Face Detection on Pre-modern Japanese Artworks for semi-automatic annotation

Alexis Mermet, EPFL

Collaborators and datasets

Asanobu Kitamoto, ROI-DS Center for Open Date in the Humanities, NII

Chikahiko Suzuki, ROI-DS Center for Open Date in the Humanities, NII

Akira Takagishi, University of Tokyo

“KaoKore Dataset”, collected by CODH from multiple organizations

"Kouhon Dataset", owned by Shojo-Kouji

“mmdetection library”, open source object detection toolbox developed by Multimedia Laboratory, CUHK.

Plan of the presentation

1. Introduction to the research.
2. The KaoKore collection.
3. Deep Learning Architectures.
4. Evaluation metrics.
5. Image Patching method.
6. Experiments and results
7. Labelling a new collection: the Kouhon collection.
8. Conclusion

Introduction

GOAL: propose an automated face detection on pre-modern Japanese artworks to reduce the time needed by art historians to fully annotated a new collection.

Source: KaoKore collection

Introduction

SOLUTION
detector co
Image Patch

+ Effici
  and r
+ Time

Source: “KaoKore collection”

The KaoKore collection

Source: KaoKore collection

https://github.com/rois-codh/kaokore

The KaoKore dataset consists of:

- 8573 high quality annotated faces from pre-modern Japanese picture scrolls and picture books.
- These faces were manually cropped from 1470 original drawings.

About the collection:

- SMALL!
- Contains only already cropped faces
- The data is not uniformly distributed.

The KaoKore collection (2)

To use the KaoKore collection for face detection → transform the original data into full artworks

Source: “KaoKore collection”

The KaoKore collection (3)

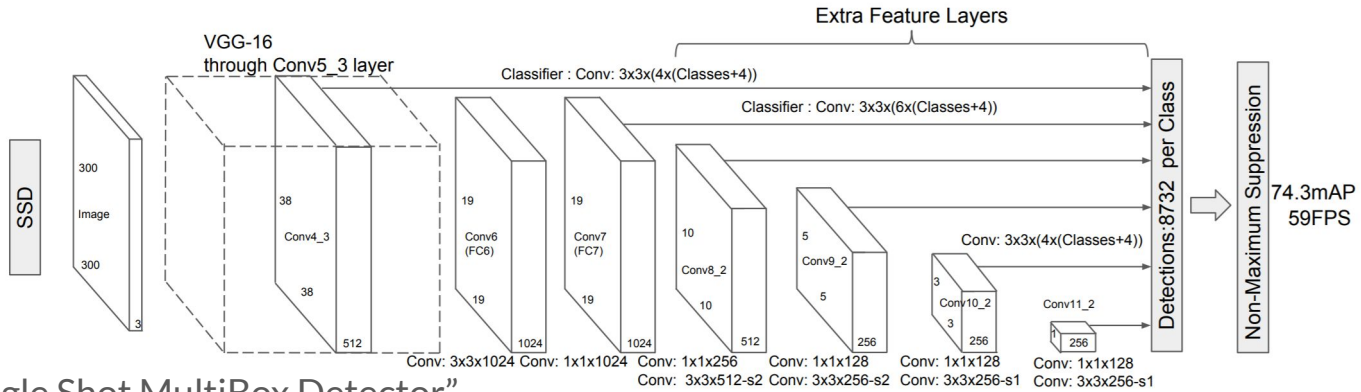
Since the data is unbalanced: Find ways to correctly split the data to train and test DL face detectors.

