Detecting Morphological Filtering of Binary Images

Abstract—This paper presents a new method for detecting morphological filtering on binary images, commonly used in tasks like noise removal and structure detection. The approach utilizes mathematical properties of erosion and dilation operators to detect the application of the filter and estimate the structuring element's shape. The technique is shown to be effective and robust in experimental tests against common binary image document operations.

Introduction

This paper addresses the problem of detecting forgeries in digital documents, particularly the application of morphological erosion or dilation to binary images. Such filters are often used to remove artifacts and hide previous operations, making them a hint of possible malicious modifications. The proposed algorithm detects not only the application and type of the filter, but also the shape of the structuring element used. The approach utilizes some properties of the filtered signal that allow distinguishing it from an original one in a deterministic way, without relying on statistical approaches or machine learning. Experimental results demonstrate the effectiveness and robustness of the technique.

PROPOSED APPROACH

Some basic overview of Mathematical Morphology Theory.

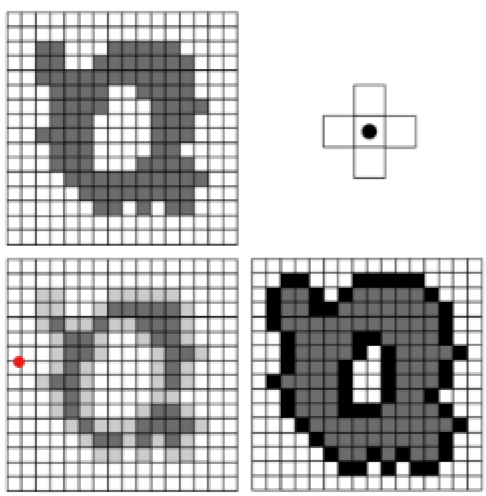
Mathematical morphology is a set of nonlinear filters based on two fundamental operators, erosion and dilation, and a binary mask called structuring element. The shape of the structuring element determines the effect of the filter on the image. While originally designed for binary images, mathematical morphology has been extended to grayscale pictures. Erosion and dilation are defined as the set of all points where the structuring element fits within the image and the set of all points where the structuring element overlaps with the image, respectively.

For given an binary image A and a structuring element B, the two basic operators, erosion and dilation, respectively, are defined as follows:

$$A \ominus B = \{x \ s.t. \ B_x \subseteq A\}$$
$$A \oplus B = \{x \ s.t. \ B_x^S \cap A \neq \emptyset\}$$

where Bx represents the structuring element B with the reference point positioned in x, while B_x^S represents the reflective rotation of B positioned in x.

Below figure shows an example of erosion and dilation of a character with the depicted structuring element.



Two composite morphological filters as combination of erosion and dilation name as **open** and **close** operators. It is the sequence erosion-dilation and dilation-erosion, respectively:

Open is given as $\mathbf{A} \circ \mathbf{B} = \mathbf{A} \ominus \mathbf{B} \oplus \mathbf{B}$

Close is given as $\mathbf{A} \bullet \mathbf{B} = \mathbf{A} \oplus \mathbf{B} \ominus \mathbf{B}$.

Where **A** is an image and **B** is a structuring element.

Some important properties of Morphological operators.

- **Translation invariance**: The translation of the filtered image depends on the position of the reference point only.
- **Commutativity of the dilation**: $A \oplus B = B \oplus A$.
- **Associativity**: The cascade of multiple filters can be transformed into a single filter whose structuring element is the morphological combination of the original elements, according to the following equations:

  $A \ominus B1 \ominus B2 = A \ominus \{B1 \oplus B2\}$

  $A \oplus B1 \oplus B2 = A \oplus \{B1 \oplus B2\}$
- **Idempotence of open and close**: The iteration of open and close operators with the same structuring element does not produce any further modification of the image:

  $A \circ B \circ B = A \circ B$

  $A \bullet B \bullet B = A \bullet B$.

Our proposed approach follows the two basic theorems as below .

Theorem 1: Let $C = A \ominus B$, then $C \bullet B = C$. Equivalently, let $C = A \oplus B$, then $C \circ B = C$.

Here C is an eroded image with structuring element B.

The statement of Theorem 1 can be rewritten as follows:

$A \ominus B = A \ominus B \oplus B \ominus B$

and equivalently.

$A \oplus B = A \oplus B \ \ B \oplus B$.

Theorem 2: Let $C = A \ominus B$, then for all D such that there exist E such that $D \oplus E = B$ we have that $C \bullet D = C$.in the same way , given $C = A \oplus B$, then $C \circ D = C$.

