




# Confidential Whiteboard Contents



## Group D

*Fenno Boomgaarden*  
*Hauke Redemann*  
*Keno Rott*



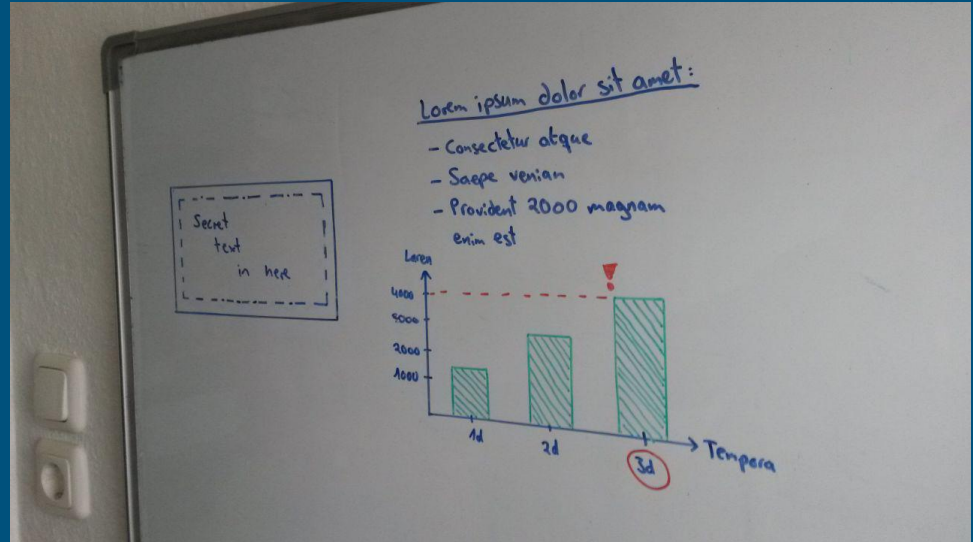
# Overview

---

1. Task Definition
2. Use Cases
3. Processing Chain
4. Conclusion
5. Outlook: Deep Learning

# Task Definition

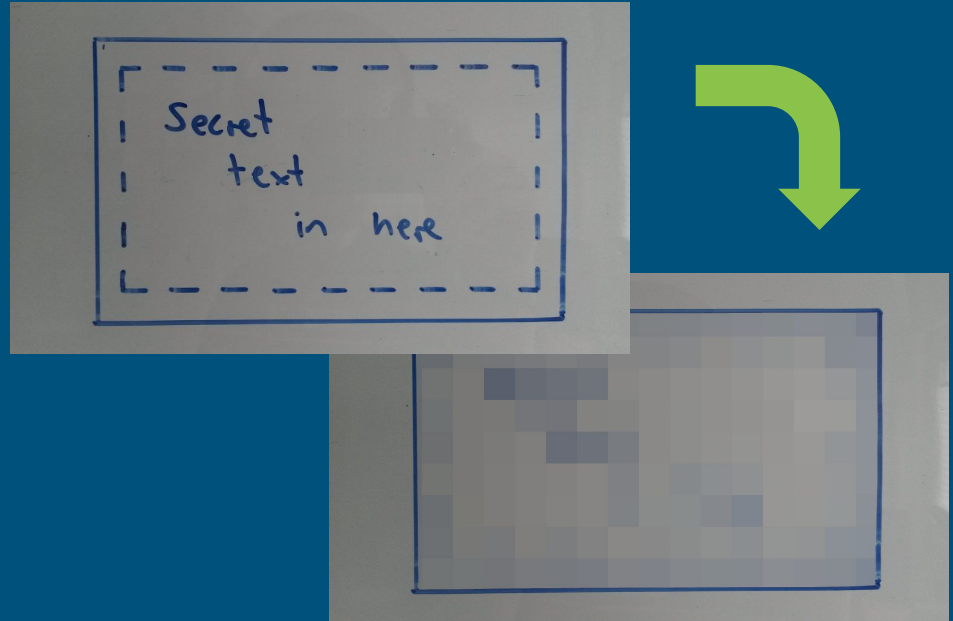
- Protect handwritten notes
- Secret pattern
  - Outer rectangle
  - Dashed rectangle inside
- Blur the insides of the pattern



# Task Definition

---

- Protect handwritten notes
- Secret pattern
  - Outer rectangle
  - Dashed rectangle inside
- Blur the insides of the pattern



# Use Case 1

Lecture recording  
software



# Lecture recording software

---

- Fully automated systems
- Time-based
- Upload without review

⇒ Practical solution

“

Das Passwort für die  
Unterlagen steht an der Tafel.

