N° d'ordre :……
N° de série :……

People's Democratic Republic of Algeria
Ministry of Higher Education and Research Scientist

UNIVERSITY ECHAHID HAMMA LAKHDAR EL OUED

FACULTY OF EXACT SCIENCES

Computer Science department

END-OF-STADY DISSERTATION Mémoire de fin d'étude

Presented for the Diploma of

ACADEMIC LICENSE

Field: **Mathematics and Computer Science**

Domain: **Computing**

Specialty: **Computer Systems**

Theme

Artificial Text Détection in Image and Video Frame

Presented by:

- **Cherif Ahmed Cherif**
- **Sebata Younes**
- **Mennai Abdelkrim**

Under the supervision of:

GUIA Sana Sahar

Sustained on …-09-2020

Board of Examiners:

Chairman/ President: University of El Oued
Supervisor: University of El Oued
Examiner: University of El Oued

Academic Year: 2019/2020

Abstract

The Text presented in videos and images contains important information for content analysis, indexing, and retrieval of digital videos libraries. However, it can be very challenging since the text is often embedded in a complex background.

This report presents a method to localize artificial text in images and videos using image-processing techniques, namely mathematical morphology operations.

This method is robust and has good performance on video text detection, which was shown on publicly available test data sets.

KEYWORDS:

Image processing, video sequence text detection, mathematical morphology

ملخـــــــــص

يحتوي النص المعروض في مقاطع الفيديو والصور على معلومات مهمة لتحليل المحتوى وفهرسة واسترجاع مكتبات الفيديو الرقمية. ومع ذلك ، قد يكون الأمر صعبًا للغاية نظرًا لأن النص غالبًا ما يكون مضمّنًا في خلفية معقدة.

يقدم هذا التقرير طريقة لاكتشاف النص الاصطناعي في الصور ومقاطع الفيديو باستخدام تقنيات معالجة الصور .

هذه الطريقة قوية ولها أداء جيد في اكتشاف نص الفيديو ، والذي تم عرضه في مجموعات بيانات الاختبار.

مفاتيح:
معالجة الصور ، كشف نص تسلسل الفيديو ، علم التشكل الرياضي

Acknowledgement

Before presenting this work, we would like to begin by thanking Almighty God, for allowing us to reach this level of studies, as well as blessing us with patience and courage.

We want to thank our dear parents, who supported us emotionally and financially to help secure our futures.

I would like to thank everyone who has helped us to complete this work and overcome all obstacles especially our professor "GUIA Sana Sahar" who has always given us advice and good guidance. We also extend our gratitude to the people who have given us the honour to participate in the juries of this thesis.

Thank you to all those who have contributed in the accomplishment of this job.

Table of Contents

General Introduction

Introduction

With the rapid development of the internet, communication technology, and digital devices, images and videos are now popular media in our daily lives.

Texts, which are often embedded in images and videos, contain lots of semantic information useful for video comprehension, especially text within an image is of particular interest as:

- It is very useful for describing the contents of an image
- It can be easily extracted compared to other semantic contents
- It enables applications such as keyword-based image search, automatic video logging, and text based image indexing.

Text detection, as prerequisite of the subsequent processes, plays a critical role in the whole procedure of textual information extraction and understanding.

In general, text displayed in a video can be classified into scene text and artificial text. Compared with scene text, the artificial text can provide concise and direct description of video content, which is important for understanding. For instance, caption text in news videos often describe the speaker's name or event information; subtitles in sport videos often highlight the information about scores or players.

In this report, we mainly focus on artificial text detection issues for video images.

Structure of the report

This report consists of three chapters.

Chapter One is devoted to the description of basic concepts and techniques that are used in the text detection algorithm.

Chapter Two presents the detection algorithm to localize artificial text in images and videos based on morphological operation in image processing.

Chapter Three describes the development environment first, then details the implementation of the text detection algorithm, finally shows examples of text detection results on TV news test set dataset.

Chapter One

Digital image processing

1.1 Introduction

Digital image processing have been undergoing a vigorous growth as a subject of interdisciplinary study and research, we briefly present a subset of some basic concepts and image processing techniques that are most frequently used in artificial text detection.

1.2 Basic concepts

At its most basic level, a digital image is simply a discrete two-dimensional array of values, much like a matrix, each element of the array is known as a pixel which is short for ‘’pixel element ‘’. These pixels contain information corresponding to the brightness or colour of a particular location in the image accessed by a pair of coordinates (x, y).

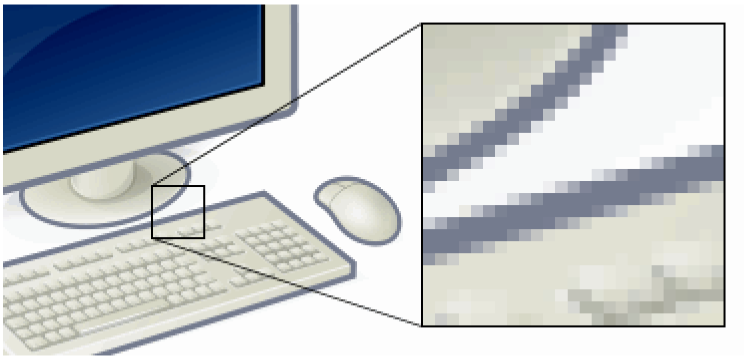


Figure 1.1: an image with a portion greatly enlarged, in which the individual pixels are rendered as small squares and can easily be seen.

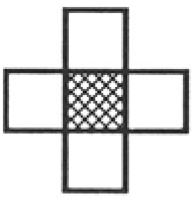
In some cases, the information associated with each pixel may be a single value, representing the grayscale brightness or intensity of that point in a scene [2].

In the RGB colour image the pixel values are triples containing the amount of light captured in the three colour channels: red, green, and blue.

A digital image has several properties, in the following, we introduce some useful properties:

1.2.1 Pixel adjacency:

The four horizontal and vertical neighbours of a pixel p are called 4-neighbours of p [3].



8-neighbors of a pixel p are its vertical horizontal and 4diagonal neighbours

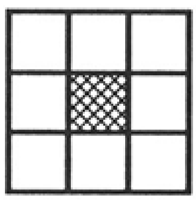


Figure 1.2 Pixel neighbourhoods.

- 4-adjacency: Two pixels p and q are 4-adjacent if they are 4-neighbors and their grey levels satisfy some specified criterion of similarity
- 8-adjacency: Two pixels p and q are 8-adjacent if they are 8-neighbors and their grey levels satisfy some specified criterion of similarity.

1.2.2 Paths

A path from pixel P to pixel Q is a sequence of points A_1, A_2 , … , A_n where A_1 = P, A_n = Q, and $A_{i+1}$is a adjacent of A_i, i = 1, ... ,n – 1.

We can define 4- and 8-path based on type of adjacency used

1.2.2 Connectivity

Two pixels p, q are [4]:

- 4-connected, if there is a 4-path from pixel p to pixel q
- 8-connected, if there is a 8-path from pixel p to pixel q

A Region is a set of pixels in which there is a path between any pair of its pixels, all of whose pixels also belong to the set.

The boundary of the region R is the set of pixels in the region that have one or more neighbours that are not in R.

1.3 image processing techniques

Digital image processing techniques are used today in a variety of problems

1.3.1 Image filtering

Image filtering is one of the most important and widely used operation in image processing and makes possible several useful tasks. Filtering image data is a standard process used in almost every image processing system. The choice of filter depends on the filter behaviour and type of data.

In image processing, a Gaussian blur (also known as Gaussians mouthing) is the result of blurring an image by a Gaussian function (named after mathematician and scientist Carl Friedrich Gauss). That is used typically to reduce image noise and reduce detail.
Gaussian filter function is defined as:

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

1.3.2 Binarization

Binarization is the process of converting a pixel image to a binary image black and white (0 and 1).

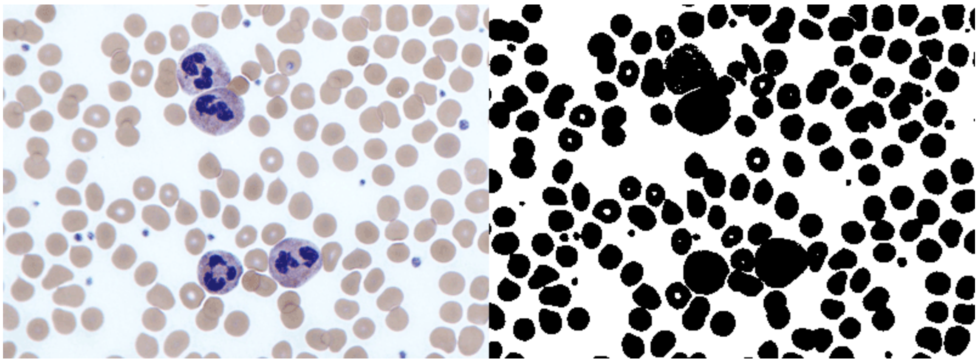


Figure 1.3 Binarization provides sharper and clearer contours of various objects present in the image.

At first, the image is converted into grayscale, then in the process of image binarization a threshold value is chosen, and all pixels with values above this threshold are classified as white, and all other pixels as black.

1.3.3 Mathematical morphology

A morphological operation is a set of image processing algorithms that provides tools for the representation and description of image regions. It provides techniques for pre- and post-processing of an image (morphological thinning, pruning, filtering). Its principles are based on set theory.

In this chapter we consider binary images. Image pixels can assume the value 1 or 0

- **Structuring element (kernel) :**

A morphological operation acts on image pixels using pre-defined kernels. These kernels, known as structuring elements, define patterns that are used to process images. Figure 4shows some examples of common structuring elements.

