I. J. Engineering and Manufacturing, **2025, 3, 32-42**
Published Online on June 8, 2025 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijem.2025.03.03

Forensic Software Tool for Detecting JPEG Double Compression Using an Adaptive Quantization Table Database

Iyad Ramlawy
Department of Natural Engineering and Technology Sciences, Arab American University (AAUP), Ramallah, Palestine
E-mail: iyad.ramlawy@gmail.com
ORCID iD: https://orcid.org/0000-0002-7791-3452

Yaman Salem*
Department of Natural Engineering and Technology Sciences, Arab American University (AAUP), Ramallah, Palestine
E-mail: y.salem3@student.aaup.edu
ORCID iD: https://orcid.org/0000-0002-4737-5112
*Corresponding author

Layth Abuarram
Department of Natural Engineering and Technology Sciences, Arab American University (AAUP), Ramallah, Palestine
E-mail: laythgmi9@gmail.com
ORCID iD: https://orcid.org/0000-0001-8341-5209

Muath Sabha
Department of Data Science, Arab American University (AAUP), Ramallah, Palestine
E-mail: muath.sabha@aaup.edu
ORCID iD: https://orcid.org/0000-0002-0496-6502

Received: 04 September, 2024; Revised: 01 November, 2024; Accepted: 19 January, 2025; Published: 08 June, 2025

Abstract: Most digital forensic investigations involve images presented as evidence. One of the common problems of these investigations is to prove the image's originality or, as a matter of fact, its manipulation. One of the guaranteed approaches to prove image forgery is JPEG double compressions. Double compression happens if a JPEG image is manipulated and saved again. Thus, the binaries of the image will be changed based on a "previous" quantization table. This paper presents a practical approach to detecting manipulated images using double JPEG compression analysis, implemented in a newly developed software tool. The method relies on an adaptive database of quantization tables, which stores all possible tables and generates new ones based on varying quality factors of recognized tables. The detection process is conducted through image metadata extraction, allowing analysis without the need for the original non-manipulated image. The tool analyzes the suspected image using chrominance, and luminance quantization tables utilizing the jpegio Python library. The tool recognizes camera sources as well as the programs used for manipulating images with the related compression rate. The tool has demonstrated effectiveness in identifying image manipulation, providing a useful tool for digital forensic investigations. The tool identified 96% of modified images whereas the other 4% identified as false positives. The tool fixes the false positives by extracting the software information from the image metadata. With a rich sources database, forensic examiners can use the proposed tool to detect manipulated evidence images using the evidence image only.

Index Terms: Digital Image Forgeries, JPEG, Double Compression, Quantization Table (QT), Database, Forensic Software Tool

1. Introduction

In the fourth industrial revolution (IR 4.0), hundreds or even thousands of images are exchanged daily, in addition, billions of images are shared each day on social networks, moreover, millions of cameras are spread everywhere

recording a massive number of images [1], further integrating the Internet of Things (IoT) with cameras and images [2], however, the integrity of digital images cannot be trusted anymore because of the advance image editing tools, therefore, image forensics gained attention from lots of researchers to improve and develop techniques for image forgeries detection [1].

There are many techniques used by investigators to identify tampered images, basically, image authentication techniques are categorized into two types: passive authentication and active/blind authentication [3]. Passive authentication confirms image authenticity by analyzing the intrinsic information of images. On the other hand, active authentication, confirms image authenticity by inserting additional data into images [3]. The passive technique is divided into two categories: the first one is source device identification based, which depends on a device fingerprint left on an image by acquisition steps such as lens aberration techniques, CFA Interpolation techniques, and sensor imperfections techniques. The second type is the tampering operation which is further divided into dependent and independent. The dependent base depends on the type of forgery carried out on the image, on the other hand, the independent base depends on artifact traces left during image processing such as compression. Depending on the way used in producing forged images, image manipulation techniques can be categorized into three categories: copy-move forgery, image splicing, and image retouching [4,5]. Fig. 1 concludes the categories of image forgery approaches and algorithms.

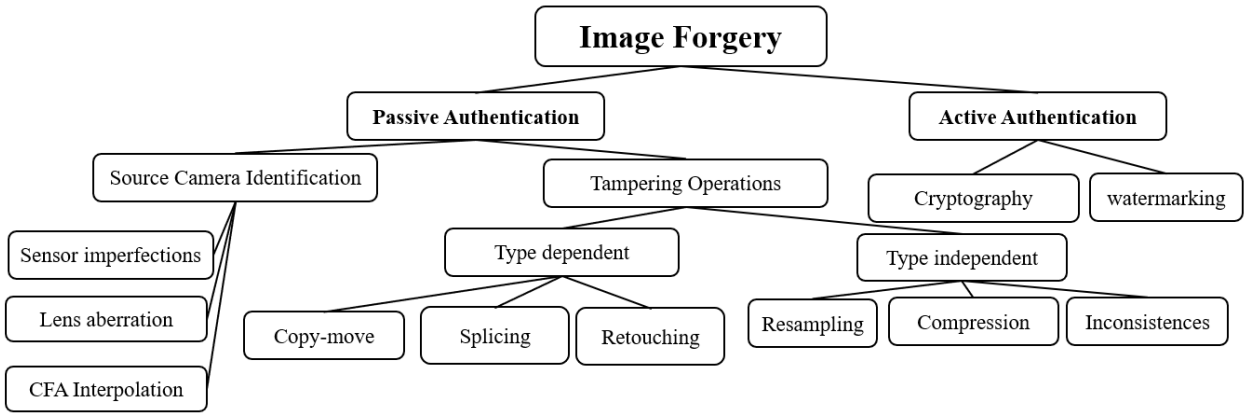


Fig. 1. Categories of image forgery approaches

The primary contribution of this paper lies in the development of a software tool, publicly available and accessible [6], for detecting manipulated images through double JPEG compression analysis, leveraging an adaptive database of quantization tables that dynamically expands to account for varying quality factors, thereby enhancing the accuracy and practicality of image forgery detection.