Proof of theorem 2: $C = A \ominus B = A \ominus D \oplus E$ (from theorem2 $D \oplus E = B$)

$A \ominus E \oplus D = A \ominus E \ominus D$ using Commutativity property $D \oplus E = E \oplus D$ and Associativity property $A \ominus E \ominus D = A \ominus \{E \oplus D\}$

Now, given $F = A \ominus E$, for Theorem 1 we have: $C = F \ominus D = \{F \ominus D\} \bullet D$

$\Rightarrow F \ominus D \oplus D \ominus D$

$\Rightarrow A \ominus E \ \ominus D \oplus D \ominus D$

$\Rightarrow C \oplus D \ \ominus D$

$\Rightarrow C \bullet D$.

According to Theorem 2, if the open or close operations are carried out with any structural element D that can be dilated to yield B, the equality established by Theorem 1 still holds true.
We will specifically see that Theorem 1 offers a useful method to identify the most recent erosion/dilation operation applied to an image, whereas Theorem 2 permits significantly reducing the computational complexity of the detector by reducing the number of structuring elements to be taken into account during the detection process.

Detection Algorithm

The idea is the following: given an unknown binary image I we want to determine whether it has been filtered with a morphological erosion, a morphological dilation, or none.
Let suppose that I is the result of an erosion operator with a structuring element B, then we will have $I = I' \ominus B$,where I is in general not available. Given Theorem 1, and assuming that we know B, we can state that:

$$I \bullet B = I \oplus B \ominus B \Rightarrow I' \ominus B \oplus B \ominus B \Rightarrow I' \ominus B \Rightarrow I .$$

This offers a useful method to determine whether I is the outcome of an erosion with B: simply apply a series of dilation and erosion with B to I and determine whether the result is pixel wise equal to I. The appropriate dilation detector is given by reversing the erosion and dilation operators in the above equation. It is obvious that the result will generally differ from I if the erosion detector is applied to a dilated or unfiltered image, and the same is true if the detection is carried out using a structuring element B distinct from B.

The passage describes the use of morphology as a regularization tool for image forensics, specifically in detecting possible matches between a document and a set of masks. However, since knowing the exact mask is unrealistic, the detector needs to scan through the set of masks. The passage then introduces Theorem 2, which allows for an optimization in the search process by defining a set of implications for each mask that can be obtained by dilating it. If a no-match is found for a particular mask, all the masks in its implication set can be excluded from further analysis. This reduces the solution space and the number of tests to be performed. The implications for each mask in the set of 36 structuring elements used in the paper are also listed in Table I. The passage suggests arranging the sequence of checks starting from smaller filters and progressively excluding larger ones that fulfill the implication rules.

STRUCTURING ELEMENTS EXCLUSIONS IMPLIED BY THEOREM 2: FOR EACH B_i THE CORRESPONDING SET $\hat{\Omega}_i$ IS REPORTED, ALONG WITH THE GENERATING MORPHOLOGICAL OPERATIONS (IN PARENTHESIS)

B_i	$\hat{\Omega}_i$	
B_1	B_{15} $(B_1 \oplus B_2)$ B_{27} $(B_1 \oplus B_{15})$ B_{35} $(B_1 \oplus B_2 \oplus B_5)$	B_{16} $(B_1 \oplus B_1)$ B_{28} $(B_1 \oplus B_{17})$ B_{36} $(B_1 \oplus B_{28})$
B_2	B_{15} $(B_2 \oplus B_1)$ B_{27} $(B_2 \oplus B_{16})$ B_{35} $(B_2 \oplus B_1 \oplus B_5)$	B_{17} $(B_2 \oplus B_2)$ B_{28} $(B_2 \oplus B_{15})$ B_{36} $(B_2 \oplus B_{27})$
B_3	B_{18} $(B_3 \oplus B_3)$	B_{29} $(B_3 \oplus B_1')$
B_4	B_{19} $(B_4 \oplus B_4)$	B_{30} $(B_4 \oplus B_2')$
B_5	B_{26} $(B_5 \oplus B_5)$	B_{35} $(B_5 \oplus B_{15})$
B_6	B_{36} $(B_6 \oplus B_{15})$	
B_7	B_{31} $(B_7 \oplus B_{14})$	
B_8	B_{32} $(B_8 \oplus B_{13})$	
B_9	B_{34} $(B_9 \oplus B_{11})$	
B_{10}	B_{33} $(B_{10} \oplus B_{12})$	
B_{11}	B_{34} $(B_{11} \oplus B_9)$	
B_{12}	B_{33} $(B_{12} \oplus B_{10})$	
B_{13}	B_{32} $(B_{13} \oplus B_8)$	
B_{14}	B_{31} $(B_{14} \oplus B_7)$	
B_{15}	B_{27} $(B_{15} \oplus B_1)$ B_{35} $(B_{15} \oplus B_5)$	B_{28} $(B_{15} \oplus B_2)$ B_{36} $(B_{15} \oplus B_{15})$
B_{16}	B_{27} $(B_{16} \oplus B_2)$	B_{36} $(B_{16} \oplus B_{17})$
B_{17}	B_{28} $(B_{17} \oplus B_1)$	B_{36} $(B_{17} \oplus B_{16})$
B_{27}	B_{36} $(B_{27} \oplus B_2)$	
B_{28}	B_{36} $(B_{28} \oplus B_1)$	

