Ministry Category : Department of Defence Production, Ministry of Defence (MOD)
Problem Statement : Real-Time Image Processing & Forensic Verification of Documents **Problem Code :** #MOD5
Team Leader Name : Chaitanya Tejaswi **College Code :** #2451

Idea Description

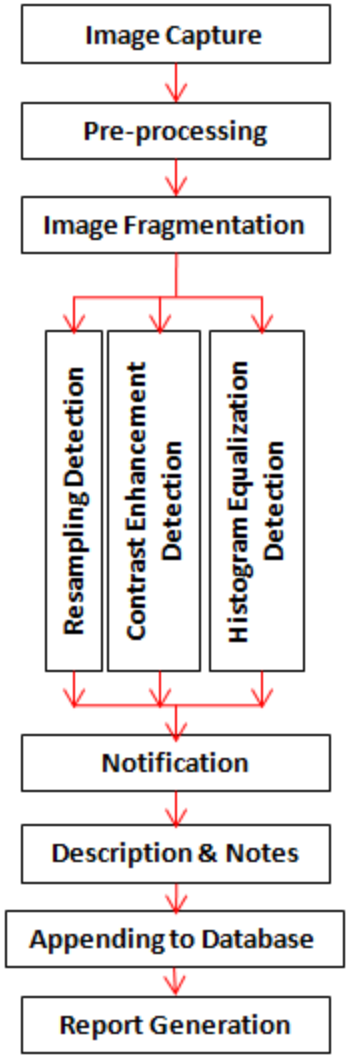
The user will be able to detect & verify computer-based manipulations (if any) on a set of digitized images, using a GUI application on a PC.
In common image forgeries, such as copy paste, region duplication, image splicing, etc. , basic image operations are often involved. So, if we can find significant evidence in favour, it can be infered that the image is altered.
Our work is based on this paper. The application development will be done mainly in Python.
The 8 main stages are described below –

1. **Image Capture** - capturing the image on a flat-bed scanner.
- Images obtained using a smartphone camera suffer from angular-defects, likely caused due to tilted position of the camera, when the image is taken. Compensation for this has to be provided, which results in poor performance & significant increase in code-size.
- Images obtained using a (flatbed) scanner don't need this. Also, most work in OCR technology has based on such images. So it's helpful if the images are scanned directly.
- One more reason for this is that we can choose to save the images in PNG/TIFF formats (instead of JPEG), which is useful for OCR detection. However, JPEG files are also processed.
- Also, if smartphone images are flattened using Android apps (e.g. CamScanner), the applied changes will be detected as "altered" (in later stages).
- In short, we need scanned document images.

2. Pre-processing - making the image more suitable feature detection.
- We are using the opencv-python module for cropping images, feature extraction and image enhancement of textual regions.

3. Image Fragmentation – based on the image resolution, it will be divided into 3x3 blocks. The main algorithm will be applied to each block concurrently.

Flow Diagram



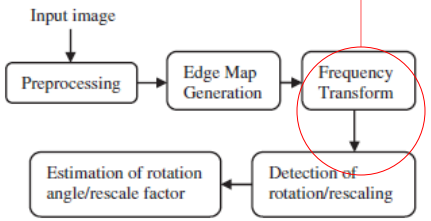
4. Main Algorithm –

This step consists of 3 main techniques:

A. Resampling (Rotation + Rescaling) Detection

B. Contrast Enhancement Detection

C. Histogram Equalization



The steps in re-sampling detection method.