This study focuses on passive image authentication methods as the investigator can examine the targeted images without the help of any additional information, the proposed approach detects image manipulation with the help of a quantization table of a suspected image and a database of quantization tables. This study has a holistic approach as chrominance and luminance quantization tables were included in the analysis process, moreover, it recognizes camera sources as well as the programs used for editing images with related compression rates. An additional process is created to verify results by extracting related META data from the image.

The rest of the paper is structured as follows: section 2, states a background, and section 3 states related work. This is followed by the proposed approach in section 4. In section 5, the results and discussion are highlighted. Finally, section 6 contains the conclusion and future work.

2. Background

IoT devices and electronic devices have developed greatly, and their type has varied depending on the target group for which the devices are manufactured [7]. With this great prosperity in the positive development and facilitating the lives of individuals, it is not without the world of digital crime [7]. Criminal activities in which electronic devices and digital information are used as tools or means are known as digital crimes [8]. Most of these cyber security threats originate from users' unconcerned behaviors, and attackers are well aware of this attitude, exploiting it to break the security chain [9]. As a result, privacy and personal data have become prime targets, with individuals increasingly vulnerable to data breaches, identity theft, and unauthorized access [10]. Dealing with digital crimes and trying to elicit digital evidence stored in storage places for analysis is a key focus of digital criminal investigations [8]. Digital forensic analysis has several forms and classifications, and each classification has specialists in this aspect working on the analysis of evidence such as Computer forensics, mobile forensics, Multimedia Forensics, and others [11]. The computer had a very prominent role in the beginning of the digitization world in the development of many programming and storage systems and in the development of other digital devices as well, with this, however, digital

crimes were not free from the negative exploitation of the capabilities of computers and their use as a tool to commit crime to become known later as computer crimes [12]. Keeping pace with the era of globalization, digitization, and technological development, the need to investigate computers and extract evidence stored in them on hard disks or even on Live memory forensics has become a necessity in order to collect evidence and link the sequence of events from existing or deleted stored evidence and retrieve it, this science became known as computer forensics [10,12].

Text, images, graphics, audio, and video play an important role today in the daily life of the individual, as the graphical interfaces with which users of digital devices interact have become a factor that attracts them to interact electronically with them and communicate and communicate with people. Old and DVDs even different interactive applications to merchandising methods and design ads in content to attract customers to their products [14].

With the great increase and development in the field of multimedia and its multiple uses, but it was not without pouring various criminal activities and exploiting the capabilities of multimedia in manipulation and modification for malicious purposes such as plagiarism, theft, forgery, and others, which prompted the specialists of digital criminal investigation to open the door to deepen the structure of multimedia from a change The tone of the voice to the fabrication of the video to the steganography, especially the images from them to investigate and know the places of manipulation by several means, including trying to retrieve the original image through the modified image and examining the places of compression in the image and where the modification was done using the principles of mathematics and programming to find new solutions in detecting fraud and modification, the origin of the image and the ownership of its owners. The process of digital forensic analysis of multimedia is centered on recovering lost parts of it by developing new algorithms, techniques, and tools to detect the manipulations that were carried out on it or to try to identify the source device. The researchers call them both the multimedia manipulation and modification detection scenario and the identification scenario [15].

In a multimedia digital forensic analysis, it is generally assumed that the forensic investigator has no knowledge of the presumed origin. These so-called blind methods usually exploit two important sources of digital artifacts:

- The acquisition device characteristics could be checked for the scenario of manipulation detection or for identification
- Previous processing operations artifacts that could be detected in the scenario of manipulation detection [15]

Images differ from texts as images represent an effective and natural visual means of communication for humans, due to their immediacy and ease of understanding their content. The evolution of a digital picture can be depicted as a series of stages: acquisition, coding, and editing. During the acquisition process, light falls on the object and is then reflected back to the digital camera where it is focused using the lenses on the camera sensor either CMOS or CCD, where the digital image signal is generated.

The light is usually filtered and separated using a color filter array, which is a layer that allows a specific component of light to pass through it to the sensor, such as the cons and the rods in the eye. In order to obtain a digital color image, the sensor output is satisfied to obtain each of the red, green, and blue colors for each pixel respectively. This process is called demosaicing. Additional balancing is done in the camera on the captured image, such as white processing, contrast optimization, image sharpening, and gamma correction. In coding, the processed signal is stored in the camera's memory. In most digital cameras, the image is stored with a compression called first compression, through which researchers know whether the image has been modified or not to detect theft or even try to retrieve it to know the changes that have occurred to the image, usually, JPEG is the format used to save and compress images. In the editing, there may be certain modifications to the images using editing programs such as Photoshop, including rotation, zooming, measurements, etc. as well as color modifications such as contrast, opacity, clarity, or cloning, such as repeating the same part in the image or moving apart from one place to another place [3].

2.1. JPEG Compression

The Double compression [16] of JPEG means that the image was previously compressed as JPEG and was compressed again as JPEG, which is what happens nowadays a lot. Both the first compression and double compression play an important role in the process of ensuring the authenticity, reliability, integrity, and origin of the image, as once it is discovered that the image has been double compressed, this means that the image may have been modified and that it is not original [17].

Compression can be categorized into two categories, the first is Lossy and the second is lossless, In lossless compression, the image is not affected, does not undergo any change, does not lose its information, and the quality of the image is the same as the original without affecting it, medical imagery is a main application of, In Lossy compression, the image loses some of its information, its quality decreases, and it is affected by the compression process [18] Lossy compression can reduce the required data storage space, and it also reduces the time spent at data checkpoints by optimizing the bandwidth for input or output [19].