1. Random split: → not good enough because of data unbalanced.
2. Inter-books split: → Realistic setup allowing to evaluate transfer of face detection methods.
3. Intra-books split: → Allows to span all the style present in the collection and help to have a more generalized model

DL face detector architectures

For this work, we used 3 different Deep Learning backbones:

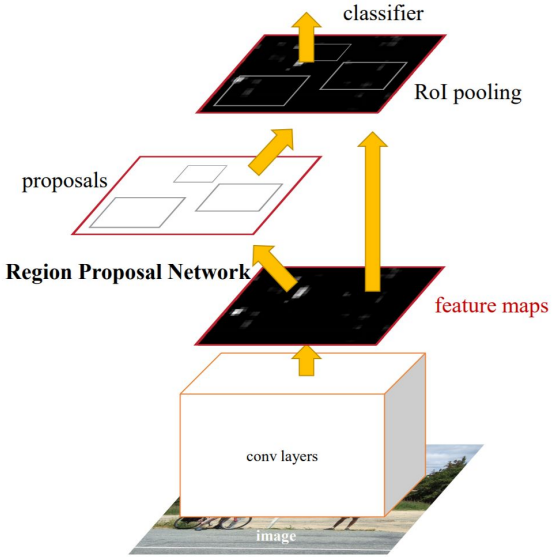
1. Single Shot Detector (SSD) with input size 300 (called SSD300).



Source: "SSD: Single Shot MultiBox Detector"

DL face detector architectures(2)

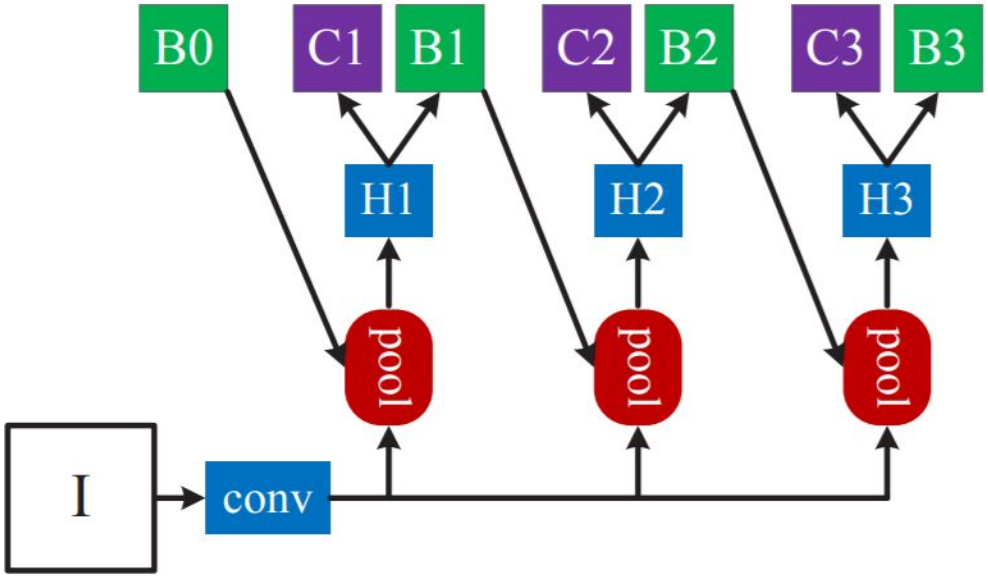
2. Faster R-CNN (Region with CNN features).



Source: “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”

DL face detector architectures(3)

3. Cascade R-CNN.



Source: "Cascade R-CNN: Delving into High Quality Object Detection"

Image Patching approach

Data originates from drawings, where color palettes, shapes, styles and materials are variable from one book/scroll to another.

→ High resolution artworks!

Before entering any detection backbone → artworks are reshaped to be passed to the input layer of the detectors.

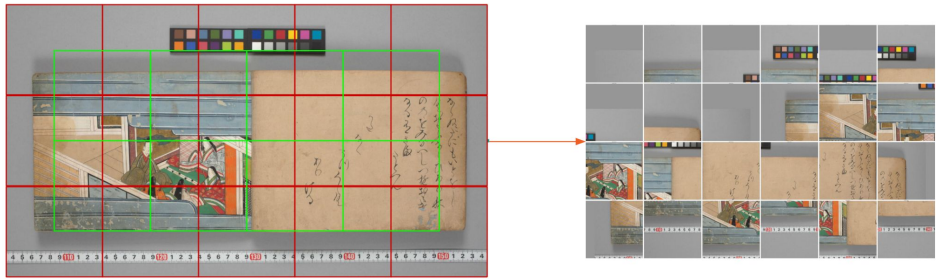
→ Resising induces details loss!

