Decoder-Encoder Autoencoders for Unsupervised Decomposition into Visual Parts

Superviser: Prof. Remi Emonet, Prof. Thierry Fournel, Prof. Amaury Habrard

Mohammad Sadil Khan

UNIVERSITÉ DE LYON

UNIVERSITÉ JEAN MONNET SAINT-ÉTIENNE

2nd July, 2021

Contents:

Project Details

- The project is a part of ANR Roli (Rey’s Ornament Image Investigation).
- The ROIi (Rey's Ornament Image investigation) project brings together researchers in the fields of the history of ideas, literature and book history (IHRIM) on the one hand and computer vision and machine learning (Hubert Curien Laboratory) on the other.
- The project is centred around the ornaments found in the famous books published in 18th Century by Marc Michel Ray, a publisher in France during 18th Century.
- The aim is to decide whether some ornaments (thus the books) can be attributed to Marc-Michel Ray.
- Ornaments are basically composed of vignettes.

Project Details

- As publishing was subject to a regime of censorship and booksellers resorted to anonymity, use of false addresses and forgeries were common and book authentication became a challenge.

A Vignette

Figure 1

An Ornament

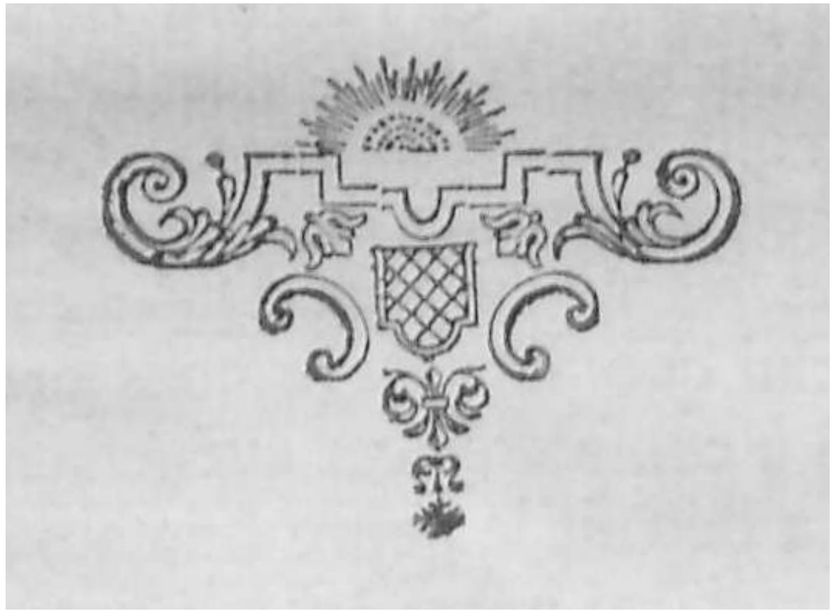
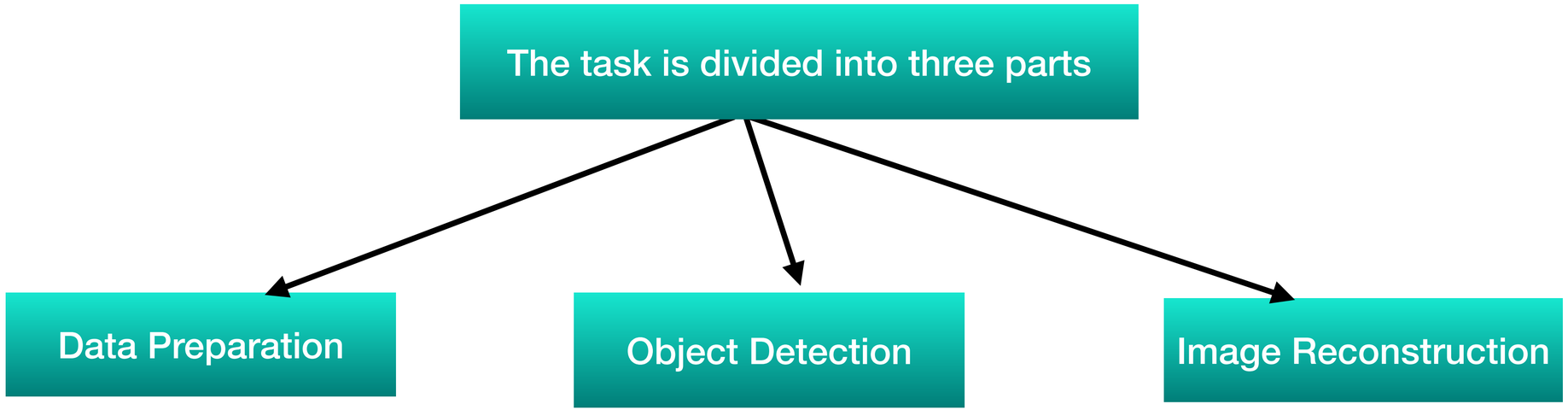


Figure 2

Internship Task

- The objective of this project is to design a tool to help authenticate books published under fictitious or counterfeit names or addresses in the eighteenth century, through the analysis of ornaments.

- Our goal is to create a novel Autoencoder ,*Detector-Encoder Autoencoder(DAE)* which will detect anomalies or sub-patterns in the ornaments which will help identify the authenticity of the ornaments in an unsupervised manner.

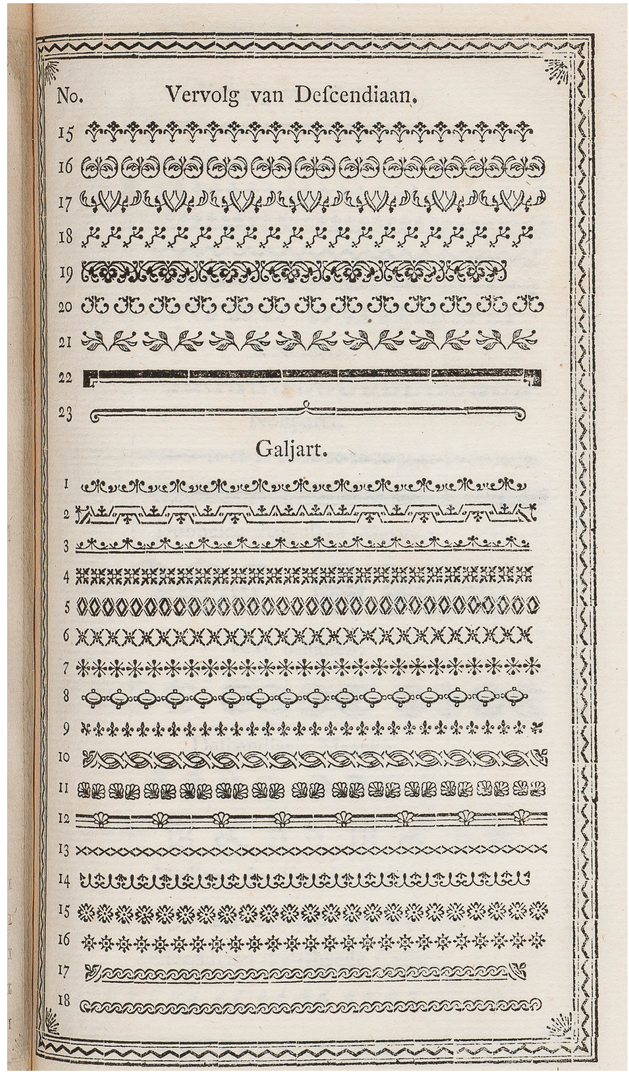


Data Preparation

- Data Collection is the most important step in Deep Learning projects.
- The quality and quantity of the data alone can greatly impact the results of the model.
- In the dataset we have 18 images, with the catalogues of vignettes.
- Since the first part of the DAE is an Object Detector, we need to create images with bounding box around the objects.

