Optical Character Recognition for videos

Amad Salmon 1990455
Arturo Calvera 1989337

Why do we need video OCR?

Information is everywhere.

We can exploit it for:

- Video captioning & annotating
- Text is used as visual indicators for navigation and notification in scenes.
- Video indexing:
  - Media archiving based on content
  - Video cataloging based on content relevance
- More accessibility for the visually impaired

Example of a commercially available product

Optical Character Recognition Glasses: OrCam MyEye

A device capable of reading out loud text for visually impaired people in real time.

Uses a region based approach as the user is required to point with their finger the general region where the text is located so the device can read it.

"Text Detection, Tracking and Recognition in Video: A Comprehensive Survey"

Xu-Cheng Yin, Ze-Yu Zuo, Shu Tian, Cheng-Lin Liu - 2016

[link]

A Unified Framework for Video Text Detection, Tracking and Recognition

Video Text Detection and Recognition Using Individual Frames

Video Text Detection, Tracking and Recognition Using Multiple Frames

Text in video

Layered caption text

Embedded caption text

Scene text

Video Text Detection and Recognition Using Individual Frames

- **Connected component based methods**
  - Character extraction by connected component analysis
  - **Pros:** Performs well for captions that have uniform color and regular spacing
  - **Cons:** trouble with color bleeding and the low contrast.

- **Region based methods**
  - Binary classifier searches for text regions over windows of multiple sizes
  - **Cons:** Needs a preprocessing step
  - **Pros:** Fast and overcomes low contrast problems but may produce false positives with the background

Video Text Detection and Recognition Using Multiple Frames

Focus on using the spatial and temporal information acquired from multiple frames: The dependencies between adjacent video frames.

- Tracking With Template Matching
- Tracking With Tracking-by-Detection
- Tracking in the Compressed Domain

Challenges

Background-related

- Complex scenes with structures similar to text
- Complex background behind the text
- Visual deterioration from uneven lightning

Text-related

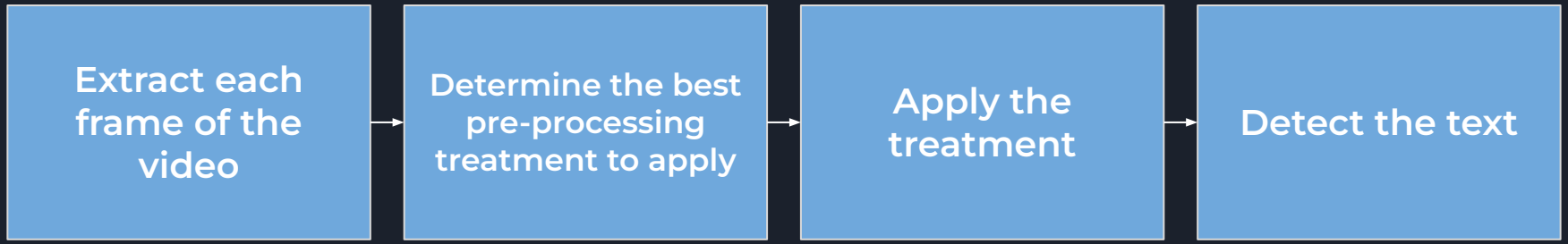
- Blurring from defocusing, motion or low resolution.
- Skew: text captured in different orientations.
- Naturally curved text
- Text sizes, fonts and shapes

Video-related

- Low resolution
- Complex non-linear motion (zooming in or out, rotating … etc) ⇒ motion blur
- Real-time processing: tracking algorithms need high computational efficiency

Our Approach

Frame by frame and region-based system



Our Approach

OpenCV

- Includes several hundreds of computer vision algorithms
- Powerful image processing modules *(linear and non-linear filtering, transformations, etc.)*
- Easy-to-use interface to video capturing and video codecs

Tesseract OCR engine

- New neural network based
- Can recognize more than 100 languages "out of the box"
- Easy-to-use PyTesseract wrapper for Python

Challenges

Challenges faced that cause poor performance

- Text on top of complex background structures
- Skew and curved text
- Poor quality from uneven lightning

How we've tackled them

- Pre-processing techniques to improve performance.
- Trying to figure out each frame's needs for pre-processing

Used pre-processing techniques

Binarization through thresholding:

- Simple thresholding
- Adaptive thresholding
- Otsu's thresholding

Used pre-processing techniques

Noise reduction:

Noise is random variation of brightness or colour in an image, that can make the text of the image more difficult to read.