In DA Method, the magnitude of DFT is calculated for each row of the edge map and then the average is taken over all the rows to get the horizontal spectrum. Assume that $E(m,n), m \in [1,M],\ n \in [1,N]$ are the entries of the edge map and F is the Discrete Fourier Transform. The DA method can be expressed as follows:

$$E_{DA} = \frac{1}{M}\sum_{m=1}^{M}|F[E(m,n)]| \quad (1)$$

In AD Method, the average of all rows of the edge map is calculated to form a horizontal row and then the magnitude of DFT is calculated to get the horizontal frequency spectrum. The AD method can be defined as follows:

$$E_{AD} = \left|F\left[\frac{1}{M}\sum_{m=1}^{M}E(m,n)\right]\right| \quad (2)$$

A. For ***Resampling Detection*** we need 2 frequency plots –

- DFT, then Averaging = DA Method
- Averaging, then DFT = AD Method

*DFT = Discrete Fourier Transform

The reason for having such plots is because if there are peaks only in the DA plot, a rotation has occurred. If there are peaks in both DA & AD plots, the image has been rescaled (zoomed). The rotation angle (Θ) & rescale factor (R) are determined by equations –

$$f_{rot1} = \begin{cases} 1-\cos\Theta, & 0^\circ < \Theta \le 60^\circ \\ \cos\Theta, & 60^\circ < \Theta < 90^\circ \end{cases}$$

and

$$f_{rot2} = \begin{cases} \sin\Theta, & 0^\circ < \Theta \le 30^\circ \\ 1-\sin\Theta, & 30^\circ < \Theta < 90^\circ \end{cases}$$

$$f_{res} = \begin{cases} 1-1/R, & 1 < R \le 2 \\ 1/R, & R > 2 \end{cases}$$

or

$$f_{res} = 1/R - 1, \quad R < 1$$

Obviously, forged images have combined rescaling & rotation. Similar equations for such cases are also available.

B. A similar approach is taken for ***Contrast Enhancement Detection***.

- First, if it's a colour image, RGB channels are separated.
- Histogram for two of these channels is computed separately.
- Magnitude of each such histogram is calculated, and used to obtain a frequency plot.

If there are sudden peaks/zeroes in the plot, the image is said to be contrast-enhanced. Why? If images are contrast enhanced, their pixel values are increased. So is their energy content. Since Energy Frequency, the enhanced images have high frequency components.

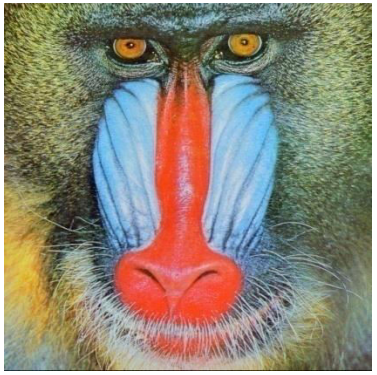
C. ***Histogram Equalization***, also a form of contrast-enhancement, provides sharp peaks in frequency-plots of colour-enhanced images. If this is applied locally to the whole image by dividing into 3x3 blocks, we can estimate regions with the most alterations.

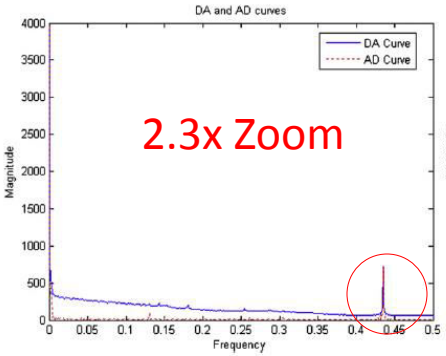
Steps **5-8** are **Notification**, **Description**, **Appending-to-Database** & **Report Generation**.

1. The user is *notified* about the detected alterations. In response, the user can add a *description* specifying what it is that has been altered.
2. The description for all the images is processed an stored into a database. A report for the same can optionally be generated.

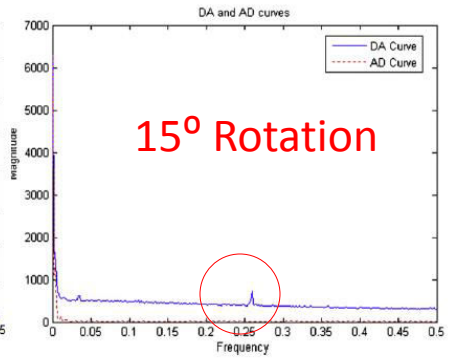
Example

DA & AD plots for Rescaled/Rotated Image





The DA and AD curves of rescaled Baboon image.