To forge an original image, the image must first be uncompressed (or first compression from the original source), then apply modifying techniques such as Copy-Move, Splicing, etc., and then recompress the modified image, when JPEG images are compressed, the DCT coefficients are affected and therefore can be exploited to find out that the image has been manipulated [16].

When the input can be represented by a Gauss–Markov source with high correlation, the discrete cosine transform (DCT) is utilized in most picture and video coding standards because the coding performance approaches that of the Karhunen–Loeve transform (KLT)-based coding [20]. By using spectral similarity, Discrete Cosine Transformation optimizes compression output while maintaining picture quality. It entails dividing an n*n picture into sub-images of the same size. Each sub-picture is given a unitary transform. DCT Bit rate is decreased in JPEG to maintain quality as compression ratios in the average level are attained. To begin, the image is partitioned into variable-size blocks in both the vertical and horizontal directions. A bit plane of each picture block is used to reduce statistical redundancy. Post-filtering can be used to remove blocking artifacts in picture decompression. The compression results are far better than JPEG and other methods [18].

Fig. 3 shows the jpeg compression process key steps of the grayscale image special case. The DCT, quantizer, and entropy encoder are the three major components of the JPEG encoder. The original spatial domain signal is converted into the frequency domain signal by applying 2D DCT to the nonoverlapping 8*8 blocks. The quantizer receives the computed DCT coefficients and quantizes them using a defined quantization table. In a zigzag scanning order, the quantized DCT coefficients are ordered and pre-compressed on direct current (dc) coefficients using differential pulse code modulation (DPCM) on ac coefficients, as well as run-length encoding (RLE). To produce the final compressed bit stream, the symbol string is Huffman-coded. We get the final JPEG file after prepending the header [21].

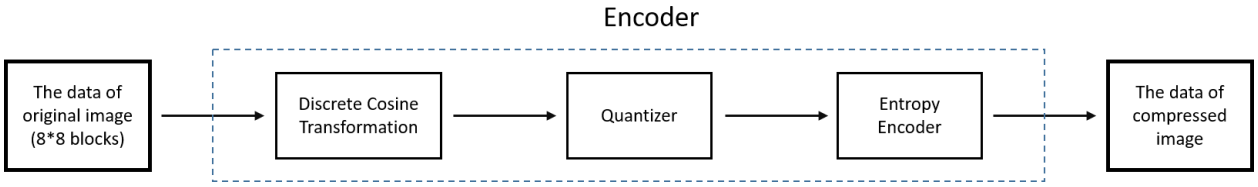


Fig. 2. JPEG compression process

JPEG compression is a lossy technique; hence the output image is not an exact copy of the original one. However, JPEG compression leads to artifacts to detect forged images, the following Fig. 33 concludes JPEG compression and de-compression processes [22].

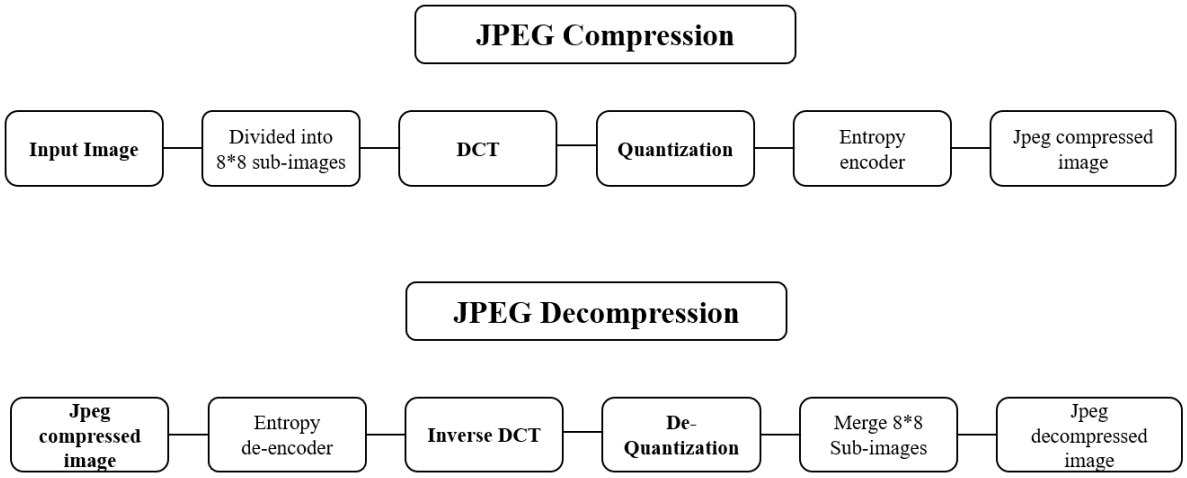


Fig. 3. JPEG compression and De-compression process

2.2. Quantization Table Types

Quantization Tables (QTs) are a fundamental part of the JPEG compression process. QTs help the balance between compression and image quality. By adjusting these tables, the application can control how much detail is preserved and how much the image is compressed [21]. There are two primary quantization tables in JPEG: Luminance (Y) and Chrominance (C). Y is used for the brightness while C is used for the color channels. When the jpeg image is first created, it is compressed using the source imaging device QT. Jpeg double compression refers to the scenario where the image that has already been compressed using JPEG is compressed again using JPEG, which means it is manipulated [22].

JPEG compression employs quantization tables to compress a range of values into a single quantum value. The flexibility of the JPEG standard allowed the customization of quantization tables. Based on a study by [23] the author classified images based on quantization tables into categories: standard tables, extended standard tables, custom fixed tables, and custom adaptive tables.

1) Standard Tables

Use scaled versions of the quantization tables published in the International JPEG Group standard. Can be scaled using Q={1, 2, ..., 99} to create 99 separate tables. Scaling with Q = 0 would produce grossly unusable images. Scaling with Q = 100 would produce a quantization table filled with all ones. Such a table would be indistinguishable from any other base table scaled with Q =100.