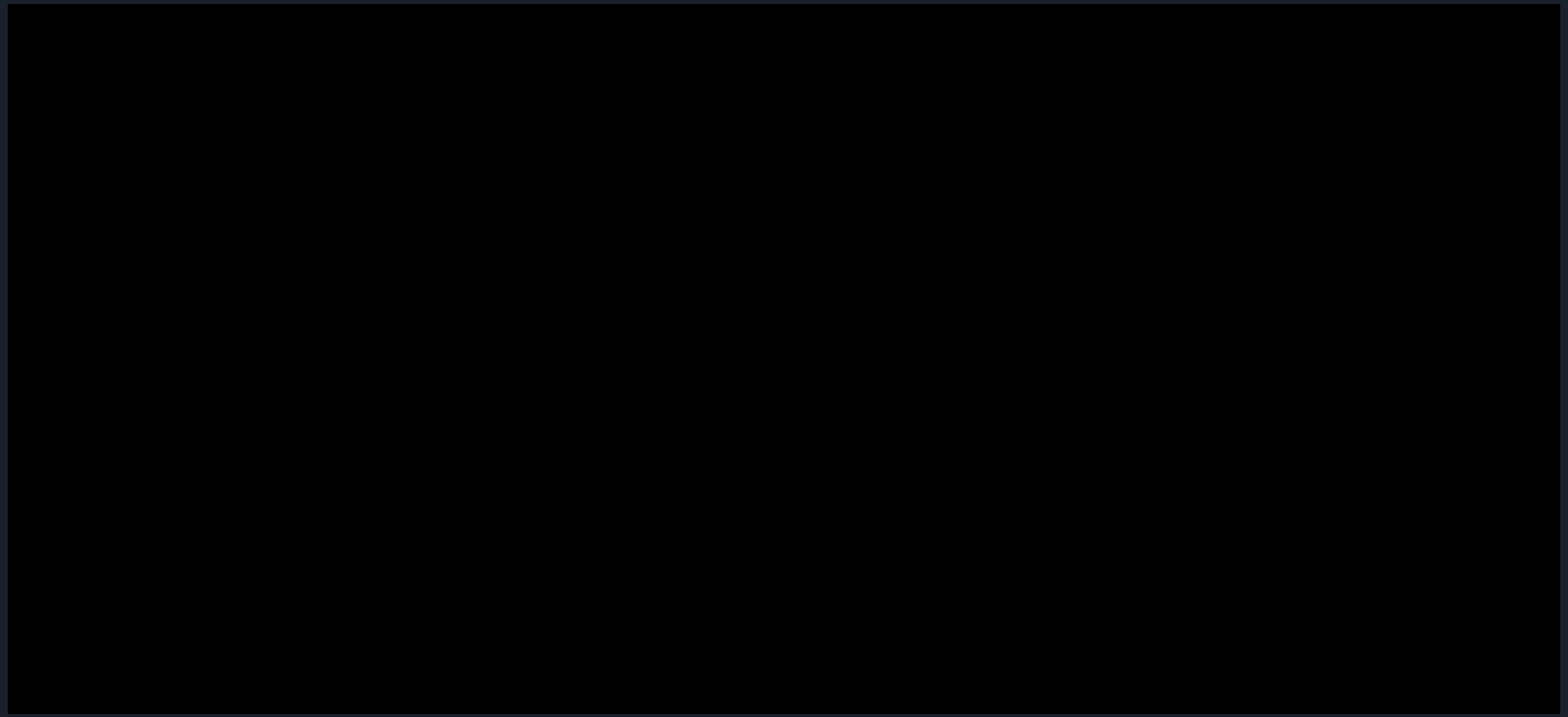
Dilation:

Opposite of erosion, object area increases. It is also useful in joining broken parts of an object.