Figure 1

An image



AutoLabelMe

- AutoLabelme is an Automatic Image Annotator created in Python and it's an extension of LabelMe the open-source image Annotator.*
- The central idea of AutoLabelme is that it takes as input a template and a search space and tries to match with similar objects and associates a bounding box and a label.
- It uses Normalized Cross-Correlation to check whether two templates are similar.

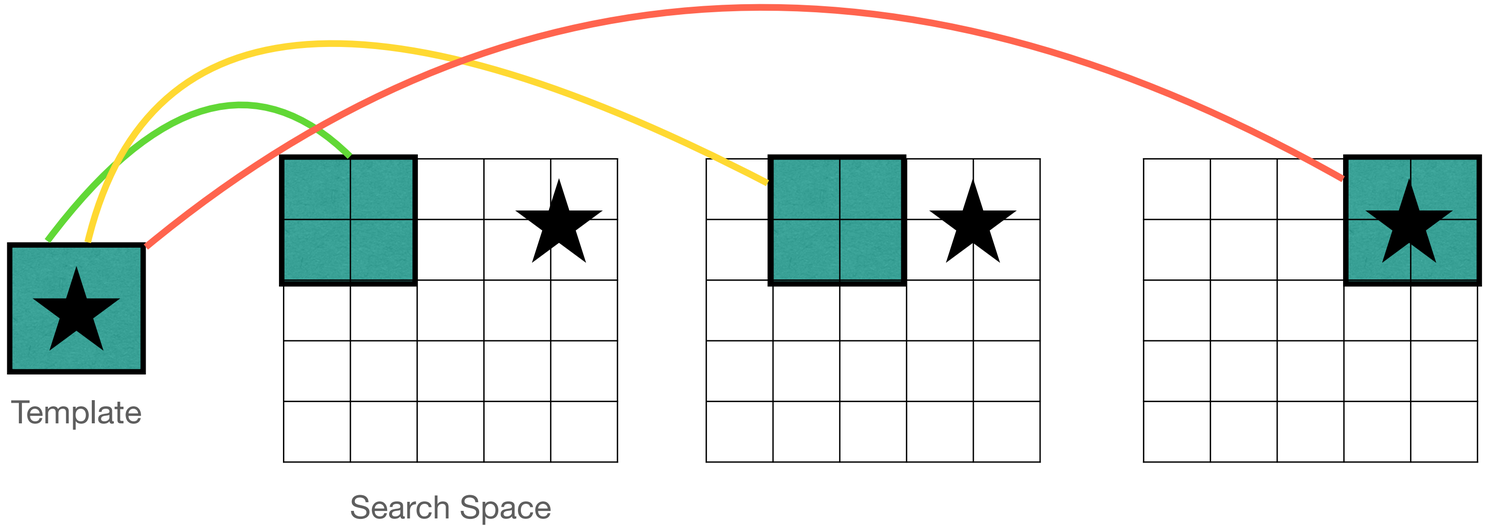


Figure 1

Normalized Cross Correlation

$$R(x,y) = \frac{\sum_{x',y'} (T'(x',y') \cdot I'(x+x',y+y'))}{\sqrt{\sum_{x',y'} T'(x',y')^2 \cdot \sum_{x',y'} I'(x+x',y+y')^2}}$$

Figure 2

*https://github.com/wkentaro/labelme

AutoLabelMe- Properties

- AutoLabelme* is simple to use. After LabelMe annotation just run Autolabelme.
- It's fast as we reduced the search space to the neighbourhood of the template which made it efficient for our project.
- AutoLabelme can detect rotated, scaled and flipped templates. No need of manual annotation.
- Save meta information about the boxes.

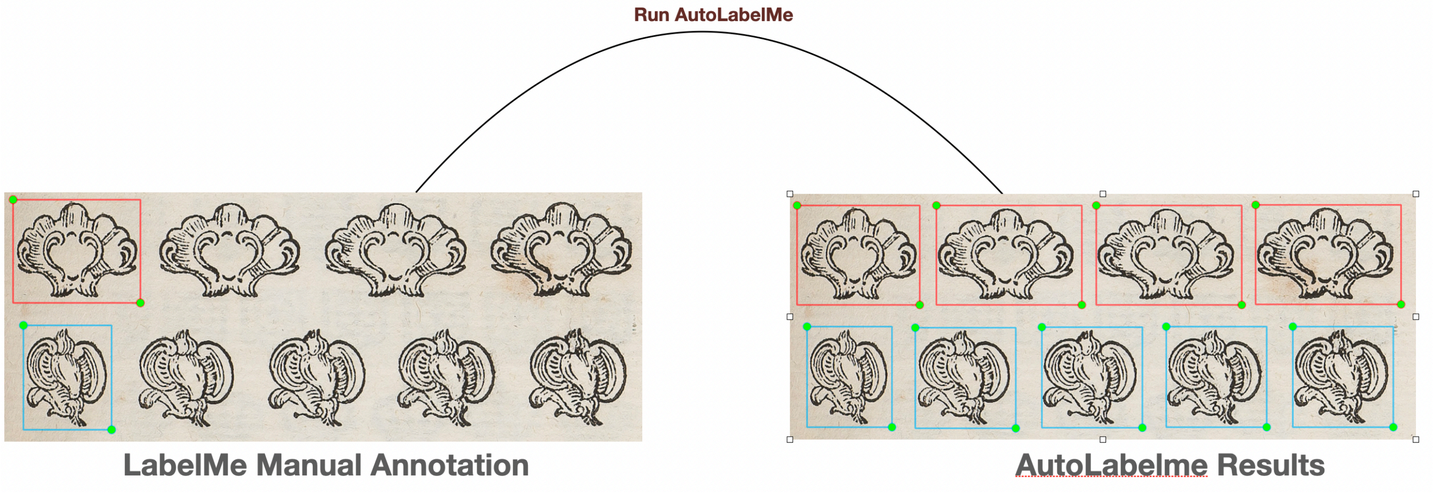


Figure 1

LabelMe Annotation

AutoLabelMe Results

Figure 2

LabelMe Annotation

AutoLabelMe Results

Figure 3

*https://github.com/SadilKhan/internship-hubert-curien-2021/tree/main/Data%20Generation

Ransac Flow GUI

- The images that are used are taken from books resulting in the projective transformation of the vignettes on the left side.
- To fix this, we used Ransac-Flow*, a two stage image alignment process to align the transformed images.

