Research Article

Atom Structure Analysis Approach for Video Forgery Identification

Punam Sunil Raskar*[1], Sanjeevani Kiran Shah[2]

[1*]Research Scholar, Sinhgad College of Engineering, (E&TC), Savitribai Phule Pune University, Pune, (Maharashtra), India.
[2]Professor and HOD (PG), Smt. Kashibai Navale College of Engineering, (E&TC), Savitribai Phule Pune University, Pune, (Maharashtra), India.

Article History: Received: 11 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 23 May 2021

Abstract: In this high-tech world, editing of the video is made possible effortlessly by using advanced video editors. Misuse of these editors leads to manipulating the videos for hiding the truth. Traditional methods are computationally complex and slow for detecting the authenticity and integrity of such videos. The paper presents a fast method to detect manipulative videos by exploiting the atom structure analysis of the suspicious video. File format and metadata are another wellspring of scientific proof; however less consideration by researchers and investigators paid to these elements. The proposed work focuses on video forgery identification (VFI) method for MPEG-4 videos by investigating detailed file format information. The proposed system gives fast and accurate results as the examination of complete video frames is not required by this method. Thus tradeoff between accuracy and speed is optimized by this method. Implementation of the proposed system is performed on forged video dataset. Manipulated videos are created by using six different video editors. Experimental results demonstrate that the proposed system works efficiently for rendering binary output for all video editing software selected for experimentation. As well as, the system also correctly classifies frame addition and removal type of forgery.

Keywords: Atom Structure analysis, Digital Forensics, Manipulated videos, MPEG4, Tamper Detection, Video Forgery.

Introduction

Digital multimedia and strong internetworking has tremendous power, which influences human life easier. The power is huge enough to make extremely unnerving impacts on people's personal life. Social networking sites like Facebook, Twitter, Whatsapp, Instagram, Snap-chat, etc. have a major role to spread a piece of information to every common individual within a fraction of seconds. This information may contain text, image, audio, video, etc. The security of these kinds of messages or information is one of the important issues in this advanced technology world. An intruder can easily get access to this media and perhaps can manipulate the information present in a message with some advanced techniques. Thus integrity and truthfulness of image, audio, video lead to one more serious issue which needs to be addressed. In this widely interconnected world, fake images and videos also get forwarded easily. Hence, innocence has to pay a big penalty because of this fake forwarding.

The content of an image or video is very important to prove in various fields. Surveillance cameras are one of the important sources to get evidence. Due to advancements in technology with lower cost, wide use of the CCTVs are seen at different places such as banking, rail-way stations, airports, public places, hospitals, etc. Low cost and easily available software's are usually used to alter the information contained in an image or video. People with the wrong intention try to manipulate the contents for the wrong purpose, for an instance, to destroy the proof against crime erasing the originality. The overuse of the software for the wrong purpose increases the necessity of image/video forgery detection techniques. Initially, researchers started working to develop techniques to detect fraud in images. Growth in this field has risen with the birth of the 21st century. In the beginning, active techniques; which need references such as digital watermarking or digital signatures were developed to detect the tampered images. Later on, methods that don't need any reference was developed to detect tampering, such techniques known as passive techniques or 'Blind Forensics'. Numerous techniques have been developed for detecting image forgeries based on this primary classification. Researchers even successfully tried to detect many possible attacks that have been used to tamper images. However, forgery detection in videos is still unwrapped for people working in the digital forensic sector.

Techniques for video forgery detection primarily fall into two major types: Intra frame forgery and inter-frame. The specific frame is targeted for tampering in the case of intra-frame forgery. Such forgery has occurred when an attacker wishes to remove or insert some objects within the frame. Copy-move attack, region duplication are major examples of intra-frame forgeries. However, inserting a completely new set of frames in video or deleting some frames from video falls under inter-frame forgery. Several approaches are discovered by scientists and forensic investigators for detecting both types of tampering. (Ng et al., 2006) projected a study on various parameters for detecting tampered images. The proposed scheme specifically works for the identification of doctored video. This scheme is suitable where an immediate authenticity check is needed, in the application where detailed knowledge of forgery is not important. Like, before forwarding a video clip, it is important to know whether it is fake or not. So,

one can restrict the forwarding of fake videos. The proposed scheme is the extension of the work proposed by (Song et al., 2016). They worked on a signature analysis of AVI (Audio-Video Interleave) video files and a few MP4 (MPEG-4) files by investigating file format information. We have worked on a similar line for MP4 videos. The main contribution of this paper, we have extended the work to detect the inter-frame type of forgery detection based on detailed atom structure analysis.

Related Work

Tamper detection in the video is detected by the popular technique Optical Flow. (Chao et al., 2018); (Wang et al., 2014);worked to detect inter-frame forgery by a method based on this popular technique. This technique calculates the consistency of photosensitive movement throughout the video. (Bidokhti, n.d.); (Jia et al., 2018)worked to find intra-frame forgery using the same technique. Another widely used technique is based on estimating motion to detect forgeries in the videos. (Conotter et al., 2012);(Raj & Nair, 2013);(Stamm et al., 2012);(Chen et al., 2016);(Feng et al., 2017) adapted motion estimation to discover the altering in videos. (Singh & Aggarwal, 2017) contributed work to detect copy-move attacks by using three techniques based on sensor pattern noise, Color Filter Array, and Hausdorff distance. (D'Amiano, L., Cozzolino, D., Poggi, G., 2015; D'Amiano et al., 2019) suggested a method for the recognition of copy-move occurrence especially for additive and occlusive changes, feature extraction is carried out with densely filled method, and then feature matching is applied followed by post-processing. This patch match based technique is implemented and tested on a handcrafted dataset. (Wei et al., 2019) suggested content-based analysis and done with single and multiple scales.

(Ghimire et al., 2020) proposed hashing based block chain technology for integrity verification videos. The system was found more robust against two attacks with excellent computational complexity. But this technique belongs to an active approach because reference video hash is needed for integrity identification. (Li et al., 2019) suggested a correlation method for the detection and localization of inter-frame video forgery. Feature extraction and tamper point detection was carried out using k-means clustering. However, this system fails to detect frame deletion precisely. The new GOP-based features extraction method is proposed by (Jiang et al., 2019) for tamper recognition in HVEC compressed videos. Classification of these GOP features is carried out by using a multilayer perceptron method. The system was found more sustainable in several encoding circumstances. (Zampoglou et al., 2019) suggested a filter-based approach to discover video tampering. Deep learning is used for the classification of tampered and original videos. However, this system was tested for limited conditions under the hypothesis.

