IMAGE COMPRESSION & IMAGE STEGANOGRAPHY DETECTION TOOL

MINI PROJECT REPORT

Submitted By

VINITHA A T (Reg No : MCT21MCA 2047)

APJ Abdul Kalam Technological University

In partial fulfilment of the requirement for the award of the

Degree of

MASTER OF COMPUTER APLLICATIONS

DEPARTMENT OF COMPUTER APPLICATIONS

MOHANDAS COLLEGE OF ENGINEERING AND TECHNOLOGY

Anad, Nedumangadu, Thiruvananthapuram

695544

2022 - 2023

.

DEPARTMENT OF COMPUTER APPLICATIONS

MOHANDAS COLLEGE OF ENGINEERING AND TECHNOLOGY

Anad, Nedumangadu, Thiruvananthapuram-695544

CERTIFICATE

This is to certify that the report entitled **"IMAGE COMPRESSION & IMAGE STEGANOGRAPHY DETECTION TOOL"** submitted by **VINITHA A T(Register No : MCT21MCA 2047)**to **APJ Abdul Kalam Technological University** in partial fulfillment of the requirement for the award of the degree **MASTER OF COMPUTER APPLICATION** is bonafied record of the project work carried out by her under my guidance and supervision.This report in any form has not been submitted by any other University or Institute for any purpose.

Head of the Department

Project Coordinator

DECLARATION

I hereby declare that the project report **"IMAGE COMPRESSION & IMAGE STEGANOGRAPHY DETECTION TOOL"**, submitted for partial fulfillment of the requirements for the award of degree of Master of Computer Application of the APJ Abdul Kalam Technological University of Kerala.It is a bonafide work done by me under supervision of Ms Jayanthi T. I have adequately and accurately cited and referenced the original source. I also declare that I have adhered to ethics of academic honestly and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be cause for disciplinary action by the institute and or the University and can also evoke penal action from the source which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of the degree, diploma or similar title of any other University

Place :Thiruvananthapuram **VINITHA A T**

Date :

ACKNOWLEDGEMENT

At the outset, thank God Almighty for standing by me throughout the project and making it possible for me to complete the project within the stipulated time. I wish to record my deep sense of gratitude to our Director, **Dr. ASHALATHA THAMPURAN**, for her extensive support and guidance throughout the course of my project. I wish to record my deep sense of gratitude to our Principal, **Dr. S. SHEELA**, for her extensive support and guidance throughout the course of my project. I am greatly thankful to our **HOD Ms. Sreeja. K**. (Department of Computer Applications) for her kind co-operation and guidance guidance throughout my project.

I am also thankful to our Project Guide **Ms. Jayanthi T** (Department of ComputerApplications) for him kind co-operation and guidance throughout the course of my project. I also extend my sincere thanks to all other faculty members of Department of Computer Applications and our friends for their co-operation and encouragements.

With Gratitude

VINITHA A T

ABSTRACT

With the rapid development of digital media editing techniques, digital image manipulation becomes rather convenient and easy. While it benefits to legal image processing, malicious users might use such innocent manipulations to tamper digital photograph images. Currently, image forgeries are widespread on the Internet and other security-related applications such as surveillance and recognition that utilize images are therefore impacted. The event and scene information delivered in images might become no longer believable. In the applications such as law enforcement and news recording.

The project deals with the forensic department detecting compressed image even if it contains the presence of noise signal. Also it detects whether the image has undergone any compression or if it is embedded with a file.It has two parts forensic section and fraud section. Fraud section tries to fool the forensic analyst by hiding the traces of compression.frauds can add noise ,hide date,hide image.Forensic part reveals the noise percentage,hidden data and image.It is also necessary to verify the originality and authencity of digital Images,and make clear the image manipulation history to get more information.

TABLE OF CONTENTS

1.INTRODUCTION

The project entitled "**IMAGE COMPRESSION & IMAGE STEGANOGRAPHY DETECTION TOOL**" aims with detecting whether the image has undergone any compression or if it is embedded with a file. Forensic department mainly focuses on detecting anykind of malpractices done in the image, whereas the anti-forensic department triesto fool the forensic analyst by hiding the traces of compression. JPEG is the mostcommonly used image standard. JPEG has a property; it follows lossy compression which does not preserve all the bit values. Thus, it leaves traces afterthe compression process. This enables the forensic analyst to determine whether the image is anti-forensically compressed by analysing the histograms of original and suspected images. The histogram of original image exhibits a comb-shape which makes it different from the histogram of original image. Anti-forensic department further works to make the histograms same by adding a noise signal. The project deals with the forensic department detecting compressed image even if it contains the presence of noise signal. With the rapid development of digital media editing techniques, digital image manipulation becomes rather convenientand easy. While it benefits to legal image processing, malicious users might use such innocent manipulations to tamper digital photograph images. Currently, image forgeries are widespread on the Internet and other security-related applications such as surveillance and recognition that utilize images are therefore impacted. The event and scene information delivered in images might become nolonger believable. In the applications such as law enforcement and news recording, it is also necessary to verify the originality and authenticity of digital images, and make clear the image manipulation history to get more information.

2.SYSTEM ANALYSIS

2.1 EXISTING SYSTEM

The widespread availability of photo editing software has made it easy to create visually convincing digital image forgeries. To address this problem, there has been much recent work in the field of digital image forensics. The field of anti- forensics seeks to develop a set of techniques designed to fool current forensic methodologies. The footprints left by JPEG compression play an important role in detecting possible forgeries, since JPEG is by far the most widely used imagecompression standard.The footprints left by JPEG compression play an important role in detecting possible forgeries, since JPEG is by far the most widely used image compression standard. To achieve lossy compression, a JPEG encoder quantizes each *Discrete Cosine Transform* (*DCT*) coefficient of an image to multiples of a quantization step size, specified by the JPEG quantization matrix. When an imageis decoded, the distribution of reconstructed DCT coefficients differs from the original which might reveal the original quantization matrix. This fact enables several forensic analysis tasks, including the identification of which camera tooka picture, or the detection of double JPEG compression. The statistical footprintsof JPEG compression can be removed by adding a properly designed dithering noise signal to the quantized DCT coefficients of a JPEG-compressed image. The distribution of the dithering noise signal is such that the resulting coefficients are approximately distributed as those of the uncompressed original image. Using this technique, it is demonstrated that many of the aforementioned forensic techniques based on JPEG footprints can be fooled.