The DA and AD curves of rotated Baboon image.

Technology Stack

- PyQt4 module *(for GUI application)*
- numpy, scipy, opencv-python modules *(for image processing)*
- matplotlib module *(for graph plotting)*
- MySQL API *(for managing image database)*
- Notepad++ Editor

Users

Office Staff, who will upload the images for forensic verification.

Use-Case

- The User will specify the set of images, using “Upload Images”. (See the GUI application template to the side)
- After a while, the “Previous/Next Image” button become active.
- Next, the “Issues” section starts listing all alteration-detected fragments for the current image. The “Check” button, which now becomes active, allows the user to verify these alterations
- Using the “Notes” section, the user can add details about the detected alterations.
- Once all the images have been processed, the user will click “Submit”. All detected information will be appended to database.
- Optionally, a report can be generated for the session.

Dependencies

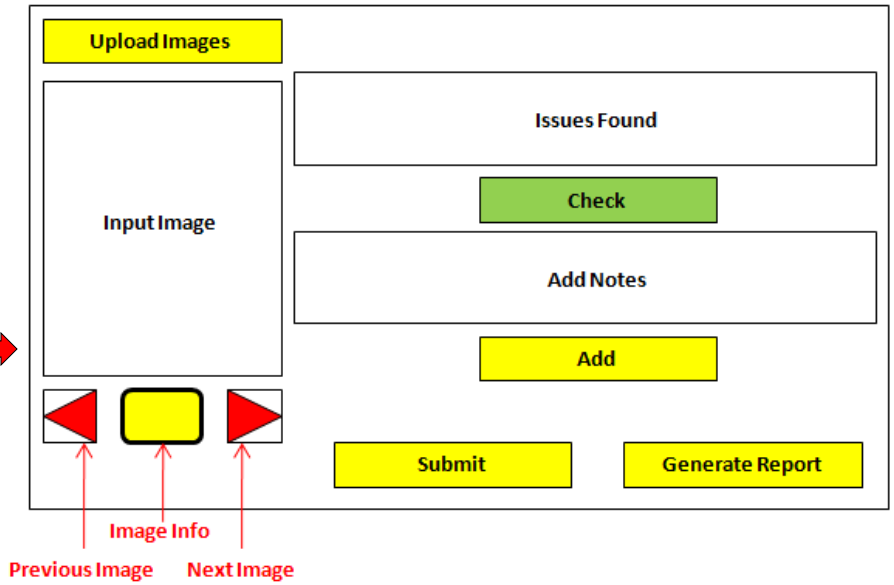
- Most of our application would be in Python, and the primary system dependency for it would be a Windows 7 system, with at least 1 GB of RAM.