Methods presently in use check the entire frames present in the video, which takes much of the time to check the decency of the video. Such methods are more useful in cases where people wish to know the exact tampering and detail information about the forgery i.e. tamper detection and localization. Such techniques need to solve the case of fake video clips in courtrooms, where the exact location and detection of tampering need to know. On the other hand in some situations where a quick result is expected, just identifying the trustworthiness of video is required. For example, 'What's app'- the main medium of forwarding images and videos. People just need to check whether the video is fake or real before clicking on the "Forward button". In such cases only identification of forgery is important. This paper focuses on the second case, where a quick result about forgery is important and no in-depth forgery location and detection is necessary. The proposed approach works on the VFI ("Video Forgery Identification") scheme. The next sections of the paper help to understand MPEG-4 video file format, database creation, proposed VFI scheme, experimentation, and conclusions.

MPEG-4 Video File Format

MPEG-4 videos are selected for the implementation of the proposed method. This section gives details about the MP-4 file format and structure to understand the proposed scheme in an easier way. Generally, MP4 and AVI are the most popularly used format to store video multimedia. According to (Gloe et al., 2014) file format information and metadata are another source of forensic evidence but have generally received less attention. The proposed approach explores MP4 video files since less work is carried out in this file format. The basic building block of the MP4 video file is 'box' or an 'atom'. The contents in the video file are arranged in this variety of atoms, which has some unique importance. Each atom carries specific information about the video. These atoms are recognized by unique four-ASCII character codes (generally users can identify it by lowercase letters). These ASCII character codes are further defined by a 32-bit unsigned integer to store in memory. In MP-4 file the first atom is generally 'ftype' which stores the type of the file. Another atom is 'meta' atom contains all metadata information about the file. Table 1 lists information about some popular atoms of the mp4 file in QTFF (Quick time file format).

Table 1.Basic type of atoms in MP-4 containers

Atom	Function
ftype	Stores type of the file. e.g .mp4 , .m4a etc.
mdat	This is a top-level atom that contains information such as the beginning of media data. This atom covers a large portion of the file and made up of chunks. It also provides all raw information of audio, visual stream.

moov	This atom is sub atom (child atom) of 'mdat'. It states decoding parameters essential for construing 'mdat' data stream correctly.
uuid	User data atom. It is available for the user for setting some user-defined fields. Especially it is used for editing extensible metadata information randomly.
meta	This atom has child atoms such as 'hldr' - metadata handler, 'keys'- metadata item keys, 'list'- metadata atom list, 'Ctry'-Country name, 'Lang'- Language, 'free'- free space atom.
udta	It is data atom for user stores extended information like movie's window position, playback characteristics.
free	This atom is mainly used for padding purposes. In metadata atom, it is used to spare the space for future add-ons.
data	This carries the actual contents of that atom. Some atoms don't have data.

Figure 1 shows the general structure of an atom layout. Atoms comprise of a header, trailed by data information of atom. The header contains the name or type of the atom along with the entire size of the atom. Type and size filed take fixed 4 bytes to store this primary information whereas the data filed of an atom contains actual information or contents of video and audio, thus has variable size according to present data. A string of multiple atoms is called an 'Atom Tree'.

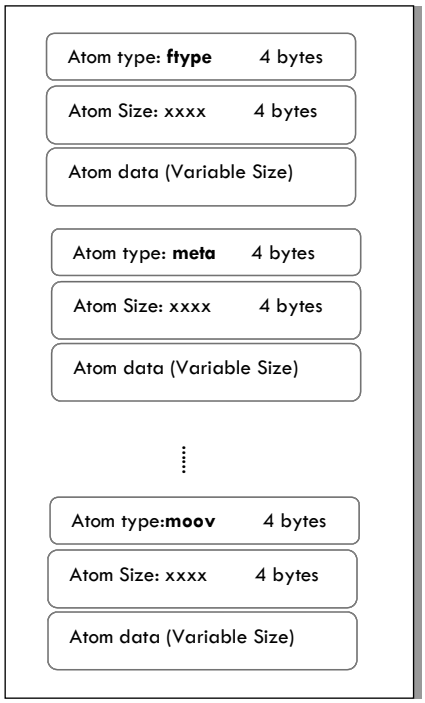

Fig. 1.General structure of Atoms Layout

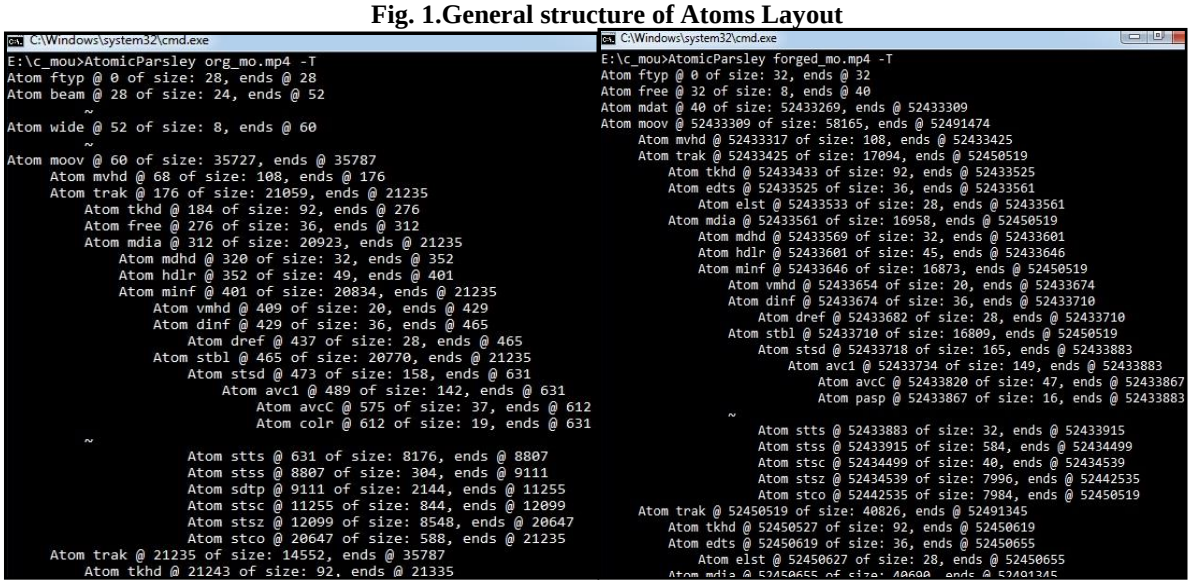

Fig. 2a Atom tree for original video file **Fig. 2b Atom tree for manipulated video file**