2.2 PROPOSED SYSTEM

To limit the hazards connected to misuse of digital images, a variety of digital image forensic techniques have been proposed by the forensic community. Forensic techniques analyze the image content in order to find traces left by specific acquisition, coding or editing operations.

The anti-forensic dither is a noisy signal which cannot replace the image content lost during quantization. It introduces visible distortion in the attacked image, which appears as a characteristic grainy noise that allows to discriminate attacked images from original uncompressed images. These traces can be analyzed in terms of distortion introduced in the tampered image. Experiments demonstrate that it is possible to correctly detect attacked images with an accuracy equal to 93%, whenexcluding the case of nearly lossless JPEG compression. The process of footprintremoval inevitably introduces new traces in the doctored image.It has mainly

two parts.

Fraud section

- A malicious user can compress the image
- He can hide text in image
- He can hide image in another image

Forensic section

- A forensic analyst can detect whether an image has undergone any compression
- Can extract the hidden data from the image
- Can extract the hidden image from the image

Advantages of Proposed System.

a) Attacked images can be correctly detected with an accuracy in the range of 93% to 99%.

b) Various hazards connected to the misuse of digital images can be minimized

c) Simple and elegant user interface

2.3 FEASIBILITY STUDY

A feasibility study is conducted to select the best system that meets performancerequirements. This entails an identification description, an evaluation of the candidate system andthe selection of the best system that matches With needs. A statement of constraints theidentification of the specific system objectives, and a description of outputs define a system'srequired performance. The analyst is then ready to evaluate the feasibility of the candidatesystem to produce these outputs. Three key considerations are involved in the feasibilityanalysis.

2.3.1 Technical Feasibility

Using various approaches to test the technical implementation of project using available hardware, software and technology it is noticed that the project can be implemented using existing technology. Also our system can adopt the technological upgrades as it is developed under the considerations of software engineering principles. Moreover it uses object oriented approach of programming which can enhance the upgrading with new classes and modules as per requirement.

2.3.2 Operational Feasibility

This mode of operational feasibility analysis includes the operational analysis of overall system. The system is tested under several circumstances with varying inputs in unit approach of testing to integrated approach of testing.

2.3.3 Economic Feasibility

Since, our system uses simple hardware components which are easily available. Thus, the overall system is economically feasible to be implemented by users.

3.SYSTEM REQUIREMENTS

3.1 Hardware Requirements

Processor	:	Intel i3 or equivalent CPU
Processor Speed	:	2.6 Ghz& above
RAM	:	4 GB & above
Hard Disk Capacity	:	50GB & above

3.2Software Requirements

Language	:	C#.Net
IDE Used	:	Visual Studio 2019
Operating System	:	Windows 10

4. SOFTWARE SPECIFICATIONS

C#.Net

C-Sharp is a multi-paradigm programming language encompassing *imperative*, *declarative*, *functional*, *generic*, *object-oriented* (*class-based*), and *component- oriented programming disciplines*. It was developed by Microsoft within the .NET initiative. C# is one of the programming languages designed for theCommon Language Infrastructure.C# is intended to be a simple, modern, general-purpose, object-oriented programming language.

Features of C#.Net

By design, C# is the programming language that most directly reflects the underlying *Common Language Infrastructure* (*CLI*). Most of its intrinsic types correspond to value-types implemented by the CLI framework.

The language specification does not state the code generation requirements of thecompiler: i.e., it does not state that a C# compiler must target a *Common Language Runtime*, or generate *Common Intermediate Language* (CIL), or generate any other specific format. Theoretically, a C# compiler could generate machine code like traditional compilers of C++ or FORTRAN.

Some notable distinguishing features of C# are:

- There are no global variables or functions. All methods and members must be declared within classes. Static members of public classes can substitute for global variables and functions.
- Local variables cannot shadow variables of the enclosing block, unlike C andC++.
- C# supports a strict *Boolean* data type, *bool*. Statements that take conditions,such as while and if, require an expression of a type that implements the true operator, such as the *boolean* type. While C++ also has a boolean type, it canbe freely converted to and from integers. C# disallows the "*Integer Meaning True Or False*" approach on the grounds that forces programmers to use

  expressions that return exactly bool that could prevent certain types of common programming mistakes in C or C++.
- In C#, memory address pointers can only be used within blocks specifically marked as unsafe, and programs with unsafe code need appropriate permissions to run. Most object access is done through safe object references,which always either point to a "live" object or have the well-defined null value; it is impossible to obtain a reference to a "dead" object or to a randomblock of memory. An unsafe pointer can point to an instance of a value-type, array, string, or a block of memory allocated on a stack. Code that is not marked as unsafe can still store and manipulate pointers through the System.

TOOLS USED

MICROSOFT VISUAL STUDIO

Microsoft Visual Studio is an *Integrated Development Environment* (IDE) from Microsoft. It can be used to develop console and graphical user interface applications along with Windows Forms applications, web sites, web applications, and web services in both native-code

together with managed code for all platforms supported by *Microsoft Windows*, *Windows Mobile*, *Windows CE*, *.NET Framework*, *.NET Compact Framework* and *Microsoft Silverlight*.

Visual Studio includes a *Code Editor* supporting IntelliSense as well as *Code Refactoring*. The *Integrated Debugger* works both as a source-level debugger anda machine-level debugger. Other built-in tools include a *Forms Designer* for building GUI applications, *Web Designer*, *Class Designer*, and *Database Schema Designer*. It accepts plug-ins that enhance the functionality at almost every level including adding support for source control systems (*Subversion* and *Visual SourceSafe*) and adding new toolsets like *Editors* and *Visual Designers* for domain-specific languages or toolsets for other aspects of the software development lifecycle.

Visual Studio supports different programming languages by means of language services, which allow the code editor and debugger to support nearly any programming language, provided a language-specific service exists. Built-in languages include C/C++ (via Visual C++), VB.NET (via Visual Basic .NET), C# (via Visual C#), and F# (as of Visual Studio 2010). Support for other languages such as M, Python, and Ruby among others is available via language services installed separately. It also supports XML/XSLT, HTML/XHTML, JavaScript and CSS. Individual language-specific versions of Visual Studio alsoexist which provide more limited language services to the user: Microsoft VisualBasic, Visual J#, Visual C++.

