JPEG Metadata:A complete study

Jincy J. Fernandez
Research Scholar, SCSE,
VIT University,
Chennai, India
jincyj.fernandez2015@vit.ac.in

Dr. Nithyanandam Pandian
Professor, SCSE,
VIT University,
Chennai, India
nithyanandam.p@vit.ac.in

***Abstract*— JPEG is one of the widely used image format. As like other image formats, JPEG has its own special bytes called JPEG markers or metadata that gives relevant information about the image. In applications such as watermarking, image theft identification and digital forensics, where some kind of confidential information is embedded into the image either for copyright protection or image authentication. Usually this embedding is done on DCT coefficients. But with the use of quantization technique and compression nature of JPEG, the extraction/ recovery of embedded data may not be possible all the times. A solution to this is to use quantized DCT coefficients for embedding. In order to decode the compressed data into quantized DCT coefficients, the encoding method used and also the type of chroma-subsampling used are to be known. This information is stored in the JPEG marker segment along with other relevant information. This paper detailed about the various types of JPEG markers and has shown experimental results. This will help young researchers to get the basic idea about the JPEG metadata.**

Keywords*— *JPEG, JFIF, EXIF, Huffman coding, Minimum coded unit, Chroma- subsampling.

I. INTRODUCTION

Joint Photographic Experts Group (JPEG) was formed by the International Telegraph and Telephone Consultative Committee for the standardization of the encoding procedures of continuous tone images. Finally JPEG came out with a set of guidelines specified in ITU-T T.81 [1]. In 1992, JPEG File Interchange Format (JFIF) is introduced by including or embedding some metadata information in the image [2]. JPEG is a compressed image format [3] which generally does not use RGB colour space for processing. Instead, it uses a variant of RGB called YCbCr that separates the brightness and colour information. The YCbCr colour space has three components Y, C_b and C_r: the Y component represents the brightness of a pixel, and the C_b and C_r components represent the chrominance of the pixel. The advantage of using YCbCr is that human eye tends to notice brightness information more than chrominance (colour) details. JPEG standard generally uses Discrete Cosine Transform (DCT). Four modes or variations [2] are defined by the JPEG standard for the compression of images such as sequential DCT, sequential lossless, progressive DCT and hierarchical mode. Sequential DCT is the JPEG baseline format which is used for images with 8- bit samples and the only encoding method used is Huffman coding. Sequential lossless mode is used for applications where it is desirable to have no loss in the image details during the compression process. Progressive DCT works on the same principle of baseline mode, but instead of processing information in one scan of the image, progressive mode processes the image in a series of scans. Hierarchical mode allows a collection of various resolution images to be within one JPEG file. It also uses multiple scans to define the image.

The structure of the paper is as follows. Section II gives information about the JPEG metadata and the various JPEG markers. Experimental results are given in section III and Conclusion is given in section IV.

II. JPEG METADATA

Metadata is a very powerful tool that describes context information of the multimedia such as image, audio and video [6]. Each multimedia type has its own metadata format. A JPEG file consists of header part and the compressed image data. The header part is divided into a group of segments; each segment contains data delimited by markers. Markers [7-9] breaks down JPEG file into structural parts. The structure of the JPEG file is given in Table I. JPEG markers are classified into two types: Standalone markers and Non standalone markers. Standalone markers are of just 2 bytes specifying marker identifier; and are used to identify a segment. Non Standalone markers consist of 2 bytes marker identifier followed by a set of bytes specifying some relevant details about the image. A JPEG file must begin with SOI marker and ends with EOI marker. All other markers come in between SOI and EOI markers.

The visualization of markers is needed in several applications such as digital watermarking, image theft identification [9] and digital forensics, where some useful information is embedded into the image. H. Huang and W. Fang [10] used metadata based watermarking for digital images for copyright protection. Since quantization technique used in JPEG round off the DCT coefficients [11] while using the standard decompression method, the extracted data may not be identical with the embedded one. So in such applications, the embedded information can be extracted without even a single bit change by using the information stored in each segment for e.g. DHT, DQT segments that are identified by the corresponding marker identifier.