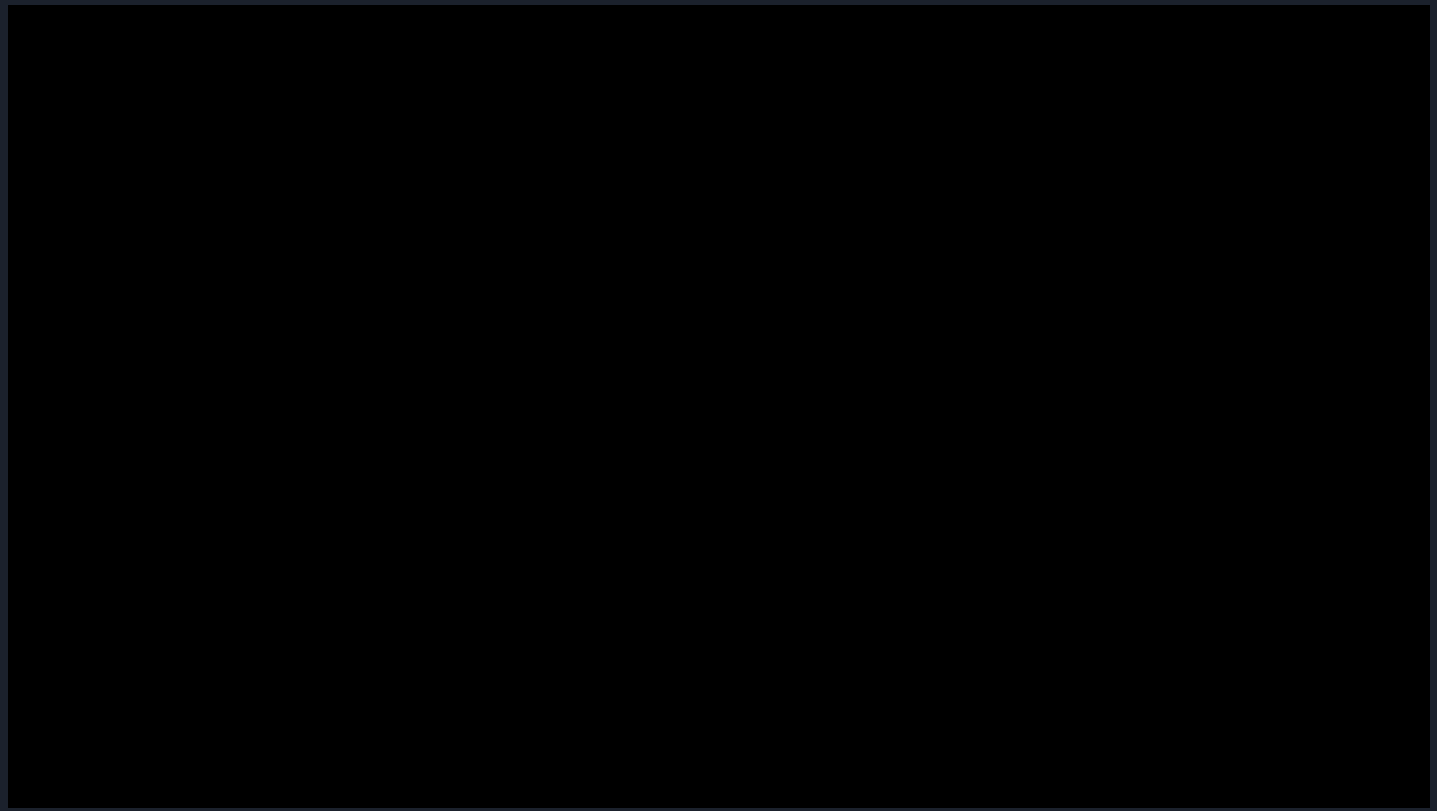
Erosion:

The basic idea of erosion is just like soil erosion only, it erodes away the boundaries of foreground object