5. MODULAR DESCRIPTION

Image preprocessing

Browse the concerned image. For an image, consider the grayscale image of each picture element. Split the original uncompressed image in pixels. Apply a DCT to blocks of pixels, thus removing redundant image data. The image data is divided up into 8*8 blocks of pixels. From this point on each color component isprocessed independently, so a pixel means a single value, even in a colour image.A DCT is applied in each 8*8 blocks. DCT converts the spatial image representation into a frequency map. The lower order-DC term represents the average value in the block, while the successful higher order (AC) terms representthe strength of more and more rapid changes across the width or height of block.The highest AC term represents the strength of a cosine wave alternating from maximum to minimum at adjacent pixels. The DCT calculation is fairly complex& is the most expensive step in JPEG compression. The DCT step itself is a lossless except for round off errors. Thus, corresponding DCT coefficient block can be obtained. Quantize each block of DCT coefficients. Perform one to one division and round off. Encode the resulting coefficients of image data

DCT HISTOGRAM GENERATION

For the generation of the histogram, RGB color values of concerned image is considered. It is transformed into luminance. The color space transformation is performed on pixel-by-pixel basis. DCT coefficient calculated value can be plotted with the DCT coefficient frequency. The histogram of a gray-scale imageconsists of a discrete array of bins, each representing a certain gray-level range and storing the number of pixels in the image whose gray-level falls into that range, defining a discrete function that maps a gray-level range to the frequency of occurrence in the image. *Scatter-based histogram* generation consists of two sub-tasks: *Bin selection for each input pixel* and *accumulation of bin contents*. Itrenders one point primitive for each input pixel. Then compute the bin index in the vertex shader and convert it to an output location that maps into our 1D bin texture. The fragment that is rasterized into our desired bin location in the histogram render target is accumulated by configuring the hardware blend units to add the incoming fragment to the contents of the render target. After scatteringand accumulating all points in this manner, the output render target will contain the desired histogram. By looking at the histogram for a specific image a viewerwill be able to judge the entire tonal distribution at a glance.

QUANTIZATION MECHANISM

To discard appropriate amount of information, each DCT output value is dividedby quantization coefficient. If quantization coefficient is large, more data is lostbecause actual DCT value is represented less accurately. Each of 64 positions of DCT output block has its own quantization coefficient. Separate *Quantization Table Matrices* are employed for *Luminance* and *Chrominance Data* with *Chrominance Data* being quantized more heavily than the *Luminance Data*. This allows JPEG to exploit further the eye's differing sensitivity to luminance andchrominance. *Quantization mechanism* controls the quality setting of most JPEG compressors. Here the input is an image. So, the anti-forensic user has to select an image which is already available in his/her system. In order to select an image for processing, it is necessary to create a search button. Then by clicking on thebutton he can select the image that he is needed. All the properties of image can be verified here. Make sure that extension of image that he is loaded is JPEG. Image matrix is a 2-dimensional matrix that can be generated on the basis of pixel values of the original image. After generating the matrix, draw the *Histogram* ofthe original image. *Histogram* is a graph showing the number of pixels in an image at each different intensity value found in that image. The *Histogram* of an unaltered image (original image) typically conforms to a smooth envelope.In this phase, the contrast of the original image that the user uploaded is enhanced. *Contrast* is defined as the separation between the darkest and brightest areas of the image. *Increasing Contrast* means you increase the separation between dark and bright, making shadows darker and highlights brighter. *Decreasing Contrast*means you bring the shadows up and the highlights down to make them closer to one another. *Adding Contrast* usually makes an image look more vibrant while decreasing contrast can make an image look duller.

After enhancing the original image, generate its matrix and based on that, draw the histogram. Then observe the differences in histogram of both original image and contrast enhanced image. The histogram of original image conforms to a smooth envelope and while that of enhanced image presents peak/gap artifacts. After enhancing the contrast of original image, the resulting new image can be saved in a new folder for further use.

COMPARISON OF MATRICES

In this step with the comparison of matrices, we can find the incorrectly classifiedimages. It can be done with comparing of image properties, histogram values, DCT comparison, noise presence etc. Image properties can have the comparisonwith the dimension, extension of the images. In histogram comparison, RGB values are compared. Next DCT values are

compared, the original DCT values and modified DCT values are compared and find the differences. Finally check the presence of noise, if the image is anti-forensically treated, there would be differences in the image matrices values, otherwise the image is original. It involves two novel algorithms. First, *Global Contrast Enhancement Detection Algorithm* and second, *Identify Source Enhanced Composite Image Algorithm*. Itinvolves