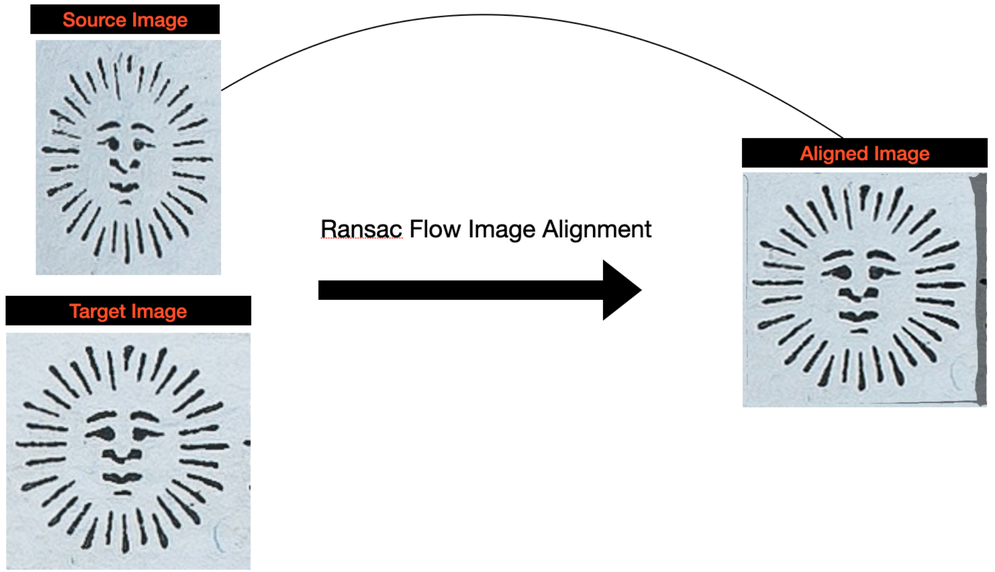
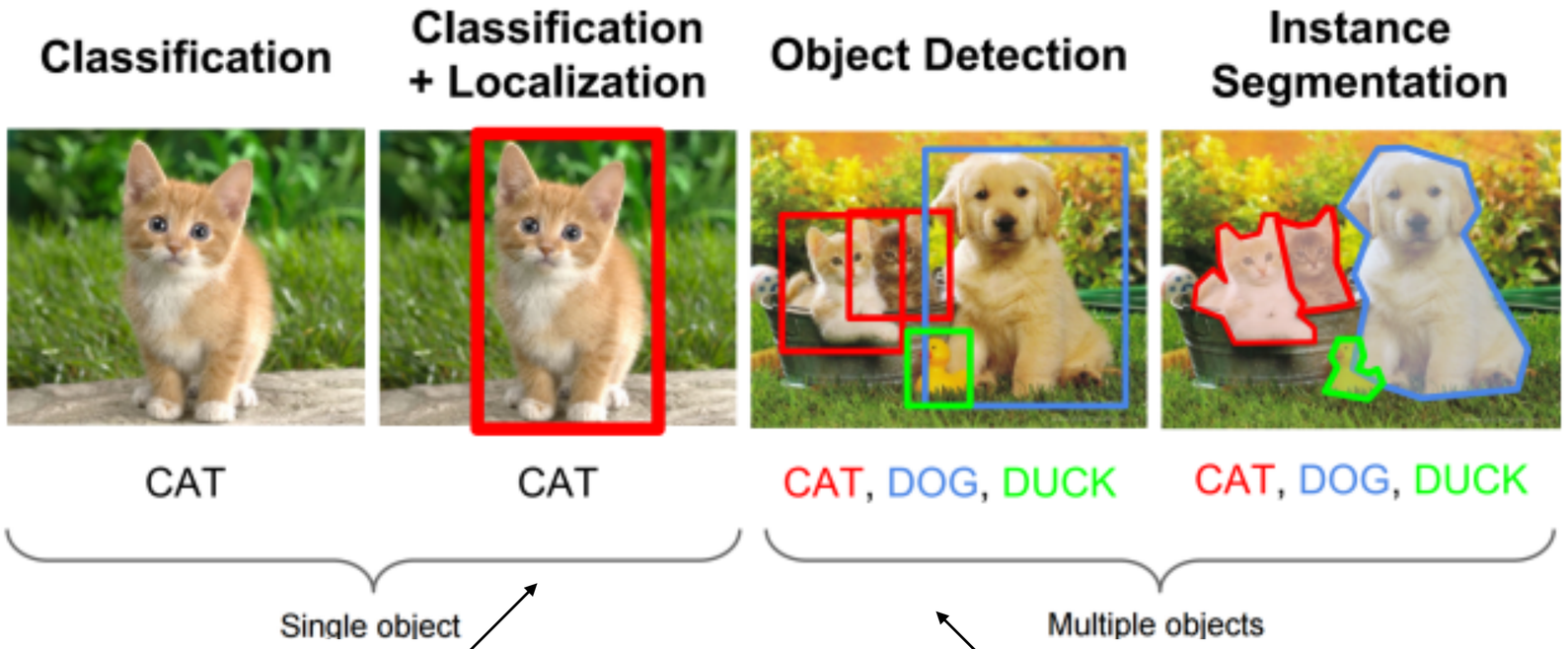


Figure 1

*https://github.com/XiSHEN0220/RANSAC-Flow

Object Detection



*

Locate one instance of an object category

In “object detection” we focus on locating all instances of a category in a given image

*https://www.themtank.org/a-year-in-computer-vision

Object Detection- SSD

- Object Detectors can be divided into two categories.
  - Two Stage Detectors:- High localization and object recognition accuracy but not end-to-end trainable. (Faster RCNN)
  - One Stage Detectors:- High inference Speed and end-to-end trainability.*

- We choose SSD300* because of its speed and object recognition accuracy.*
- SSD approach is based on a feed-forward convolutional network that produces a fixed size collection of bounding boxes and scores for the presence of object class instance in those boxes, followed by a non-maximum suppression step to produce the final detections.
- SSD300 has total 8732 boxes.
- SSD loss function - Localization Loss(L1 Loss) and Classification Loss(Categorical Cross-Entropy).

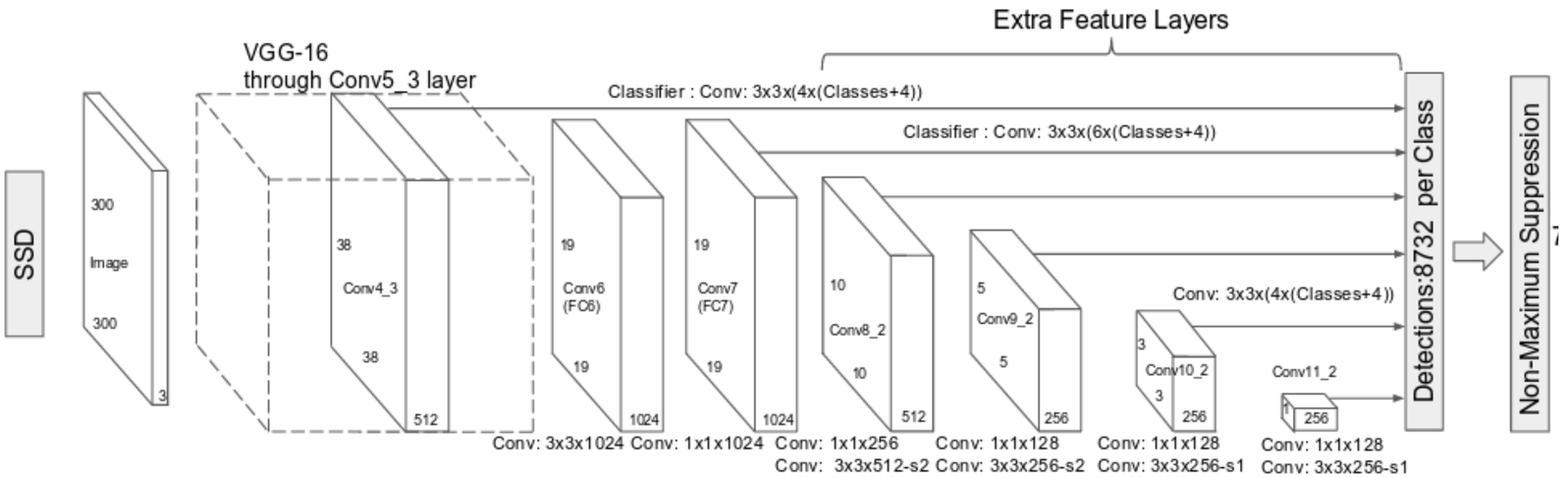
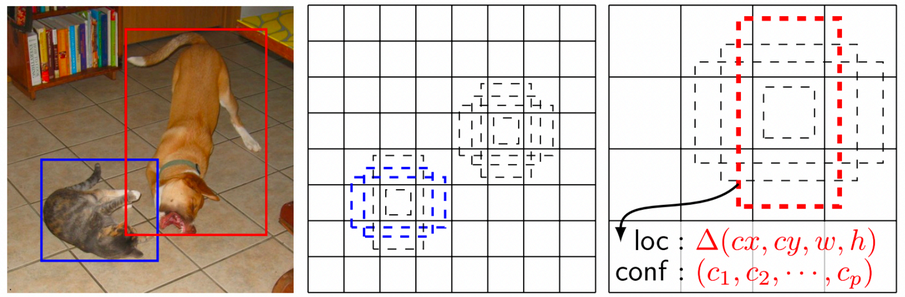


Figure 1



(a) Image with GT boxes (b) 8×8 feature map (c) 4×4 feature map

Figure 2

*SSD: Single Shot MultiBox Detector, https://arxiv.org/abs/1512.02325

Why not IOU as a Loss function?