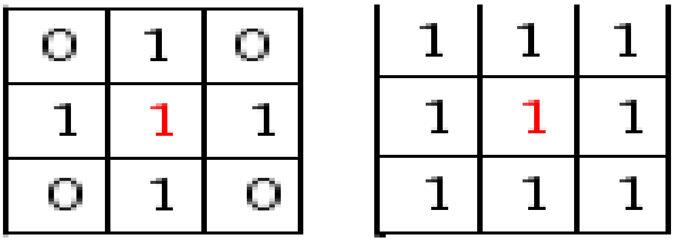


Figure 1.4 Structuring element

Structuring elements [5] are small sets or sub images used to examine the image under study; it is expressed with respect to a local origin. The origin of the **structuring elements** visits every pixel in an image. It is then decided if the visited pixel will belong to the resulting set or not following a criterion.

- **Fitting, Hitting and Missing**

I is an Image and S is The Structuring Element [6].

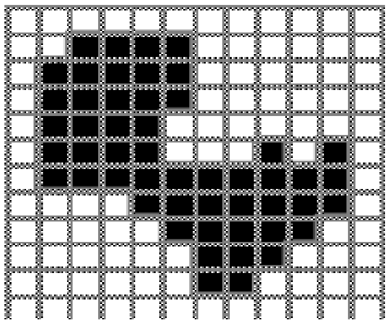
* S hits I at x if $\{y: y = x + s, s \in S\} \subset I$

* S hits I at x if $\{y: y = x + s, s \in S\} \cap I \neq \emptyset$

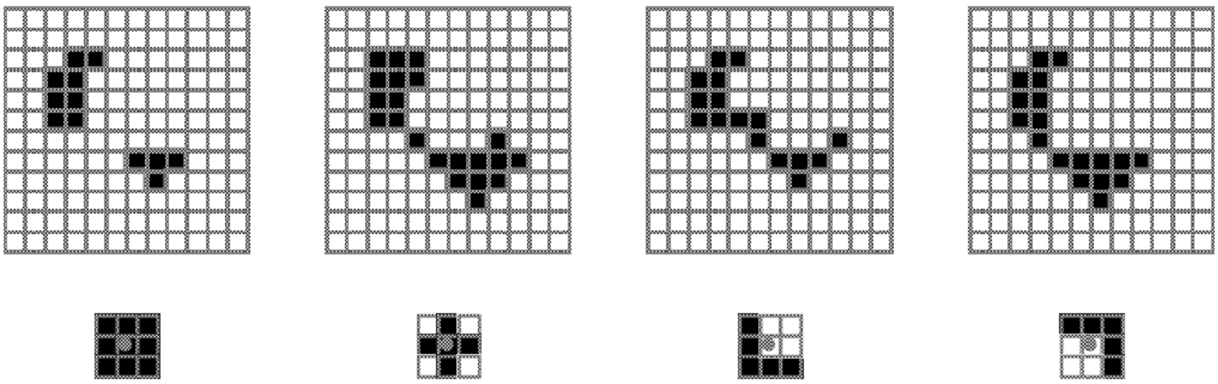
* S misses I at x if $\{y: y = x + s, s \in S\} \cap I = \emptyset$

- **Somme basic opérations :**
  - **Erosion :**

Erosion has the effect of shrinking objects and deteriorating the boundary pixels of foreground elements. Precisely a pre-defined kernel is passed over the image as a 2D convolution. The input pixel will be 1 if all the pixels under the kernel is 1. Otherwise, it gets a 0.



(a)



(b)

Figure1.5: (a)Original Image (b)image after Erosion by different structuring elements

The image E:

$E = I \ominus S$

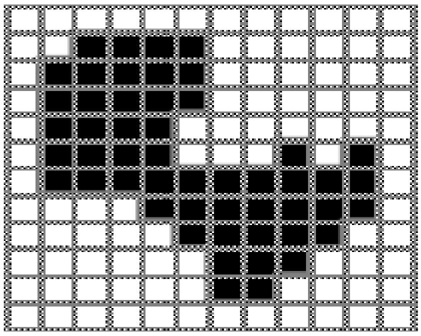
Is the Erosion of image I by structuring element S [6].

$$E(x) = \begin{cases} 1, \mathit{if\ S\ its\ I\ at\ x} \\ 0 \quad , \mathit{otherwise} \end{cases}$$

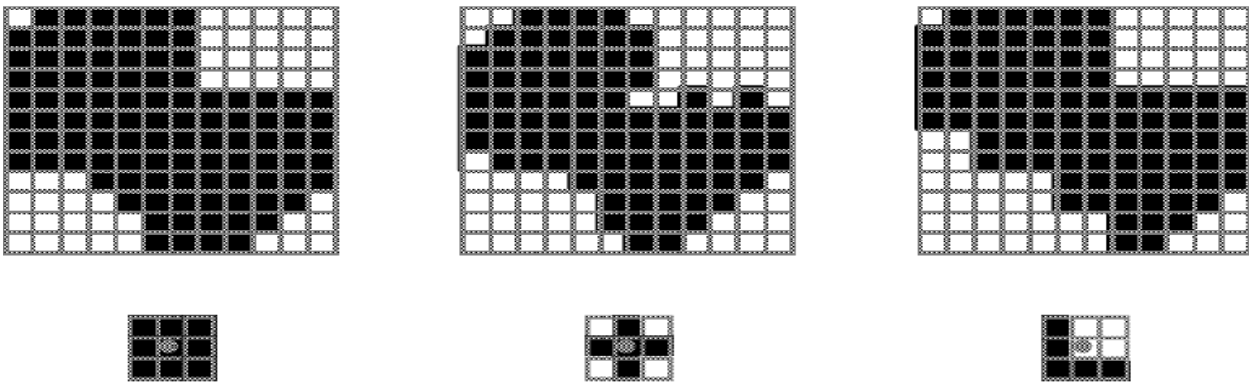
$E = \{ y : y = x + s \in I \text{ for every } s \in S \}$

↘ Dilation :

Dilation acts in the opposite way to erosion, it enlarges objects and expands the boundaries of foreground elements, and therefore, given an input pixel, the output is one if at least one of the input pixels (under the kernel) is one. Otherwise, the output is zero.



(a)



(b)

Figure 1.6: (a) Original Image (b) image after Dilation by different structuring elements

The image D:

$D = I \oplus S$

is the Dilation of image I by structuring element S [6].

$$D(x) = \begin{cases} 1, \text{if } S \text{ hits } I \text{ at} \\ 0, \text{otherwise} \end{cases}$$

$D = \{y: y = x + s \in I \text{ for any } s \in S\}$

↘ Opening :

Erosion followed by dilation creates an important morphological transformation called opening.Notice that the same structuring element is used as well for the erosion as for the dilation. Opening is widely used as a noise removal transformation. It is used to smooth object boundaries, and to remove thin connections between objects.

- **Closing :**

The morphological closing is defined as a dilation followed by an erosion. Closing is used to get rid of small gaps within the foreground objects. In addition, it has the effect of merging narrow separation between objects.

1.4 Conclusion

Extraction of text from images and videos attracts a large number of researchers; there exist already a high number of contributions. In the next chapter, we present an algorithm to localize artificial text in images and videos using image-processing techniques including morphological processing.

Chapter Two

Text Detection Methodology

Chapter 2 : Text Detection Methodology

2.1 Introduction

A high number of contribution exist in the research domain of artificial text detection in image and video [7]. In this chapter, we present the text extraction algorithm beginning from several processing steps until the last step of regrouping text into rectangular bounding box.

2.2 Description of the Text detection problem

Computer Vision is the process of getting pragmatic data from single or collection of images by scrutinizing the image and extracting the required information, which can further be used for various purposes [8].

2.2.1 Definition

Text processing from images or video frame involves two major functionalities namely, Text detection and Text recognition. Text detection is the process of extracting the text objects from the image and Text recognition is the process of identifying the character forming meaningful words (Fig 2.1).

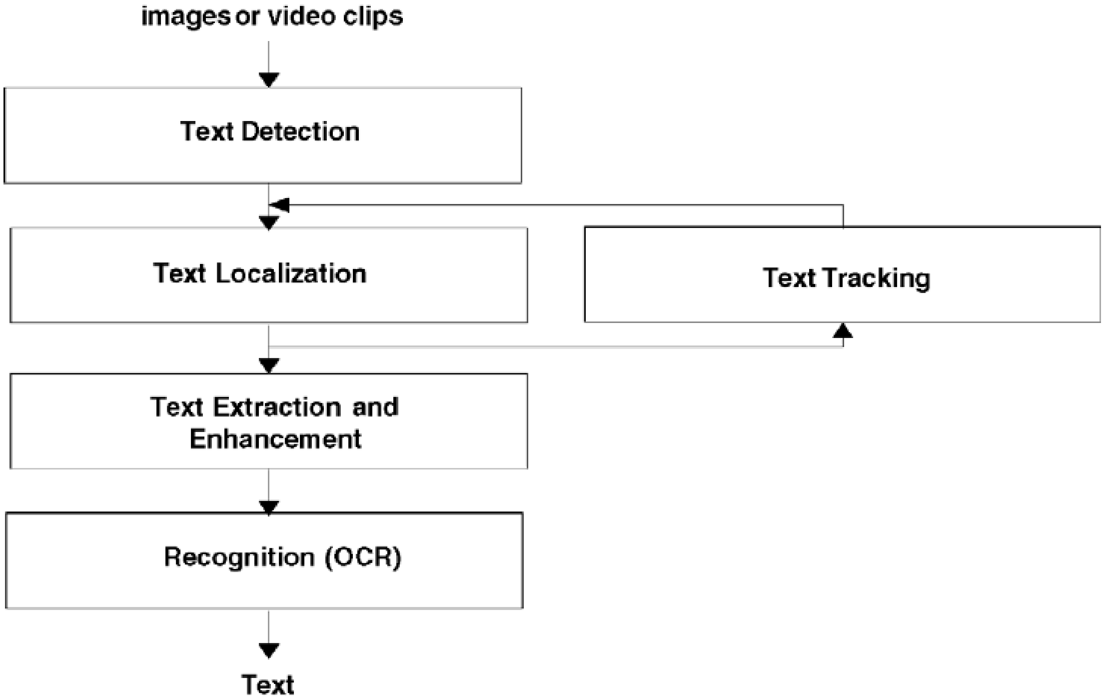


Figure 2.1: Architecture of text information extraction

Generally speaking, text in video images can be classified into scene text which exists naturally in the image(Fig2), or artificial text which is artificially overlaid on the image (Fig3) [9].