2) Extended Standard Tables

These are a special case of the standard tables. They use scaled versions of the JPEG tables but have three tables instead of the two in the standard. The third table is a duplicate of the second.

3) Custom Fixed Tables

Some programs have their own non-JPEG quantization tables. For example, when Adobe Photoshop saves an image as a JPEG it allows the user to select one of 12 quality different settings. Some devices use their custom base quantization table with the JPEG scaling method.

4) Custom Adaptive Tables

These images do not conform to the JPEG standard. They may change, either in part or, between images created by the same device using the same settings.

3. Related Work

Many studies have been proposed to detect manipulated or double-compression images using different approaches, for example, a recent study by [24] depended on double compression artifacts to propose an end-to-end system that detects and localizes spliced areas. It estimated the primary quantization table to recognize spliced areas that come from different images.

A study by [23] proposed a methodology to recognize image source identification based on quantization tables, the researcher mentioned that the method is not perfect and it could be enhanced using the known program signatures.

The researchers in [22] proposed a fast method to detect tampered JPEG images via DCT coefficient analysis, they focused on JPEG images and showed tampered image detection by examining the double quantization effect hidden among histograms of the DCT coefficients, this approach located tampered regions automatically.

The study [25] proposed a model for estimating the first quantization table steps from double-compressed JPEG images. Depending on the fact that, in the JPEG compression standard, the bigger the quantization step is, the lower the quality of the image is. The researchers' approach assumed that the second quantization step is lower than the first one, exploiting the effects of successive quantization followed by de-quantization.

The research [26] proposed a method for predicting the first quantization table from the second quantization table, the study approach investigated image with no need for a training phase and no need for to original image, it explained a technique based on extensive simulation, with the aim to infer the first quantization for a certain numbers of Discrete Cosine Transform (DCT) coefficients exploiting local image statistics.

Another study by [27] exploited the JPEG compression artifacts that came from a single image to detect multiple compression, it analyzed Discrete Cosine Transform (DCT) histograms and estimated the previous quantization table. The researchers stated that its approach was simple with relatively low computational complexity compared to machine learning techniques thus it didn't need to training phase. On the other hand, several studies depended on machine learning techniques for image forensics. For instance, a study by [28] recognized image compression history using deep learning based technique, another study by [29] proposed a CAT-Net model based on convolutional neural network and JPEG compression artifacts for splicing detection and localization. In [30], proposed an image splicing detection and localization scheme that depends on a deep convolutional neural network, the result showed that it has robustness against JPEG compression and it is has a high detection accuracy.

In [31], stated a method for estimating quantization factors from a previous JPEG compressed image, DCT coefficient histogram was involved in the estimation process. A study by [32] stated a method for JPEG compression detection based on the DCT coefficient, the result showed that the proposed method was robust to anti-forensic operations and post-processing.

The proposed approach of this study is different and effective, where the researchers analyze the suspected image using chrominance, and luminance quantization tables, moreover, it recognizes camera sources as well as the programs used for manipulating images with the related compression rate. In addition, it includes a rich database of quantization tables of camera sources and editing programs, this database can be adaptive in an effective approach. The proposed approach is validated by means of extensive experiments showing its superior performance in detecting double compression.

4. The Proposed Approach

The main aim of this study is to detect image double compression or manipulation, the detection approach is conducted using quantization tables analysis to recognize that the image is original which came from a camera, or has been edited by a program, the following Fig. 4 clarifies the detection approach.

The proposed approach is implemented using Python code [6], the following steps clarify the effective approach:

1) ***Step 1****: building a database that contains:*
   - Luminance and chrominance QTs of camera sources.
   - Luminance and chrominance QTs for compressed images by Photoshop with levels (8-9-10-11-12)
   - Luminance and chrominance QTs for compressed images by GIMP with Quality factors (50-60-70-80-85-90-95-99)
   - Luminance and chrominance QTs for compressed images by Krita with Quality factors (80-85-90-95-99)
2) ***Step 2****: extract Q-table of the suspected image and compare it with the databases.*
3) ***Step 3:*** *show the result of image detection*
   - Camera source if it is an original image.
   - Edited program if it is manipulated image with the related quality factor (QF).
4) ***Step 4:*** *add to the database the new quantization tables if not detected.*
5) ***Step 5****: show meta data extraction including the Model and Software.*

The following Table 1 clarifies the camera types used for capturing original images and the software used for compressing them with different quantization tables

Table 1. Camera types and software used

Captured Images camera types	Software used
Samsung s8 (mobile)	Photoshop with levels (8-9-10-11-12) GIMP with QF (80-85-90-95-99) Krita with QF (80-85-90-95-99)
Realmi i6 (mobile)	
Samsung Note20 Ultra (mobile)	
Canon EOS 5D Mark III	
NIKON D7100	

The advantages of this approach are that it can detect double-compressed or manipulated images and, moreover, recognize the program that is used for editing with the quality factor. Additionally, it extracted the image metadata and it depends on an efficient adaptive database. Table 2 clarifies the Sudo Code of the proposed approach.