—

# Potential problems

---

- Access credentials
- Copyrighted material
- Solutions for exam exercises

⇒ Hide sensitive information **without** manually editing the video

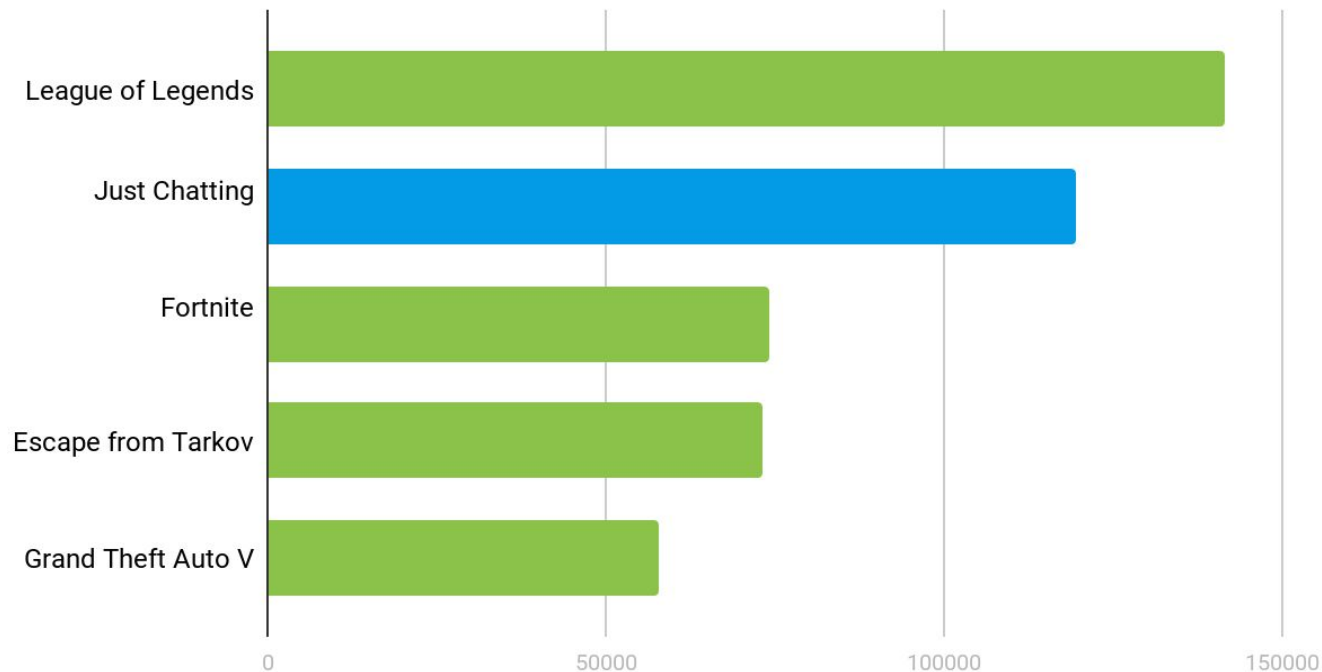


## Use Case 2

Real-life  
video streaming



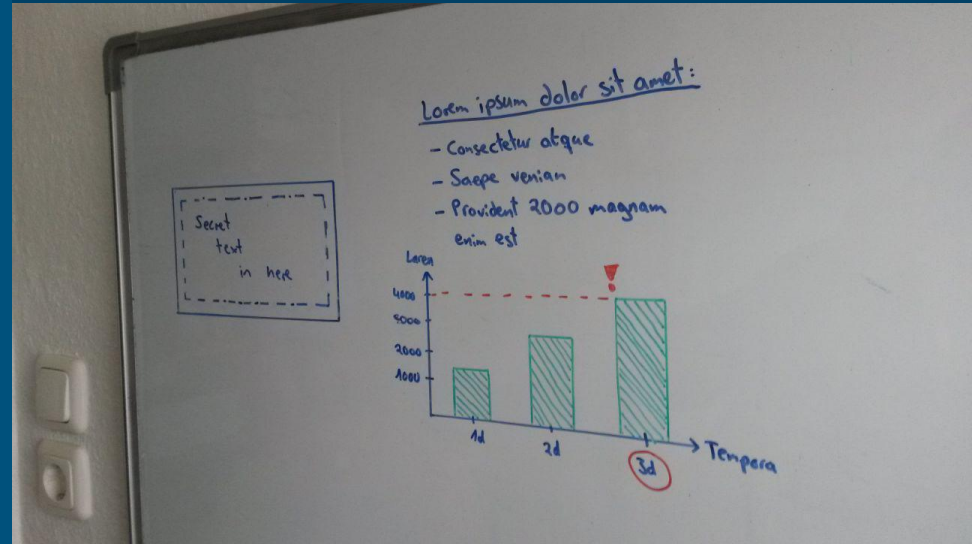
## Twitch: Top Categories



<https://twitchtracker.com/statistics/games> (2020-01-20)

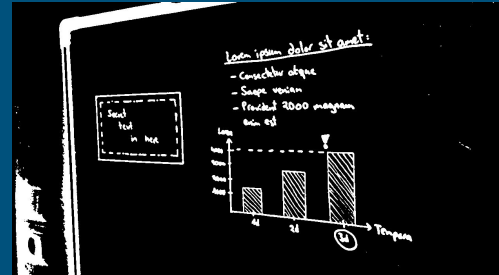
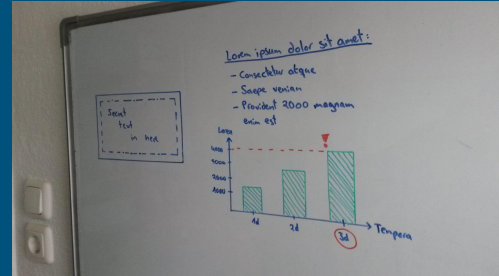
# Processing Chain

Solving the task



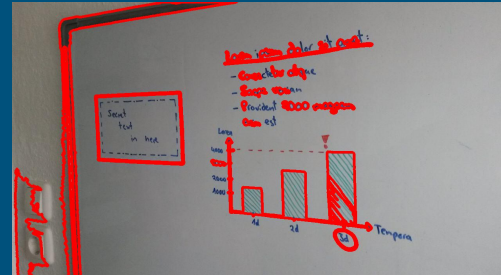
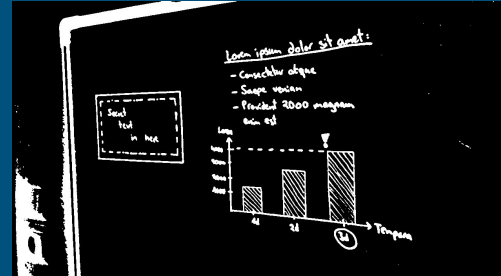
# Processing chain

1. Preprocessing
2. Contour Detection
3. Rectangle Recognition
4. Detection of dashed lines
5. Image Manipulation



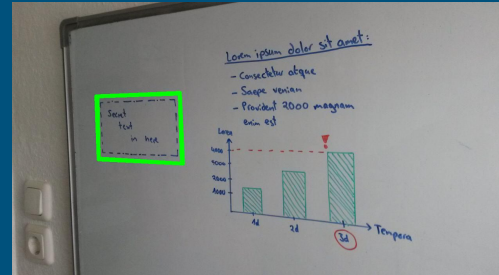
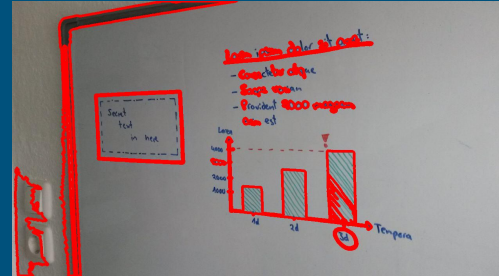
# Processing chain

1. Preprocessing
2. **Contour Detection**
3. Rectangle Recognition
4. Detection of dashed lines
5. Image Manipulation



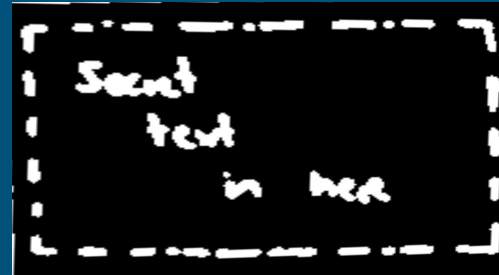
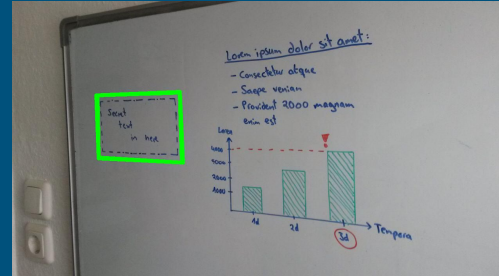
# Processing chain

1. Preprocessing
2. Contour Detection
- 3. Rectangle Recognition**
4. Detection of dashed lines
5. Image Manipulation



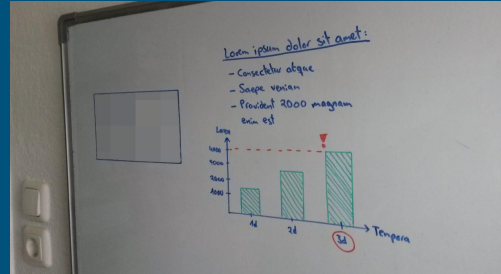
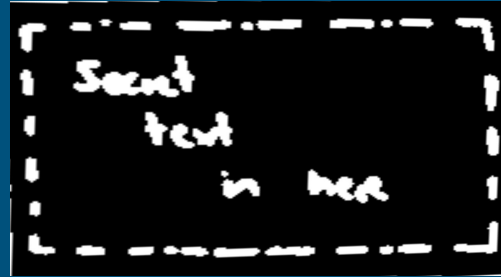
# Processing chain