- Bounding box regression is one of the most important fundamental components in Object Detection.
- Precise location of vignettes is crucial for DAE.
- We use IOU (Intersection over Union) to compare the similarity between two boxes (Predicted and ground).

$$IOU = \frac{\text{area of overlap}}{\text{area of union}} =$$

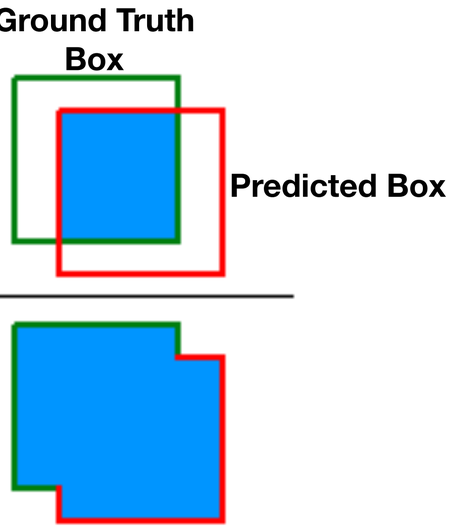


Figure 1: IOU definition

- The metric is IOU but the loss is l_n norms.
- But there is no correlation between minimising l_n norms and improving IOU values.*

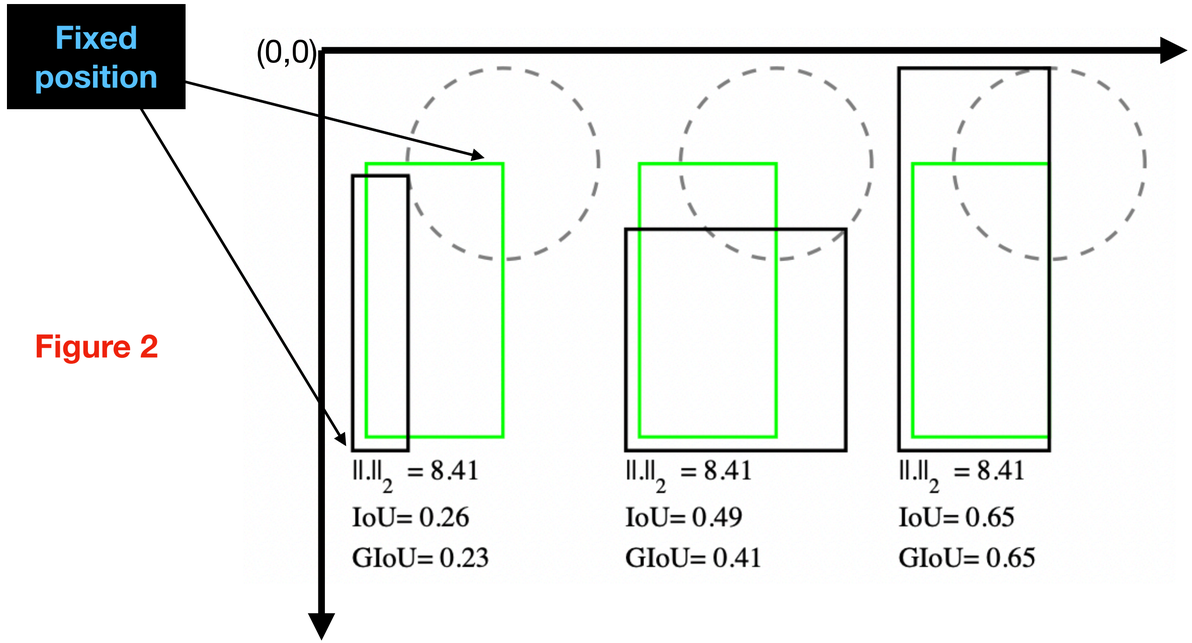


Figure 2

*Generalized Intersection over Union: A Metric and A Loss for Bounding Box Regression, Hamid Rezatofighi et al

IOU Loss and GIOU Loss

- For two boxes, G (ground truth) and P (predicted box),
  $L_{Iou} = 1 - \frac{|G \cap P|}{|G \cup P|}$.

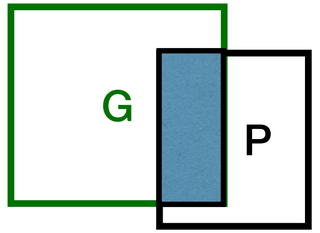


Figure 1: Overlap. $IoU \neq 0$

- If two objects doesn't overlap, IoU doesn't tell how far the two shapes are from each other.
- L_{Iou} has a disadvantage. If $|G \cap P|$=0, $IoU(G, P) = 0$. In this case, gradient is zero and can't be optimised.

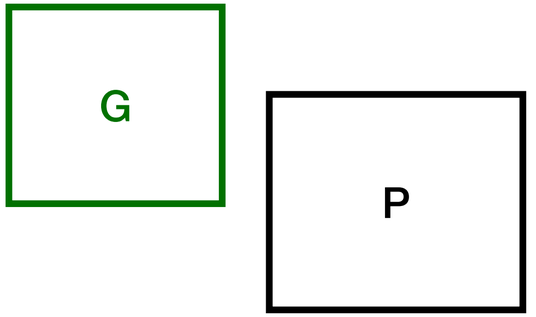


Figure 2: No Overlap. $IoU = 0$

- For two boxes G (ground truth) and P (predicted box),
  $GIou = IoU - \frac{|C \backslash (G \cup P)|}{|C|}$, where C is the smallest enclosing box of G and P.
- $L_{GIoU} = 1 - GIoU$