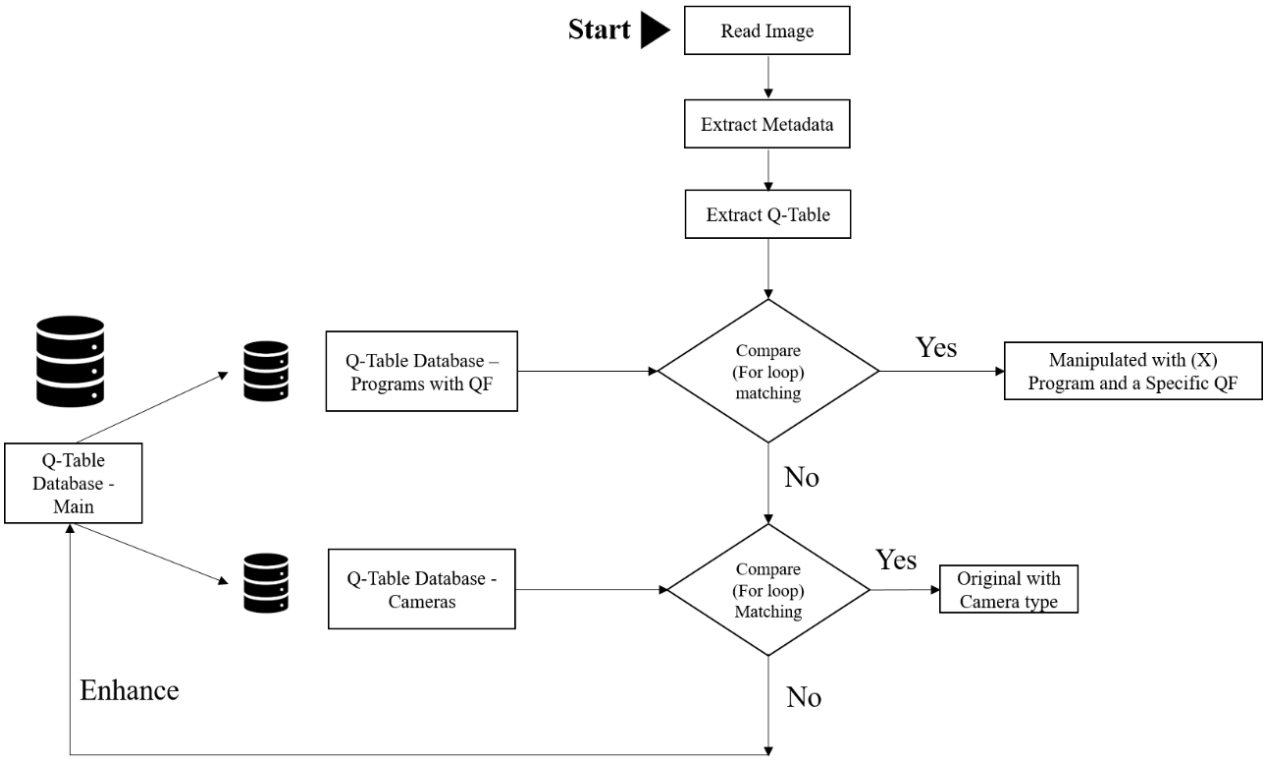


Fig. 4. The proposed detection approach framework

Table 2. Sudo Code of the proposed approach.

Sudo Code of the proposed approach
Build JSON database of Q-tables: Input: Jpeg image: I Output: JSON record: R # Build and load Apps Database: contains all Q-tables of editing App. With the related Q-factor if (I → Manipulated): app ← Software Name QF ← Quality Factor used R ← I[Y], I[C], app, QF Write R to Apps Database #Build and load Sources Database: contains all Q-tables of sources cameras Else if (I → Original): Source ← Device Name Model ← Device Model R ← I[Y], I[C], Source, Model Write R to Sources Database

Input: Read Image (I)
#Extract Chrominance Q-Table of Image
I[C] = {qc1, qc2, ….qc64}
 	
#Extract Luminance Q-Table of Image
I[Y] = {ql1, ql2, ….ql64}
 	
#Compare the chrominance and luminance Q-table of I with the Q-table in database of Programs, if matched then the image is compressed.
 	
For { i =1 to 64
If I[Ci] == I[C] (database of Apps) && I[Yi] == I[Y] (database of Apps)
 	
Print (the image is compressed with a specific Q-factor)
 	
#Compare the chrominance and luminance Q-table of I with the original Q-table in database of cameras, if matched then the image is compressed.
 	
Else {
 	
If I[Ci] == I[C] (database of cameras) && I[Yi] == I[Y] (database of cameras)
Print (the image is original)
 	
}
 	
Read EXIF (I) → Extract Metadata of image (I)
Print (the information from the EXIF)
 	
If the quantization table of suspected image is not found in any of databases {
give the user options to add the Q-table to an appropriate database.
}

The following Table 3 clarifies samples of images taken from Canon and Nikon.

Table 3. Sample of original images before and after compression using different software

Original image	Camera type: Canon EOS 5D Mark III
Compressed images by Photoshop with levels (8-9-10-11-12)	 8 9 10 11 12

compressed images by GIMP with QF (80-85-90-95-99)	 80 85 90 95 99
compressed images by Krita with QF (80-85-90-95-99)	 80 85 90 95 99

Original image	Camera type: NIKON D7100
Compressed images by Photoshop with levels (8-9-10-11-12)	 8 9 10 11 12
compressed images by GIMP with QF (80-85-90-95-99)	 80 85 90 95 99
compressed images by Krita with QF (80-85-90-95-99)	 80 85 90 95 99

5. Results and Discussion

The proposed approach can successfully identify double-compressed or manipulated images. This is based on Quantization Tables (QTs) comparison. The image QTs are compared with databases of QTs of image editing software and QTs of image sources. An additional step is used to verify the results. The verification step includes extracting the metadata of the image as well as presenting the related part of these data to support the QT check result. This step is found to be useful in cases where few applications share the same quantization tables.

The database can be adapted using the tool. The database has one data file for image source QTs collected using nonmodified images captured by devices and another data file for QTs of image manipulation apps collected by nonmodified images created using the apps. The forensic examiner can use the tool to add new image sources QTs records for tools or imaging devices if they are not found in the database and are not detected as double compression. The Apps database record contains the Luminance and Chrominance QTs extracted by the tool, the application name, and the quality factor entered by the examiner. The image sources database record contains the Luminance and Chrominance QTs extracted by the tool, the device name, and the model entered by the examiner.