i. Image pre-processing

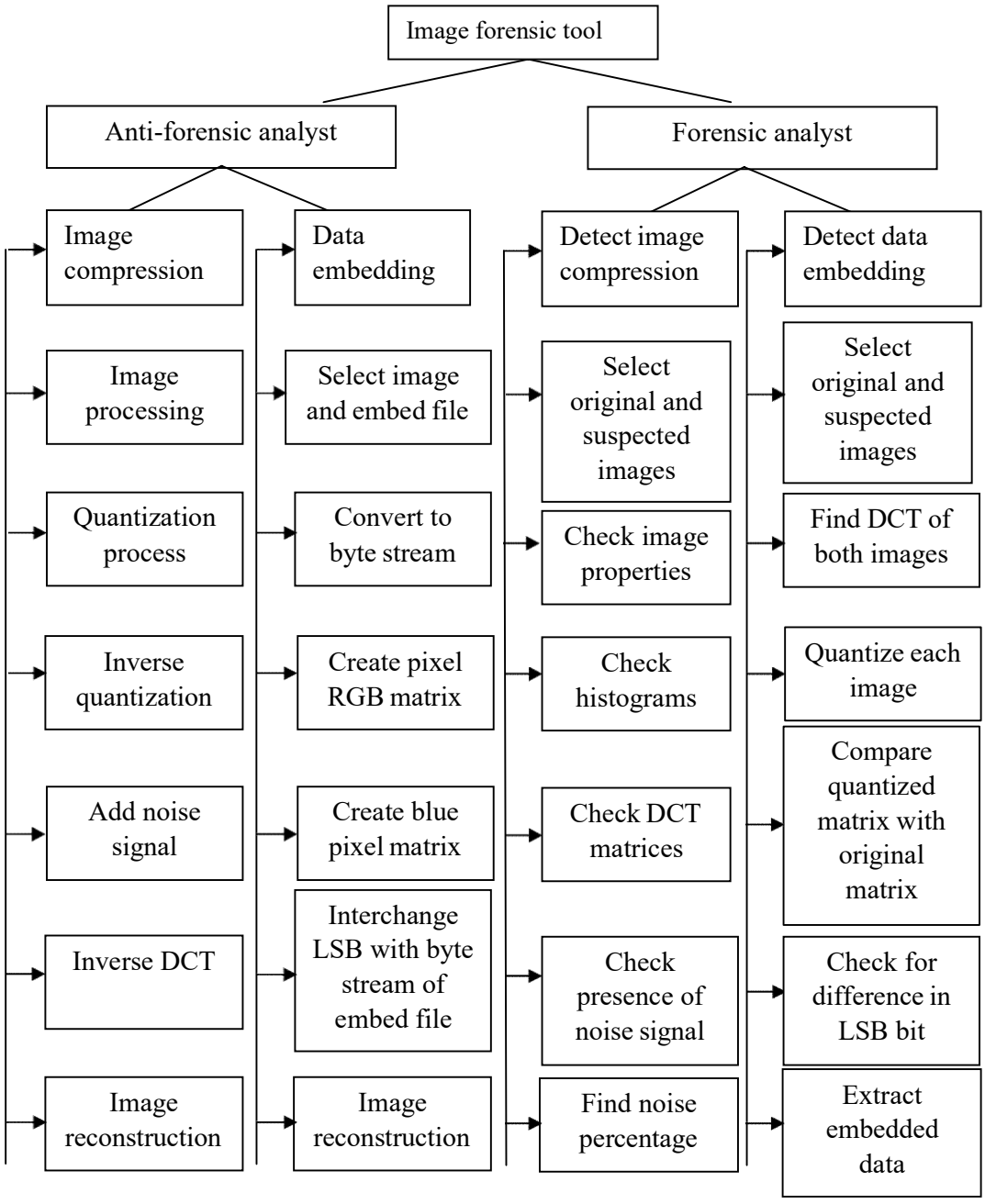
ii. Global contrast enhancement detection algorithm

- Peaks/pits detection
- Zero-height gap bin detection

iii. Source enhanced composite image identification algorithm

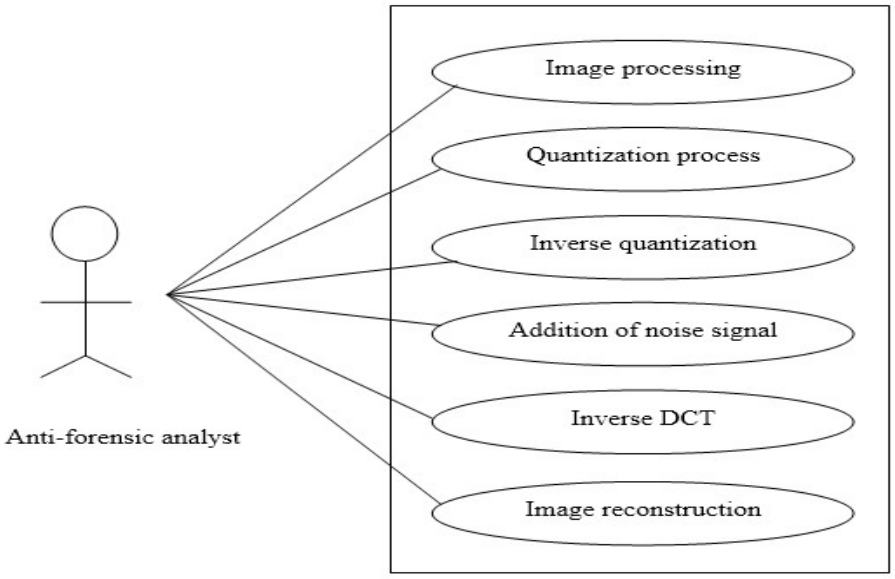
- Peak/gap based similarity measure
- Similarity maps fusion