TABLE I. STRUCTURE OF JPEG/ JFIF FILES [2]

Marker Abbreviation	Marker Name
SOI	Start of Image
APP0	Application Marker
DQT	Quantization table
DHT	Huffman table
DRI	Restart Interval
SOF	Frame header
SOS	Scan header
	Compressed data
EOI	End of Image

A. SOI: Start of Image Marker

Start of Image (SOI) marker is a standalone marker used to identify whether it is a JPEG file. It is identified by the 2 bytes marker identifier 0xFFD8.

B. APP: Application Specific Marker

Application markers stores application specific information. The information about APP0 segment is given in Table II. APP0 marker stores details about JPEG/JFIF image format whereas APP1 segment stores JPEG/EXIF [4, 5] details. APP0 segment starts with marker identifier 0xFFE0. It is followed by 2 bytes length field. Next is the 5 bytes long file identifier field whose value in hexadecimal is 4A 46 49 46 00 and is same as "JFIF" in ASCII. The version of the JFIF is specified then by 2 bytes. The first byte specifies the major version and second byte specifies the minor version of the JPEG/JFIF format. Units' field identify the unit of measurement for representing image resolution as given in Table III.

TABLE II. APP SEGMENT

Field	Size (in bytes)	Description
Marker Identifier	2	0xFFE0
Length	2	It must be >= 16
File Identifier Mark	5	Identifies JFIF
Major revision number	1	Major version of the JPEG/JFIF format
Minor revision number	1	Minor version of the JPEG/JFIF format
Units for x/y densities	1	Unit of measurement
X-density	2	Horizontal resolution
Y-density	2	Vertical resolution
X- Thumbnail	1	Horizontal dimension of the thumbnail image
Y-Thumbnail	1	Vertical dimension of the thumbnail image
Thumbnail image	N	Uncompressed 24 bit RGB thumbnail data

TABLE III. UNITS OF MEASUREMENTS

Units (in hexadecimal)	Measurements
01	Dots per inch
02	Dots per centimeter
00	Pixel aspect ratio

X-density and Y-density each of 2 bytes long are the horizontal and vertical resolution of the image, which must be a nonzero value. X-thumbnail and Y-thumbnail give the dimension of the embedded thumbnail image. A thumbnail image is a smaller representation of the image in the main JPEG data stream. If no thumbnail image is embedded, this field will be zero. This is followed by thumbnail image of size (3xn) bytes, where n= X-thumbnail * Y-thumbnail.

C. DQT: Define Quantization Table Marker

A quantization table is specified in the DQT segment. A quantization table is a 8×8 matrix of bytes ordered after zigzag principle. The luminance and chrominance components have separate quantization tables. The information about DQT segment is given in Table IV. A DQT segment begins with 2 bytes marker identifier 0xFFDB, followed by quantization table length. Next is a byte in which 4 MSB bits represents precision and 4 LSB bits represents quantization table information. The possible values for quantization table information are given in Table V. It is followed by the bytes in the quantization table.

TABLE IV. DQT SEGMENT

Field	Size (in bytes)	Description
Marker Identifier	2	0xFFDB
Length	2	the length of the Quantization Table
QT information	1	bit 0..3: number of Quantization Table bit 4..7: precision of Quantization Table
Bytes	N	Quantization Table values

TABLE V. QT INFORMATION

QT Precision	QT Index	Content type	Component
0	0	Bytes (8 bits)	Luminance component
0	1	Bytes (8 bits)	Chrominance component
1	0	Words (16 bits)	Luminance component
1	1	Words (16 bits)	Chrominance component

D. SOF: Start of Frame Marker

Start of Frame (SOF) segment specifies details about the frame as shown in Table VI. It starts with the 2 bytes marker identifier. The marker identifier 0xFFC0 represents baseline