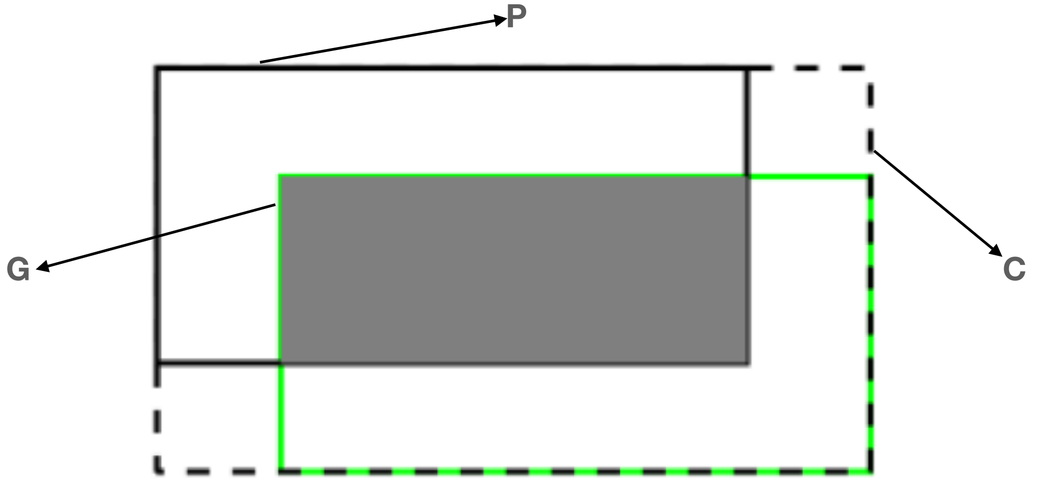
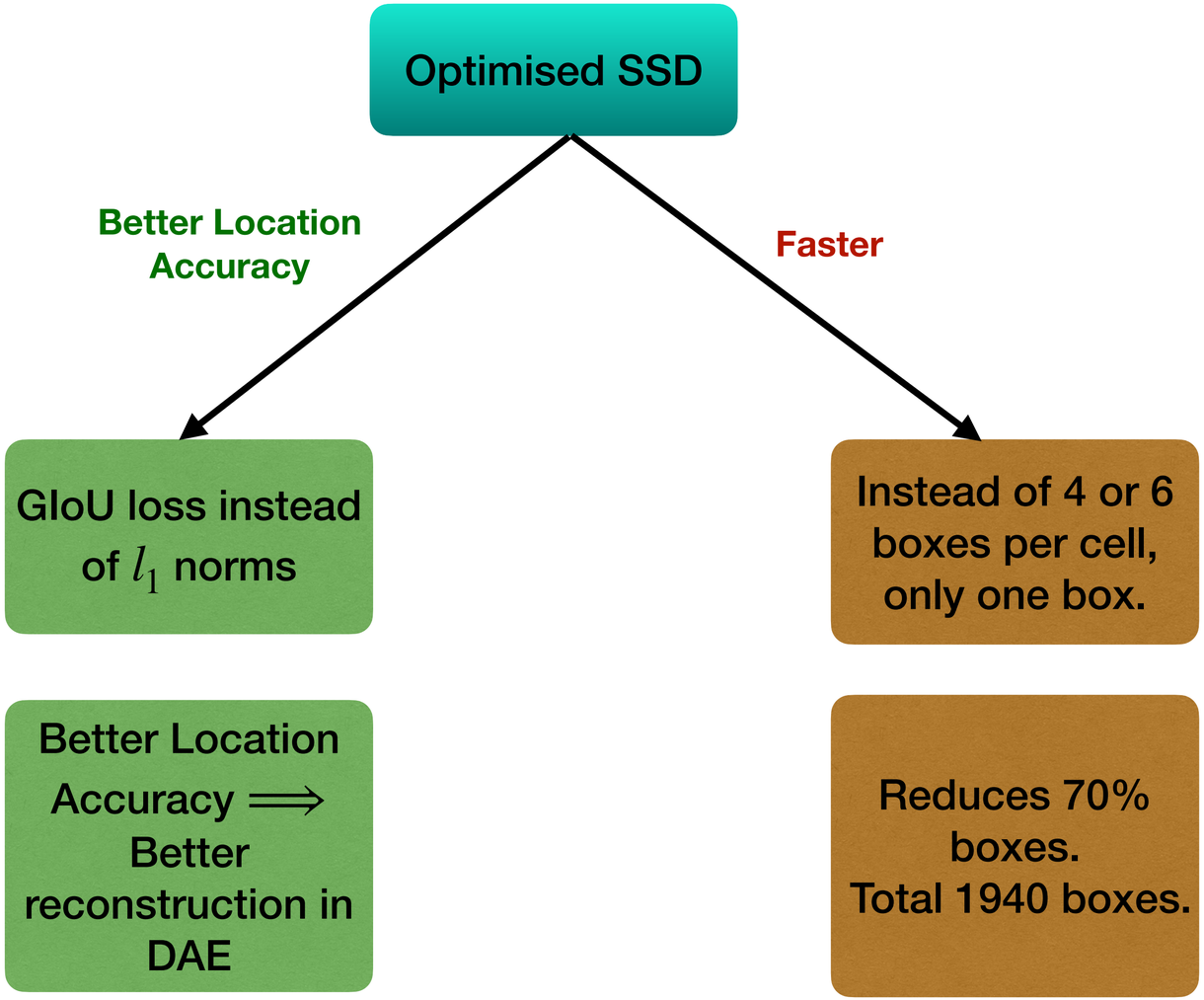


Figure 3: GIOU

Optimised SSD



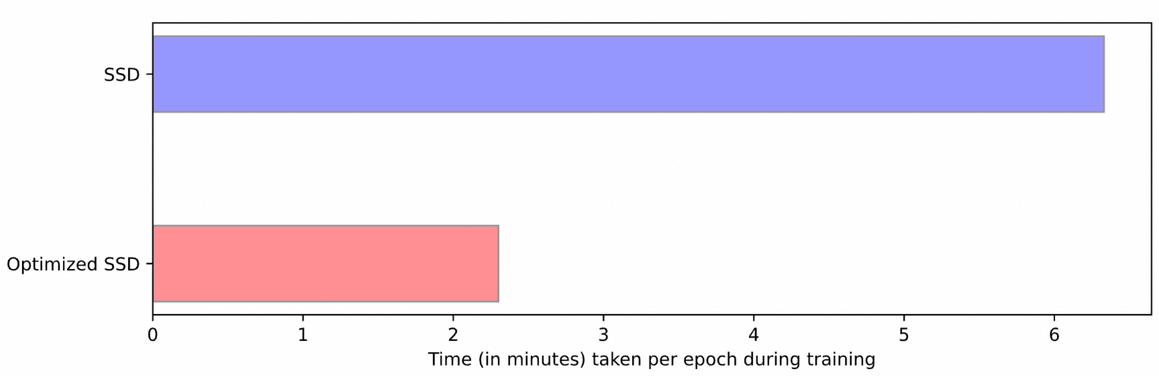
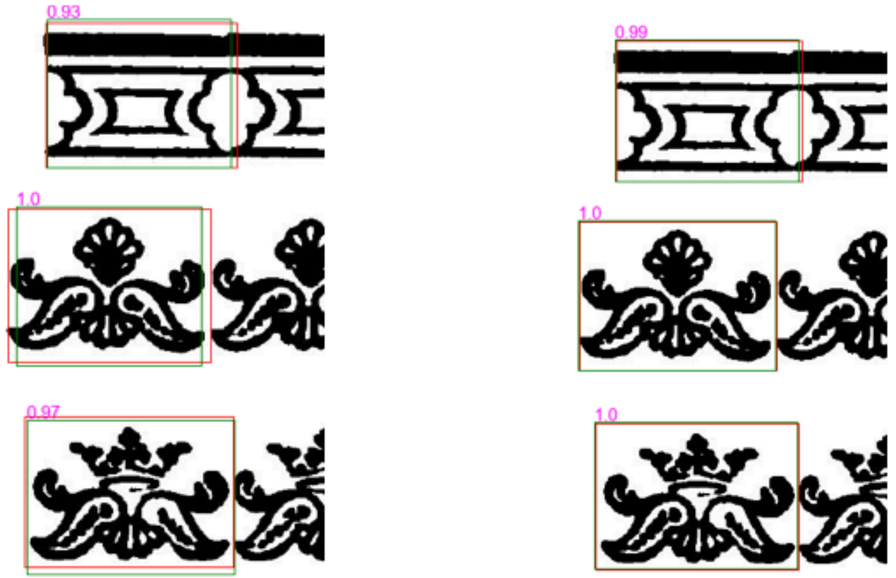


Figure 1: Time for every epoch.



L1 Loss

GIOU Loss

Figure 2: Results of Optimised SSD. Left L1 loss and Right Giou Loss

Autoencoder

1. Autoencoders are an unsupervised learning technique in which neural networks are trained for the task of representation learning.
2. The network can be viewed as consisting of two parts: an **encoder** function and a **decoder** function.*