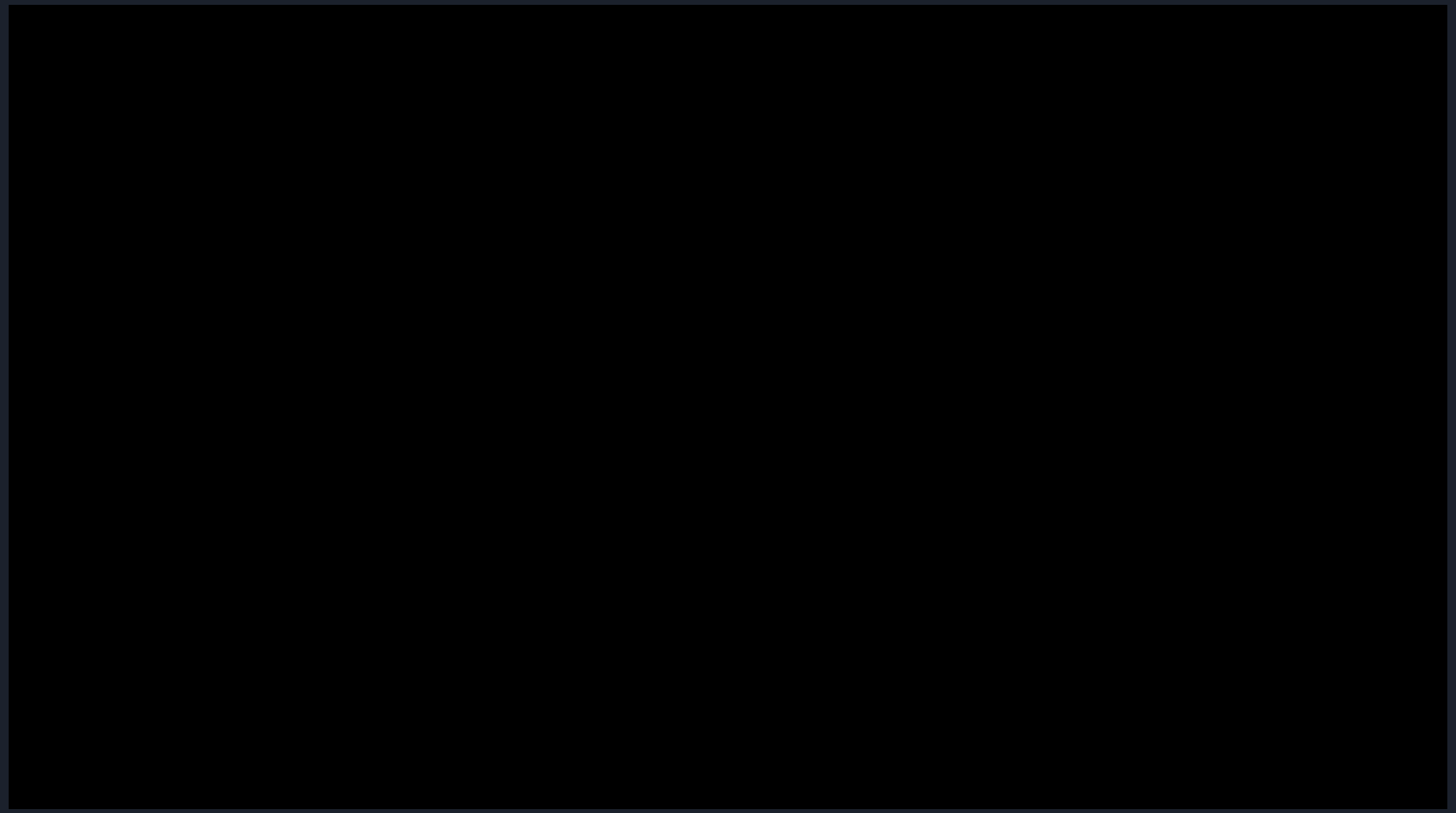
Final video output - Simple example



Final video output - Medium example



Final video output - Complex example



Future improvements

- Work out relations between frames to find text in the following frames.
- Real-time OCR.
- Tackle better text embedded in real life objects.

Questions?