Figure 2.2 : Scene text images

Figure 2.3: Artificial text in video frame

In this report, we will focus on identifying the horizontal artificial texts in the digital images extracted from videos (Fig 2.4), These horizontal artificial texts have certain features such as (designed to be read easily, highly contrasted, not occulted, long and horizontal) [10]

.

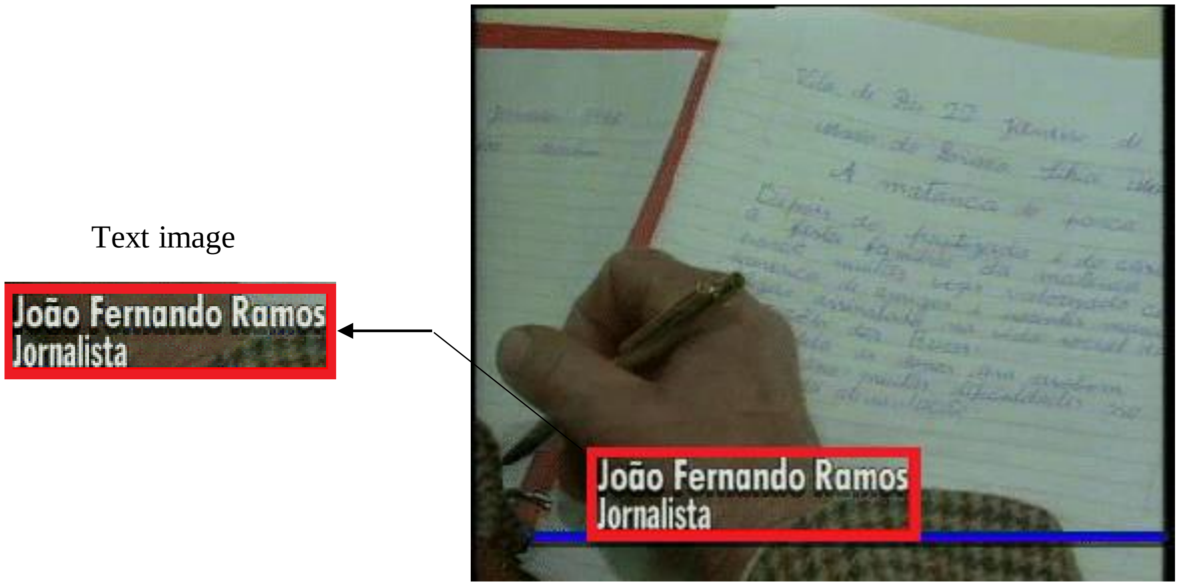


Figure 2.4: Text detection in image or video frame

2.2.2 Challenges

Although the algorithm is designed with logical steps to detect horizontal artificial texts, sometimes there are significant obstacles that makes video and image text detection a extremely challenging tasks [11], in which an exact modelling of all types of text becomes almost impossible, and the main challenges are:

- Variations in languages,

-complex backgrounds.

- Low quality of the original image i.e. (low resolution, strong noises, low contrast, etc…) [12].

-Text appears in videos in a wide range of writings, fonts , styles, colours, sizes, etc.,

All of these obstacles lead to more complexity for detecting text.

2.2.3 Goals

Today more and more audio and visual information is captured, stored, delivered and managed in digital forms. The wide usage of digital media files provokes many new challenges in mobile information acquisition and large multimedia database management. Among the most prominent are [13]:

1. Automatic broadcast annotation: creates a structured, searchable view of archives of the broadcast content;
2. Digital media asset management: archives digital media files for efficient media management;
3. Video editing and cataloguing: catalogues video databases on basis of content relevance;
4. Library digitizing: digitizes cover of journals, magazines and various videos using advanced image and video OCR;
5. Mobile visual sign translation: extracts and translates visual signs or foreign languages for tourist usage, for example, a handhold translator that recognizes and translates Asia signs into English or French.

2.3 Methodology of Text Detection

The horizontal artificial texts detection task is to draw a bounding box or rectangle box for each line of horizontal artificial text in the digital images, There are many ways to detect texts in the image in different fields such as (deep learning), but we have taken a certain method, depending on the morphological operations.

2.3.1 Text detection algorithm

The detection of horizontal artificial texts in a digital image consists of six stages, They are [10] :

- Compute the image of vertical gradients,
- Binarize the gradient image,
- Detect regions highly dense in vertical edges,
- Make the region more compact,

- Remove spurious regions,
- Localize detected regions inside bounding boxes).

An overview of the algorithm is shown in Figure 2.5 and more detailed descriptions of the algorithm are provided in the next subsection.

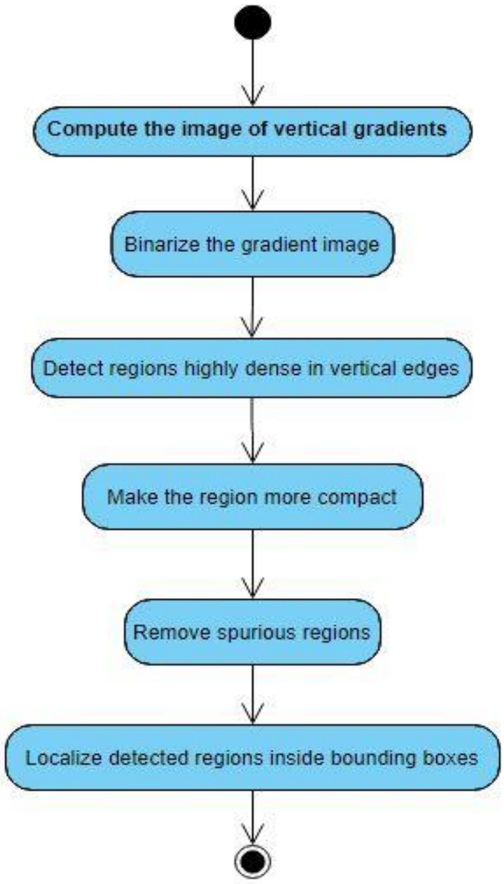


Figure 2.5: Activity diagram of the text detection algorithm.

2.2.2 Description of the text detection algorithm

We will explain the six stages of the algorithm.

Note:

These operations are sequential; each operation is performed on the output image of the previous operation.

- **Compute the image of vertical gradients**

a) Convert the image to a grayscale image since colour is not an essential feature of the text.

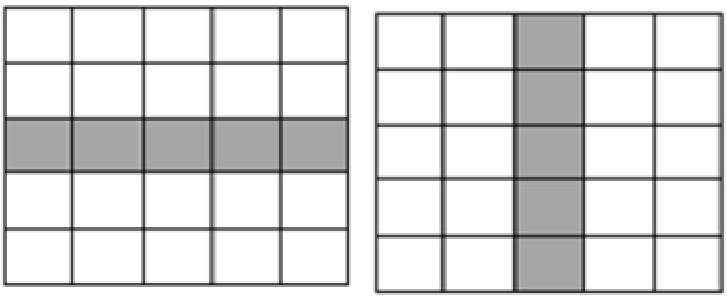
b) Apply a Gaussian filter to reduce noise in the image. This is achieved by convolving the 2D Gaussian distribution function with the image.

c) Calculate the vertical gradient g(x,y), which is the difference between horizontal dilation and horizontal erosion of the image I(x,y), to detect the horizontal edges and output the highest values for the horizontal structures on the image.

Horizontal dilation is a special case of dilation with a linear and horizontal structuring element $_L$.

The same as the horizontal erosion which is a special case of erosion with a linear and horizontal structuring element $_L$.

$$g(x, y) = \text{dilation}_L(I(x, y)) - \text{erosion}_L(I(x, y))$$



(a) (b)

Figure 2.6: (a) 5 x 5 horizontal structuring element, (b) 5 x 5 vertical structuring element

- **Binarize the gradient image**

The appropriate results of binary image analysis depend on the correct binarization. The key point is the choice of a proper threshold, in order to maximize the information gain. The threshold is calculated based on the entropy of the histogram; the entropy is a criterion for finding the “optimal” threshold to distinguish between the background and foreground of an image [14].

In this step the binarization is used to detect the extreme values of the gradient, we use the entropy value to calculate the threshold, and the threshold value is determined as the grey level value which increases the total amount of information provided by the background and foreground (Objects) separately [15].

- **Detect regions highly dense in vertical edges**

a) A vertical closing is a vertical dilation applied to the image, followed by a vertical erosion.