An example of the source database record will be like: {"QTableY": [[3, 3, 3, 3, 3, 2, 3, 3], [3, 3, 3, 3, 3, 2, 3, 3], [3, 3, 3, 3, 2, 3, 3, 4], [3, 3, 3, 3, 3, 3, 4, 5], [3, 3, 2, 3, 4, 5, 6, 7], [2, 2, 3, 3, 5, 7, 9, 11], [3, 3, 3, 4, 6, 9, 12, 14], [3, 3 , 4, 5, 7, 11, 14, 19]], "QTableC": [[4, 5, 4, 4, 4, 4, 5, 6], [5, 4, 4, 4, 4, 4, 5, 6], [4, 4, 4, 4, 4, 5, 6, 7], [4, 4, 4, 5, 5, 6, 7, 8], [4, 4, 4, 5, 7, 8, 10, 11], [4, 4, 5, 6, 8, 11, 13, 16], [5, 5, 6, 7, 10, 13, 17, 21], [6, 6, 7, 8, 11, 16, 21, 28]], "Source": "Samsung", "Model": "Galaxy Note20 Ultra"}

The proposed approach showed a successful result, where it can recognize the double compression or manipulated image. The JPEG editing software and quality factor were effectively identified. After conducting experiments for several modified and original images, the result can be summarized as follows:

- The database contains a set of quantization tables for several cameras and for several image editing programs with different quality factors. The database stores both luminance and chrominance quantization tables. All Q-tables were stored using JSON NoSQL database.

- Fig. 5 clarifies sample images were modified in both Photoshop and GIMP and were identified as double compressed with their related quality factor value and program name. The success rate was 100%.

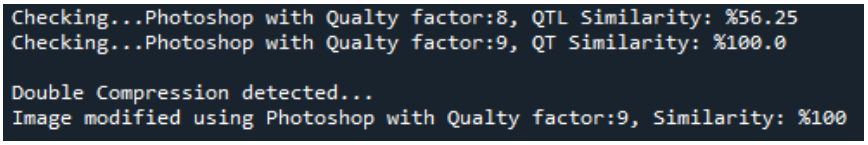


Fig. 5. Detecting double compressed jpg by Photoshop

- All photos taken using the Samsung Galaxy S8, Samsung Note20 Ultra, Canon, and NIKON were recognized. The photo was found to be an original unedited photo and the success rate was 100% as shown in Fig. 6.

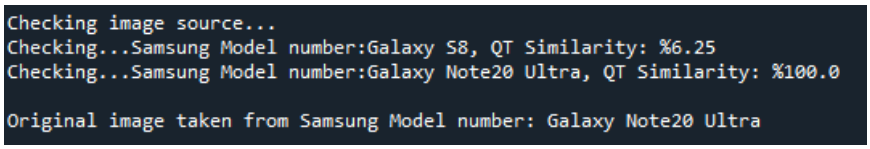


Fig. 6. Identifying image source

- All photos taken using the Realme i6 phone were recognized as modified photos using the Gimp program with a Quality factor of 95. This is due to the use of the same quantization tables in Gimp and for the mobile camera. The solution can identify this case and similar cases as false positives with a warning to notify the forensic analyst. This is shown in the figure below.

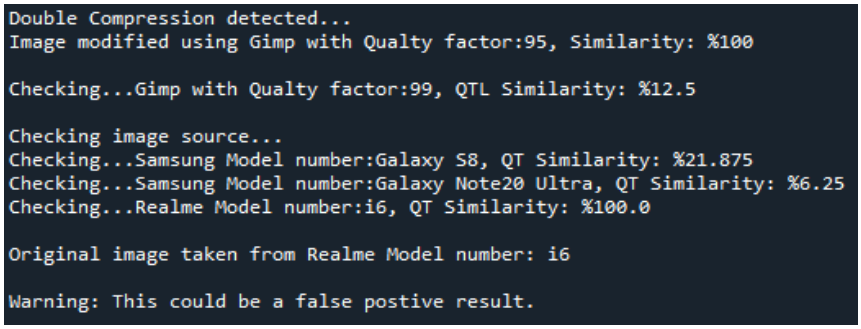


Fig. 7. Reamle i6 false positive result

- To identify the source of the image, especially in false positive results such as in the above point, software information is extracted from the image metadata and displayed. This can help in forensic investigations.

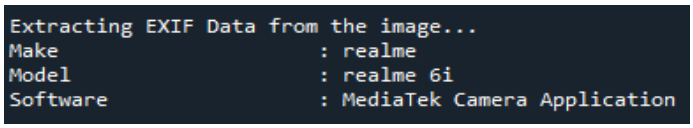


Fig. 8. Metadata extraction

- Several images were modified using the Krita program, the result shows, the quantization tables for Krita and Gimp are identical for all images. Metadata extraction can help in such cases.

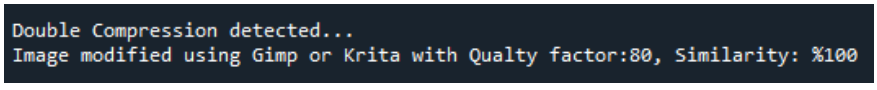


Fig. 9. Gimp and Krita editing software use the same quantization tables

6. Conclusion and Future Work

Double-compressed images can be identified based on their quantization tables. Luminance and chrominance quantization tables' involvement in the similarity comparison process provides more accurate results when compared with databases of QTs. A software tool is developed to simulate the proposed approach. Through the experimental results obtained, it is concluded that the proposed approach is powerful. The tool identified 96% of images whether they were modified/ double compressed. This is based on comparing the quantization tables of the image and the

quantization tables stored in the database. 4% exceptional cases obtained for images of Realme i6 phone. The images of this phone were recognized as modified using the Gimp software. A warning will be displayed for such cases. To cover this type of possible false positive result, the image made, model, and software metadata are extracted from the image file. This can support the results of the QT comparison. The developed tool is useful for forensic investigations, more likely in cases where it is not possible to identify the original images. They can use this tool to prove the integrity of images provided as evidence.