1. Preprocessing
2. Contour Detection
3. Rectangle Recognition
4. **Detection of dashed lines**
5. Image Manipulation



# Processing chain

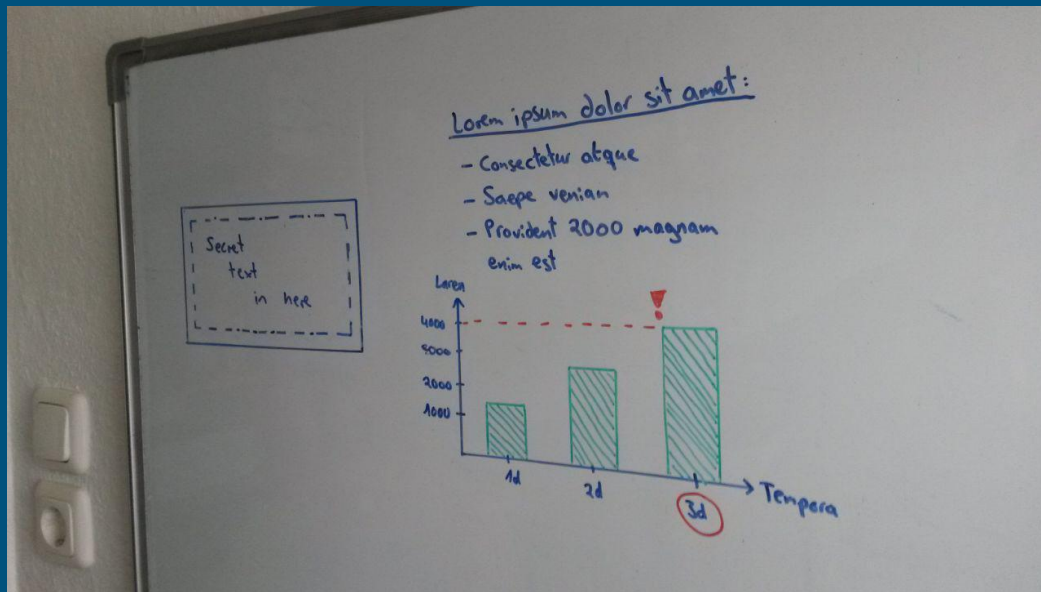
1. Preprocessing
2. Contour Detection
3. Rectangle Recognition
4. Detection of dashed lines
5. **Image Manipulation**





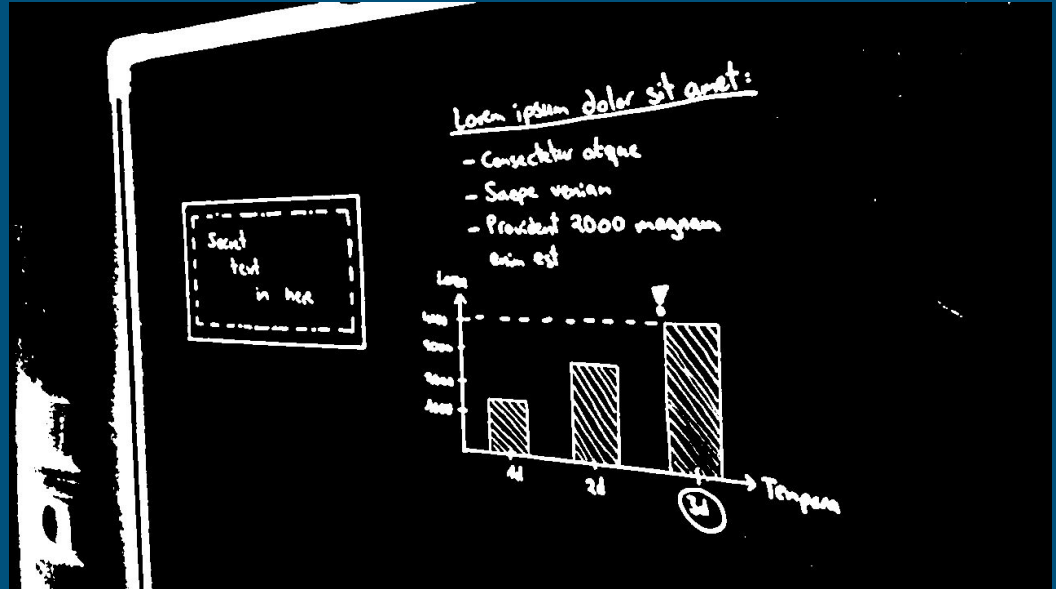
# Preprocessing

- Differentiate drawing and background
- Filter out light reflections
- Close small gaps



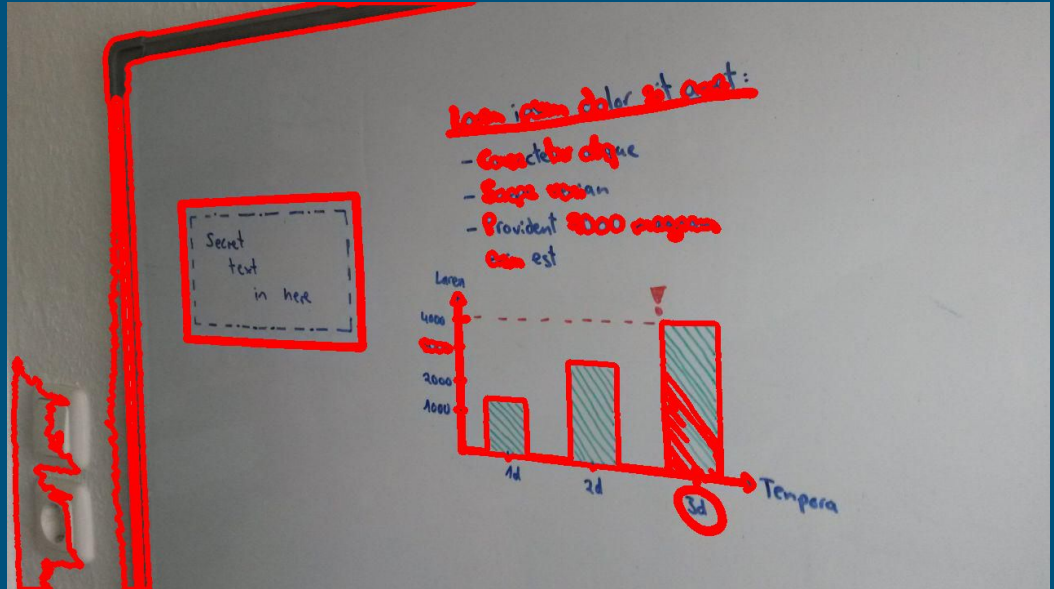
# Preprocessing

- Differentiate drawing and background
- Filter out light reflections
- Close small gaps