Figure 2a illustrates the atom tree for the sample MP4 video files. We have used the existing Atomic Parsley for getting the tree structure of atoms in the file. The first atom is 'type'; the second atom is 'beam' and so on. The proposed algorithm works on the fact that the tree structure of the atoms gets altered when any atom is removed or added to the tree. Basically 'mdat', 'scot' and 'free' atoms are accountable for the same. When any frame is added or removed from the video these atoms modifies the atom tree structure. Fig 2b shows a modified atom tree for the same original file manipulated with a MOVAVI video editor named 'forged_mo.mp4'. According to the analysis, it is seen that the second atom for both files is different. Fig, 2a has 'beam' as a second atom while fig. 2b shows 'free' atom is in the second position. Thus subsequent atoms are also get changed which disturbs the tree structure of the video file. To investigate and analyze this fact, we have developed our Matlab code to extracts the atoms of the input video file. The results of this experimentation are discussed in the 5th section along with a comparison with AtmicParsley software.

Database Creation

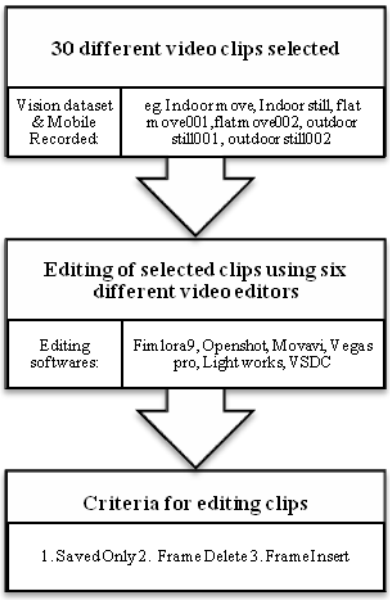

Fig. 3.Forged video database creation process

Two kinds of databases are created for the analysis and implementation of the proposed algorithm.

A. Forged Video dataset

The first database is composed by creating manipulated videos using different video editing software. 10 mp4 video clips from the benchmark dataset are selected for experimental analysis. Six different popularly used video editing suits have been selected for manipulating these original videos. Editors used for proposed work forged dataset construction are Fimlora9, Openshot, Movavi, Vegas pro, Light works, and VSDC. Figure 3 briefs about steps to organize a forged video database. All the source videos are collected from the benchmark VISION dataset proposed by (Shullani et al., 2017). This dataset is exclusively made for forensic investigation and experimentation. Further, these selected original clips are edited by applying three conditions using said video editors. Editing of original videos is carried out by applying three types like frame insertion, frame deletion, and saved only for producing video forgery. Thus, 10 original mp4 videos passed through three editing criteria using six different editors. Hence, 10 x 3 x 6 = 180 manipulated videos are made for analysis and experimentation.

B. Signature Dataset (Atom Pattern Dataset)

The second database is important, which stands as a crucial step in the proposed VFI scheme. This dataset is created after a detailed analysis of the atom tree structure. As we know, every video has a unique atom tree. For the construction of this dataset, the first manual analysis is carried out by creating an excel sheet. This sheet contains atom trees of original and manipulated videos. To be more specific, the atom tree of all original video and its manipulated version for three criteria are recorded in the sheet. This record is maintained for every single video editor selected for implementation. So, a total of six sheets (one for each editor) containing 180atom trees for each criterion are maintained. Further investigation is carried out based on these maintained records to generate a unique atom pattern. This unique pattern is called 'Signature', which is figured out for a specific editor by analyzing the atom pattern structure for each clip passed by that particular editor. Figure 4 shows the scheme to create a database of the atom pattern of manipulated videos. For composing this database we have used the same code developed for discovering atoms as discussed in section 2. This code extracts atoms of manipulated videos. Then feature extraction is carried out by looking at the details of the tree structure and pattern analysis. The signature is generated based on the unique pattern of atoms. Finally, this signature is stored in the database. Consequently, the same process is used to make the signature of forged video for particular editing software.

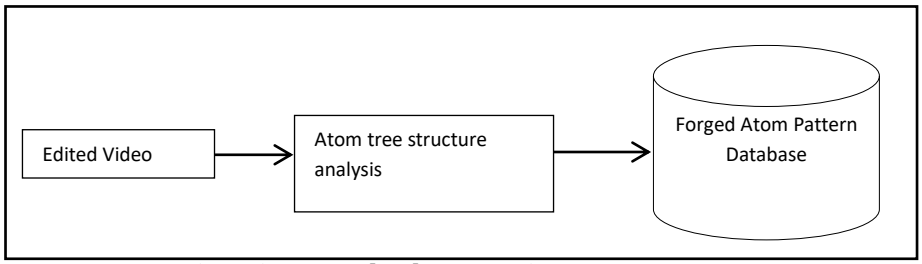

Fig.4.Signature database creation process

Proposed Video Forgery Identification Process

a) ***Proposed system block schematic***

The block diagram of the proposed scheme is depicted in figure 5. The proposed video forgery identification is a scheme carried out in two stages. The first system is designed to generate Boolean output using a forgery identification algorithm. The input video file is passed through the proposed VFI algorithm for atom extraction and atom structure analysis. At the second stage type of forgery will get classified by performing second-level atom pattern recognition. The complete system is designed to generate editor specific output for both stages.