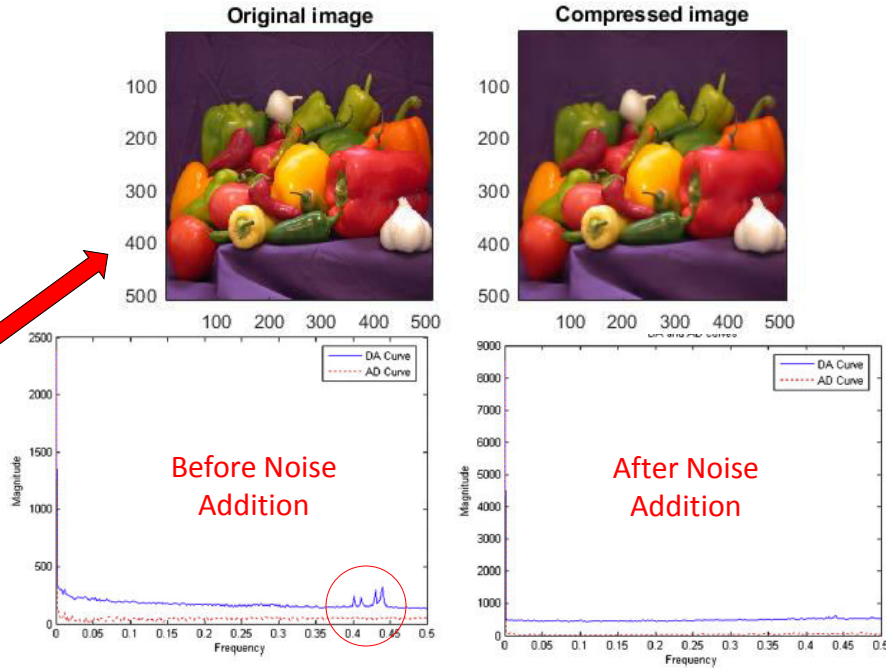
Showstoppers

JPEG images

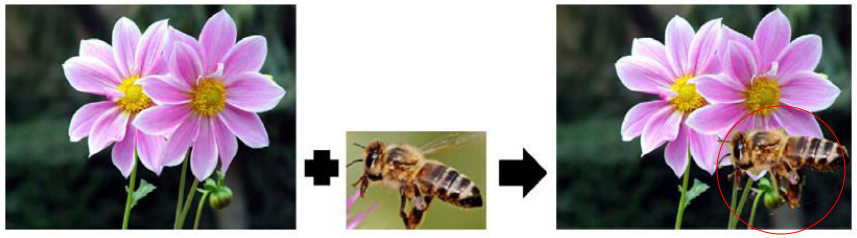
- The proposed technique indicates the presence of re-sampled image regions in an image. However, it is susceptible to “JPEG attacks” – periodic JPEG blocking artifacts coinciding with periodic patterns introduced by resampling.
- This is suppressed by adding Gaussian noise to the JPEG image.

User Application Window

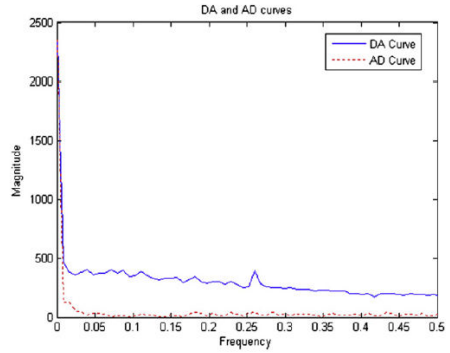




Examples

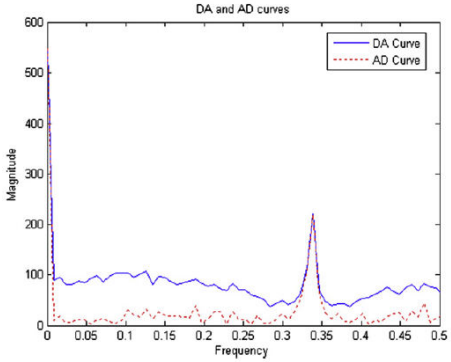


The original flower and bee images (first and second) and the forged image (third) obtained by rotating the bee image by 15° and pasting it to the flower image.



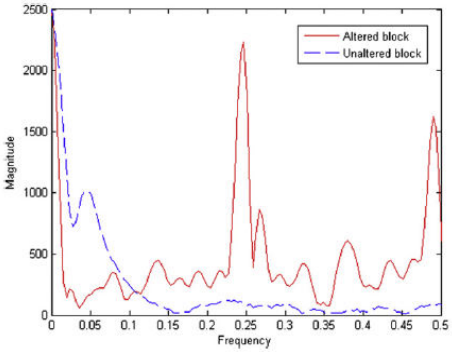
The DA and AD curves of one of the fake image blocks of the forged image. A peak appears at 0.26 in DA method because of rotation by 15°.

The original image (left) and the forged image (right) obtained by copying and rescaling an original image block and pasting it within that image.



The DA and the AD curves of one of fake image blocks of the forged image. A peak appears at 0.33 in both methods because of rescaling by a factor of 3.

The two original images (first and second) used for creating the forged image (third) which is an example for local histogram equalization.



The frequency spectrum plots of unaltered block and one of the altered blocks of the forged image. A striking peak appears in the plot of altered image block at 0.25.