso, F.: Space-time-separable graph convolutional network for pose forecasting. In: Proceedings of the IEEE/CVF In- ternational Conference on Computer Vision. pp. 11209–11218 (2021)
[30] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2015)
[31] Xu, X., Wang, T., Yang, Y., Hanjalic, A., Shen, H.T.: Radial graph convolutional network for visual question generation. IEEE transactions on neural networks and learning systems **32**(4), 1654–1667 (2020)
[32] Yang, Z., Guo, Y., Wang, J., Huang, D., Bao, X., Wang, Y.: Towards video anomaly detection in the real world: A binarization embedded weakly-supervised network. IEEE Transactions on Circuits and Systems for Video Technology (2023)
[33] Ye, H., Wu, Z., Zhao, R.W., Wang, X., Jiang, Y.G., Xue, X.: Evaluating two- stream cnn for video classification. In: Proceedings of the 5th ACM on International Conference on Multimedia Retrieval. pp. 435–442 (2015)
[34] Yusuf, A.A., Chong, F., Xianling, M.: An analysis of graph convolutional networks and recent datasets for visual question answering. Artificial Intelligence Review **55**(8), 6277–6300 (2022)
[35] Yusuf, A.A., Chong, F., Xianling, M.: Evaluation of graph convolutional networks performance for visual question answering on reasoning datasets. Multimedia Tools and Applications **81**(28), 40361–40370 (2022)