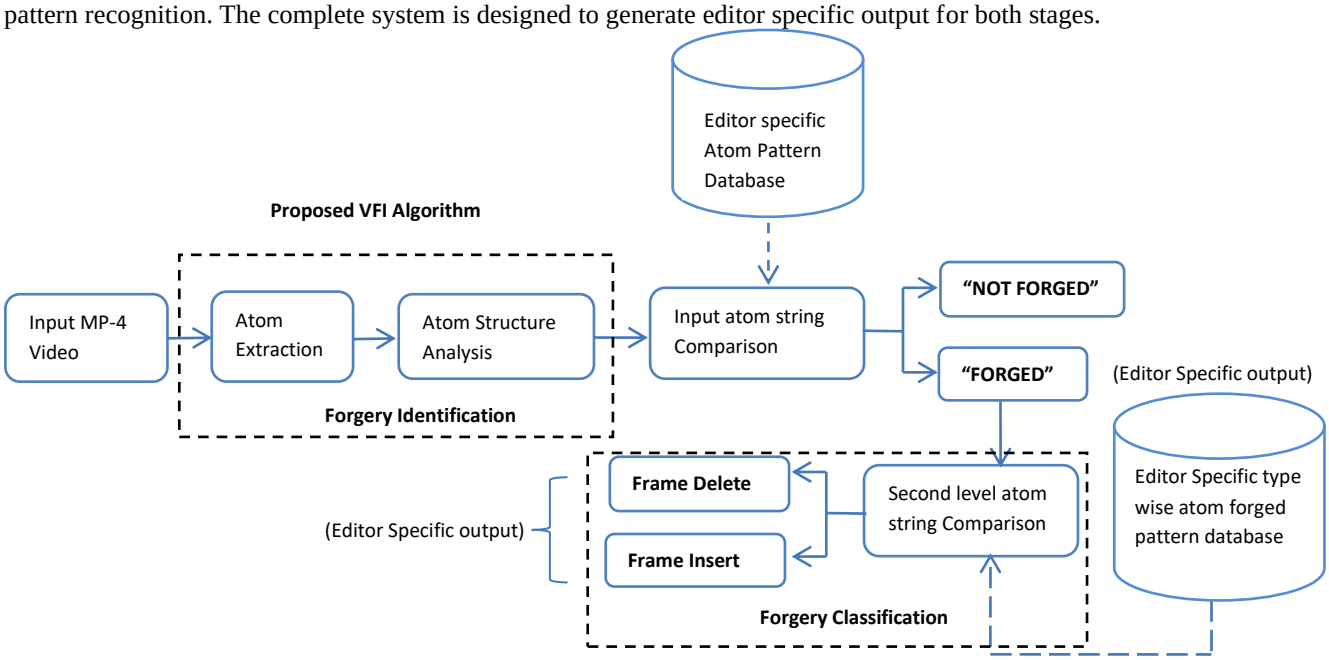

Fig. 5.Proposed System Block Schematic

b) Three-level checks for forgery identification

Three-level checks are applied by the proposed algorithm for the identification of forgery in the given input video file. Mathematical rules discussed in this section are concerning only one video editor. The same checks are applied for the remaining five editors used for experimental analysis of the proposed algorithm. Let FS1, FS2, FS3 are three signatures (forged atom pattern database figured out by doing atom structure analysis) for one of the editors out of six. SV1, SV2, SV3 are sizes of the signatures respectively for FS1, FS2, FS3. Let IPS is the input string of the current input video file.

1. First level check

At this level initially, IPS is get matched with each block of FS1. If the comparison is successful then string matched is declared otherwise, IPS are compared with the rest two strings FS2 & FS3.

2. Second level Check

Three sub-rules checks are performed by comparing the sizes of the signature and input file. ***Check*** & ***check1*** are final checks used to declare the final result. Here, check and check1 are used for counting checks.

a) If SV1 > IPS_size, Then rule1 satisfies and ***Check*** is incremented otherwise ***check1***
b) If SV1 < IPS_size, Then rule2 is checked and ***check1***is incremented otherwise ***check***
c) If SV1= IPS_size but last blocks are not matching & SV1= IPS_size & last blocks are matching then ***Check*** is incremented otherwise ***check1***

Thus, the same steps are repeated for the other two SV2 & SV3 signature matches.

3. Third level Check

At this stage, final count checks are performed by checking the last block with 'mdat' and ' moov' atoms.
If the last block of the input string is > mdat and < moov of FS1, FS2, and FS3 Then finally,

If ***Check ≠ 0*** then the final result is declared as VIDEO IS FORGED and If ***Check1==3***then the final result is declared as VIDEO IS NOT FORGED

c) Proposed VFI Algorithm

Fig. 6 depicts the proposed VFI algorithm to identify the authenticity of a video based on the signature of the forged video database. In the beginning, we have stored the signature of all six video editing software's as discussed in section 4. When we are giving input files, the program reads all atoms and check for the validation of the mp4 file. If the file is in a format other than mp4 then it flashes an error. After validation, it compares the atom string with the stored pattern of the atom database (signature of editing software). If the atoms of the input video match with the database signature, Then it determines that the video has been altered. This output is further classified and displayed according to the specific editor.

Algorithm (Forgery identification and Forgery Classification)

Input: mp4 Video
Output: 1. Forged/Not Forged
2. Frame Insert/Frame Delete

Steps:

1: Start
2: Open file
3: Read .mp4 file
4: Search for atoms of .mp4 file
5: Sort the atom strings
6: Check for parent & child atoms (find subset of parent atom)
7: Concatenate parent & child (subset) atoms
8: Validate step 8 output with stored .mp4 atoms
9: Is string validated?
No - ***Flash Error***
Yes- Go to step no. 10

Validation of mp4 format

10: Compare extracted features with trained model (Signature of editing software)
Comparison Successful?
11: No – Display ***'VIDEO IS NOT FORGED'***
Yes- Display ***'VIDEO IS FORGED'***
12: If video is forged go for classification
13: Compare extracted features with trained model (Signature of editing software: Type wise)
14: Display classified output according to corresponding match as:
1. Frame Insertion 2. Frame Deletion
15: End

Classification & Identification

Fig. 6.Proposed VFI algorithm

Experimental Results and Discussion

The proposed algorithm is executed using Matlab 2018, Core i3 Processor, and 4GB memory. The experimentation is carried out at three different phases; Atom discovery comparison, Forgery identification, and Forgery classification. In the starting phase of research, we have started experimentation by extracting atoms for analysis. Then we have developed an atom structure analysis code for the training signature database. The same database we have extended for a specific editor for frame delete and insert videos.

a) Comparative scheme for atom discovery

At the outset, we have started the experimentation by discovering atoms of the video. Initially, atoms are extracted using existing software and then by our Matlab code. Then the results of the proposed algorithm are matched with existing 'AtomicParsley' software. Basically 'AtomicParsley' is a lightweight command-line program for reading, parsing, and setting metadata into MPEG-4 files. Figure 7 shows the results of AtomicParsley and the proposed algorithm for the discovery of atoms for input mp4 video. For the investigation, a total of 25 mp4 video files were selected from the VISION standard dataset. As shown in figure 5 video input '6original.mp4' is given as input for both the software. More atoms are discovered using Matlab code than that of existing software AtomicParsley. It is seen that AtomicParsley discovers 50 atoms for the input file whereas Matlab code discovers more atom discovery i.e. 55 atoms for the same video input file. Table 2 and figure 8 shows a comparison for sample 10 different mp4 videos.