ARCHITECTURAL DIAGRAM

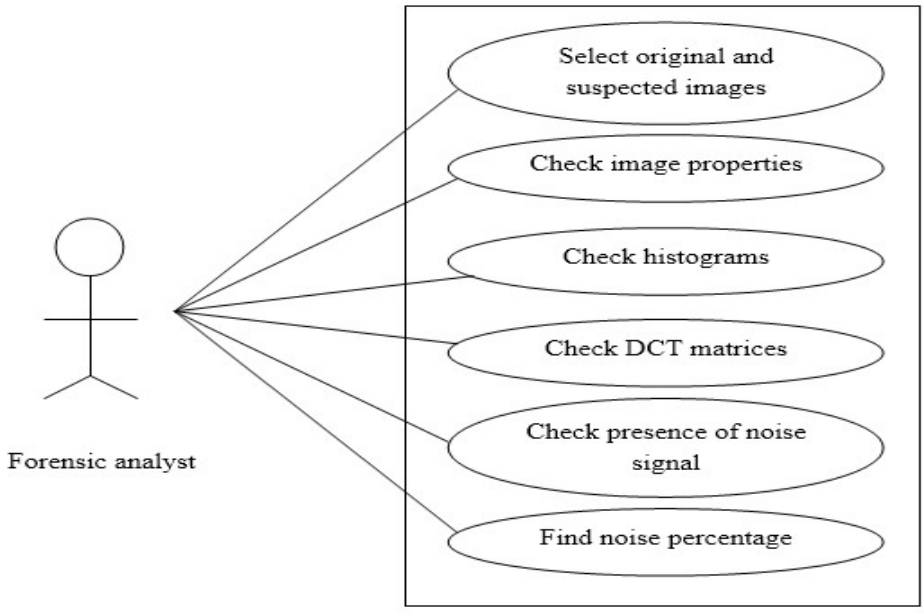


UML DAIGRAM

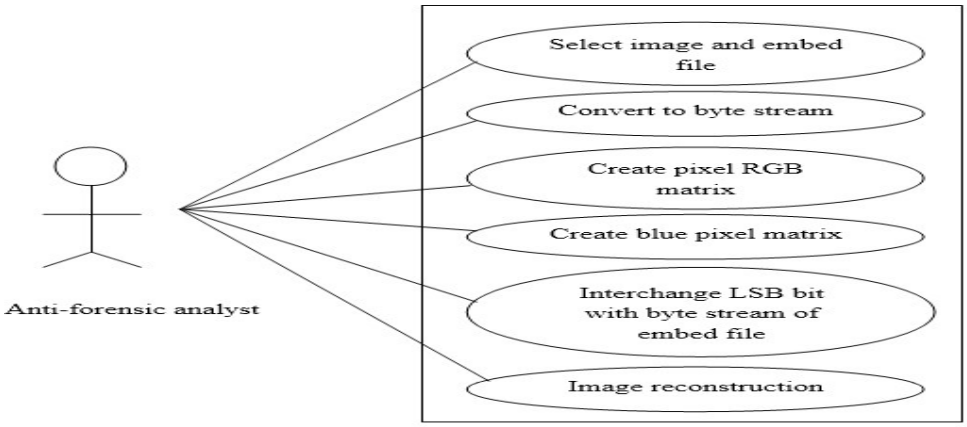
The *Unified Modelling Language* (UML) is a standard language for specifying, visualizing, and documenting the artifacts of software system, as well as for business modelling and other non-software systems. The UML represent the collection of the best engineering practices that have proven successful in the modelling of large and complex systems. The UML is a very important part of developing object-oriented software and the software development process. The UML uses mostly graphical notations to express the design of the software projects. Using the UML helps project teams communicate, explore potential designs, and validate the architectural design of the software.



Use Case Diagram for Anti-forensic analyst (image compression)



Use Case Diagram for Forensic analyst (detecting image compression)



Use Case Diagram for Anti-forensic analyst (embedding data on image)

ALGORITHMS USED

The coding step is the process that transfer design into programming language. It translates a detail design representation of software into a programming language realization. The translation process continues when a compiler accepts source code as input and produces machine-dependent object code as output. Quality of source code can be improved by the use of structured coding techniques; good coding style and readable, consistent code format. We are using C#-.Net with the.NET *Common Language Runtime* (CLR) garbage collection, which some feel is necessary in an object-oriented language. It also supports the notion of indexers, which in simplified terms lets to manipulate objects as arrays and delegates, which can be thought of as a method-call-backs on steroids

Algorithm 1: Global Contrast Enhancement Detection Algorithm

a) Peak/Gap Artifacts Incurred by JPEG Compression

The two factors affect the de facto presence of histogram peak bins in a JPEG image: 1) the flatness; 2) JPEG quality factor. The larger flat regions and larger DC quantization step would cause more apparent peak bins. Even for the block with sparse non-zero quantized AC coefficients, after decompression the number of pixel gray levels still decreases to some extent. So the global histogram is still prone to discontinuity. Strength of the histogram peak/gap artifacts incurred by JPEG compression can also be measured by the metric F.

b) Peak/Gap Artifacts Incurred by Contrast Enhancement

There exists notable difference between the peak/gap artifacts from contrast enhancement and those from JPEG compression. The gap bins with zero height, where no primary pixel values are mapped to, always appear in enhanced images. On the contrary, the zero-height gap bins are absent in compressed images since there is lack of a distinct pixel value mapping applied to all pixels. A regular pixel value mapping relationship exists in flat regions, but not in other regions. Therefore, the zero-height gap feature can be used to detect global contrast

enhancement in both uncompressed and compressed images.

c) Zero-Height Gap Bin Detection

The zero-height gap feature can be used to detect global contrast enhancement indigital images. The gap bins with zero height are always appearing in contrast enhanced images.Detect the bin at *k* as a zero-height gap bin if it satisfies the below pseudo code:

$$\begin{cases} h(k) = 0 \\ \min\{h(k-1),\ h(k+1)\} > \tau \\ \frac{1}{2w_1+1} \sum_{x=k-w_1}^{k+w_1} h(x) > \tau. \end{cases}$$

Here, the first sub-equation assures that the current bin is null. To define a gap bin, the second sub-equation keeps two neighbouring bins larger than the threshold τ. To exclude the zero-height gap bins which may be incorrectly detected in histogram trail-ends, the average of neighbouring $(2w1+1)$ bins should be larger than τ , as constrained by the third sub equation. Then, count the numberof detected zero-height gap bins, denoted by $N\boldsymbol{g}$. If it is larger than the decision threshold, contrast enhancement is detected, else not.

Algorithm 2: Algorithm For Image Compression

1) Start.
2) Select the image to be compressed.
3) Process the image by finding the image properties and forming the histograms(RGB format).
4) Form the RGB matrix (hex value matrix) based on the pixel values.
5) Find the DCT matrix (dividing into 8*8 matrix).
6) Quantize DCT matrix with the standard jpeg quantization matrix.
7) Round off the resultant quantization matrix.
8) Perform inverse quantization (multiplication with standard jpeg quantizationmatrix) to obtain the modified DCT.
9) Compare modified DCT matrix with the original DCT matrix in order to findthe difference and then add a noise signal to nullify the difference.
10) Perform inverse DCT on the matrix to get modified RGB matrix.
11) Reconstruct the image from the modified RGB matrix.
12) Save the image.
13) Stop.

Algorithm 3: Algorithm For Detecting Image Compression

1) Start.
2) Select original image and suspected image.
3) Check whether image properties are same and if so, proceed to next step or else, the images are different.
4) Check whether histograms are same and if so, proceed to next step or else, theimages are different.
5) Check whether DCTs are same and if so, proceed to next step or else, the images are different.
6) Check presence of noise signal and if so, images are not same and has gone through an anti-forensic compression method and find out the percentage of noise signal added; or else the images are same.
7) Stop.

Algorithm 4: Algorithm For Embedding Data On Image

1) Start.
2) Load the image and select the file to be embedded.
3) Convert to byte stream.
4) Create pixel RGB matrix.
5) Retrieve and create the blue pixel matrix.
6) Interchange the LSB bit with the byte stream of the file to be embedded.
7) Reconstruct the image from the modified matrix.
8) Save the image.