DCT mode whereas 0xFFC2 represents progressive DCT mode. Identifier is followed by 2 bytes representing frame length. The data precision represents the number of intensity levels that can be represented and is usually 8 for baseline JPEG. The information about the image height and width, each of 2 bytes are retrieved from the SOF segment. The 1 byte 'number of components' field is followed by 3 bytes component specific information. The component identifier specifies the type of the component such as Y, Cb, Cr, I, Q based on the type of colour spaces (or colour models). A colour model is a mathematical model describing the range of colours as tuples of numbers. There are a variety of colour spaces, such as RGB, YCbCr, CMYK, HSV, HIS. JPEG generally uses YCbCr colour space. Component Identifier is followed by the sampling factors used by the JPEG encoders for Chroma subsampling. Chroma subsampling[12-14] is a process of encoding images by specifying less resolution for chrominance (colour) information than for luminance (brightness) information, because human eyes are less sensitive to colour information than brightness. With the use of Chroma subsampling, the amount of data can be reduced by sharing the colour information among adjacent pixels and that too without affecting the visual quality of the image.

TABLE VI. SOF SEGMENT

<table>
<tr><th>Field</th><th>Size (bytes)</th><th colspan="2">Description</th></tr>
<tr><td>Marker Identifier</td><td>2</td><td colspan="2">0xFFC0 / 0xFFC2</td></tr>
<tr><td>Length</td><td>2</td><td colspan="2">Frame length</td></tr>
<tr><td>Data precision</td><td>1</td><td colspan="2">Representation of intensity levels</td></tr>
<tr><td>Image height</td><td>2</td><td colspan="2">Height of the image</td></tr>
<tr><td>Image width</td><td>2</td><td colspan="2">Width of the image</td></tr>
<tr><td>Number of components</td><td>1</td><td colspan="2">1 for Grayscale, 3 for Color (YCbCr to YIQ) 4 for Color (CMYK)</td></tr>
<tr><td rowspan="3">Component specific information</td><td rowspan="3">3</td><td>Component Identifier:
1 for Y, 2 for Cb, 3 for Cr,
4 for I, 5 for Q</td><td>1 byte</td></tr>
<tr><td>Sampling factor</td><td>1 byte</td></tr>
<tr><td>Quantization table identifier</td><td>1 byte</td></tr>
</table>

The sampling pattern can be specified by two numbers: vertical sampling factor and horizontal sampling factor. The 1 byte sampling factor information is divided into 4 LSB bits specifying vertical sampling factor and 4 MSB bits specifying horizontal sampling factor. Horizontal sampling factor tells the number of pixels in the horizontal direction that shares 1 chrominance information. Vertical sampling factor tells the number of pixels in the vertical direction that shares 1 chrominance information. These sampling factors are used to decide the size of Minimum Coded Unit (MCU) [15].

In JPEG encoding, the image data is partitioned into a number of blocks called MCUs. Each MCU has a collection of data units, which are the smallest unit of image data that can be processed in a JPEG mode of operation. For e. g., for DCT based mode, each data unit is of size 8×8 pixels. For a non-interleaved scan, i.e. a scan with only one component, each MCU contains one data unit whereas for an interleaved scan, each MCU is a collection of data units. The size of an MCU is determined from the largest sampling factor of all components. Let 'hmax' and 'vmax' be the maximum horizontal and vertical sampling factors. The MCU size is determined as given in eqn. (1).

$$\begin{aligned} hsize &= 8 * h\max \\ vsize &= 8 * v\max \end{aligned} \tag{1}$$

For a JPEG image of size $M \times N$ and with three components (Y, Cb, Cr), the number of data units in a MCU is determined based on

(a) The horizontal and vertical sampling factors

(b) The number of MCUs in the horizontal and vertical direction for the image.