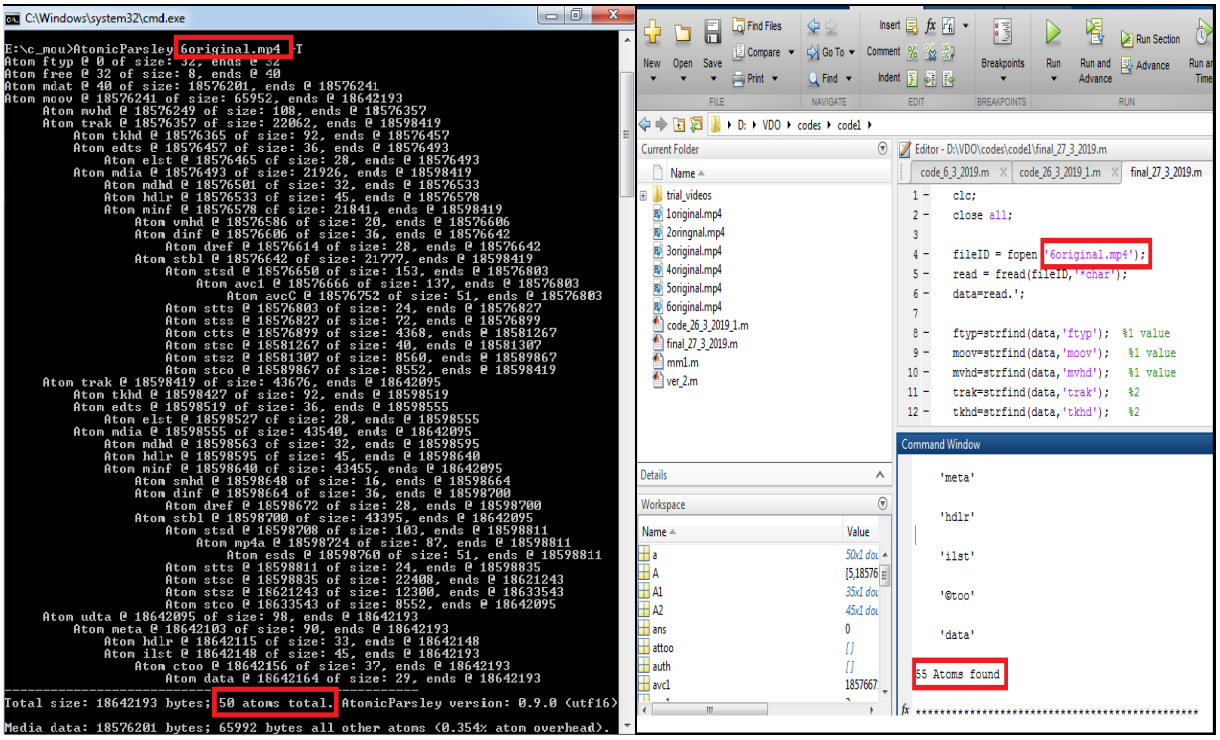

Fig. 7.Atom Discovery AtomicParsley Vs Proposed algorithm

Table 2.Comparative scheme for Atom Discovery

Sr. No	Video Clip (Vision Dataset)	AtomicParsley	Proposed algorithm
1	D01_V_indoorWA_move_0001	44	47
2	D01_V_indoorWA_panrot_0001	44	47
3	D01_V_indoorWA_still_0002	44	48
4	D01_V_indoorWA_panrot_0001	44	47
5	D01_V_indoorWA_still_0002	44	48
6	D12_V_flat_move_0001	55	59
7	D12_V_flat_move_0002	55	63
8	D12_V_outdoorYT_move_0001	50	64
9	D12_V_outdoorYT_still_0001	50	63
10	D12_V_outdoorYT_still_0002	50	55

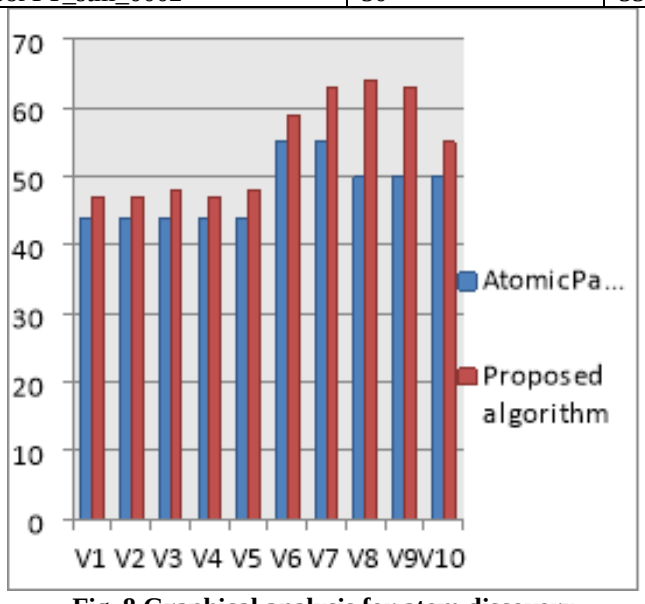

Fig. 8.Graphical analysis for atom discovery

b) Forgery Identification

The proposed algorithm is implemented by testing a total of 50 video sequences for the identification of video forgery. The algorithm works successfully for all input videos by generating editor specific output for different six editors involved in experimentation. Song et al. mainly focused on metadata fields and header file information for investigation targeting AVI file formats rendering Boolean output. Also, they have performed a few experiments on MP-4 video files. We have carried out the expansion of MP4 videos on a similar line to that of previous work. However, the proposed system is completely targeting MP4 file formats by investigating a complete atom tree structure analysis. In addition to this, we have extended this research for the classification of the type of forgery. Moreover, the results of the proposed system are validated and tested by comparing it with existing software. The proposed system found better atom discovery than AtomicParsley with minimum time evaluation. Figure 9 depicts sample results for Sony-Vegas (Popular video editor) and Figure 10 depicts sample results Movavi (Video editor) output with the time elapsed. The system also correctly identifies the original video for all input video sequences. The sample result for correct identification of the original video is shown in figure 11.Thus the accuracy of the proposed algorithm is 83%.