Algorithm 5: Algorithm For Detecting Data Embedded On The Image

1) Select original image and suspected image.
2) Find the DCT of both images.
3) Quantize each of the images.
4) Compare the original matrix with quantized matrix.
5) Check the difference in the LSB bits.
6) Find the difference and extract the data.

6.PRODUCT BACKLOG

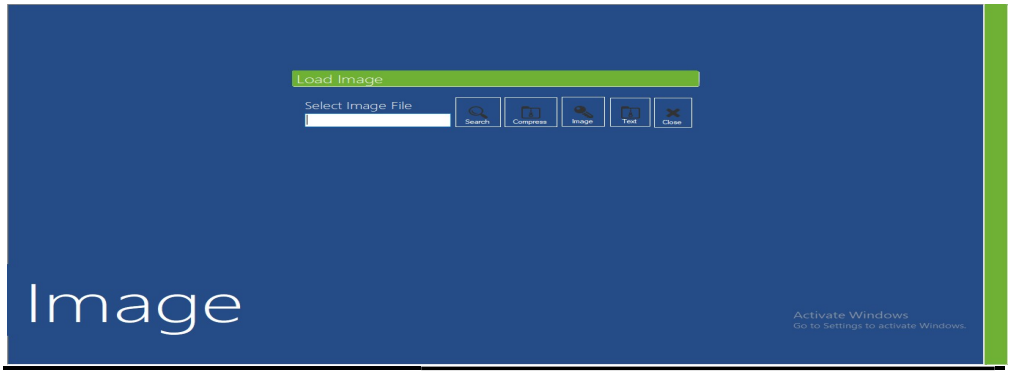
SL NO	USER STORIES	PRIORITY	COMMENT FROM SCRUM MASTER	COMMENT FROM PRODUCT OWNER
1.	User want to login	High		
2.	Should be able to select image	High		
3.	Image compression	High		
4.	Addition of noise signal	High		
5.	Image reconstruction	High		
6.	User want to embed data	High		
7.	User want to embed image	High		
8.	User want to embed image	High		

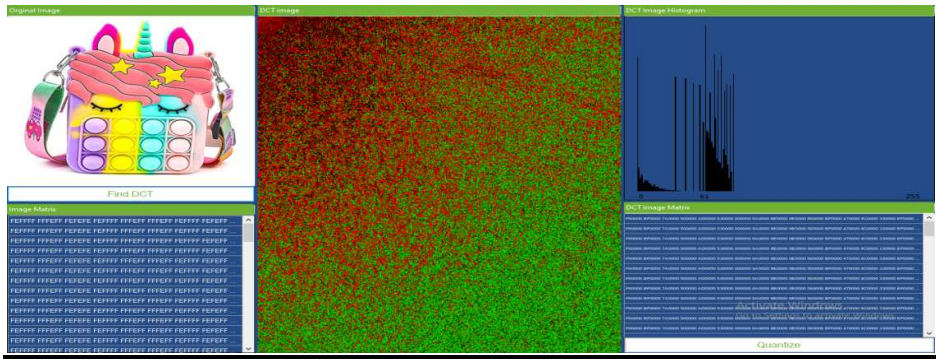
SL NO	USER STORIES	PRIORITY	COMMENT FROM SCRUM MASTER	COMMENT FROM PRODUCT OWNER
1.	User want to login	High		
2.	User want to select original and detected images	High		
3.	Check image properties	High		
4.	Want to check presence of noise signal	High		
5.	User want to display noise percentage	High		
6.	User want to display hidden data	High		
7.	Display hidden image	High		

7. SPRINT BACKLOG

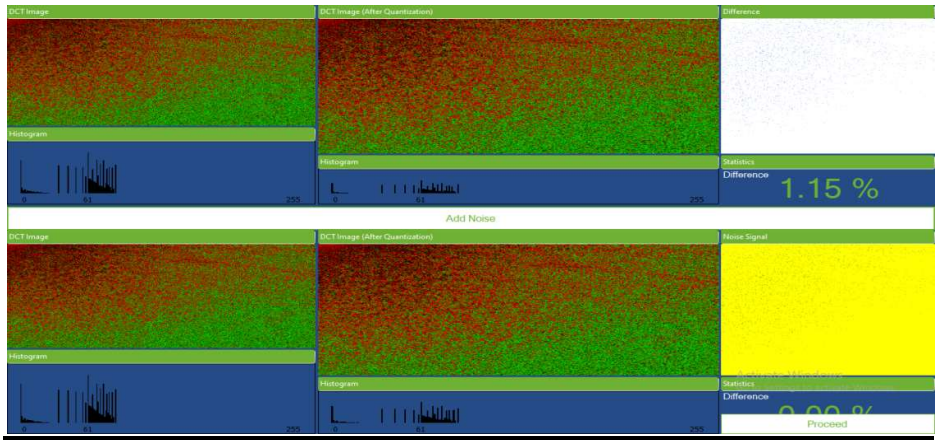
Sprint No	Tasks	Status
1	Data Collection	Completed
2	Image preprocessing	Completed
3	DCT histogram generation	Completed
4	Quantization mechanism	Completed
5	Comparison of matrices	Completed
6	Data embedding	Completed
7	Data extraction	Completed
8	Image embedding &extraction	Completed

8.SCREENSHOTS




Orginal Image
DCT image
DCT Image Histogram
Find DCT
Image Matrix
DCT Image Matrix
Quantize


Quantization
DCT Image
Quantization Matrix
DCT Image (After Quantization)
DCT Image Matrix
Quantize
Quantization
DCT Image Matrix (After Quantization)
DCT Image Histogram
Proceed
DCT Image Histogram (After Quantization)


DCT Image
DCT Image (After Quantization)
Difference
Histogram
Statistics
Difference
1.15 %
Add Noise
Noise Signal
Proceed