Let 'xmcu' and 'ymcu' represents number of MCU's in the horizontal and vertical direction respectively and are determined as given in eqn. (2).

$$\begin{aligned} xmcu &= \left(\frac{width}{hsize}\right) \\ ymcu &= \left(\frac{height}{vsize}\right) \end{aligned} \tag{2}$$

Let 'hsf' and 'vsf' be two arrays representing horizontal and vertical sampling factors for each component. The number of data units in one MCU is the sum of products of horizontal and vertical sampling factors of all the components as given in eqn. (3).

$$No.\ of\ \ data\ units = \sum_{i=1}^{n} hsf[i] * vsf[i] \tag{3}$$

Where n is the number of components.

Example:

Let width and height of an image be 188 and 268 respectively and the sampling factors of each component are given in Table VII.

TABLE VII. EXAMPLE- SAMPLING FACTORS

Component id	Hsf	vsf
1	2	2
2	1	1
3	1	1

The maximum sampling factor is 2×2. So $h\max = 2$ and $v\max = 2$ Then, $hsize = 16$ and $vsize = 16$. So the size of each MCU is 16×16. The number of MCU's in both directions is:

$xmcu = 12\ and\ ymcu = 17$

The number of data units in each MCU= $(2*2)+(1*1)+(1*1)=6$, i.e. four Y component data units, one C_b component data unit and one C_r component data unit.

E. DHT: Define Huffman Table Marker

JPEG standard gives a number of tables of Huffman codes that describe the various ways in which the quantized DCT coefficients are to be encoded. DHT segment helps to generate the Huffman tables which are needed for the entropy decoding process. The information about DHT segment is given in Table VIII. Huffman coding in JPEF differentiates the table based on the

(a) Type of coefficient: AC or DC coefficient

(b) Type of the component: Luminance or Chrominance component

TABLE VIII. DHT SEGMENT

Field	Size (in bytes)	Description
Marker Identifier	2	0xFFC4 to identify DHT marker
Length	2	Length of the Huffman table
HT information	1	Specifies the coefficient and component type
No. of symbols	16	No. of symbols with codes of length 1 to 16.
Symbols	N	Symbols in the order of increasing code length.

Four Huffman tables are defined for colour images whereas two tables for grayscale images. The AC and DC coefficients are coded separately and similarly luminance and chrominance components are coded separately. The 1 byte HT information separates these details as DHT class and DHT id. The least significant four bits specify the DHT class i. e. identifying whether it is AC or DC coefficient. The four most significant bits specify DHT id i. e. luminance or chrominance component. The identification of DHT class and id is given in Table IX.

TABLE IX. DHT CLASS and DHT ID

DHT class	DHT id	Type
0	0	Represents DC coefficients of Luminance component
1	0	Represents AC coefficients of Luminance component
0	1	Represents DC coefficients of Chrominance component
1	1	Represents AC coefficients of Chrominance component

The DHT table contains a number of codes or symbols of each bit-string length, followed by a sequence of all code-words. The actual bit-strings are not specified in the DHT; instead it lists out the length and code values. The HT information is followed by 16 bytes specifying the number of codes of each length. The actual symbols are specified after this. Let 'n' be the number of Huffman values, then the total number of bytes in the DHT segment is 19+n.