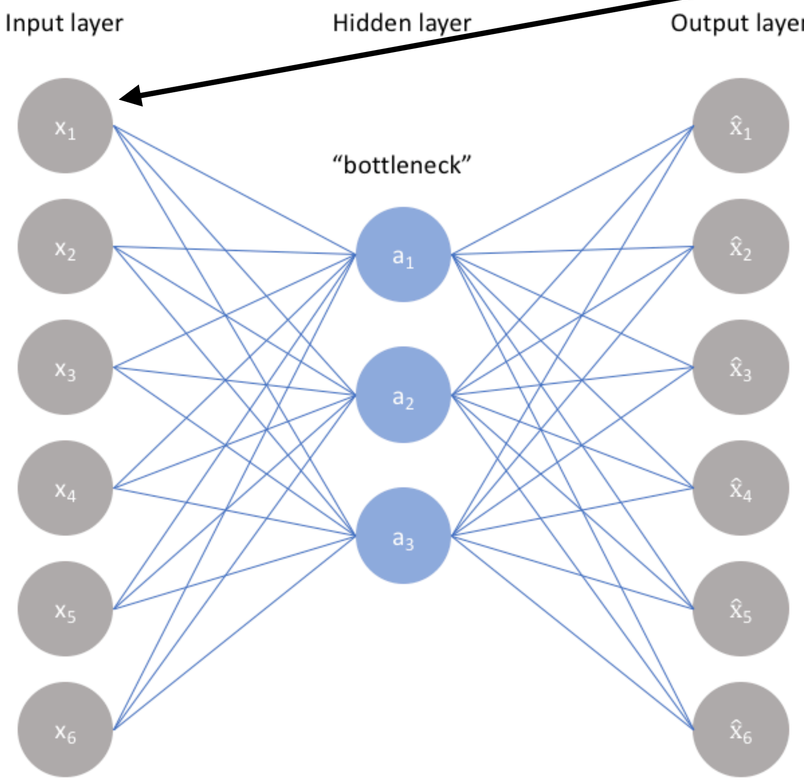


Figure 1

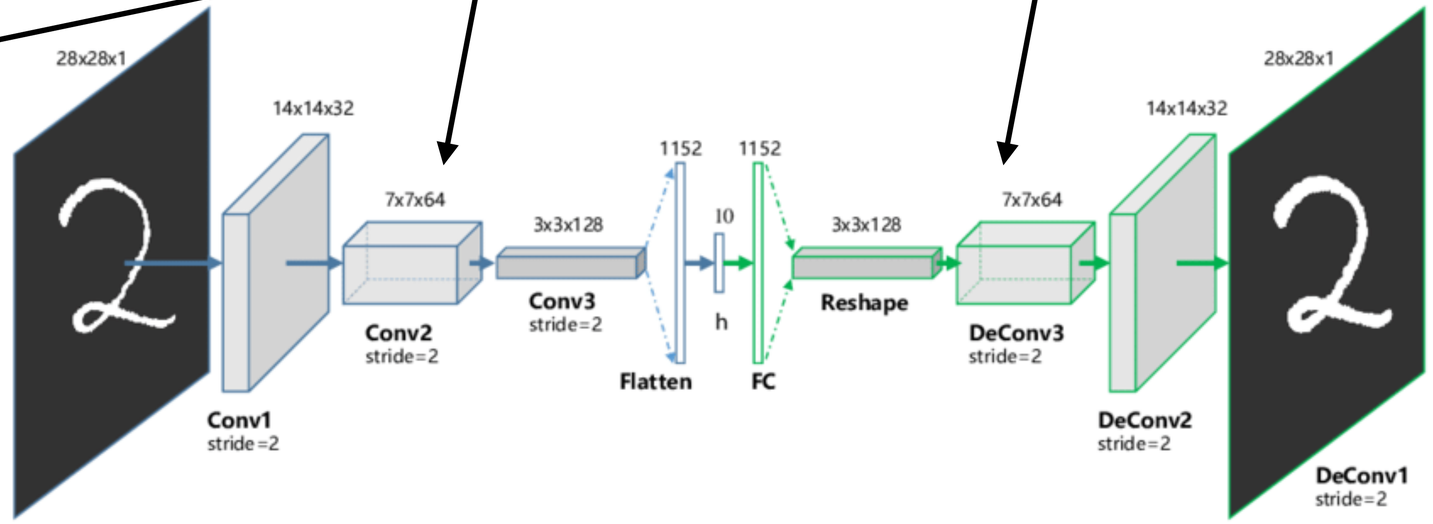


Figure 2: Convolutional AutoEncoders

*https://www.jeremyjordan.me/autoencoders/

Detector-Encoder AutoEncoder

Encoder

- In Autoencoder the encoder learns to encode the image using which decoder reconstructs the image.
- Our goal is to detect the vignettes which are anomaly in an ornament in an self-supervised manner.
- Using Autoencoder we can detect whether an ornament has anomaly or not but it will create a blurry image.
- Using Detector, we can detect the vignettes that don't have anomaly but it can't be self-supervised.

- To address this issue, we replaced the encoder with SSD.
- SSD is pre-trained with the catalogue of vignettes and if in an ornament, a vignette has an anomaly, it won't be detected.
- SSD has several feature maps for the purpose of detection. All these encode information about the image as well as their spatial information.

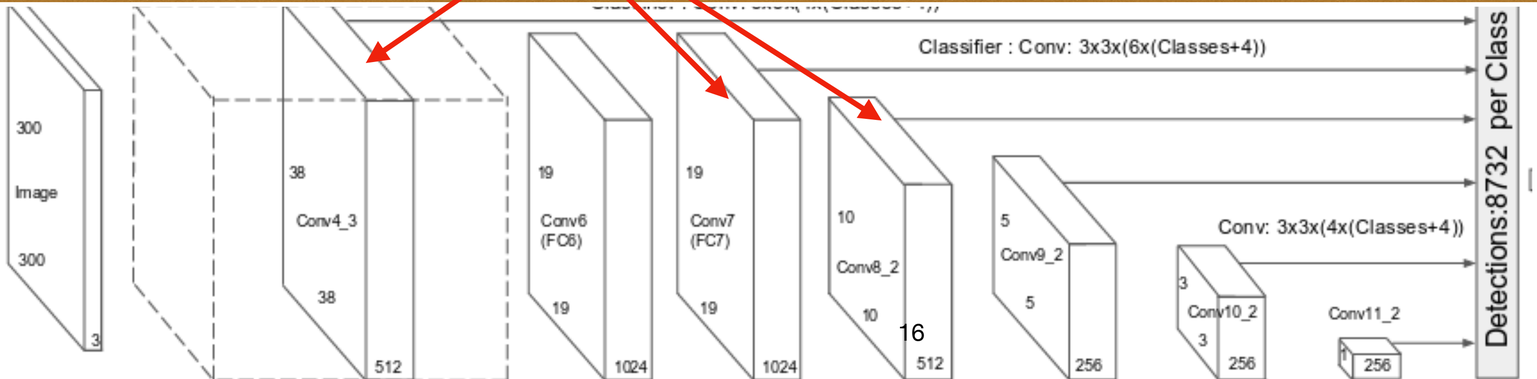


Figure 1

Detector-Encoder AutoEncoder

Encoder

- In each of the feature maps, the detected boxes specify the position of the object in original image.
- The spatial information is the detected boxes which we call ROI(Region of Interest).
- But the ROI's are of different shape and aspect ratios.
- So we use RoIAlignment*, first introduced in Mask RCNN to make all the ROI's of the same shape.
- This is the encoded information about the vignettes which we can use.