# Contour detection

- Find all contours in the image
- Pre-filtering is required



# Pre-filtering

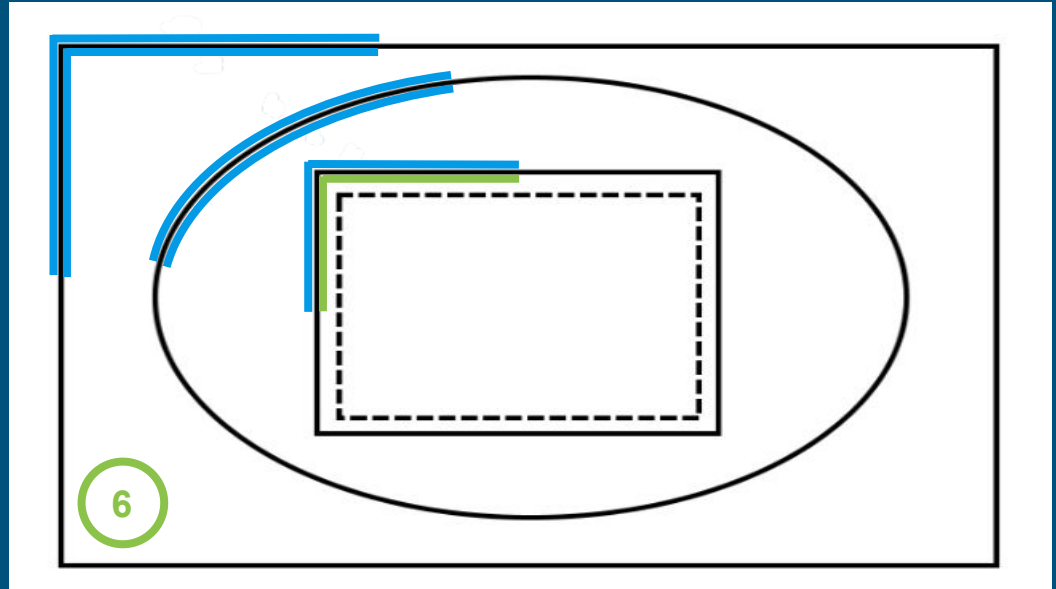
- OpenCV returns **inner** and **outer** contours
- Remove all outer contours