We use a vertical closing of half-size 2pixels (5x5 structuring element)to reconnect the vertical lines with a maximum of five pixels between them, since the vertical gradient produces discontinuous lines for lines not strictly straight.

b) Then, a horizontal closing is used to merge close edges into one unique region; the closing uses a horizontal structuring element of half-size 4, which merges edges that are at a maximum distance of nine pixels.

- **Make the region more compact**

A vertical closing is used to close the small holes inside the foreground objects, or small black points on the object, the closing uses a vertical structuring element of half-size 2 which closes holes that are at a maximum vertical size of 5 pixels.

- **Remove spurious regions**

Concept:

Rectangularity is the property of being shaped like a rectangle

Specifically, rectangularity is the ratio of a region area to the area of the minimum bounding rectangle; the maximum value is 1.0 for a square [16].

a) The horizontal opening uses a horizontal structuring element of half-size 2 pixels, which removes the regions that are smaller than five pixels.

b) And after that we use “area-opening” to remove all regions of surface area less than 100 pixels.

c) After calculating the convex hull for each region, the convex hull regions with a rectangularity factor less than 70% are eliminated.

d) Then we reapply the horizontal opening again using a horizontal structuring element of half-size 10 pixels to remove regions of width less than 21 pixels, since artificial text are quite wide regions.

- **Localize detected regions inside bounding boxes**

a) A bounding-box locates the texts because the text is a horizontal rectangular area, and then creates colour-filled bounding boxes around the texts.

b) We apply dilation to expand the dimensions of the bounding boxes horizontally and vertically to ensure that the bounding boxes contain all character pixels.

c) Finally, we use Superimposition to draw the red bounding boxes around the text in the original image.

The algorithm uses the following class diagrams:

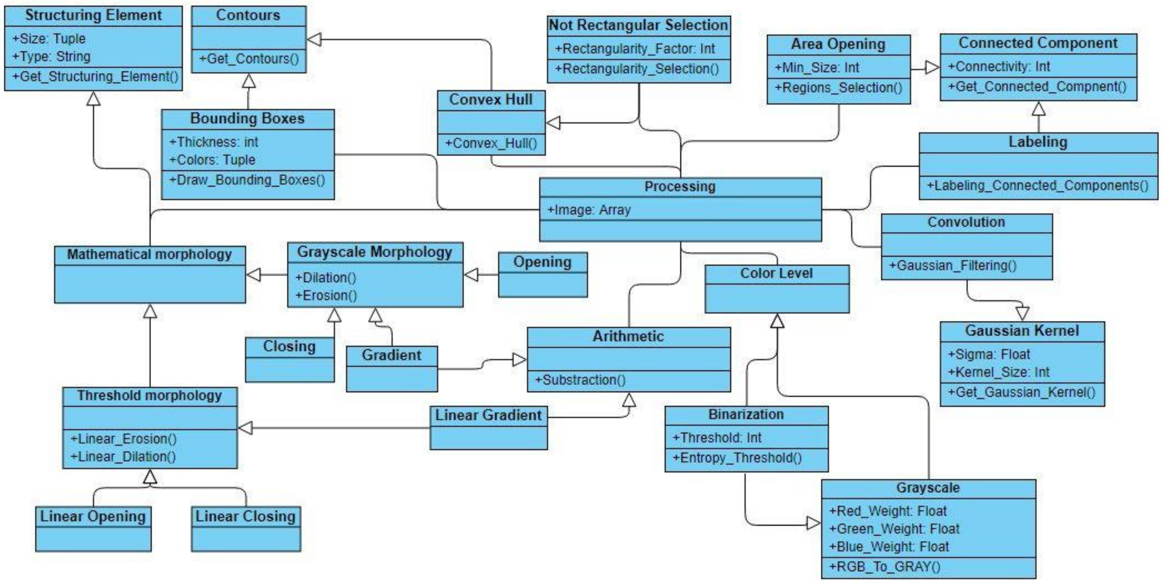


Figure 2.7 : Class diagram

The UML Class diagram is a graphical notation used to construct and visualize object oriented systems. A class diagram in the Unified-Modelling- Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

1.4 Conclusion

In this chapter, we presented a scheme for detecting text in images and video frames. The implementation of the algorithm explained in this chapter and the presentation of the experimental results are the subject of the next chapter.

Chapter three

implementation

Chapter 3 : implementation

3.1 Introduction

Finally, in this third chapter we will describe the work environments, then present the results obtained through the application of several different image-processing operations.

3.2 workspace environment

Our computer must meet the hardware and software environments that are listed below when attempting to run Project of text detection.

3.2.1 Hardware Environment

- Summary:

Operating system:	Windows 7
CPU:	Intel(R) Core(TM) i3 CPU M 380 @ 2.53GHz
Memory:	2.00 GB RAM
Hard disk:	250 GB

3.2.2 Software Environment

Python:

is a popular programming language, it was created by Guido van Rossum, and released in 1991, it is used for (web development, software development, mathematics, system scripting) [17], A key benefit of using Python is its active developer community and the large amount of available software packages available from "pypi.org" [18].

Python works on different platforms (Windows, Mac, Linux, etc.),it has a simple syntax similar to the English language, which allows developers to write programs with fewer lines than most programming languages.

Python runs on an interpreter system, meaning that code can be executed as soon as it is written, This means that prototyping can be very quick, Python can be treated in (a procedural way, an object-orientated way or a functional way) [17].

OpenCV is an open-source computer vision library that can be used to process videos and images.

3.3Description of the application

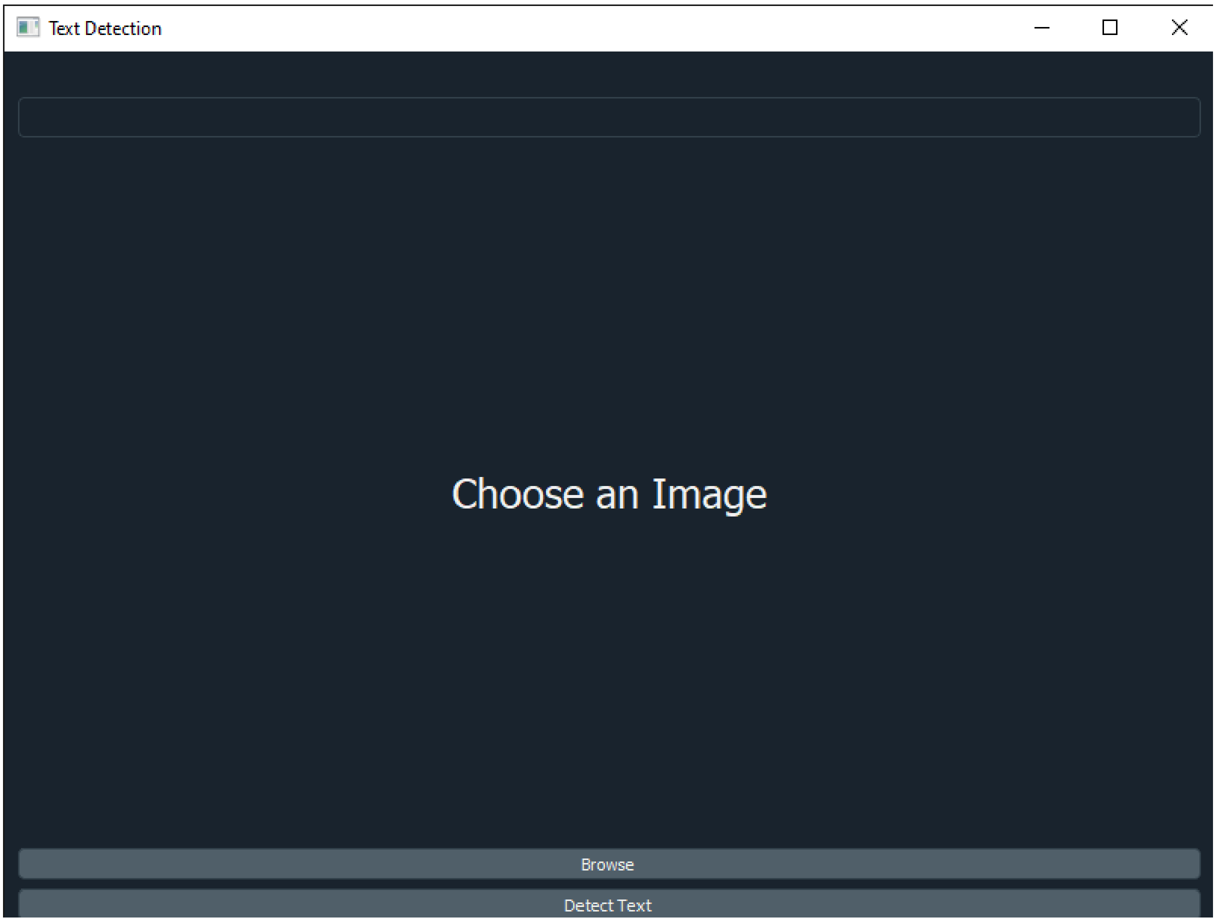


Figure 3.1: The main interface of the program

This application is a program written in the Python programming language designed to implement several morphological operations with certain settings And some other image processing operations such as (grayscale, binarization, subtraction, Gaussian filtering) on a specific sequence to allow you to detect horizontal artificial texts.

3.4 Implementation of the algorithm

In this part, we will see the details of applying the stages of the horizontal artificial texts detection algorithm, in each stage, its sub-steps are explained with a picture of its result.