GOAL: allow images to be passed to a detection architecture while avoiding as much as possible the resizing process.

Image Patching approach (2)

Data: List of Japanese artworks I, and patching size s
Result: Bounding boxes for the faces contained in each artwork
for *image i in I* **do**
  Pad image i according to patching size s;
  Crop all patches of size s from image i;
  Add these patches to the list of patches, P;
  for *patch p in P* **do**
    Detect faces in patch p;
    Add found faces to the list of found faces F;
  end
  Translate faces in F into i's coordinate system;
  Non-Maximum Suppression to remove overlapping bounding boxes;
end

Algorithm 1: Image Patching approach on never seen artworks



Source: KaoKore collection

Evaluation Metrics

$$\text{recall} = \frac{tp}{(tp + fn)}$$

$$\text{precision} = \frac{tp}{(tp + fp)}$$

$$mAP = \int_0^1 p(r)dr$$

$$p_{\text{interp}}(r) = \max_{\tilde{r} > r} p(\tilde{r}) \text{ and } mAP = \sum_n (r_{n+1} - r_n) p_{\text{interp}}(r_{n+1})$$

$$\text{IoU} = A \cap B / A \cup B$$

Experiments: setup

For the following experiments, we fixed the thresholds to the following values:

- ϕ = 0.5, prediction score threshold.
- γ = 0.3, Jaccard Overlap threshold used by NMS to find overlapping boxes.

- Θ = 0.4 for most experiments. Will be fine tuned later on. → affects detectors quality.

All the experiments were run on the same Intra-Books split of the KaoKore collection for comparabilty.

Our detectors were all pretrained on the ImageNet dataset to speed up the training.

Most of the experiments were conducted using the mmdetection library that implements an easy framework to setup multiple PyTorch object detection networks.

Experiments: setup (2)

Our backbones were implemented such that they had the following input dimensions:

1. SSD300: 300*300.
2. Faster R-CNN: 1333*800.
3. Cascade R-CNN: 1333*800.

Training was done using an SGD optimizer with the following parameters:

1. SSD300: 24 epochs, learning rate of 1e-4 that changes at the 16th and 22nd epochs. Weight decay of 5e-4 and momentum of 0.9
2. Faster R-CNN: 12 epochs, learning rate of 2e-3 that changes at the 8th and 11th epochs. Weight decay of 5e-4 and momentum of 0.9
3. Cascade R-CNN: Same as faster R-CNN

Experiments: study of the KaoKore collection

In this experiment, we wanted to have a look at how the KaoKore's data could be used in a face detection setup.

We train multiple face detectors with the Image Patching method on each of the different training/testing splits of the data. For each split setups, we made 5 independent runs with different training and testing sets.

Split type	Random	Inter-books	Intra-books
MaP (%)	71.2	70.3	73.5

Experiments: study of the KaoKore collection (2)

1. Our approach seems to generalize well on our data-set.
2. Transfer between already seen style, materials, color palettes and shapes on never seen ones is possible inside of our data-set.
3. Our model, when able to train on any available type of books (any type of images colors, shape, style, materials, etc...), is able to well detect faces on any images of our data-set.