For future work, it is possible to enrich the applications and image source databases with more quantization tables of other software and devices. Other image forgery techniques can be engaged with this approach to solve false positive results other than metadata extraction. This may include techniques for evaluating the first quantization table.

References

[1] Y. Salem and M. M. N. Hamarsheh, "Forensically analyzing IoT smart camera using MAoIDFF-IoT framework," *Forensic Science International: Digital Investigation*, vol. 51, no. September, p. 301829, 2024, doi: 10.1016/j.fsidi.2024.301829.

[2] Y. Salem, M. Owda, and A. Y. Owda, "Towards Digital Forensics 4 . 0 : A Multilevel Digital Forensics Framework for Internet of Things (IoT) Devices," *International Journal of Wireless and Microwave Technologies (IJWMT)*, vol. 14, no. 2, pp. 27–54, 2024, doi: 10.5815/ijwmt.2024.02.03.

[3] A. Piva, "An Overview on Image Forensics," *ISRN Signal Processing*, vol. 2013, pp. 1–22, 2013, doi: 10.1155/2013/496701.

[4] K. B. Meena and V. Tyagi, *Data, Engineering and Applications*, no. April. 2019.

[5] A. H. Saber, M. A. Khan, and B. G. Mejbel, "A survey on image forgery detection using different forensic approaches," *Advances in Science, Technology and Engineering Systems*, vol. 5, no. 3, pp. 361–370, 2020, doi: 10.25046/aj050347.

[6] GitHub, "JPEG Image Double Compression Detection using QT Code." [Online]. Available: https://github.com/iyadRamlawynew/JPEG "(accessed Jan. 28, 2025)".

[7] Y. Salem, M. Owda, and A. Y. Owda, "A Comprehensive Review of Digital Forensics Frameworks for Internet of Things (IoT) Devices," in *2023 International Conference on Information Technology: Cybersecurity Challenges for Sustainable Cities, ICIT IEEE 2023 - Proceeding*, 2023, pp. 89–96, doi: 10.1109/ICIT58056.2023.10226145.

[8] Y. Yusoff, R. Ismail, and Z. Hassan, "Common Phases of Computer Forensics Investigation Models," *International Journal of Computer Science and Information Technology*, vol. 49, no. 3, pp. 17–31, 2011, doi: 10.5121/ijcsit.2011.3302.

[9] Y. Salem, M. Moreb, and K. S. Rabayah, "Evaluation of Information Security Awareness among Palestinian Learners," in *2021 International Conference on Information Technology (ICIT)*, 2021, pp. 21–26, doi: 10.1109/icit52682.2021.9491639.

[10] Y. Salem and E. Y. Daraghmi, "GDPR-BLOCKCHAIN COMPLIANCE for PERSONAL DATA: REVIEW PAPER," *Journal of Theoretical and Applied Information Technology*, vol. 99, no. 24, pp. 5867–5877, 2021.

[11] Y. Salem, M. Owda, and A. Owda, "An Experimental Approach for Locating WhatsApp Digital Forensics Artifacts on Windows 10 and the Cloud," *International Journal of Electronic Security and Digital Forensics*, vol. 15, no. 1, pp. 281–300, 2023, doi: 10.1504/ijesdf.2023.10051774.

[12] D. Bem, F. Feld, E. Huebner, and O. Bem, "Computer Forensics - Past , Present and Future," *Journal of Information Science and Technology*, vol. 5, no. 3, 2008.

[13] L. Wang, R. Zhang, and S. Zhang, "A model of computer live forensics based on physical memory analysis," *2009 1st International Conference on Information Science and Engineering, ICISE 2009*. pp. 4647–4649, 2009, doi: 10.1109/ICISE.2009.69.

[14] S. M. Rahman *et al.*, *Multimedia Technologies: Concepts, Methodologies, Tools, and Applications*. .

[15] R. Böhme, F. C. Freiling, T. Gloe, and M. Kirchner, "Multimedia forensics is not computer forensics," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5718 LNCS, pp. 90–103, 2009, doi: 10.1007/978-3-642-03521-0_9.

[16] V. L. L. Thing, Y. Chen, and C. Cheh, "An improved double compression detection method for JPEG image forensics," *Proceedings - 2012 IEEE International Symposium on Multimedia, ISM 2012*, pp. 290–297, 2012, doi: 10.1109/ISM.2012.61.

[17] C. Chen, Y. Q. Shi, and W. Su, "A machine learning based scheme for double JPEG compression detection," *Proceedings - International Conference on Pattern Recognition*, pp. 6–9, 2008, doi: 10.1109/icpr.2008.4761645.

[18] M. Rehman, M. Sharif, and M. Raza, "Image compression: A survey," *Research Journal of Applied Sciences, Engineering and Technology*, vol. 7, no. 4, pp. 656–672, 2014, doi: 10.19026/rjaset.7.303.

[19] S. W. Son, Z. Chen, W. Hendrix, A. Agrawal, W. K. Liao, and A. Choudhary, "Data compression for the exascale computing era - survey," *Supercomputing Frontiers and Innovations*, vol. 1, no. 2, pp. 76–88, 2014, doi: 10.14529/jsfi140205.

[20] N. I. Cho and S. K. Mitra, "Warped discrete cosine transform and its application in image compression," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, no. 8, pp. 1364–1373, 2000, doi: 10.1109/76.889021.

[21] F. Huang, X. Qu, H. J. Kim, and J. Huang, "Reversible Data Hiding in JPEG Images," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 9, pp. 1610–1621, 2016, doi: 10.1109/TCSVT.2015.2473235.