We will apply the algorithm on this original image:

Figure 3.2: Original Image [19]

First, the image is converted to grayscal after that, we apply a Gaussian filter to the grayscale image to reduce the noise

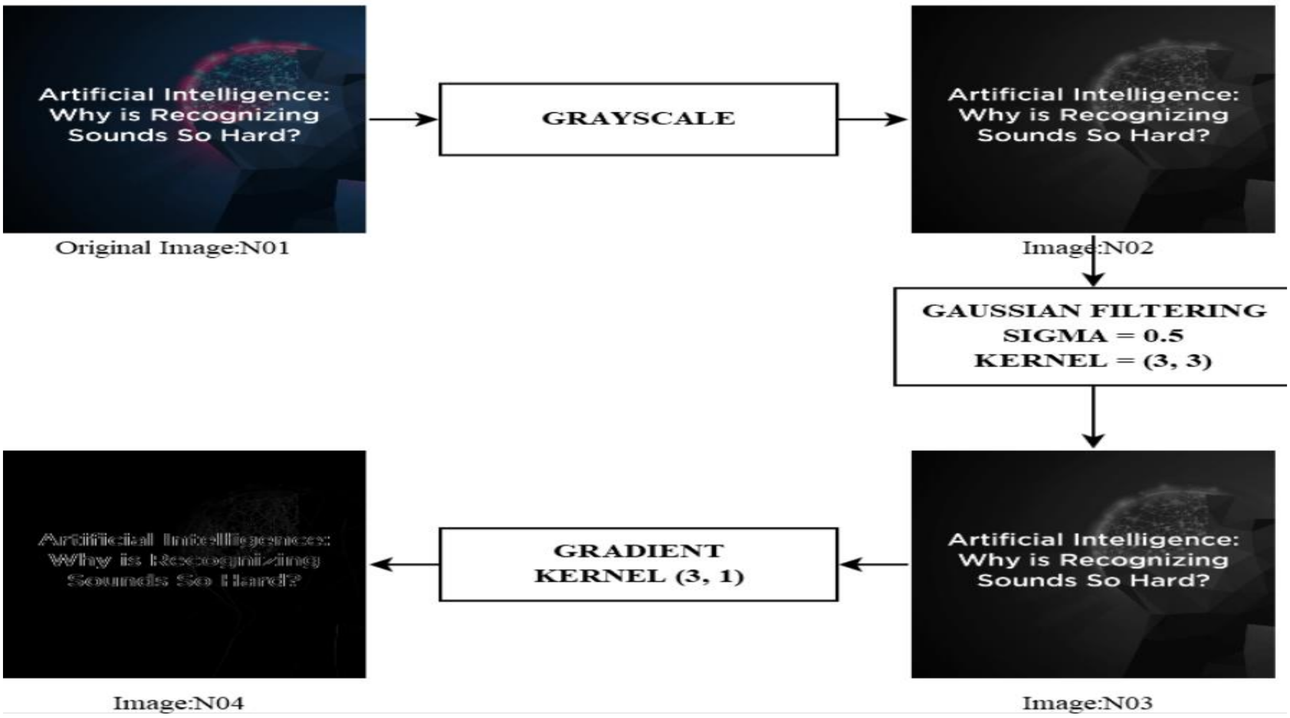


Figure 3.3: Compute the image of Vertical gradients

Then we continue with the rest of the other stages of processing.

3.4.1 Calculate the image of vertical gradients :

- Gradient (Gaussian Image)

  - Dilation (Structuring Element, Gaussian Image)

* Result: Temporary Image N°01

  - Erosion (Structuring Element, Temporary Image N°01)

    * Result: Temporary image N°02

- Difference (Temporary Image N°01, Temporary Image N°02)

  * Result: Temporary Image N°03

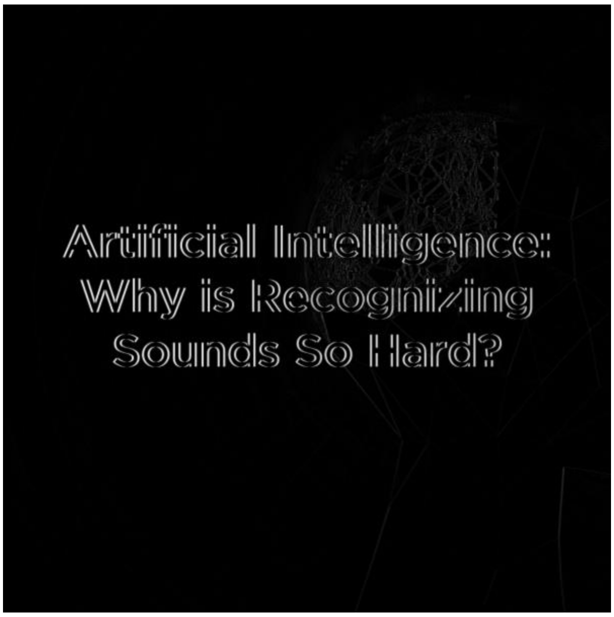


Figure 3.4: Temporary Image N°03 Result of the 1st stage

3.4.2 Binarize the image of the gradients:

- Binarization(Temporary Image N°03)

  - Entropy_Threshold(Temporary Image N°03)

    * Result: Temporary Image N°04

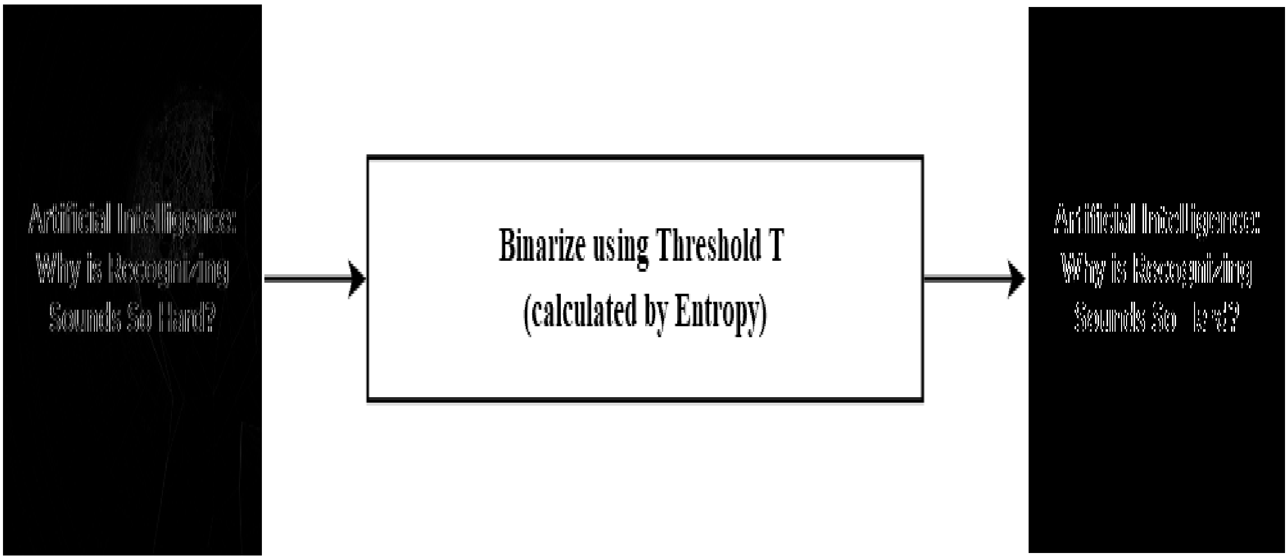


Figure 3.5: Binarize the gradients image

3.4.3 Detect dense regions in vertical contours:

- Closing (Temporary Image N°04)

- Dilation(Vertical Structuring Element, Temporary Image N°04)
  * Result: Temporary Image N°05
- Erosion(Vertical Structuring Element, Temporary Image N°05)
  * Result: Temporary Image N°06

- Closing (Temporary Image N°06)

- Dilation(Horizontal Structuring Element, Temporary Image N°06)
  * Result: Temporary Image N°07
- Erosion(Horizontal Structuring Element, Temporary Image N°07)
  * Result: Temporary Image N°08

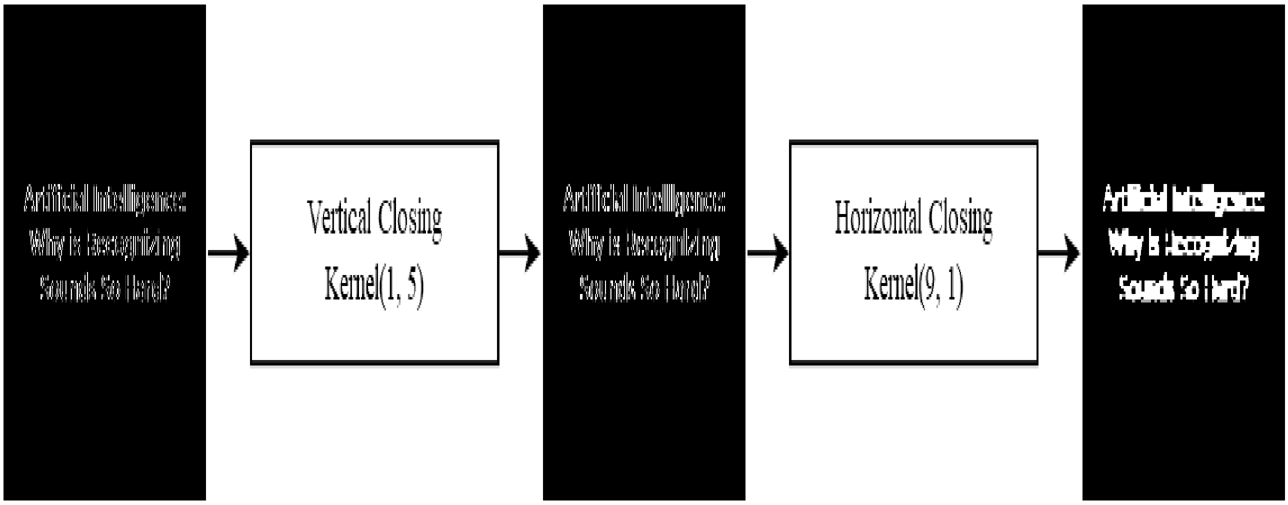


Figure 3.6: Detect regions highly desne in Vertical Edges

3.4.4 Make regions more compact:

- Closing (Temporary Image N°08)