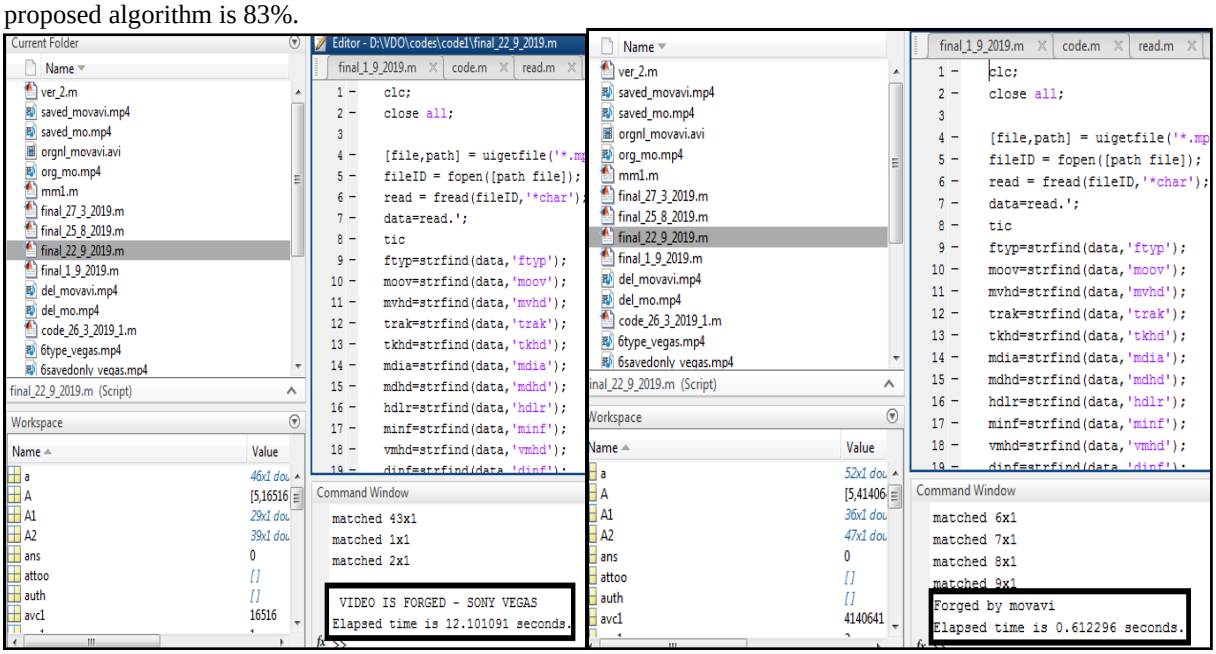

Fig. 9.Editor specific output for Sony Vegas **Fig. 10.Editor specific output for Movavi**

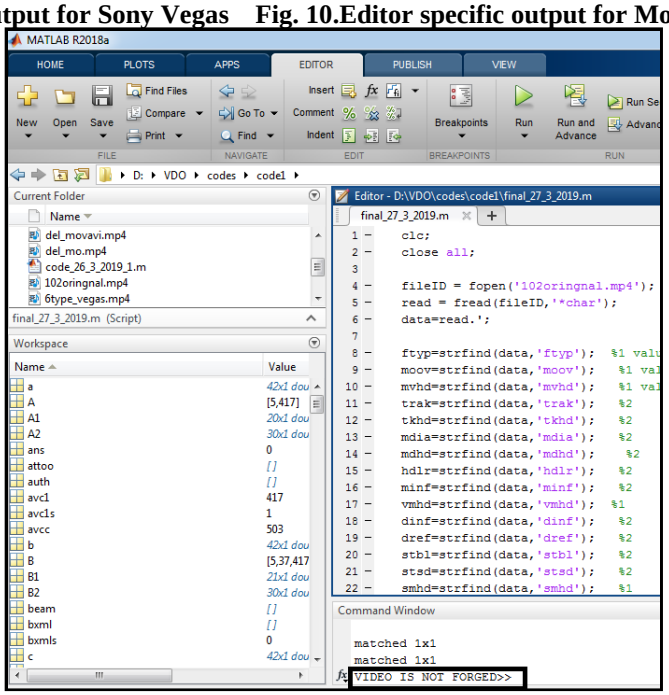

Fig. 11.Identification of Original Video

c) Forgery Classification

The inter-frame forgery is made for manipulating video using well-known video editing software's. The proposed algorithm is extended further to classify the type of forgery. This is a unique method to classify the type of forgery based on atom tree structure analysis. As discussed earlier for manipulation of videos we have adopted frame insertion and frame deletion forgery. Figure 12 illustrates sample results for frame deletion for the Open shot editor. Another sample result is shown in figure 13 for the Filmora9 video editor with frame insertion type forgery. We have tested the proposed algorithm for a total of 100videos. The algorithm correctly classifies for both types of inter-frame forgery for all six video editors.

