Detecting Copy-Move Forgery Using Matrix Singular Value Decomposition

Group Members:

Sandra Nettey

Wofa Yamoah Attafuah

Kwabena Arhinkorah

Department of Computer Science and Information Systems, Ashesi University

Discrete Structures and Theory

Dr. Ayawoa Dagbovie

April 22nd, 2022

Introduction

Forgery in multimedia in the modern age has become the norm, and images are very popular, but they are pretty easy to manipulate. This may have harmful consequences, significantly when the editing changes important content in the pictures, from teens making their selfie photos more flattering to evidence being altered for criminal purposes. Therefore, detecting traces of changes in digital images is an urgent need. Image processing involves the extraction, filtering, and enhancement of images using mathematical operations. “In image processing, various alterations are performed on an image by performing mathematical operations on the corresponding matrix.” (Prabhune et al.,2017).

Every digital image can be stored as a matrix. The matrix has as many rows and columns as the image has pixels horizontally and vertically, respectively. Each cell in the matrix corresponds to the color and color intensity of a particular pixel in the image. Various mathematical operations are performed on these matrices to alter their corresponding images. Detecting tampered regions and proving the authenticity and integrity of a digital picture have become increasingly important in digital forensics and multimedia security. Although there are different ways to ensure that forgery in digital images is suppressed, such as using watermarks as digital signatures, the majority of images do not contain digital watermarks, so there is a need for another solution. The solution we have chosen is to identify copy-move image tampering by applying the singular value decomposition (SVD). Copy-move forgery is a type of image tampering where a section of an image is copied and pasted on another part of the same image in order to produce a brand-new image. Singular Value Decomposition (SVD) is an orthogonal matrix transformation that divides the matrix into three different matrices and concentrates image energy into several fixed elements of the transformed matrices.

The decomposition of a matrix by Singular Value Decomposition is as follows:

A m × n matrix, S can always be broken down to three matrices:

A = U × ∑ × V, where U is an m × m orthogonal matrix,

V is an n × n orthogonal matrix, and V^T is the matrix transposition of V.

∑ is a diagonal matrix, i.e., $\sum_{ij}$ = 0 for i ≠ j. The diagonal entities of ∑ are called the singular values of the matrix A and $\sum_{11} \geq \sum_{22} \geq \ldots \geq \sum_{kk}$, where k = min {m, n}. (Nguyen, H. C., & Cao, T. L., 2019).

Methodology

This section presents the step-by-step procedure for employing Singular Value Decomposition in detecting copy-move image forgery and further provides our simplified code demonstration of the process.

Suppose you have an image you believe is authentic, O, and another image you believe has been forged, F. To verify the authenticity of F, you would need to carry out the following steps:

- O and F are converted from their coloured RGB forms to a YUV colour model where the Y-components (brightness or luminance) of their colours are extracted, transforming them into grayscale (black and white) images.
- The gray-level images, O and F, are each divided into n blocks of equal sizes that do not overlap to obtain a set of tiles from O, $\{O_1, O_2, \ldots, O_n\}$ and another set of tiles from F, $\{F_1, F_2, \ldots, F_n\}$.
- Each chunk from both images is then converted into matrices, and Singular Value Decomposition is applied to each individual chunk matrix to obtain the characteristic information of that chunk.

- For O, after each chunk matrix decomposition, the largest singular values of that chunk are extracted from its diagonal matrix $\sum$ and added to an overall singular value matrix D_o. The exact process is repeated to F to obtain a matrix D_F.
- Singular Value Decomposition is again performed on both matrices, D_O and D_F, to obtain a singular value vector for each.
- The elements of the singular value vectors are, however, massive and thus, need to be normalized by the formula, $\eta_i = \frac{\sigma_i}{\sigma_1}, i = 1,2,\ldots b$ where σ_i represents a singular value element and i represents the index of that singular value element. The normalized singular values of O and F form new vectors known as feature vectors. At this stage, the feature vectors of O and F contain information characteristic of each image, and the comparison can now begin.
- In order to quantify the differences between the two images, the Euclidean distance of each pair of values at every index between the feature vectors of O and F are then computed using the formula $D(Q,P) = \sqrt{\sum_{i=1}^{b} (q_i - p_i)^2}$.
- The similarity measure (SM) of the two images is then calculated according to the formula $\textbf{SM}\ (Q,P) = \frac{1}{1 + D(Q,P)}$. If SM is above a predetermined level between 0 and 1, we can conclude that F is a forged or counterfeit version of O.