- Dilation(Vertical Structuring Element, Temporary Image N°08)

  * Result: Temporary Image N°09

- Erosion(Vertical Structuring Element, Temporary Image N°09)

  * Result: Temporary Image N°10

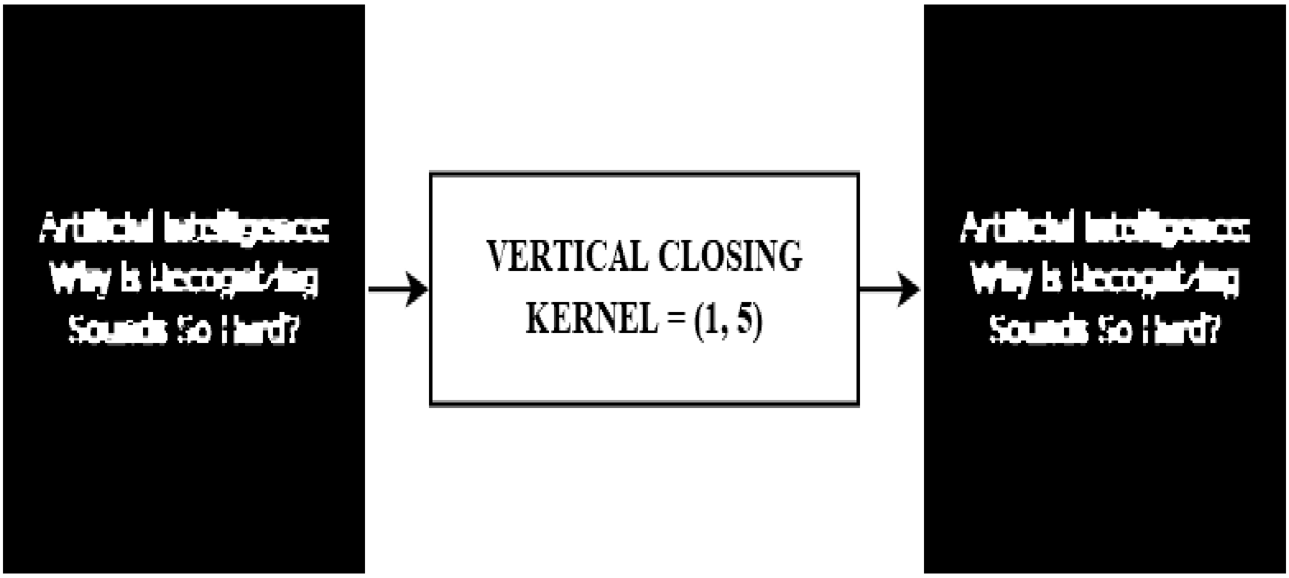


Figure 3.7: Make the regions more compacte

3.4.5 Remove spurious regions

- **Removing Too Small Regions**

  - Opening (Temporary Image N°10)

  - Erosion(Horizontal Structuring Element, Temporary Image N°10)
    * Result: Temporary Image N°11
  - Dilation(Horizontal Structuring Element, Temporary Image N°11)
    * Result: Temporary Image N°12

  - Area-Opening (Temporary image N°12)

  - Min_Size_Selection(Min Size, Connectivity, Temporary Image N°12)
    * Result: Temporary Image N°13

- **Removing Non Rectangular Regions**
  - Labelling (Connectivity, Temporary Image N°13)
    * Result: Temporary Image N°14

  - Convex-Hull (Temporary Image N°14)
    * Result: Temporary Image N°15
  - Rectangularity-Selection (Rectangularity Factor, Temporary Image N°15)

* Result: Temporary Image N°16

- **Removing Too Thin Convex Hulls**

  - Opening (Temporary Image N°16)

  - Erosion(Horizontal Structuring Element, Temporary Image N°16)

    * Result: Temporary Image N°17

  - Dilation(Horizontal Structuring Element, Temporary Image N°17)

    * Result: Temporary Image N°18

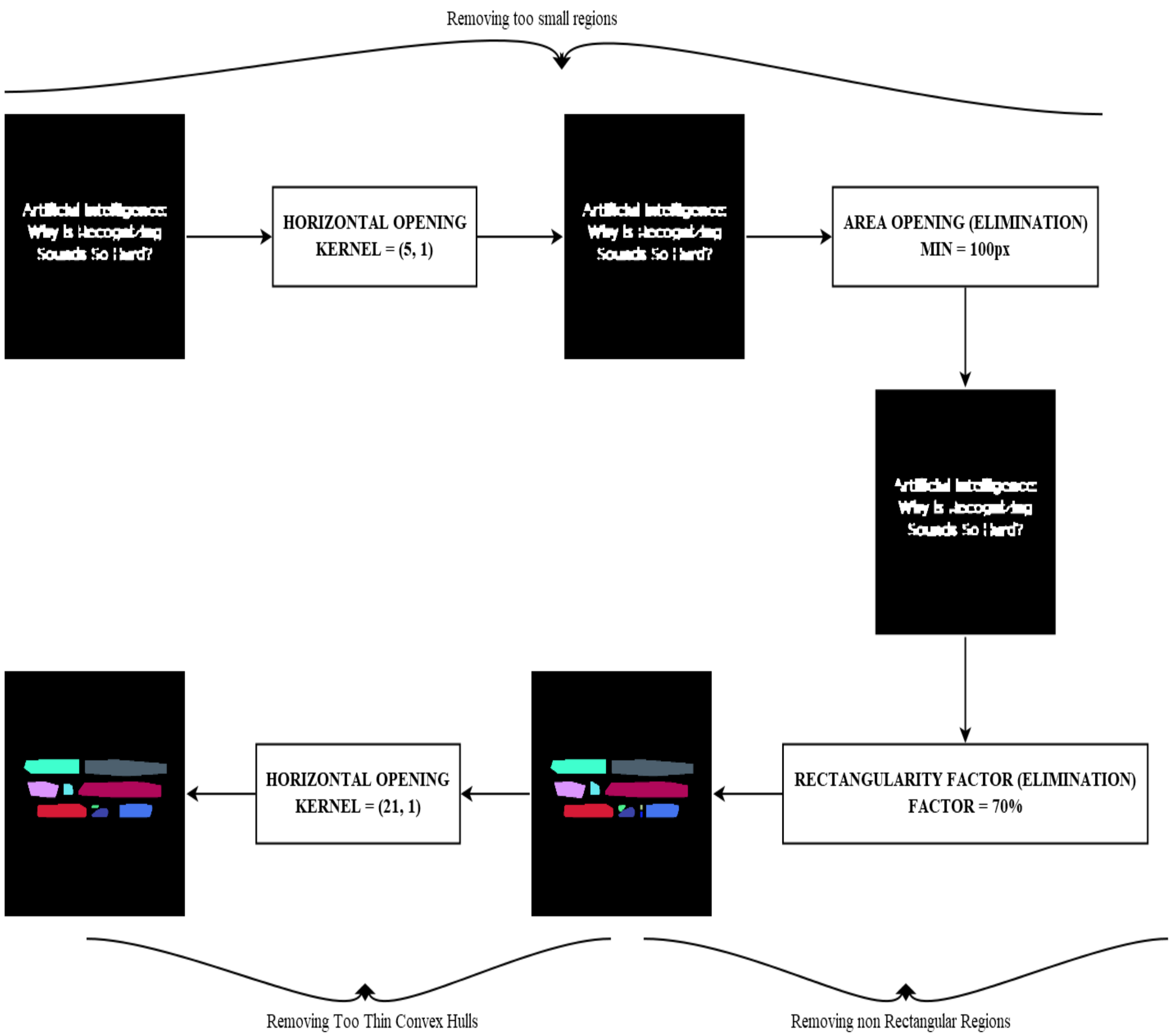


Figure 3.8: Remove spurious regions

3.4.6 Locate the regions found in bounding boxes

- **Localizing the Text Regions**

  - Bounding-Box (Thickness True, Temporary Image N°18)

    * Result: Temporary Image N°19

  - Dilation (Vertical Structuring Element, Temporary Image N°19)

    * Result: Temporary Image N°20

- **Superimposition of the Bounding Boxes to the Original Image**

  - Superimposition (Temporary Image N°20, Original Image)

  - Bounding-Box(Thickness False, Temporary Image N°20, Original Image)