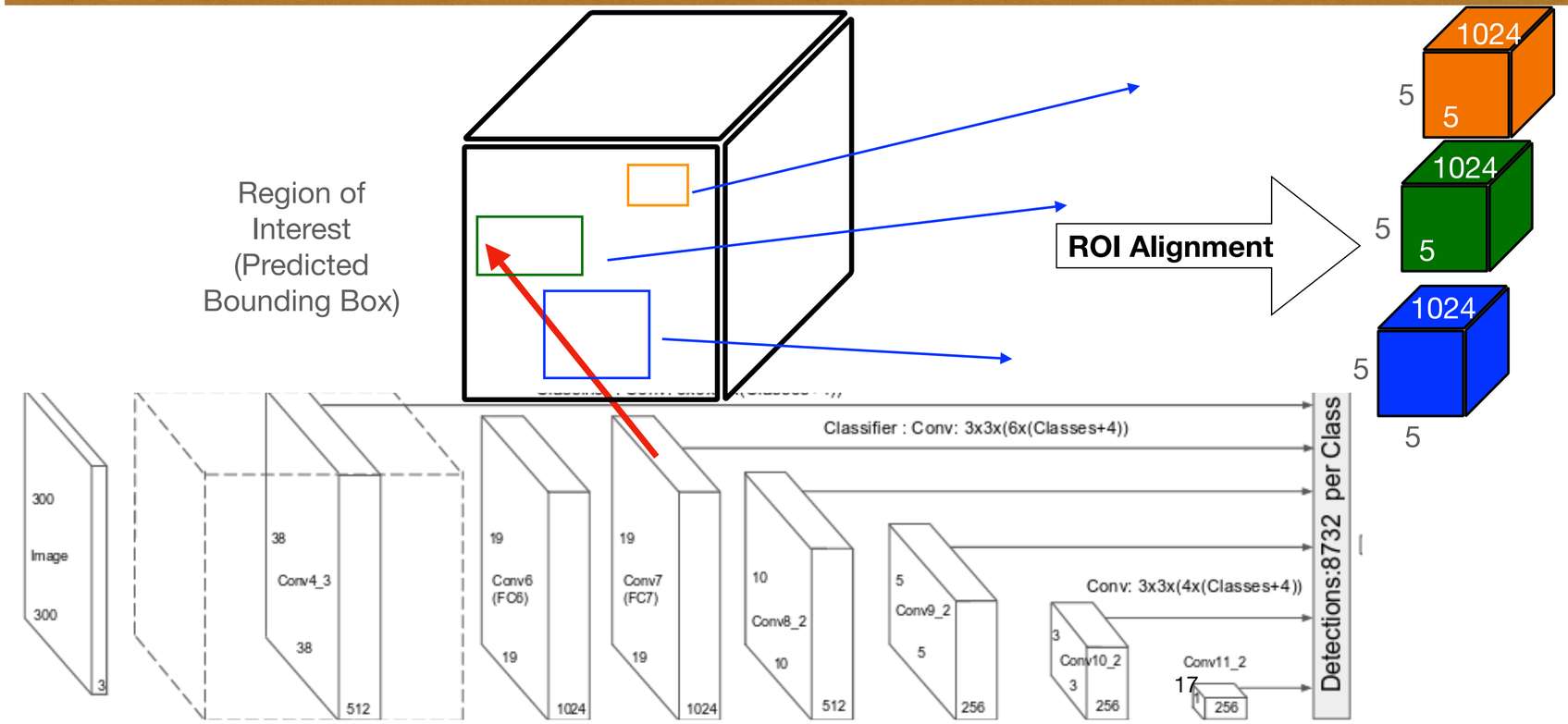


Figure 1: ROI Alignment

*Mask R-CNN, Kaiming,Georgia et al.

Detector-Encoder AutoEncoder

Decoder

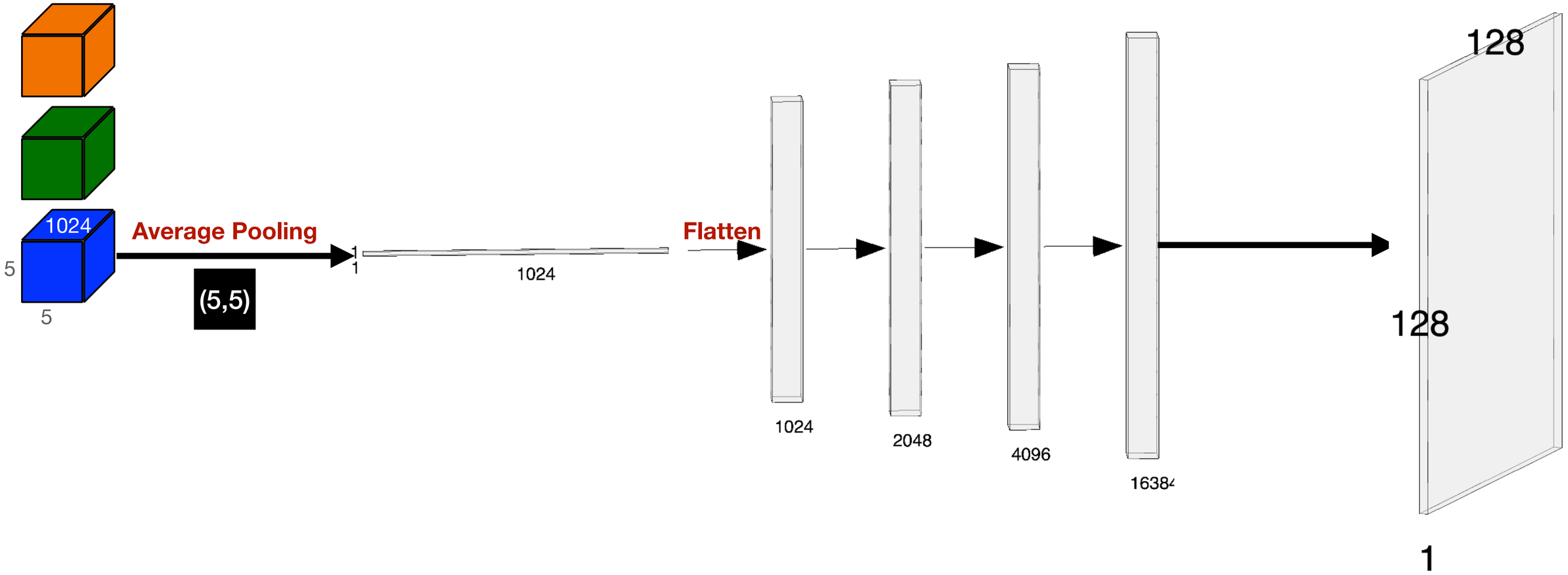
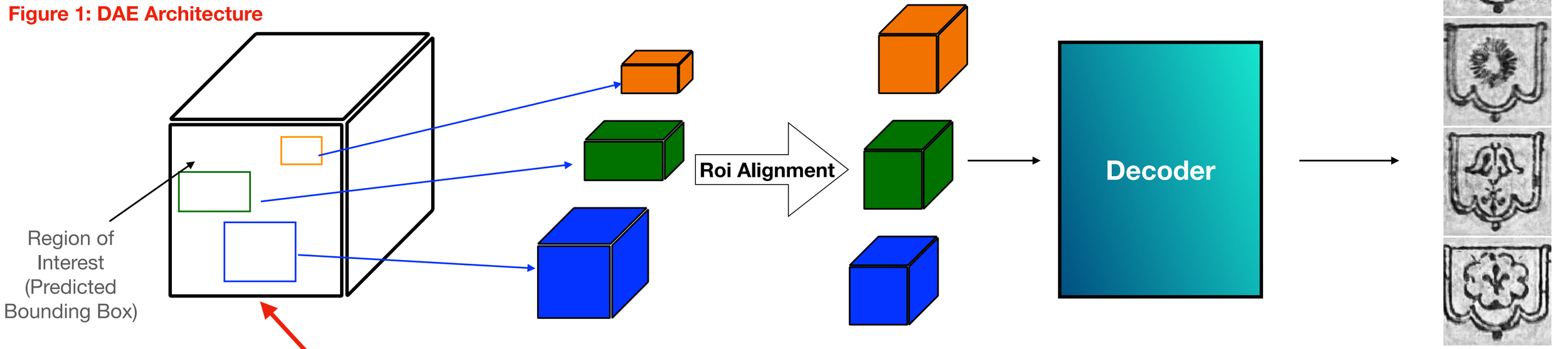
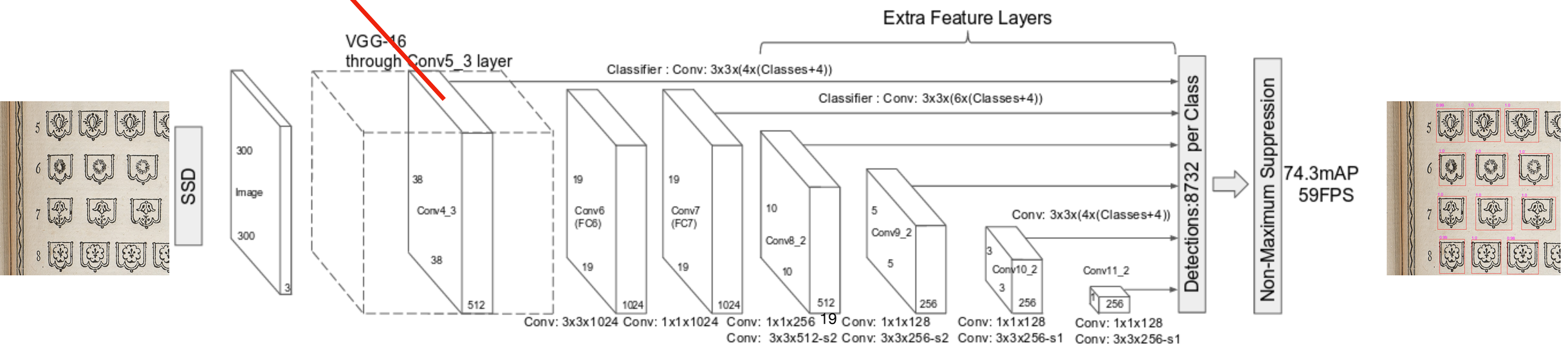