⇒ Minimize workload



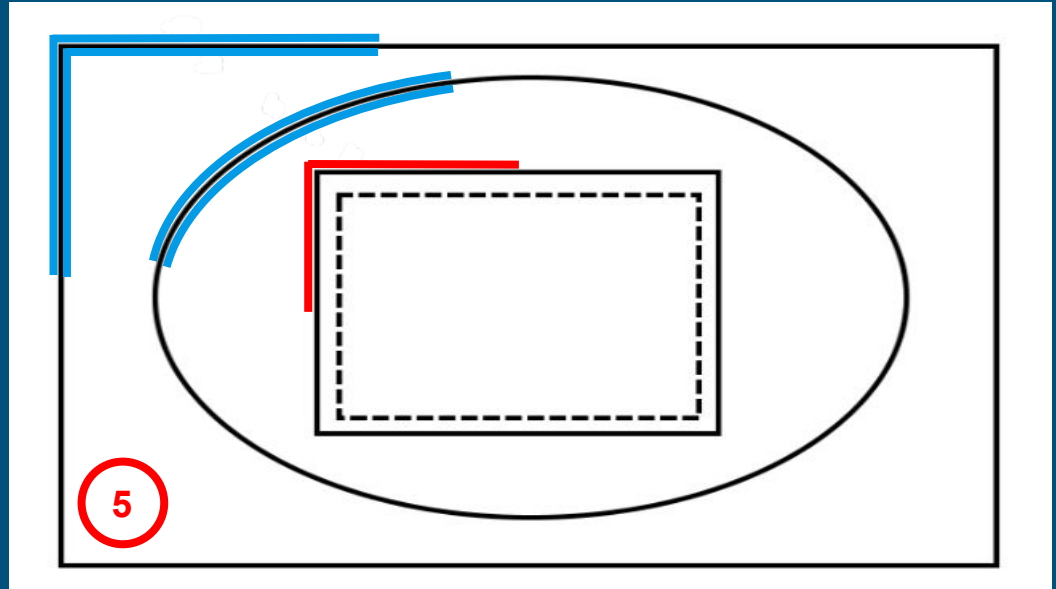
# Pre-filtering

- Inner contour
- Even depth in the hierarchy tree



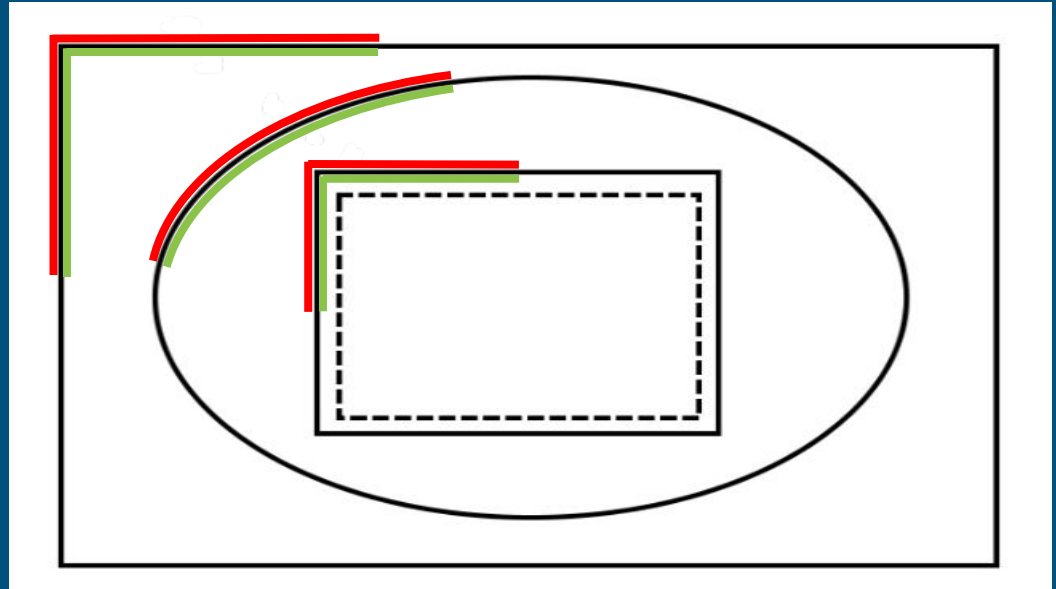
# Pre-filtering

- **Outer** contour
- Uneven depth in the hierarchy tree



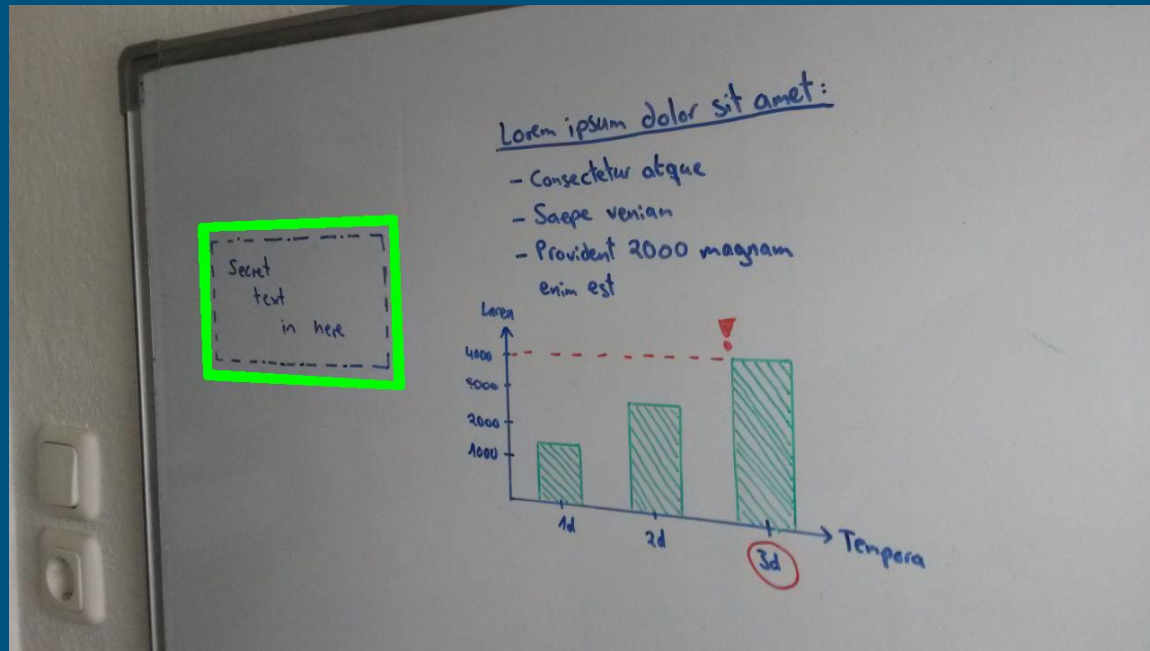
# Pre-filtering

- Contours can be classified
- Additionally: Remove contours with no children



# Rectangle Recognition

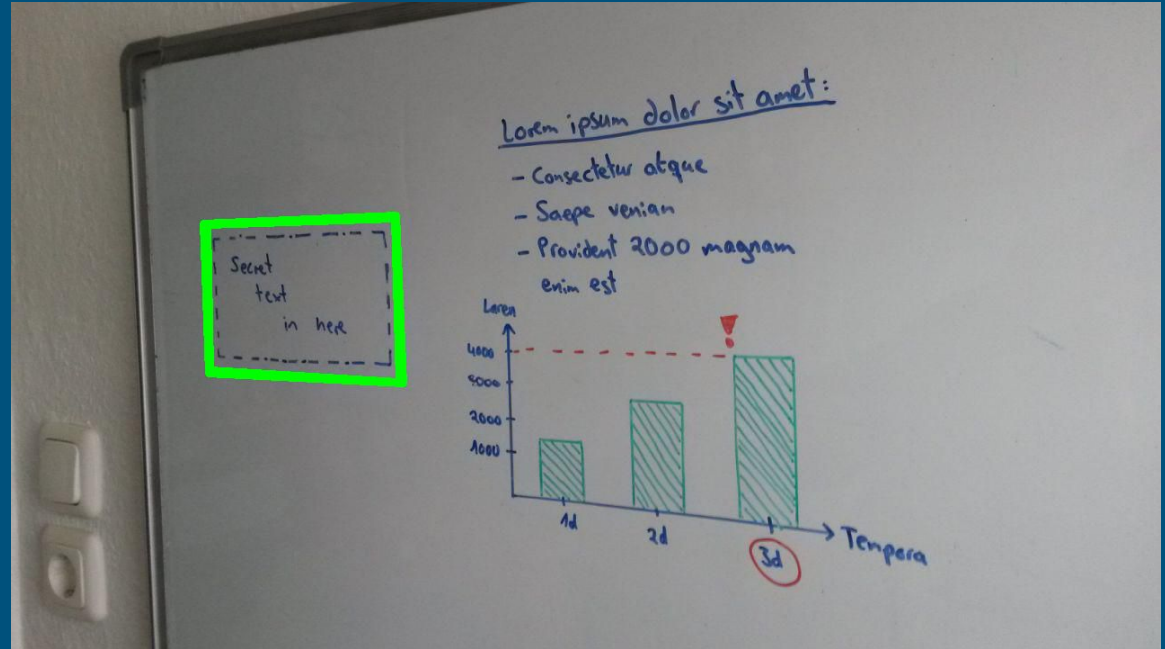
- Approximated contour has four vertices
  - Convex shape
- ⇒ All **rectangles** are potential matches