    * Result: Original Processed Image

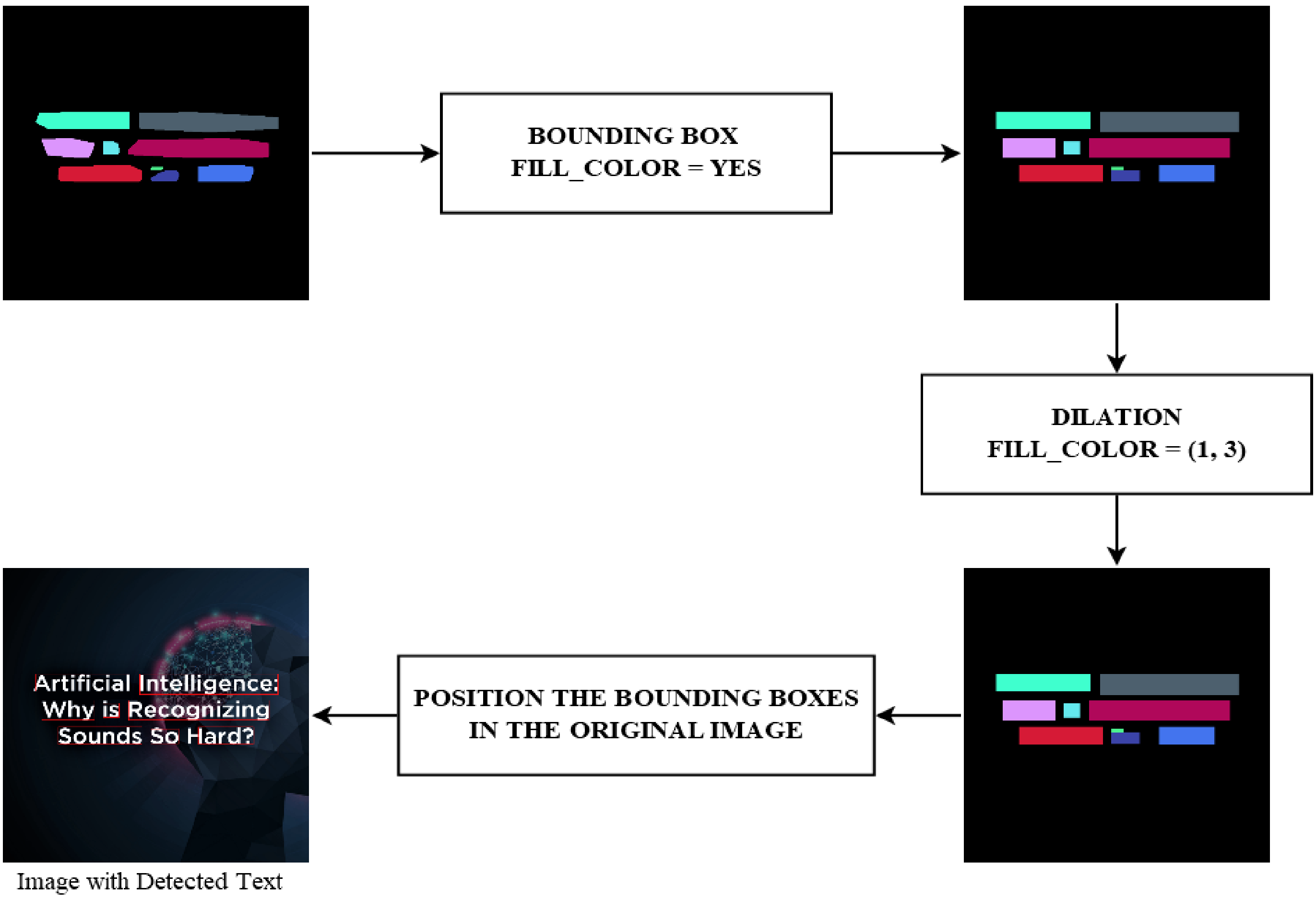


Figure 3.10: Localize detected regions inside bounding boxes

Figure 3.10: The Final Result

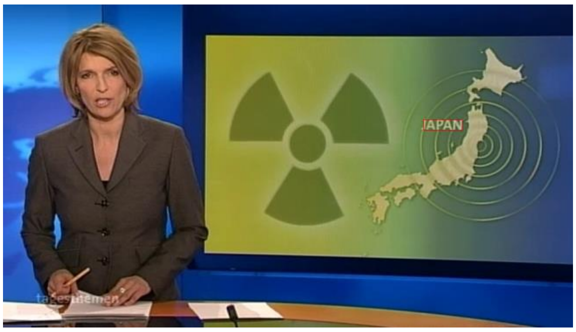
Once all the stages are completed, we get the original image containing bounding boxes around the horizontal artificial texts.

Figure 3.11 show examples of text detection results on publicly available test dataset: TV news test set [20].

(a) (b)

(c)

(d)



(e)

(f)

(g)

Figure 3.11 Examples of text detection results. Detected text lines are bounded with red boxes.

3.5 Source Code :

```
# USAGE
# PY 19.PY -I IMAGE.*
IMPORT CV2
IMPORT NUMPY AS NP
IMPORT RANDOM AS RNG
IMPORT ARGPARSE

AP = ARGPARSE.ARGUMENTPARSER()
AP.ADD_ARGUMENT("-I", "--IMAGE", HELP = "PATH TO THE (OPTIONAL) IMAGE
FILE")
ARGS = VARS(AP.PARSE_ARGS())

DEF KERNEL(ELEMENT,TUPLE):
  ELEMENT_DIC = {'RECT':0 , 'CROSS':1, 'ELLIPSE':2}
  RETURN CV2.GETSTRUCTURINGELEMENT(ELEMENT_DIC[ELEMENT], TUPLE)

DEF SHOW(TITLE,IMAGE):
  CV2.IMSHOW(TITLE,IMAGE)
  CV2.WAITKEY(0)

DEF GAUSSIANKERNEL(SIZE, SIGMA):
  SIZE = INT(SIZE) // 2
  X, Y = NP.MGRID[-SIZE:SIZE+1, -SIZE:SIZE+1]
  NORMAL = 1 / (2.0 * NP.PI * SIGMA**2)
  KERNEL =  NP.EXP(-((X**2 + Y**2) / (2.0*SIGMA**2)))* NORMAL
  RETURN KERNEL

DEF LABCONEBRF(IMAGE):
    CONTOURS, _ = CV2.FINDCONTOURS(IMAGE, CV2.RETR_EXTERNAL,
CV2.CHAIN_APPROX_NONE)
    HULL_LIST = []
    FOR I IN RANGE(LEN(CONTOURS)):
        HULL = CV2.CONVEXHULL(CONTOURS[I])
        HULL_LIST.APPEND(HULL)
    IMG = NP.ZEROS((IMAGE.SHAPE[0], IMAGE.SHAPE[1], 3), DTYPE=NP.UINT8)
    MASK = NP.ZEROS(IMAGE.SHAPE, DTYPE=NP.UINT8)
    FOR I IN RANGE(LEN(CONTOURS)):
        X, Y, W, H = CV2.BOUNDINGRECT(HULL_LIST[I])
        MASK[Y:Y+H, X:X+W] = 0
        CV2.DRAWCONTOURS(MASK, HULL_LIST, I, (255,255,255), THICKNESS=-1)
        R = FLOAT(CV2.COUNTNONZERO(MASK[Y:Y+H, X:X+W]))/(W * H)
        IF R > 0.70:
          COLOR = (RNG.RANDINT(0,255), RNG.RANDINT(0,255),
```

```
RNG.RANDINT(0,255))
            CV2.DRAWCONTOURS(IMG, HULL_LIST, I, COLOR, THICKNESS=-1)
    RETURN IMG

DEF EBMINSIZE(IMAGE,M,C):
    NB_COMPONENTS, OUTPUT, STATS, _ =
CV2.CONNECTEDCOMPONENTSWITHSTATS(IMAGE, CONNECTIVITY=C)
    SIZES = STATS[1:, -1]; NB_COMPONENTS -= 1
    IMG = NP.ZEROS(IMAGE.SHAPE,DTYPE=NP.UINT8)
    FOR I IN RANGE(0, NB_COMPONENTS):
        IF SIZES[I] >= M:
            IMG[OUTPUT == I+1] = 255
    RETURN IMG

DEF BOUNDINGBOX(IMAGE,I):
    GRAYSCALE = CV2.CVTCOLOR(IMAGE, CV2.COLOR_BGR2GRAY)
    _,THRESHOLD = CV2.THRESHOLD(GRAYSCALE,0,255,CV2.THRESH_BINARY)
    CONTOURS, _ = CV2.FINDCONTOURS(THRESHOLD, CV2.RETR_EXTERNAL,
CV2.CHAIN_APPROX_NONE)
    FOR J IN RANGE(LEN(CONTOURS)):
      Y_HAT, X_HAT, = CONTOURS[J][0][0][:2]
      X, Y, W, H = CV2.BOUNDINGRECT(CONTOURS[J])
      IF I ==0:
        COLOR = TUPLE([ INT(X) FOR X IN IMAGE[X_HAT,Y_HAT] ])
        CV2.RECTANGLE(IMAGE, (X, Y), (X+W-1, Y+H-1), COLOR, -1)
      ELSE:
        CV2.RECTANGLE(IMAGE, (X, Y), (X+W-1, Y+H-1), (0,0,255), 1)
    IF I==0:
      RETURN IMAGE
    ELSE:
      RETURN IMAGE

DEF ENTP(X):
    TEMP = NP.MULTIPLY(X, NP.LOG(X))
    TEMP[NP.ISNAN(TEMP)] = 0
    RETURN TEMP

DEF MAXIMUM(IMG):
  H = CV2.CALCHIST([IMG],[0],NONE,[256],[0,256])
  H = H / NP.SUM(H)
  THETA = NP.ZEROS(256)
  HF = NP.ZEROS(256)
  HB = NP.ZEROS(256)
```

```
  FOR T IN RANGE(1,255):
      HF[T] = - NP.SUM( ENTP(H[:T-1] / NP.SUM(H[1:T-1])) )
      HB[T] = - NP.SUM( ENTP(H[T:] / NP.SUM(H[T:])) )
      THETA[T] = HF[T] + HB[T]

  THETA_MAX = NP.ARGMAX(THETA)
  RETURN THETA_MAX

IMAGE = CV2.IMREAD(ARGS['IMAGE'])
# 1. COMPUTE THE IMAGE OF VERTICAL GRADIENTS.
GRAYSCALE = CV2.CVTCOLOR(IMAGE, CV2.COLOR_BGR2GRAY)
CV2.IMWRITE('GRAYSCALE.PNG',GRAYSCALE)
GAUSSIAN_FILTERING = CV2.FILTER2D(GRAYSCALE, -1, GAUSSIANKERNEL(3,0.5) )
CV2.IMWRITE('GAUSSIAN_FILTERING.PNG',GAUSSIAN_FILTERING)
GRADIENT = CV2.MORPHOLOGYEX(GAUSSIAN_FILTERING, CV2.MORPH_GRADIENT,
KERNEL('CROSS', (3,1)) )
CV2.IMWRITE('GRADIENT.PNG',GRADIENT)
# 2. BINARIZE THE GRADIENT IMAGE.
T = MAXIMUM(GRADIENT)
_,THRESHOLD = CV2.THRESHOLD(GRADIENT,T,255,CV2.THRESH_BINARY)
CV2.IMWRITE('THRESHOLD.PNG',THRESHOLD)

# 3. DETECT REGIONS HIGHLY DENSE IN VERTICAL EDGES.
FIRST_V_CLOSING = CV2.MORPHOLOGYEX(THRESHOLD, CV2.MORPH_CLOSE,
KERNEL('CROSS', (1,5)) )
CV2.IMWRITE('FIRST_V_CLOSING.PNG',FIRST_V_CLOSING)
H_CLOSING = CV2.MORPHOLOGYEX(FIRST_V_CLOSING, CV2.MORPH_CLOSE,
KERNEL('CROSS', (9,1)) )
CV2.IMWRITE('H_CLOSING.PNG',H_CLOSING)

# 4. MAKE THE REGION MORE COMPACT.
SECOND_V_CLOSING = CV2.MORPHOLOGYEX(H_CLOSING, CV2.MORPH_CLOSE,
KERNEL('CROSS', (1,5)) )
CV2.IMWRITE('SECOND_V_CLOSING.PNG',SECOND_V_CLOSING)
# 5. REMOVE SPURIOUS REGIONS.
# 5.1 REMOVING TOO SMALL REGIONS.
FIRST_H_OPENING = CV2.MORPHOLOGYEX(SECOND_V_CLOSING, CV2.MORPH_OPEN,
KERNEL('CROSS', (5,1)) )
CV2.IMWRITE('FIRST_H_OPENING.PNG',FIRST_H_OPENING)
ELIMINATE_BY_MIN_SIZE = EBMINSIZE(FIRST_H_OPENING,100,8)
CV2.IMWRITE('ELIMINATE_BY_MIN_SIZE.PNG',ELIMINATE_BY_MIN_SIZE)

# 5.2 REMOVING NOT RECTANGULAR REGIONS.
LABELED_HULL_THEN_ELIMINATE_BY_RF = LABCONEBRF(ELIMINATE_BY_MIN_SIZE)
```

```
CV2.IMWRITE('LABELED_HULL_THEN_ELIMINATE_BY_RF.PNG',LABELED_HULL_THEN_ELI
MINATE_BY_RF)

# 5.3 REMOVING TOO THIN CONVEX HULLS.
SECOND_H_OPENING = CV2.MORPHOLOGYEX(LABELED_HULL_THEN_ELIMINATE_BY_RF,
CV2.MORPH_OPEN, KERNEL('CROSS', (21,1)) )
CV2.IMWRITE('SECOND_H_OPENING.PNG',SECOND_H_OPENING)

# 6. LOCALIZE DETECTED REGIONS INSIDE BOUNDING BOXES.
# 6.1 LOCALIZING THE TEXT REGIONS.
FILLED_BOUNDING_BOXES = BOUNDINGBOX(SECOND_H_OPENING,0)
CV2.IMWRITE('FILLED_BOUNDING_BOXES.PNG',FILLED_BOUNDING_BOXES)
DILATION = CV2.DILATE(FILLED_BOUNDING_BOXES, KERNEL('CROSS', (1,3)) )
CV2.IMWRITE('DILATION.PNG',DILATION)

# 6.2 SUPERIMPOSITION OF THE BOUNDING BOXES TO THE ORIGINAL IMAGE.
RED_BOUNDING_BOXES = BOUNDINGBOX(DILATION,1)
CV2.IMWRITE('RED_BOUNDING_BOXES.PNG',RED_BOUNDING_BOXES)
```