Figure 1: Decoder

Detector-Encoder Decoder Architecture

Figure 1: DAE Architecture





Results -1

Figure 1: Example 1

Results-2

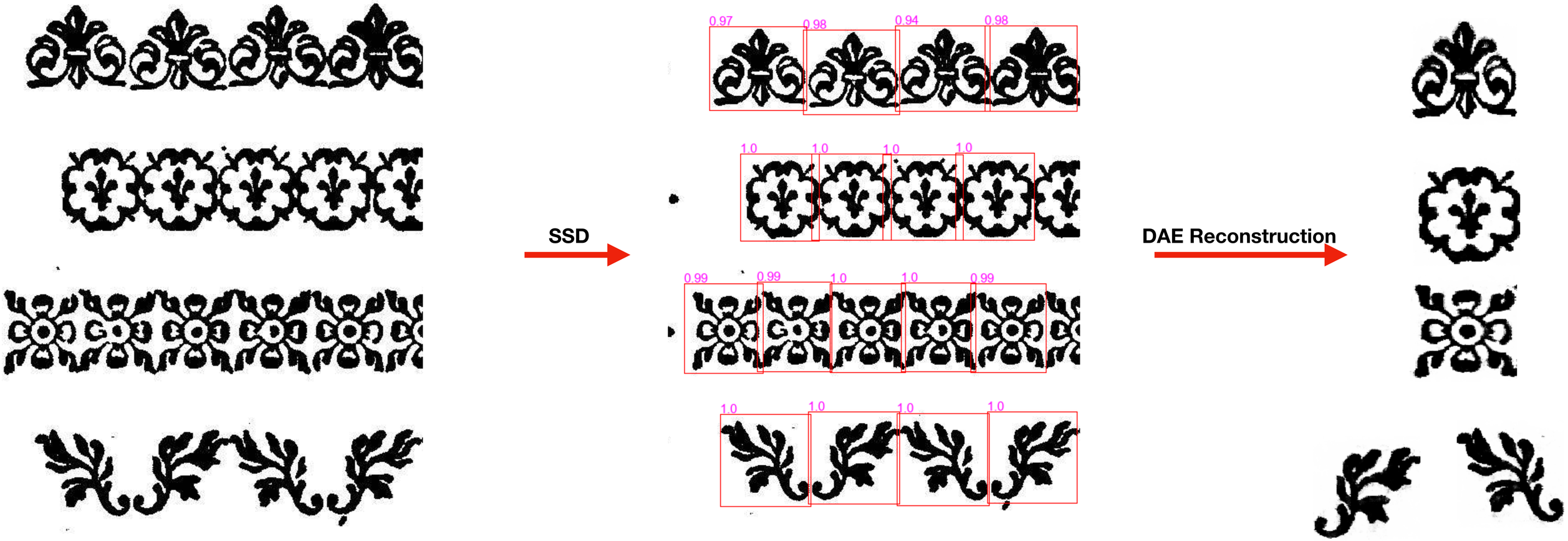


Figure 1: Example 2

Results-3

SSD

DAE Reconstruction

Figure 1: Example 3

Results-4

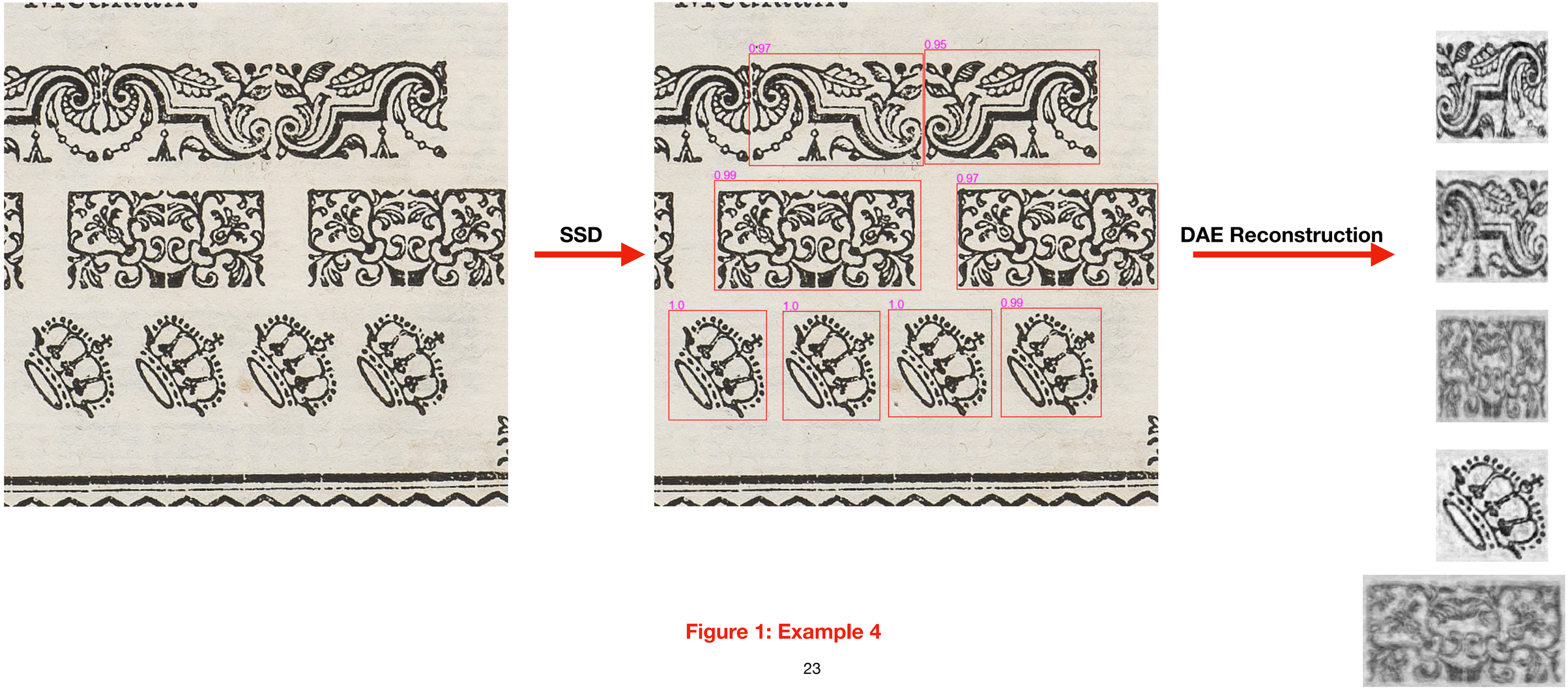


Figure 1: Example 4

Future Work

Self-Supervising DAE

- Remove the detection part and train DAE in self-supervised way.
- Faster AutoLabelMe.

Thank You