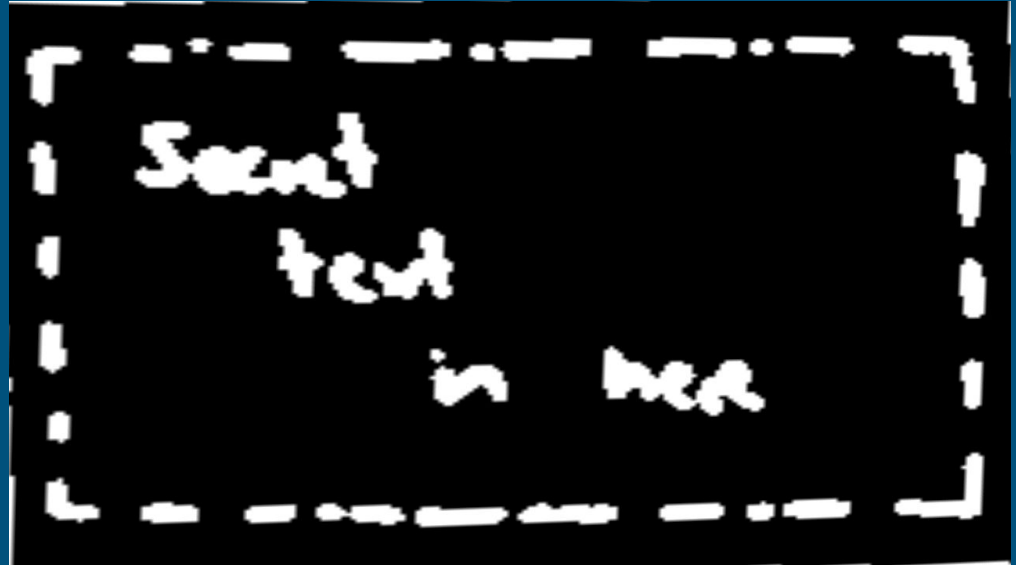
# Detection of dashed lines

- Create a sub-image for each rectangle
- Further processing on each sub-image



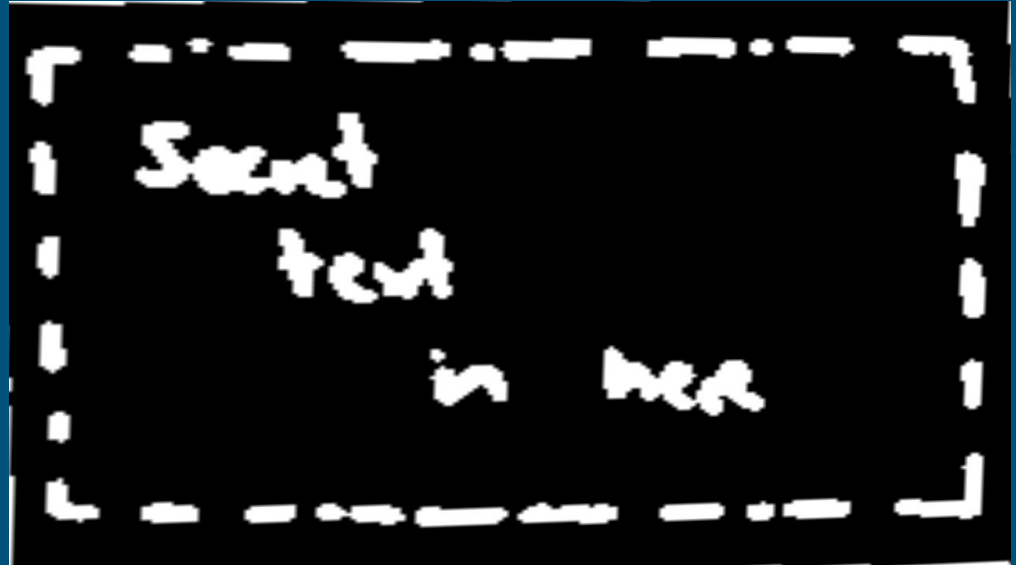
# Detection of dashed lines

- Create a sub-image for each rectangle
- Further processing on each sub-image



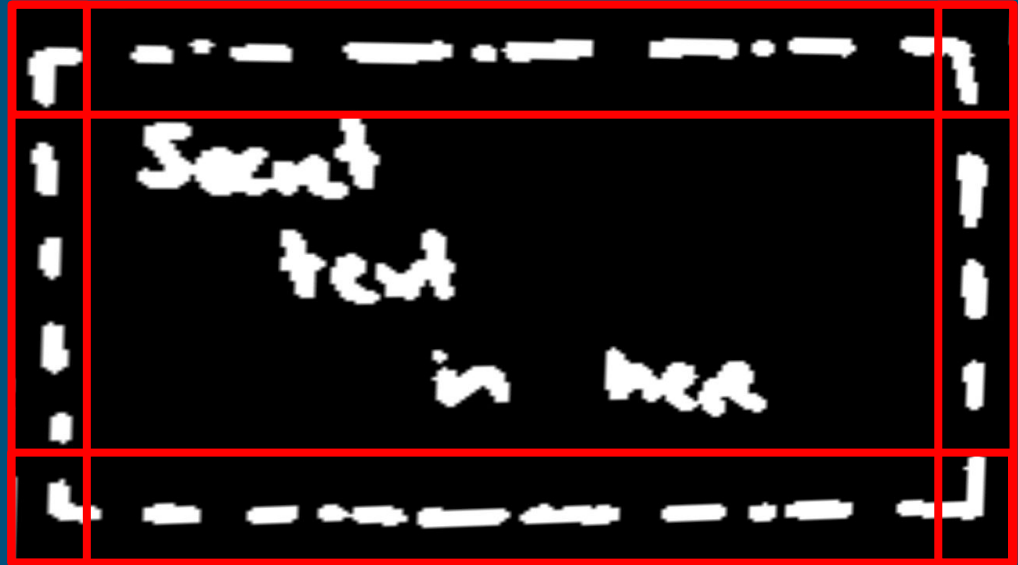
# Sub-image Slices

- Create slices from predefined margins
- Further processing on each slice



# Sub-image Slices

- Create slices from predefined margins
- Further processing on each slice



# Sub-image Slices

- Create slices from predefined margins
- Further processing on each slice



# Sub-image Slices

- Create slices from predefined margins
- Further processing on each slice



# Dashed line detection

- White pixels projected to a 1D array
- Compare **length** of each segment to a **predefined maximum**



# Dashed line detection

- White pixels projected to a 1D array
- Compare **length** of each segment to a **predefined maximum**



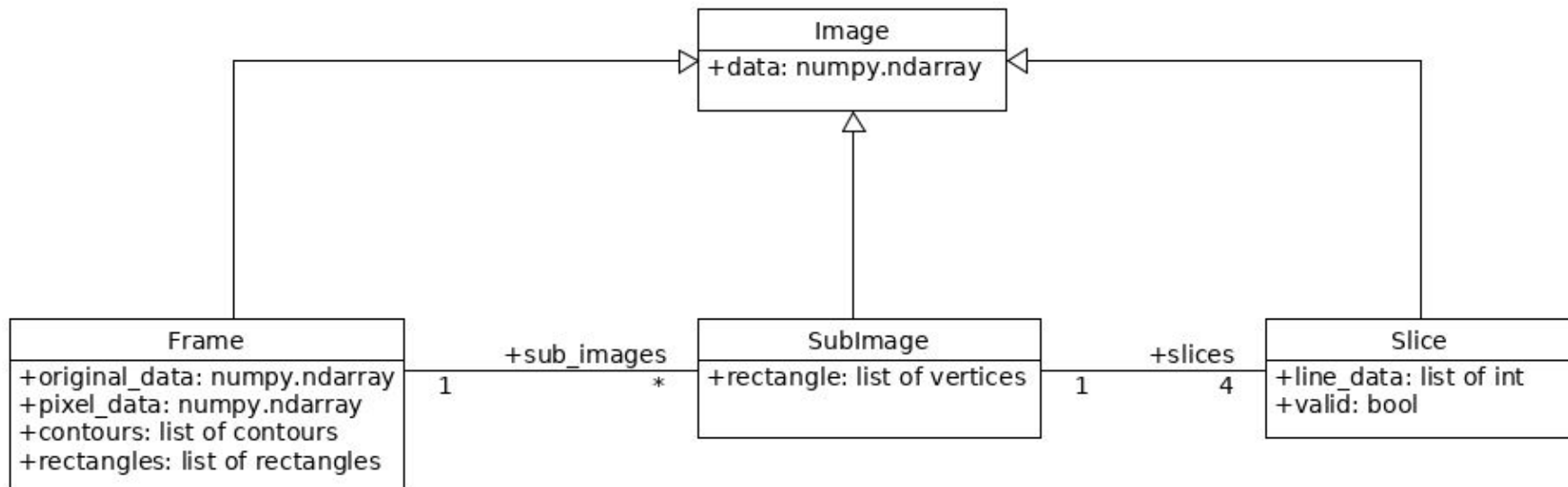


# Dashed line detection

- White pixels projected to a 1D array
- Compare **length** of each segment to a **predefined maximum**