During decompression process, the compressed bit-strings are decoded to quantized DCT coefficients after decoding the Huffman table. For decoding the Huffman table, a binary tree is built from root to leaf node by assigning symbols in the increasing order of code length. A binary tree is a tree which can have maximum of two children. The tree starts from the root node that represents level zero of the tree. Then each time, new branches are created and code-words are assigned to the leaf nodes of the tree. Usually the left branch is labelled with bit '0' and right branch with bit '1'. Level 'i' stores code-words of length 'i'. In the absence of code-words of length 'i', spawn off a left and right node from each of the nodes in the i^{th} level. The JPEG Huffman trees (binary trees) are never deeper than 16 levels. Once the binary tree has been created for all the code-words in the DHT segment, bit-strings are read for each code-word by combining the bit labels of each branch on the path from root node. The data blocks after Huffman decoding represents the quantized DCT coefficients.
Example: One of the DHT segments of the test image is
1 *0 3 1 1 1 0 0 0 0 0 0 0 0 0 0 0* 1 2 3 4 0 5
The first byte (given in boldface) specifies the HT information, which is divided into two halves representing DHT class and DHT Id. Here, DHT Id=0 and DHT class=1 specifying that it is the AC Huffman table for luminance. Next 16 bytes (given in italics) specifies the number of symbols with code lengths from 1 to 16. There is no codes of length one, 3 codes of length two, 1 code of length three and so on. This is followed by n bytes representing the Huffman symbols or values. So after Huffman decoding by creating Huffman tree, the table is generated as shown in Table X. BITS field represents the code length, HUFFVAL represents the Huffman symbols extracted from the DHT segment and HUFFCODE represents the variable length Huffman code generated from the Huffman tree. After decoding the DHT segment, the Huffman code for the AC and DC coefficients in three components (Y, Cb, Cr) are shown in Fig. 4.

TABLE X. SAMPLE DHT SEGMENT

BITS	HUFFVAL	HUFFCODE
2	1	00
2	2	01
2	3	10
3	4	110
4	0	1110
5	5	11110

F. SOS: Start of Scan Marker

Start of Scan (SOS) segment contains the compressed data. It starts with the 2 bytes marker identifier, which is followed by scan header length of 2 bytes. Next is the 1 byte field for number of components in a scan, which specifies whether it is non-interleaved or interleaved scan. It is followed by a specification of each scan component selector. From the next 1 byte, 4 LSB bits represents AC entropy coding table selector and 4 MSB bits represents DC entropy coding table selector. A zero value represents luminance component whereas a value of one specifies chrominance component. This is followed by the compressed data. SOS length does not include the length of the compressed data. If the compressed image data contains 0xFF byte, it is usually followed by either a zero byte or a marker identifier. In the case of zero byte following 0xFF, the zero byte is discarded. For JPEG baseline mode, one SOS segment is present in the JPEG file whereas for progressive mode, multiple SOS segments are available with same length.

DRI: Define Restart Interval Marker

DRI markers are recognized within the entropy coded data; but are not explicitly written to the compressed data stream. It specifies the interval between restart markers [16] in MCU.

G. EOI: End of Image Marker

Every JPEG file ends with the standalone marker, EOI. EOI have the marker identifier of 0xFFD9.

III. EXPERIMENTAL RESULTS

A collection of software is available to visualize some metadata information. In this paper, the visualization of JPEG file is done by giving a summarized version of all the markers found in the test image. Fig. 1 shows the test image. and metadata information of the JPEG image is shown in Fig. 2, Fig. 3 and Fig. 4.