Algorithm 1 Erosion Detector

Input: Test image I, structuring elements $\Omega = \{B_i\}, i \in [1, N]$
Output: Positive or negative detection of erosion on I
Method:

$\hat{\Omega} = \emptyset$, $E = \emptyset$
for $i = 1, \dots, N$ **do**
  if $B_i \in \Omega \setminus \hat{\Omega}$ **then**
    compute $I_C \bullet B_i$
    if $I_C = I$ **then** $E = E \cup B_i$ // match
    else $\hat{\Omega} = \hat{\Omega} \cup \hat{\Omega}_i$ // no match, exclude implications
end for
if $E == \emptyset$ **then return** negative
else return positive to mask B_{max}, $max = \max i$ s.t. $B_i \in E$

Where I is image on which algorithm check whether it is eroded or not. **Ω** contain the set of 36 structuring element.
Take two empty set **Ω^** and **E** . now compute $I \bullet B_i$ for each structuring element which is in **Ω** - **Ω^** one by one. Let's say $I_C = I \bullet B_i$ so now check
If $I_C == I$ add that structuring element in the set **E** otherwise add that structuring element in the set **Ω^** . at the end of iteration we check
If set **E** is empty then return negative which means image is not eroded
else return the maximum size structuring element for which it is detected positive.

For instance if masks B1, B2, B3, B4 fail, we can exclude from the check $\boldsymbol{\Omega}_1\hat{} \cup \boldsymbol{\Omega}_2\hat{} \cup \boldsymbol{\Omega}_3\hat{} \cup \boldsymbol{\Omega}_4\hat{}$ = {B15, B16, B17, B18, B19, B27, B28, B29, B30, B35, B36}. Similarly, if the first 10 small structuring elements fail, we can exclude 16 bigger masks from the test. This allows us to reduce significantly the computation

If an element Bi does not match with the image, then we can exclude all other elements in the set $\boldsymbol{\Omega}_i\hat{}$, which are the set of all masks that can be obtained by dilating B_i .This is done using Theorem 2, which states that if a structuring element Bi fails, every other structuring element that can be obtained by dilating B_i will also fail. On the other hand, if an element B_i does match with the image, this does not exclude the possibility that a larger element $B_j \in \boldsymbol{\Omega}_i$ could also match. In this case, the largest structuring element in the positive subset will be outputted as the result.

Experimental Results

On our images(50 images)

Initially , we have tested our implementation on our dataset

For the demo we have used 11 images and 20 structuring elements starting from B1 to B2