Demonstration

In order to demonstrate how Singular Value Decomposition can be used to identify copy-move image forgery, we implemented a simplified version of the above procedure in the Java programming language. We modularized our work by separating most of the tasks carried out in our code into independent functions, which allowed us to avoid running all parts of the code each time and avoid unnecessary repetitions. Using two images, “rabbits_1” and “rabbits_2,” where rabbit_1 is the original image and rabbit_2 is the image upon which copy-move forgery has bn perform, we would like to use our code to perform SVD analysis on the two images and output the sections of rabbit_1 that have been copied and pasted to create rabbit_2. See Fig. 1 for the images of rabbit_1 and rabbit_2 and the sections of rabbit_1 that were copied.

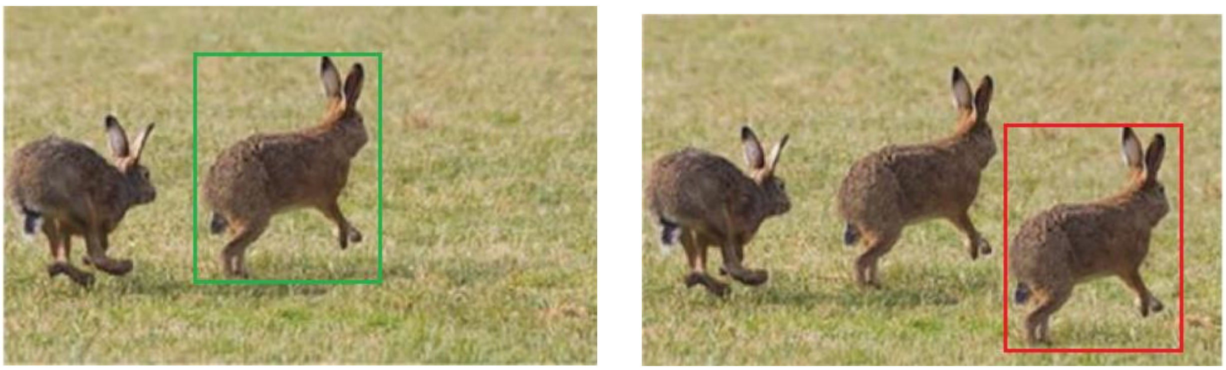

Fig. 1: The images rabbit_1 and rabbit_2, with the green border demarcating the copied section and the red border where it was pasted. (Mahmood et al., 2016).

The image to be tested for forgery can be the only input given to the program but for this demonstration, we used both the original and test images. Firstly, we imported the two images, with rabbit_1 as the original image and rabbit_2 as the test image, extracted their luminance, and split them each into 16 overlapping chunks.

```
public static BufferedImage Lum( BufferedImage current ){
        int height = current.getHeight(null);
        int breadth = current.getWidth(null);
        for (int i = 0; i < breadth; i++) {
            for (int j = 0; j < height; j++) {
                int pixel = current.getRGB(i,j);
                Color colour = new Color(pixel, true);

                int red = colour.getRed();
                int green = colour.getGreen();
                int blue = colour.getBlue();
                double Y = (0.299 * red) + (0.586 * green) + (0.114 * blue);
                int Y_ = (int) (Math.round(Y));
```

```
                Color gray = new Color(Y_, Y_, Y_);
                int gn = gray.getRGB();
                current.setRGB(i, j, gn);
            } } return current; }
```

Fig. 3: Lum function to convert images to grayscale. The function takes an image as its argument, extracts its Y component or luminance, and converts it to grayscale as a result.