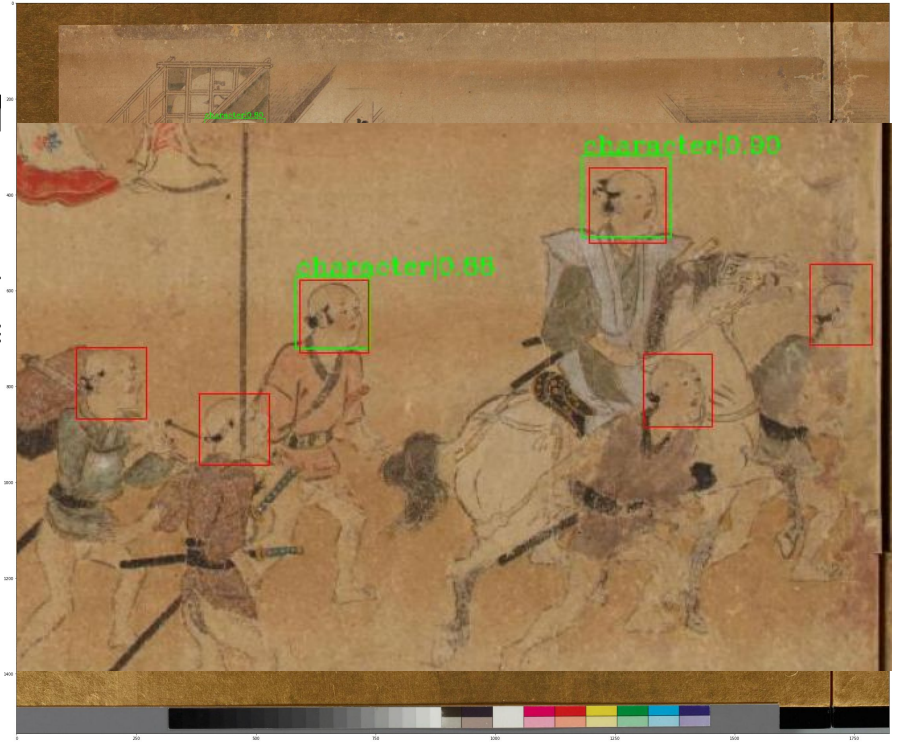
Experiments: architectures and Image Patching

architecture	patch size	mAP (%)	Recall (%)
SSD300	no patches	41.1	71.2
	600*600	72.7	83.1
Faster R-CNN	no patches	79.3	89.7
	600*600	78.6	89.9
	1333*800	80.2	90.4
Cascade R-CNN	no patches	79.0	87.6
	600*600	79.0	89.9
	1333*800	80.3	90.1

From these results, we can see that the Image Patching methods allows improvement in both mAP score and Recall.

Also, the Image Patching technique benefits the detection quality as long as the patch size is adapted to the input layer of the backbones.

Source: “KaoKore collection”



Source: “KaoKore collection”

Experiments: tuning the IoU threshold Θ

IoU threshold	0.4	0.5	0.6	0.7	No patching
mAP (%)	80.2	82.9	83.2	82.7	79.0
recall (%)	90.4	91.1	90.9	90.5	89.7

The IoU threshold Θ, that determines the positive samples, influences a Faster R-CNN model achieves its detection quality.

Face detection on a new collection

the Yugyo Shounin Engi-Emaki Shojo-Kouji Kouhon collection, that we will abridged to 'Kouhon', is a collection of picture scrolls from the 14th to the 17th centuries composed of 346 pages, split into 10 volumes.

We used the following detector:

→ Faster R-CNN with Image Patching, size 1333*800, $\Theta = 0.5$, $\phi = 0.5$ and $\gamma = 0.3$. Trained on the KaoKore collection with Intra-books split.

Source: “Kouhon collection”

Source: “Kouhon collection”

Evaluating our results on the Kouhon dataset

After having art historians check our detection results and fully label the Kouhon collection we obtained the following results:

restriction on true positives	case 1	case 2	case 3
Precision	85.99%	85.94%	84.17%
Recall	67.07%	67.02%	64.65%

Evaluating our results on the Kouhon dataset (2)

Cost wise: $\text{cost} = t_{\text{acc}} * N_{\text{cor}} + t_{\text{mod}} * N_{\text{nearly-cor}} + t_{\text{rem}} * N_{\text{incorr}} + t_{\text{add}} * N_{\text{miss}}$

Considering only t_{add} as non null,

1 - recall = ⅓ of the original cost (cost without machine).

Conclusion

- We have trained an efficient face detector on the KaoKore collection
- This detector allowed to label a new collection, reducing the cost of detection to ⅓ of its original cost → can transfer on other Japanese collections.
- We only tackled one part of the annotation process:
    - Detection
    - Labelling: we tested a ResNext labeller for status yielding an accuracy of 85%

In the future: fully automated the annotations process
both the detection and the classification could be done simultaneously or not.
→ training loop: use new currated collection to retrain the detector.