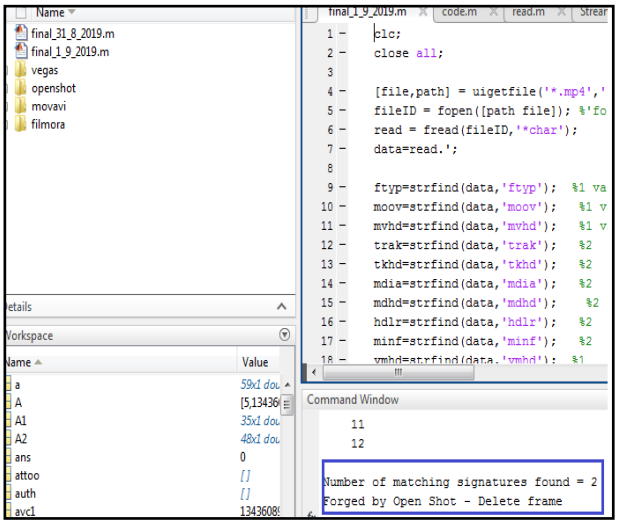

Fig. 12.Frame delete by Open shot

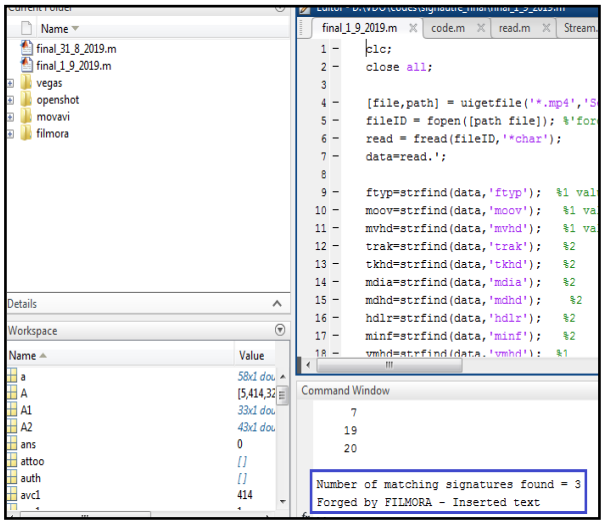

Fig. 13.Frame Insert by Filmora

Statistical and Comparative Analysis of Proposed Scheme

Evaluation of the proposed system is carried out by using confusion matrix performance parameters. Table 3 shows the parameters for performance assessment. For the boolean output of forgery identification, we have tested a total of50 videos, and accuracy is calculated for a specific editor involved in the experimental analysis. Table 4 shows editor specific accuracy of the proposed algorithm. The average accuracy forthe identification of the proposed VFI scheme is 78%. We have an extended database of forged videos for performing experimentation on forgery type classification. Table 5 and Table 6 show the evaluation of performance parameters for 'Frame Delete' and 'Frame Insertion' forgery respectively.A total of 100 of 180 videos are tested for efficiency check of proposed editor specific forgery classification technique. The system gives almost 83% accuracy for both frame delete and frame insert. Accuracy is calculated by : (TP+TN) / (TP+TN+FP+FN). The performance of the proposed scheme is depicted in figure 14 by plotting accuracy for the various editors.

Table 3.Performance parameters

Parameter	**Interpretation**
TP	We tested for manipulated videos and they found manipulated.
TN	We tested for original videos but they found manipulated.
FP	We tested for manipulated but they found original.
FN	We tested for original and they found original.

Table 4.Statistical analysis for Boolean output

Parameter	**TP**	**TN**	**FN**	**FP**	**Accuracy (%)**
Editor 1: Sony Vegas	35	5	5	5	80
Editor 2: Movavi	34	6	5	5	80
Editor 3: OpenShot	32	5	8	5	74
Editor 4: Filmora9	38	2	5	5	80
Editor 5: VSDC	33	4	6	7	74
Editor 6: LightWorks	36	5	4	5	82

Table 5.Statistical analysis for **Frame Delete**

Parameter	**TP**	**TN**	**FN**	**FP**	**Accuracy (%)**
Editor 1: Sony Vegas	77	7	7	9	84
Editor 2: Movavi	75	7	8	10	82
Editor 3: OpenShot	74	9	7	10	83

Editor 4: Filmora9	79	6	10	5	85
Editor 5: VSDC	79	7	8	10	83
Editor 6: LightWorks	80	5	5	10	85

Table 6.Statistical analysis for Frame Insert

Parameter	TP	TN	FN	FP	Accuracy (%)
Editor 1: Sony Vegas	75	7	9	9	82
Editor 2: Movavi	77	7	8	8	84
Editor 3: OpenShot	76	6	9	9	82
Editor 4: Filmora9	75	8	10	7	83
Editor 5: VSDC	74	9	7	10	83
Editor 6: LightWorks	77	5	8	10	82

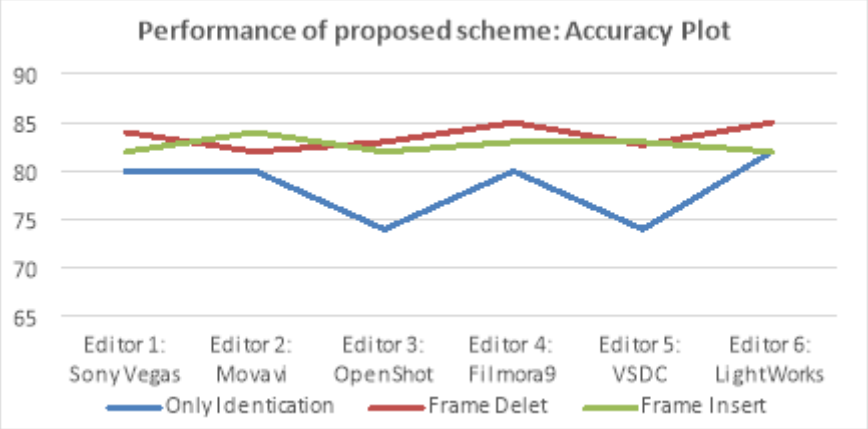

Fig. 14.Accuracy Plot: Ediotrwise Performance

The proposed research is matched with the existing method implemented by Song et al.(2016). They have proposed signature analysis for AVI and a few MP4 video file formats. The proposed scheme is implemented for mp4 file format for generating boolean output with the additional classification of forgery type. Table 7 shows the comparison of the proposed scheme with the previous research. Softwares

Table 7.Comparison with state-of-art

State-of-art	Dataset	Software's used	File Type	Output	Computational Time
Song et al. (2016)	VEDR	Sony VEGAS, Adobe Premiere, Edius, Avid MC, and Avid Studio	AVI, MP4	Binary Output	Not mentioned
Proposed scheme	VISION	Sony VEGAS, Open shot, Filmora9, VSDC, Light works, and Movavi	MP4	Binary output Plus classified forgery type (Frame Insert/Delete)	Average Time Less than 15 Seconds

Conclusions and Future Scope

This paper presents a fast video forgery identification scheme that overcomes a drawback of a traditional system by achieving both speed and accuracy. The system is useful for applications where only the identification of forgery is required. The proposed system is mainly made by keeping in mind social media applications like What's app, Facebook messenger, etc. where before forwarding any video, just integrity of video needs to check. Atom tree structure analysis is used for identification and classification. A total of six popularly used video editors are used for creating manipulated videos. 10 original videos are passed through the editor by applying three conditions. Thus a total of 10*6*3= 180 forged videos is created for experimentation. Experimentation and analysis show that the atom tree of each video has a distinctive structure. When the media contents of the file are get modified, it is reflected in the atom patterns of that file. Besides the atom tree, metadata fields also change when the video gets manipulate. The proposed system works efficiently for both fake and original video identification along with the type of forgery. The future work for this research work is looking to increase databases for more video editing software's especially editors widely used in mobile phones.