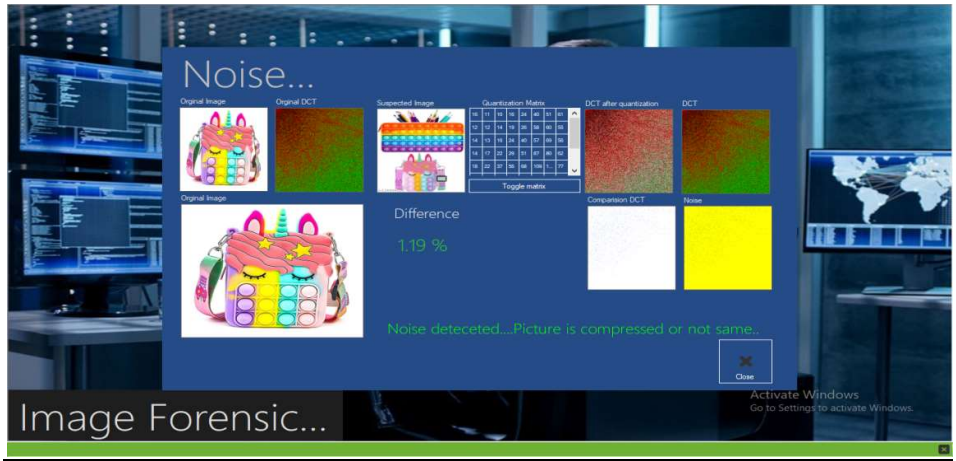

Noise...
Orginal Image
Orginal DCT
Suspected Image
Quantization Matrix
DCT after quantization
DCT
Toggle matrix
Orginal Image
Difference
1.19 %
Comparision DCT
Noise
Noise deteceted....Picture is compressed or not same..
Close
Image Forensic...


Image Upload...
Browse Image
Embed Data
Embed
Close
Image
Activate Windows
Go to Settings to activate Windows.

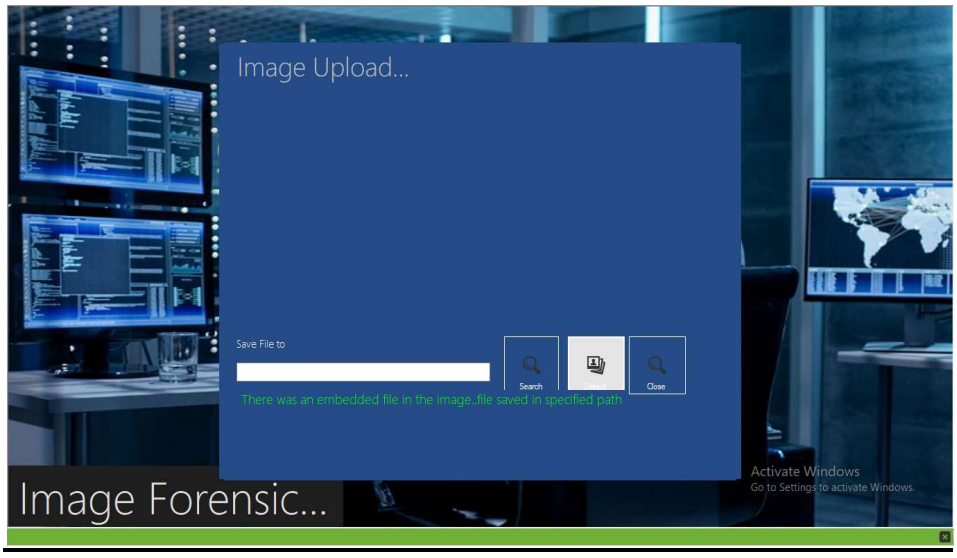
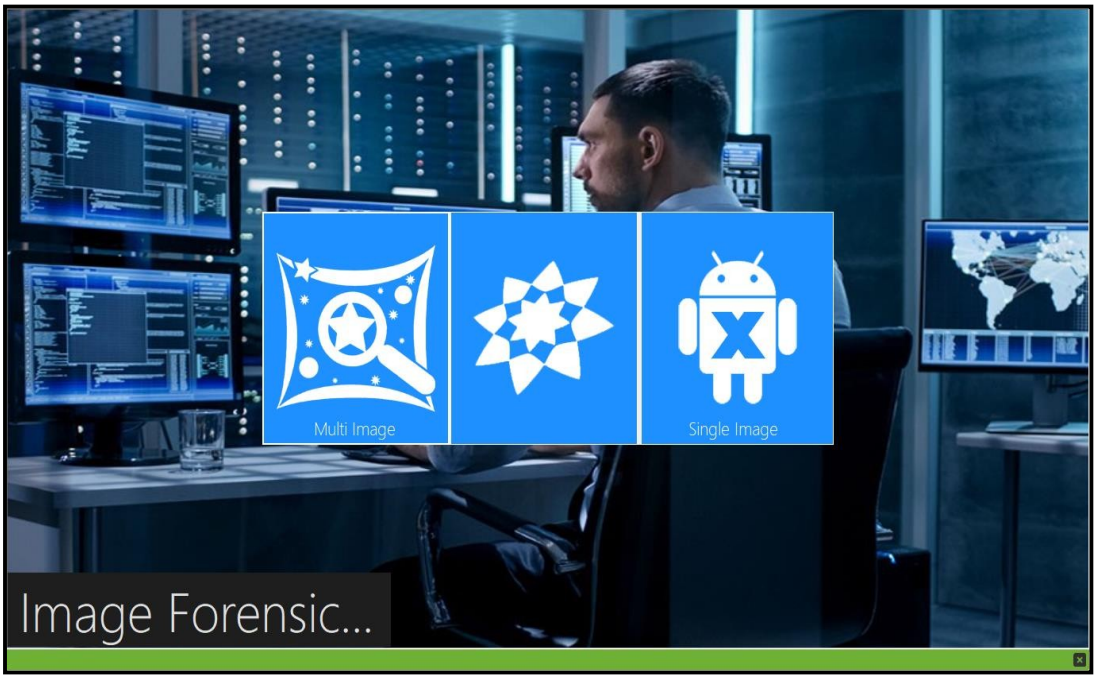
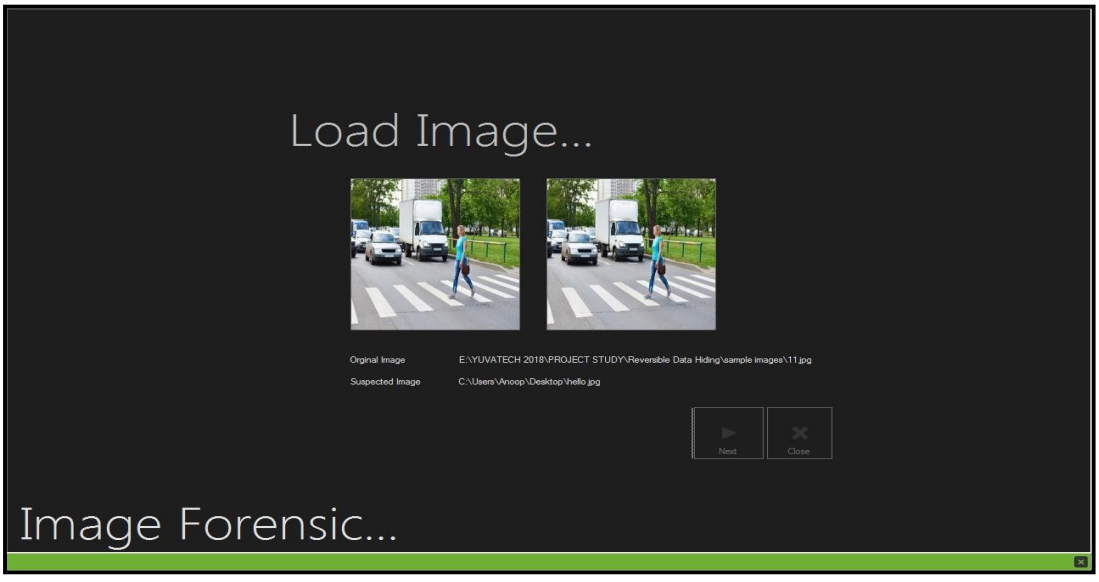