Fig. 1. Test Image

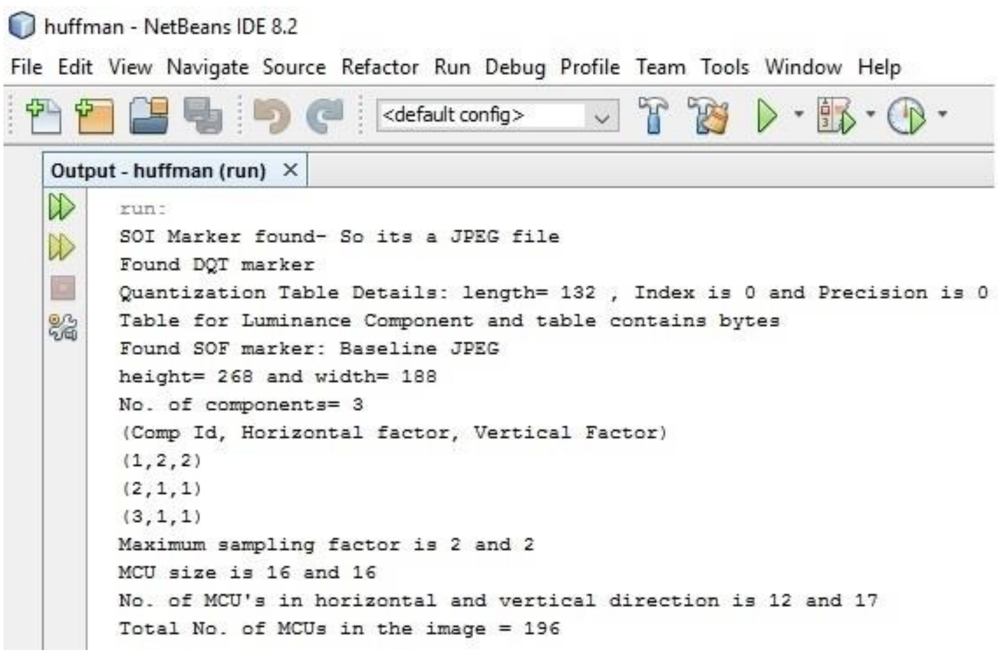

Fig. 2. SOI, DQT, SOF segment details

huffman - NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Output - huffman (run)

```
 DHT information
Offset= 173      HT length= 27   HT info = 0
Offset= 202      HT length= 56   HT info = 16
Offset= 260      HT length= 25   HT info = 1
Offset= 287      HT length= 37   HT info = 17
Processing HT info
DHT Class= 0 and DHT ID= 0: DC table for luminance
0  0  2  3  1  1  1  0  0  0  0  0  0  0  0  0  0  4  5  2  3  6  7  1  0
DHT Class= 0 and DHT ID= 1: DC table for chrominance
16 0 1 3 3 2 4 3 8 2 1 3 4 3 0 0 0 1 0 2 17 3 4 33 18 49 5 65 81 97 34 113 177 6 19 129 145 161 193
209 240 50 225 20 35 178 241 21 66 82 114 51 83 146
DHT Class= 1 and DHT ID= 0: AC table for luminance
1  0  3  1  1  1  0  0  0  0  0  0  0  0  0  0  0  1  2  3  4  0  5
DHT Class= 1 and DHT ID= 1: AC table for chrominance
17  0  2  2  3  0  3  0  2  1  5  0  0  0  0  0  0  0  1  2  17  3  33  49  4  18  65  34  81  145
19  20  50  97  113
```

Fig. 3. DHT Segment Details

huffman - NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Output - huffman (run)

Huffman codes- DHT DC-Y Component

BITS	HUFFCODE	HUFFVAL
2	00	4
2	01	5
3	100	2
3	101	3
3	110	6
4	1110	7
5	11110	1
6	111110	0

Huffman codes-DHT AC- Y Component

BITS	HUFFCODE	HUFFVAL
2	00	1
2	01	2
2	10	3
3	110	4
4	1110	0
5	11110	5

Huffman codes-DHT AC- CbCr Component

BITS	HUFFCODE	HUFFVAL
2	00	0
2	01	1
3	100	2
3	101	17
4	1100	3
4	1101	33
4	1110	49
6	111100	4
6	111101	18
6	111110	65
8	11111100	34
8	11111101	81
9	111111100	145
10	1111111010	19
10	1111111011	20
10	1111111100	50
10	1111111101	97
10	1111111110	113

Huffman codes-DHT DC- CbCr Component

BITS	HUFFCODE	HUFFVAL
2	00	1
3	010	0
3	011	2
3	100	17
4	1010	3
4	1011	4
4	1100	33
5	11010	18
5	11011	49
6	111000	5
6	111001	65
6	111010	81
6	111011	97
7	1111000	34
7	1111001	113
7	1111010	177
8	11110110	6
8	11110111	19
8	11111000	129
8	11111001	145
8	11111010	161
8	11111011	193
8	11111100	209
8	11111101	240
9	111111100	50
9	111111101	225
10	1111111100	20
11	11111111010	35
11	11111111011	178
11	11111111100	241
12	111111111010	21
12	111111111011	66
12	111111111100	82
12	111111111101	114
13	1111111111100	51
13	1111111111101	83

Fig. 4. Huffman table decoding

IV. CONCLUSION

Metadata stores various information about an image and are transmitted along with the image. Every image file format have its own metadata format. This paper detailed about the JPEG metadata or markers. In applications such as digital watermarking, image theft identification and digital forensics, some confidential information is embedded into the image. So in such applications, extraction of markers are essential in order to extract the embedded information back without even a single bit change.