References

1. Bidokhti, A. (n.d.). *Detection of Regional Copy / Move Forgery in MPEG Videos using Optical Flow*.
2. Chao, J., Jiang, X., & Sun, T. (2018). A Novel Video Inter-frame Forgery Model Detection. *Springer-Verlag Berlin Heidelberg*, 267-281.
3. Chen, S., Tan, S., Li, B., & Huang, J. (2016). Automatic Detection of Object-Based Forgery in Advanced Video. *IEEE Transactions on Circuits and Systems for Video Technology*, *26*(11), 2138-2151. https://doi.org/10.1109/TCSVT.2015.2473436.
4. Conotter, V., Brien, J. F. O., & Farid, H. (2012). *Exposing Digital Forgeries in Ballistic Motion*. *7*(1), 283-296.
5. D'Amiano, L., Cozzolino, D., Poggi, G., & V. (2015). VIDEO FORGERY DETECTION AND LOCALIZATION BASED ON 3D PATCHMATCH L . D ' Amiano , D . Cozzolino , G . Poggi and L . Verdoliva. *Multimedia & Expo Workshops (ICMEW) IEEE International Conference On*, 1-6.
6. D'Amiano, L., Cozzolino, D., Poggi, G., & Verdoliva, L. (2019). A PatchMatch-Based Dense-Field Algorithm for Video Copy-Move Detection and Localization. *IEEE Transactions on Circuits and Systems for Video Technology*, *29*(3), 669-682. https://doi.org/10.1109/TCSVT.2018.2804768.
7. Feng, C., Xu, Z., Jia, S., Zhang, W., & Xu, Y. (2017). Motion-Adaptive Frame Deletion Detection for Digital Video Forensics. *IEEE Transactions on Circuits and Systems for Video Technology*, *27*(12), 2543-2554. https://doi.org/10.1109/TCSVT.2016.2593612.
8. Ghimire, S., Choi, J. Y., & Lee, B. (2020). Using Blockchain for Improved Video Integrity Verification. *IEEE Transactions on Multimedia*, *22*(1), 108-121. https://doi.org/ 10.1109/TMM.2019.2925961.
9. Gloe, T., Fischer, A., & Kirchner, M. (2014). Forensic analysis of video fi le formats. *Digital Investigation*, *11*, S68-S76. https://doi.org/10.1016/j.diin.2014.03.009.
10. Jia, S., Xu, Z., Wang, H., Feng, C., & Wang, T. (2018). Coarse-to-Fine Copy-Move Forgery Detection for Video Forensics. *IEEE Access*, *6*, 25323-25335. https://doi.org/10.1109/ACCESS.2018.2819624.
11. Jiang, X., He, P., Sun, T., & Wang, R. (2019). Detection of Double Compressed HEVC Videos Using GOP-Based PU Type Statistics. *IEEE Access*, *7*, 95364-95375. https://doi.org/10.1109/ACCESS.2019.2928857.
12. Li, Q., Wang, R., & Xu, D. (2019). Detection of double compression in HEVC videos based on TU size and quantised DCT coefficients. *IET Information Security*, *13*(1), 1-6. https://doi.org/10.1049/iet-ifs.2017.0555.
13. Ng, T. T., Chang, S. F., Lin, C. Y., & Sun, Q. (2006). Passive-blind Image Forensics. *Multimedia Security Technologies for Digital Rights Management*, 383-412. https://doi.org/10.1016/B978-012369476-8/50017-8.
14. Raj, J., & Nair, M. S. (2013). Forgery detection in ballistic motion videos using motion estimation and modelling. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *8284 LNAI*, 328-339. https://doi.org/10.1007/978-3-319-03844-5_34.
15. Shullani, D., Fontani, M., Iuliani, M., Shaya, O. Al, & Piva, A. (2017). VISION: a video and image dataset for source identification. *Eurasip Journal on Information Security*, *2017*(1). https://doi.org/10.1186/s13635-017-0067-2.
16. Singh, R. D., & Aggarwal, N. (2017). Detection and localization of copy-paste forgeries in digital videos. *Forensic Science International*, *281*, 75-91. https://doi.org/ 10.1016/j.forsciint.2017.10.028.
17. Song, J., Lee, K., Lee, W. Y., & Lee, H. (2016). Integrity verification of the ordered data structures in manipulated video content. *Digital Investigation*, *18*, 1-7. https://doi.org/10.1016/j.diin.2016.06.001.
18. Stamm, M. C., Lin, W. S., & Liu, K. J. R. (2012). *Temporal Forensics and Anti-Forensics for Motion Compensated Video*. *7*(4), 1315-1329.
19. Wang, Q., Li, Z., Zhang, Z., & Ma, Q. (2014). Video inter-frame forgery identification based on optical flow consistency. *Sensors and Transducers*, *166*(3), 229-234.
20. Wei, W., Fan, X., Song, H., & Wang, H. (2019). Video tamper detection based on multi-scale mutual information. *Multimedia Tools and Applications*, *78*(19), 27109-27126. https://doi.org/10.1007/s11042-017-5083-1.
21. Zampoglou, M., Markatopoulou, F., Mercier, G., Touska, D., Apostolidis, E., Papadopoulos, S., Cozien, R., Patras, I., Mezaris, V., & Kompatsiaris, I. (2019). Detecting Tampered Videos with Multimedia Forensics and Deep Learning. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *11295 LNCS*(Mmm), 374-386. https://doi.org/10.1007/978-3-030-05710-7_31.