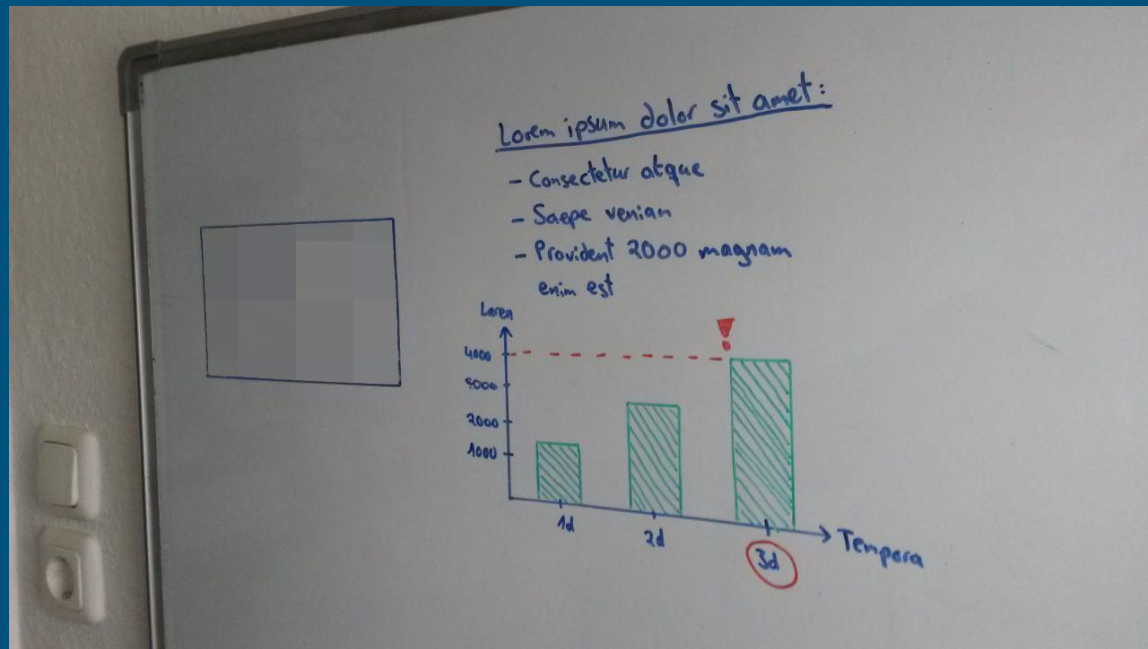


# Class structure



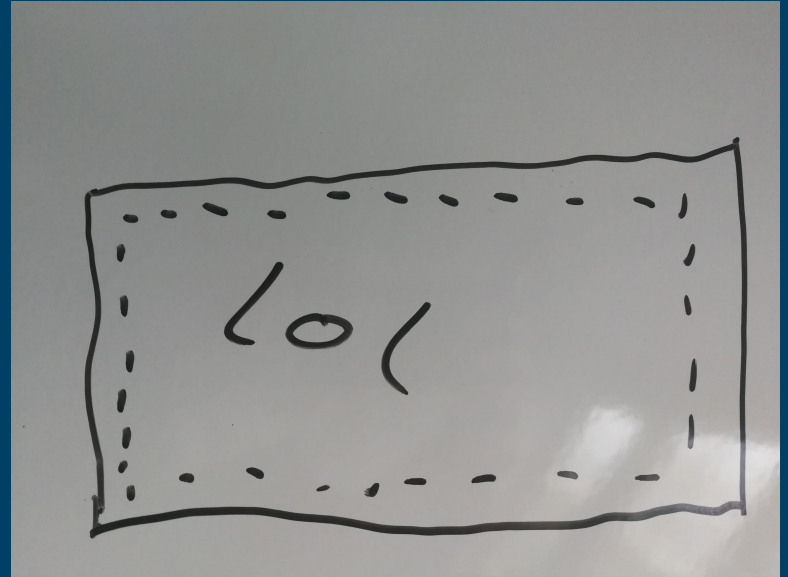
# Image Manipulation

- Create mask from original contour
- Replace insides of the mask with pixelated image data



# Conclusion

Statistics and examples



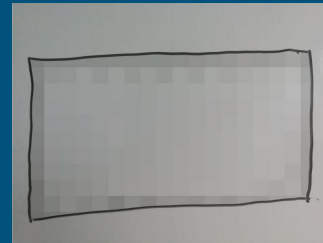
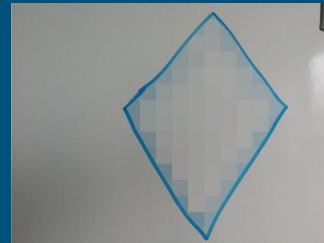
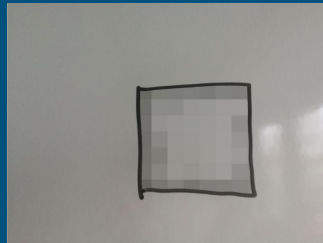
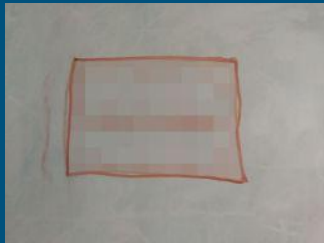
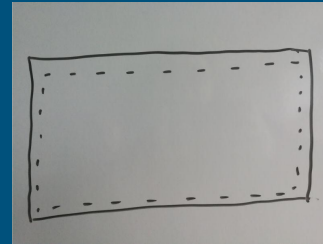
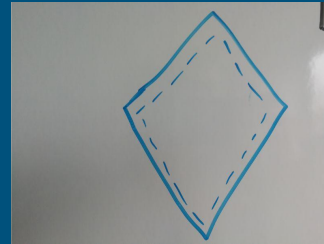
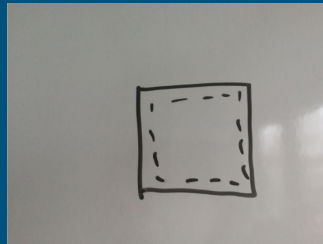
# Average execution times

---

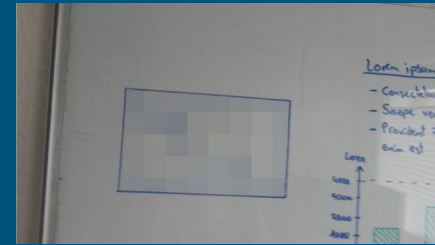
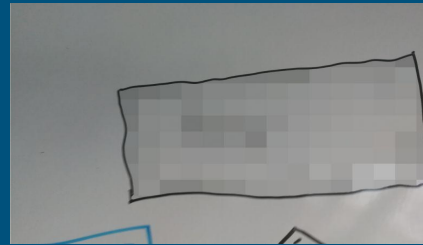
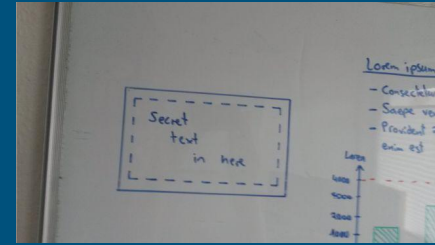
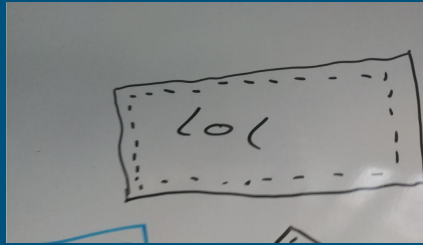
<b>Preprocessing</b>	<b>468 ms</b>
<b>Contour Detection</b>	<b>6 ms</b>
<b>Rectangle Detection</b>	<b>4 ms</b>
<b>Create sub-images</b>	<b>63 ms</b>
<b>Extract slices</b>	<b>1 ms</b>
<b>Detect dashed lines</b>	<b>1 ms</b>
Image Manipulation (blackened)	4 ms
Image Manipulation (blurred)	3000 ms
<b>Image Manipulation (pixelated)</b>	<b>37 ms</b>