REFERENCES

[1] CCITT, Information Technology – Digital Compression and Coding of Continuous-Tone Still Images –Requirements and Guidelines, Recommendation, *International Telecommunication Union, ITU T.81*; 1992, Available: http://www.w3.org/Graphics/JPEG/itu-t81.pdf, [Last Accessed: 20/05/2018].

[2] Hamilton, E.: JPEG File Interchange Format v 1.02, *C-Cube Microsystems*; 1992, Available: http://www.w3.org/Graphics/JPEG /jfif3.pdf, Last Accessed: 20/05/2018.

[3] Wallace GK., The JPEG still picture compression standard, *IEEE Transactions on Consumer Electronics*, vol. 38, no. 1, p. 18-34, 1992.

[4] JEITA CP-3451, Exchangeable image file format for digital still cameras-Exif Version 2.2, *Japan Electronics and Information Technology Industries Association* 2002, Available: http://www.exif.org/Exif2-2.PDF, [Last Accessed: 20/05/2018].

[5] P. Alvarez, "Using Extended File Information (EXIF) File headers in digital evidence analysis," *International Journal of Digital Evidence*, vol. 2, no. 3, 2004.

[6] M. W. Group et al., Guidelines for handling image metadata, Version. 2, 2010, Available: http://www.metadataworkinggroup.org/pdf/ mwg_guidance .pdf, [Last Accessed: 24/06/2018].

[7] Sencar H.T. and Memon N., "Identification and recovery of JPEG files with missing fragments," *Digital Investigation*, vol. 6, p. S88-S98, 2009.

[8] Malik Mohamad K., Deris M.M., "Visualization of JPEG Metadata," In: Badioze Zaman H., Robinson P., Petrou M., Olivier P., Schroder H., Shih T. K., *Visual Informatics:bridging research and practice*, IVIC, 2009.

[9] Jincy J Fernandez, P. Nithyanandam, Raghuvamsh Chavali, Ashwanth Kumar Appalaghe, "Image theft identification using biometric feature," *Special issue: Advances in Smart Computing and Bioinformatics, Asian Journal of Pharmaceutical & Clinical Research*, p. 385-389, 2017.

[10] H. Huang and W. Fang, "Metadata based image watermarking for copyright protection," *Simulation Modelling Practice and Thoery*, vol. 18, no. 4, p. 436-445, 2010.

[11] E. Kee, Johnson MK., H. Farid, "Digital Image Authentication from JPEG Headers," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 3, p. 1066-1075, 2011.

[12] Charles Poynton, Digital Video and HD: Algorithms and Interfaces, *The Morgan Kaufmann series in Computer Graphics*, 2012.

[13] E. Dumic, M. Mustra, S. Grgic and G. Gvozden, "Image quality of 4:2:2 and 4:2:0 chroma subsampling formats," *International Symposium ELMAR*, Zadar, p. 19-24, 2009.

[14] A. Wong and W. Bishop, "Practical content adaptive subsampling for image and video compression," *Proceedings of the 8th International Symposium on Multimedia*, 2006.

[15] Iwata M., Miyake K., Shiozaki A., "Digital steganography utilizing features of JPEG images," *IEICE Transactions on fundamentals of electronics, communications and computer sciences*, e87-a, no. 4, p.929-936, 2004.

[16] Karresand K., Shahmehri N., "Reassembly of fragmented JPEG images containing restart markers," *European Conference on Computer Network Defence,* p. 25-32, 2008.