The robustness of our implementation can be understood by the matrix we got

Making Table

11.0	2.0	1.0	1.0	1.0	2.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	2.0	2.0	2.0	1.0	1.0	1.0
2.0	11.0	1.0	1.0	1.0	2.0	1.0	2.0	1.0	2.0	1.0	1.0	2.0	1.0	2.0	2.0	2.0	1.0	1.0	1.0
2.0	2.0	11.0	1.0	1.0	2.0	1.0	2.0	1.0	2.0	2.0	1.0	2.0	2.0	2.0	2.0	2.0	1.0	1.0	1.0
2.0	2.0	1.0	11.0	1.0	2.0	1.0	2.0	1.0	2.0	1.0	2.0	2.0	1.0	2.0	2.0	2.0	1.0	1.0	1.0
2.0	2.0	1.0	1.0	11.0	2.0	1.0	2.0	1.0	2.0	1.0	1.0	2.0	1.0	2.0	2.0	2.0	1.0	1.0	1.0
2.0	2.0	1.0	1.0	1.0	11.0	1.0	2.0	1.0	2.0	1.0	1.0	2.0	1.0	2.0	2.0	2.0	1.0	1.0	1.0
2.0	2.0	1.0	1.0	1.0	2.0	11.0	2.0	1.0	2.0	1.0	1.0	2.0	1.0	2.0	2.0	2.0	1.0	1.0	1.0
2.0	2.0	1.0	1.0	1.0	2.0	1.0	11.0	1.0	11.0	1.0	1.0	2.0	1.0	2.0	2.0	2.0	1.0	1.0	1.0
3.0	2.0	1.0	1.0	1.0	2.0	1.0	2.0	11.0	2.0	1.0	1.0	2.0	1.0	2.0	2.0	2.0	1.0	1.0	1.0
2.0	2.0	1.0	1.0	1.0	2.0	1.0	11.0	1.0	11.0	1.0	1.0	2.0	1.0	2.0	2.0	2.0	1.0	1.0	1.0
3.0	2.0	1.0	1.0	1.0	2.0	1.0	2.0	1.0	2.0	11.0	1.0	2.0	1.0	2.0	2.0	2.0	1.0	1.0	1.0
3.0	2.0	1.0	2.0	1.0	2.0	1.0	2.0	1.0	2.0	1.0	11.0	2.0	1.0	2.0	2.0	2.0	1.0	1.0	1.0
2.0	2.0	1.0	1.0	1.0	2.0	1.0	3.0	1.0	3.0	1.0	1.0	11.0	1.0	2.0	2.0	2.0	1.0	1.0	1.0
2.0	2.0	1.0	1.0	1.0	2.0	1.0	2.0	1.0	2.0	1.0	1.0	2.0	11.0	2.0	2.0	2.0	1.0	1.0	1.0
11.0	11.0	1.0	1.0	1.0	2.0	1.0	2.0	1.0	2.0	1.0	1.0	2.0	1.0	11.0	2.0	2.0	1.0	1.0	1.0
11.0	2.0	1.0	1.0	1.0	2.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	2.0	11.0	2.0	1.0	1.0	1.0
2.0	11.0	1.0	1.0	1.0	2.0	1.0	2.0	1.0	2.0	1.0	1.0	2.0	1.0	2.0	2.0	11.0	1.0	1.0	1.0
2.0	2.0	11.0	1.0	1.0	2.0	2.0	2.0	2.0	2.0	2.0	1.0	2.0	2.0	2.0	2.0	2.0	11.0	1.0	1.0
2.0	2.0	1.0	11.0	1.0	2.0	1.0	2.0	1.0	2.0	1.0	2.0	2.0	1.0	2.0	2.0	2.0	1.0	11.0	1.0
2.0	2.0	1.0	1.0	1.0	2.0	2.0	2.0	2.0	2.0	1.0	1.0	2.0	1.0	2.0	2.0	2.0	1.0	1.0	11.0

As we see , each of the 11 images were eroded using each structuring elements .
If an image was eroded using B1 , it was detected using B1 . So the diagonal values are 11 similar to the matrix given in the paper where the diagonal values are 125 for 125 images

But since we used images of lower dimensions , even unfiltered original images were detected as filtered . So the non diagonal also have non zero values..

There is an interesting thing to see in the above matrix . When the images were eroded using B15 , we find that all eroded returned positive detection for B1 and B2 as the value of (B15,B1) =11 and (B15,B2) =11

This is because , B15 can be made by dilating B1 and B2

On images given in the dataset of paper

The images which are used in the paper are of higher dimensions . Hence there is no false detection i.e unfiltered original images are not detected as filtered

Since we are implementing from scratch with the hope of getting similar results as in the paper , we could not check for all the images and for all the structuring elements as our implementation is taking some time (We hope to optimize it in future).

Hence we checked our detector on individual image one by one .

We tried to check for those structuring elements which are giving interesting results .

So we checked our implementation on 125 original image using B1 ,B2 and B15 to see if our implementation is mimicking the original implementation

So we got the following result for 125 images and B1 ,B2 and B15

	B1	B2	B15
B1	125	0	0
B2	0	125	
B15	125	125	125

So let us understand the matrix.
So when 125 images are eroded using B1 , the erosion detector detects all 125 images to be eroded using B1

Similarly , we argue for B2

But when it comes to B15 , even if the images are eroded using B15 , our detector detects all images to be eroded using B1 and also using B2

This is because B15 can also be generated by dilating B1 and B2

Robustness of the implementation

We check the robustness of the implementation by tampering with the eroded/dilated image by rotating the eroded/dilated image by 90 degrees and passing it in the detector

We find that the detector is able to detect no matter what with 100 % accuracy even if the eroded image is rotated by 90 degree.
However , the structuring element returned is also roated by the same amount and in same direction

We compared our algorithm with the inbuilt implementation using cv2 , we find that cv2 implementation is not able to detect erosion or detection when the filtering is done is using asymmetrical kernel .
Hence our implementation is performing better than the cv2 implementation , but the implementation is slower .

Future improvements

We hope to make our implementation much faster using parallel processing(using CUDA).

We also hope to implement the splicing detector as we could not implement it due to lack of time .

The paper addresses the detection of morphological filtering in the binarized image .
We find it quite interesting to study about the morphological filtering in graycale and other higher channel images

Raunak kumar perform literature survey on project.
Aniket Mohathy worked on erosion detector and dilation detector.
Md Arsad worked on erosion and dilation in the coding part.
Satish Kumar Orion worked on erosion detector and dilation detector in the coding part.

Video - link -
https://drive.google.com/drive/folders/1ZQTCTOI2XznLj3wyivTV5WybcmhFjUVd?usp=share_link