3.5 Conclusion

After we explained the concepts, in the previous two chapters, in this chapter we implemented the horizontal artificial texts detection algorithm with all the stages that contain various processing operations that we saw previously.

In the end, we detect the horizontal artificial texts in the original image, the result was a modified version of the original image containing bounding boxes around the detected texts.

General Conclusion

General Conclusion

In this report, we presented an algorithm to detect and process artificial text in images and videos. A very high detection rate is obtained with a simple algorithm where only small percentage of the text boxes in the test data are missed. This method contains an integral approach for the localization of the text boxes before they are passed to a commercial OCR.

We were interested in mathematical morphology operations to analyse text detection in image and video sequence. These operators were used in most steps of text detection algorithm and the results obtained in a collection of test images are encouraging and confirm the success of the approach.

As perspective, we will be concentrated on text with fewer constraints, i.e. scene text, text with general orientations and moving text. Scene text, which has not been designed intentionally to be read easily, demands a different kind of treatment.

Bibliographie

[1] M. L. S. &. K. J. (. 2. 1.-1. Zhao, “Text detection in images using sparse representation with discriminative dictionaries.,” Image and Vision Computing, 2010.

[2] X. Y. C. W. H. W. Y. Z. S. H. W. &. L. J. Zhou, “East: an efficient and accurate scene text detector. (pp. 5551-5560).,” In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, 2017.

[3] M. H. V. &. B. R. Sonka, “Image processing, analysis, and machine vision,” Cengage Learning, 2014.

[4] S. Das, ““Computer Vision” [Lecture notes],” 2019. [Online]. Available: http://www.cse.iitm.ac.in/~vplab/courses/CV_DIP/CVHandout2019.pdf.

[5] ChristophorosNikou, ““Digital Image Processing Morphological Image Processing” [PowerPoint slides].,” 2013. [Online]. Available: http://www.cs.uoi.gr/~cnikou/Courses/Digital_Image_Processing/2012-2013/Chapter_09_Mathematical_Morphology_6spp.pdf.

[6] Brown University, [Online]. Available: https://cs.brown.edu/courses/csci1290/labs/lab_compositing/Morphology.pdf.

[7] C. &. J. M. 6. 3.-3. Wolf, “Extraction and recognition of artificial text in multimedia documents, Formal Pattern Analysis & Applications,,” 2004.

[8] Francis, L. M. and N. Sreenath, “ TEDLESS–Text detection using least-square SVM from natural scene. Journal of King Saud University-Computer and Information Sciences,,” 2020.

[9]]. K. Jung, K. I. Kim and A. K. Jain, ‘‘Text information extraction in images and video: A survey,’’ Pattern Recognition., vol. 37, no. 5, pp. 977–997., 2004.

[10] R. clouard, “Tutorial: Detecting artificial text in still images,” 2013. [Online]. Available: https://clouard.users.greyc.fr/Pantheon/experiments/text-detection/index-en.html.

[11] W. S. H. C. J. H. X. &. Y. J. (. 6. 4.-4. Lu, “A novel approach for video text detection

and recognition based on a corner response feature map and transferred deep convolutional neural network.," IEEE Access, 2018.

[12] U. Gargi, D. Crandall, S. Antani, T. Gandhi, R. Keener and R. Kasturi, "A system for auromatic text detection in videos, In the proceedings of the FIfth International Conference on Document Analysis and Recognition," ICDSR'99 IEEE, 1999.

[13] D. Chen, "Text detection and recognition in images and video sequences," 2003.

[14] L. C. H. and C. K. Lee, " Minimum cross entropy thresholding .," in *Pattern recognition,*, 1993, pp. 26(4), 617-625..

[15] RégisClouard, "pentropybinarization," 2017. [Online]. Available: https://clouard.users.greyc.fr/Pandore/programmes/en/operatorsP0/prectangularityselection.

[16] RégisClouard, "prectangularityselectionRetrieved," 2017. [Online]. Available: https://clouard.users.greyc.fr/Pandore/programmes/en/operatorsP0/prectangularityselection.html.

[17] Refsnes Data, "Python Introduction," [Online]. Available: https://www.w3schools.com/python/python_intro.asp.

[18] Princton University, [Online]. Available: https://csguide.cs.princeton.edu/software/virtualenv.

[19] audioanalytic, "audio analytic joins arms ai partner program," [Online]. Available: https://www.audioanalytic.com/audio-analytic-joins-arms-ai-partner-program/.

[20] H. Yang, B. Quehl and H. Sack, ''A framework for improved video text detection and recognition,'' Multimedia Tools Appl., vol. 69, no. 1, pp. 217–245, 2014.

[21] J. C. &. R. J. C. Russ, " Introduction to image processing and analysis," CRC press, 2017.

[22] [Online]. Available: https://www.quora.com/unanswered/How-is-data-science-helpful-for-uplifting-machine-learning-and-AI.