# Working examples

---

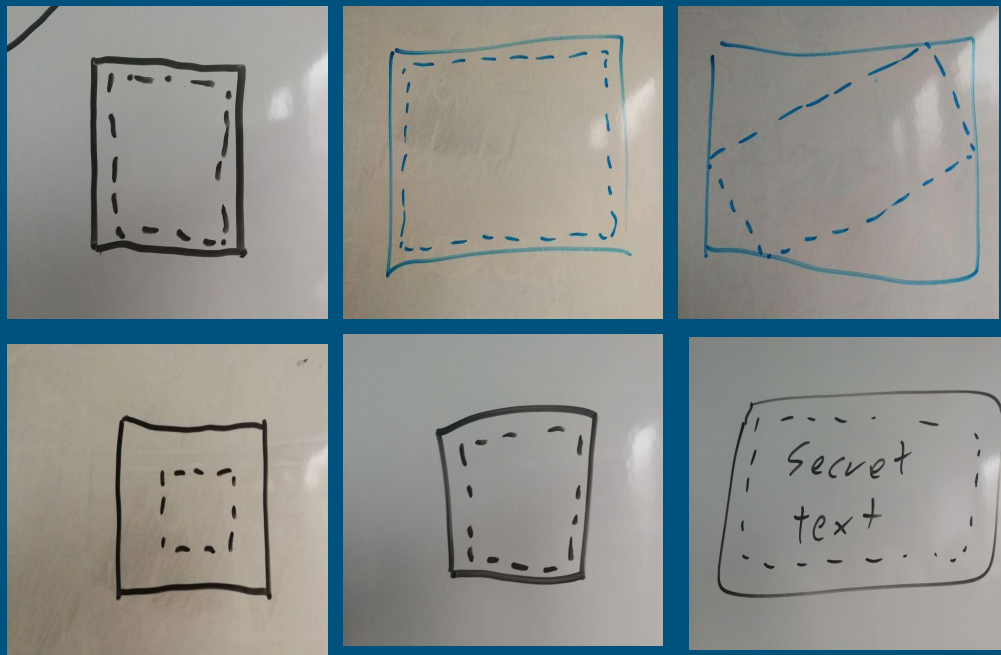


# Working examples



# Failing examples

- Long dashes ( $\geq$  threshold)
- Big holes in the outlines
- Too much distance between lines
- Curvy rectangles





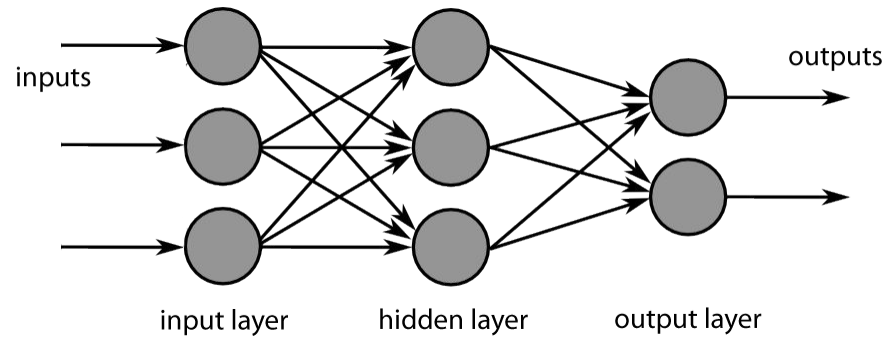
# Conclusion

---

- Great results on most images
- Robust solution
- Inflexible pattern detection
- Not (yet) real-time capable

# Outlook

## Deep Learning



# Motivation

---

**Problem:** Algorithm too inflexible for poorly drawn patterns

⇒ Deep Learning could provide more general solutions

# Motivation

---

**Problem:** Algorithm too inflexible for poorly drawn patterns

⇒ Deep Learning could provide more general solutions

**Idea:** Identify major error sources in our pipeline

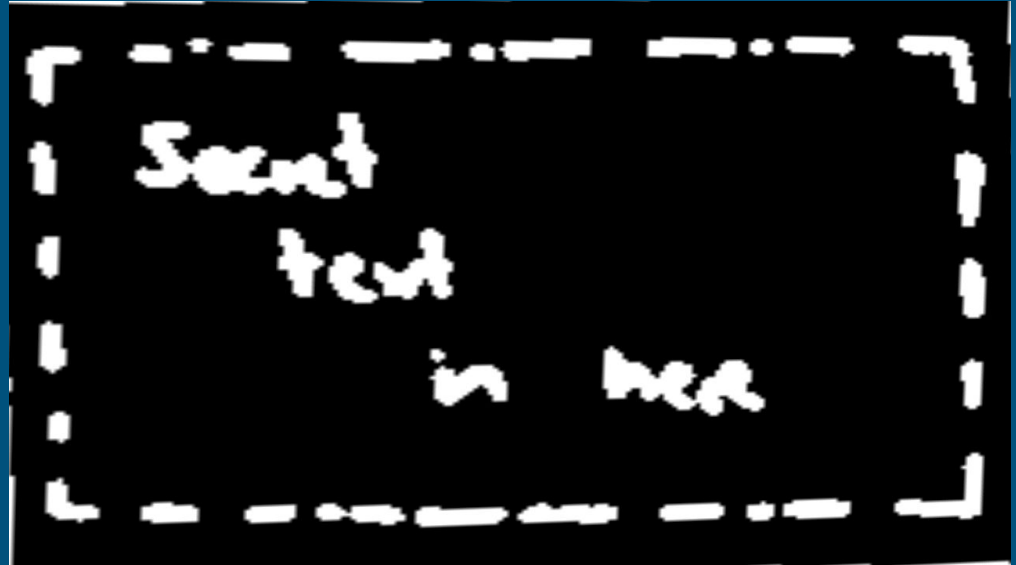
- Rectangle detection?
- Detection of dashed lines

⇒ Replace corresponding steps with CNNs

# Detection of dashed lines with CNN

---

- Construct a suitable CNN
- Train it on generated data



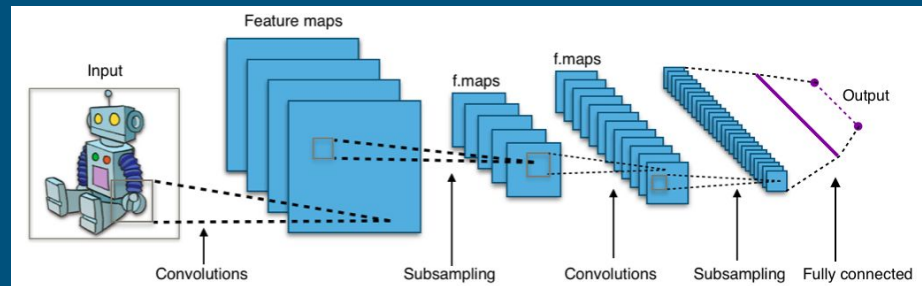
# Dashed Line Detection with CNN

**Easy task:** Classifier of dashed lines

But can we collect enough training data?

- Rectangle is already detected and transformed
- Very specific task

⇒ Data is easy to generate



Thank you for  
your attention!





# Image Sources

---

<https://www.flickr.com/photos/gpadjp/8121939408>

[https://twitch.tv/Van\\_Hinten88](https://twitch.tv/Van_Hinten88)

[https://upload.wikimedia.org/wikipedia/commons/4/47/MultiLayerNeuralNetwork\\_english.png](https://upload.wikimedia.org/wikipedia/commons/4/47/MultiLayerNeuralNetwork_english.png)

[https://upload.wikimedia.org/wikipedia/commons/6/63/Typical\\_cnn.png](https://upload.wikimedia.org/wikipedia/commons/6/63/Typical_cnn.png)