Image Upload...
Save File to
Search
Close
There was an embedded file in the image..file saved in specified path
Image Forensic...
Activate Windows
Go to Settings to activate Windows.



Load Original & Suspected Image



9. CONCLUSION

JPEG compression leaves characteristic footprints which can be potentially exploited by the forensic analyst to perform tampering detection, source identification, etc. Recently, it has been shown that an adversary might conceal such footprints by adding a properly designed dithering noise signal in the DCT domain. The project investigates the problem of JPEG-compression anti- forensics by showing how the forensic analyst can effectively counter the anti- forensic method which was originally proposed. Our analysis proves that removing traces of JPEG compression is more difficult than previously thought. Furthermore, our approach differs from conventional steganographic techniques in that it specifically targets JPEG anti-forensic dither, thus it is less prone to produce false positives when the image has been corrupted by other non- malicious kinds of noise. In addition, if the quantization matrix is a scaled versionof a known template, it is able to estimate the underlying JPEG quality factor. Future research will investigate the problem of image compression anti-forensics not only in jpeg image compression standard, but also extend to other standards such as bmp, png, tif etc. It may also become possible to detect image compression in the absence of the original image. Moreover, data embedding canbe done also with DCT embedding rather than LSB embedding

10. **REFERENCES**

- *Beginning Visual C#* -Watson,White, Wrox Publications.
- *C# Programming Bible*-Jeff Ferguson, Meeta Gupta.
- *Pro ASP.Net 4.0 in C#* -Mac Donald and Szpuszta.
- *Software Engineering*-Roger S Pressman.
- *System Analysis and Design*- James A. Senn.

1. Gang Cao, Yao Zhao, Rongrong Ni *"*Contrast Enhancement-Based Forensics in Digital Images*"* IEEE transactions on information forensics and security, vol. 9, no. 3, march 2014
2. H. Farid, "Image forgery detection," *IEEE Signal Process. Mag.*, vol. 26, no.2, pp. 16–25, Mar. 2009.
3. S. Bayram, I. Avcubas, B. Sankur, and N. Memon, "Image manipulation detection," *J. Electron. Imag.*, vol. 15, no. 4, pp. 04110201–04110217, 2006.
4. A. Swaminathan, M. Wu, and K. J. R. Liu, "Digital image forensics via intrinsic fingerprints," *IEEE Trans. Inf. Forensics Security*, vol. 3, no. 1, pp. 101–117, Mar. 2008.
5. H. Cao and A. C. Kot, "Manipulation detection on image patches using FusionBoost," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 3, pp. 992– 1002, Jun. 2012.
6. J. Fan, H. Cao, and A. C. Kot, "Estimating EXIF parameters based on noise features for image manipulation detection," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 4, pp. 608–618, Apr. 2013.
7. A. C. Popescu and H. Farid, "Exposing digital forgeries by detecting traces ofresampling," *IEEE Trans. Signal Process.*, vol. 53, no. 2, pp. 758–767, Feb. 2005.
8. G. Cao, Y. Zhao, R. Ni, and A. C. Kot, "Unsharp masking sharpening detection via overshoot artifacts analysis," *IEEE Signal Process. Lett.*,vol. 18,no. 10, pp. 603–606, Oct. 2011.
9. G. Cao, Y. Zhao, R. Ni, and A. C. Kot, "Unsharp masking sharpening detection via overshoot artifacts analysis," *IEEE Signal Process. Lett.*,vol. 18,no. 10, pp. 603–606, Oct. 2011.
10. M. C. Stamm and K. J. R. Liu, "Forensic detection of image manipulation using statistical intrinsic fingerprints," *IEEE Trans. Inf. Forensics Security*, vol. 5. Sep. 2010.

11. M. C. Stamm and K. J. R. Liu, "Forensic detection of image manipulation using statistical intrinsic fingerprints," *IEEE Trans. Inf. Forensics Security*, vol. 5, no. 3, pp. 492–506, Sep. 2010.

12. M. C. Stamm and K. J. R. Liu, "Forensic estimation and reconstruction of a contrast enhancement mapping," in *Proc. IEEE Int. Conf. Acoust., Speech Signal*, Dallas, TX, USA, Mar. 2010, pp. 1698–1701.

13. G. Cao, Y. Zhao, and R. Ni, "Forensic estimation of gamma correction in digital images," in *Proc. 17th IEEE Int. Conf. Image Process.*, Hong Kong, 2010, pp. 2097–2100.

14. P. Ferrara, T. Bianchiy, A. De Rosaz, and A. Piva, "Reverse engineering of double compressed images in the presence of contrast enhancement," in *Proc. IEEE Workshop Multimedia Signal Process.*, Pula, Croatia,Sep./Oct. 2013, pp. 141–146.

15. M. Kirchner and J. Fridrich, "On detection of median filtering in digital images," in *Proc. SPIE, Electronic Imaging, Media Forensics and Security*