Next, we took each chunk from both images, and obtained its matrix representation. SVD was then performed on each chunk matrix to obtain its singular values.

```
public static double[][] getImageMatrix(BufferedImage image) {
        int height = image.getHeight(null);
        int breadth = image.getWidth(null);
        double[][] ImageMatrix = new double[breadth][height];
        //Assigning the RGB values of each pixel to each cell in the matrix
        for (int i = 0; i < breadth; i++) {
            for (int j = 0; j < height; j++) {
                ImageMatrix[i][j] = image.getRGB(i, j);
            }
        }
        return ImageMatrix;
    }
```

Fig. 4: getImageMatrix function to convert an image to a matrix. The function takes an image its argument, creates a two-dimensional array, and assigns the RGB value of each pixel in the image to its corresponding index in the array.

```
public static double[] Singularize(BufferedImage img){
        Matrix A = new Matrix(getImageMatrix(img));
        SingularValueDecomposition S = new SingularValueDecomposition(A);
        double[] U1 = S.getSingularValues();
        return U1;
    }
```

Fig. 5: Singularize function takes in an image, converts it to a matrix using the getImageMatrix function, performs Singular Value Decomposition, and adds singular values to an array.

At this stage, the Euclidean Distance function in Fig. 6 was used to compare the singular values of every chunk of the original image to every chunk of the image being tested for forgery, in order to find similar chunks. A threshold range of 1,000,000 to 100,000,000 was found to be suitable for our purposes after multiple rounds of trial and error. Matched pairs of chunks were then output to the folder containing the code, and the program prints out whether the test image is forged or authentic.

Results

After running the images rabbit_1 and rabbit_2 through the program, the following two pairs of images were obtained:

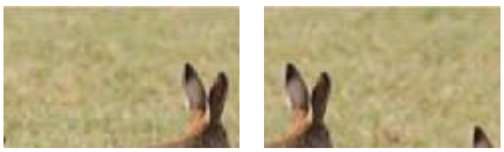

Fig. 6: First pair of matched image chunks

Fig. 7: Second pair of matched image chunks

The left image chunk of each pair was obtained from the original image, and the right image chunk was obtained from the test image, while the program printed out that the test image was forged. These results concur with the highlighted sections from Fig. 1 and prove that Singular Value Decomposition can be used to detect image forgery.

Discussion and Conclusion

The team learnt how the pixelation of images in matrices could be used to detect forgery. This is relevant in the world we find ourselves in today as the copy-pasting of pictures for various purposes has become digitally accessible. This helps us to distinguish between an authentic image and a forged one. One significant finding was that the larger the size of blocks compared to the original, the less accurate the results are. Hence, a further improvement to this project is to reduce the number of square pixel blocks that can be captured and compared with the original images to improve the accuracy of the results. A limitation of SVD is its robustness, with QR decomposition preferable in this regard (Nguyen, H. C., & Cao, T. L., 2019). An idea for a future update to SVD techniques is a modification such that it can also help detect copy-moved images in videos.

References

Compton, E. A., & Ernstberger, S. L. (n.d.). *Singular Value Decomposition: Applications to Image Processing*. 7.

Kang, X., & Wei, S. (2009). An efficient approach to still image copy detection based on SVD and block partition for digital forensics. *2009 IEEE International Conference on Intelligent Computing and Intelligent Systems*, 457–461.

Nguyen, H. C., & Cao, T. L. (2019). Using Matrix Decomposition and Frequency Transforms to Detect Forgeries in Digital Images. *2019 IEEE-RIVF International Conference on Computing and Communication Technologies (RIVF)*, 1–4.

Prabhune, O., Sabale, P., Sonawane, D. N., & Prabhune, C. L. (2017). Image processing and matrices. *2017 International Conference on Data Management, Analytics, and Innovation (ICDMAI)*, 166–171.

Zhang, T., & Wang, R. (2009). Copy-Move Forgery Detection Based on SVD in Digital Image. *2009 2nd International Congress on Image and Signal Processing*, 1–5.

Mahmood, T., Nawaz, T., Irtaza, A., Ashraf, R., Shah, M., & Mahmood, M. T. (2016). Copy-Move Forgery Detection Technique for Forensic Analysis in Digital Images. *Mathematical Problems in Engineering*, *2016*, 1–13. https://doi.org/10.1155/2016/8713202