[22] Z. Lin, J. He, X. Tang, and C. K. Tang, "Fast, automatic and fine-grained tampered JPEG image detection via DCT coefficient analysis," *Pattern Recognition*, vol. 42, no. 11, pp. 2492–2501, 2009, doi: 10.1016/j.patcog.2009.03.019.

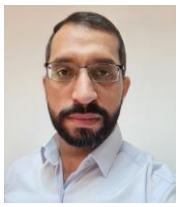
[23] J. D. Kornblum, "Using JPEG quantization tables to identify imagery processed by software," *Digital Investigation*, vol. 5, no. SUPPL., pp. 21–25, 2008, doi: 10.1016/j.diin.2008.05.004.

[24] Y. Niu, B. Tondi, Y. Zhao, R. Ni, and M. Barni, "Image Splicing Detection, Localization and Attribution via JPEG Primary Quantization Matrix Estimation and Clustering," pp. 1–14, 2021, [Online]. Available: http://arxiv.org/abs/2102.01439.

[25] F. Galvan, G. Puglisi, A. R. Bruna, and S. Battiato, "First quantization matrix estimation from double compressed JPEG images," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 8, pp. 1299–1310, 2014, doi: 10.1109/TIFS.2014.2330312.

[26] S. Battiato, O. Giudice, F. Guarnera, and G. Puglisi, "Computational data analysis for first quantization estimation on JpEG double compressed images," *Proceedings - International Conference on Pattern Recognition*, pp. 5951–5958, 2020, doi: 10.1109/ICPR48806.2021.9412528.
[27] S. Battiato, O. Giudice, F. Guarnera, and G. Puglisi, "Estimating Previous Quantization Factors on Multiple JPEG Compressed Images," *Eurasip Journal on Information Security*, vol. 2021, no. 1, 2021, doi: 10.1186/s13635-021-00120-7.
[28] I. International, W. On, M. Learning, and F. O. R. Signal, "DOUBLE JPEG COMPRESSION DETECTION FOR DISTINGUISHABLE BLOCKS IN IMAGES COMPRESSED WITH SAME QUANTIZATION MATRIX Abhinav Narayan Harish Vinay Verma Nitin Khanna Multimedia Analysis and Security (MANAS) Lab , Electrical Engineering , Indian Institute of T," pp. 0–5, 2020.
[29] M.-J. Kwon, I.-J. Yu, S.-H. Nam, and H.-K. Lee, "CAT-Net: Compression Artifact Tracing Network for Detection and Localization of Image Splicing," no. Figure 2, pp. 375–384, 2021, doi: 10.1109/wacv48630.2021.00042.
[30] Y. Rao, J. Ni, and H. Zhao, "Deep Learning Local Descriptor for Image Splicing Detection and Localization," *IEEE Access*, vol. 8, pp. 25611–25625, 2020, doi: 10.1109/ACCESS.2020.2970735.
[31] H. Yao, H. Wei, T. Qiao, and C. Qin, "JPEG quantization step estimation with coefficient histogram and spectrum analyses," *Journal of Visual Communication and Image Representation*, vol. 69, p. 102795, 2020, doi: 10.1016/j.jvcir.2020.102795.
[32] C. Kumawat and V. Pankajakshan, "A robust JPEG compression detector for image forensics," *Signal Processing: Image Communication*, vol. 89, no. September, p. 116008, 2020, doi: 10.1016/j.image.2020.116008.

Authors' Profiles



Iyad Ramlawy received his BS in computer engineering from An-Najah National University, Nablus, Palestine, in 2005. He has worked with Palestine Cellular Telecommunications Company - Jawwal since 2005 as Senior Systems Engineer, specializing in information security systems. In 2024, he moved to work at Future Information Systems - FIS as an Information Security and Network Manager. Currently pursuing his Master of Cybercrime and Digital Evidence Analysis from Arab American University, Ramallah, Palestine. His research interests include computer networking, information security, Artificial Intelligence, and Machine Learning.

Yaman Salem works as a 'Cybersecurity Consultant' at Security First Company in Dubai. Her professional journey in information security started in 2015 as an 'Associate Information Security Consultant at IT Security C&T company, progressing to the role of 'Security Awareness Content Team Leader,' managing an information security content team. She gained an MSc in Cybercrime and Digital Forensics from the Arab American University in Palestine in 2022, and she has a bachelor's in Telecommunications Engineering. She is certified in CIHE, CPTE, CCNA Security, CCNA, and Network+.

Layth Abuarram is a PhD student at Hamad Bin Khalifa University in Qatar, specializing in Computer Science and Engineering. He holds a Bachelor's degree in Cyber Security from Hebron University and a Master's in Digital Forensics and Analysis from Arab American University of Palestine. Layth's research focuses on cutting-edge advancements in cybersecurity, digital forensics, and signals intelligence in the field of signals interference. He has authored multiple papers published in esteemed conferences and journals. He is currently working on developing AI-based models to address challenges related to signal jamming and interference prevention.

Muath Sabha is an Assistant Professor in the Multimedia Technology Department at the Arab American University-Palestine. He completed his PhD in Computer Engineering, with a concentration in Computer Graphics, from the Katholieke University of Leuven, Belgium. Muath's research interests include Computer Vision, Computer Graphics, and Multimedia Technology. He has published numerous ranked papers, accessible on various research portals. Muath is a member of IEEE and ACM, and serves as a reviewer for several high-ranked academic journals and conferences.

How to cite this paper: Iyad Ramlawy, Yaman Salem, Layth Abuarram, Muath Sabha, "Forensic Software Tool for Detecting JPEG Double Compression Using an Adaptive Quantization Table Database", International Journal of Engineering and Manufacturing (IJEM), Vol.15, No.3, pp. 32-42, 2025. DOI:10.5815/ijem.2025.03.03