Siegel NDH: Unsupervised Clustering

Katharina Prasse

August - December 2021

1 General Information

Task: unsupervised clustering of image files according to seal form, material and content.
Goal: support individuals with historical interest and/or expertise in working with seals by categorizing the the material, colour and content displayed in the seal.
Results: clustering according to form and material not achievable in a cost-effective manner. Focus on clustering according to content. Most successful clustering produced 35 new labels for the category "crowned seals" and discovered all seals from the training set. The employed model was a finetuned densenet, re-trained for 200 epochs.
Data: historical seals [Archiv Nordhausen]
Employed Programming languages: Python
Git Repository: KathPra (K. Prasse)
Supported by: Prof. H. Kümper & Prof. M. Keuper

1.1 Dataset Description

Data: seal dataset of Archiv Nordhausen (Harz)

- Form [3]: round, oval, ogival
- Material [6]: lead, wax (green, red, yellow), lacquer (red), paper
- Symbols [about 20]: eagle, crowned coat of arms, full coat of arms, city wall, Mary, tools, farm animals, et al.

Number of images: about 7.500 files
Class distribution: unknown, many regional coats of arms

2 Research

2.1 Preparation

Preprocessing

- Remove all images which display multiple different seals. Identified by naming convention "..0.00.jpg" (177 images) and "..1.00.jpg (99 images): Total 276.
- Remove all non-seal images e.g. images showing ink prints of seal stamps and seal stamps (identified by folder affiliation [folder 4: StadtA NDH 9.7.397.jpg - StadtA NDH 9.7.441.jpg - 45 images; folder 22: StadtA NDH 9.7.1368.01.jpg - StadtA NDH 9.7.1370.0.03.jpg - 28 images]).

Transformations

Circular Hough Transformation identifies circular shapes within the image by looking at object contours (edges). The most prominent circles are extracted and the first, central within bounds, circle is kept and all pixel outside this shape are switched to black.
The minimum and maximum radius for circles are determined through experiments. The image resolution is decreased by a factor of 150 to retain only dominant structures in the images. The contours of these structures e.g. seal shape, ruler are then extracted using the canny edge detector and used for the circular hough transformation.
The algorithm returns in the majority of cases a perfect or near perfect cut-out of the seal. The improvement in clustering performance overcompensates for the loss of few seal images due to imperfect cut-outs.

Recommendations for future seal fotography

- Photograph only one seal at a time. When covering neighbouring seals, cover them entirely.
- Keep all non-seal content constant between images (e.g. color scale, ruler, background).
- Define clear nomenclature for image files to support easy access to seal images.

2.2 Experiments

All clusters have been inspected manually by looking and arising questions were answered by a historian.

Virtual environment: Python 3.7, SciPy, scikit-learn, PyTorch, CUDA 10.2

2.2.1 Apply pre-trained model to extract features from seal data and cluster accordingly [Inception net]

The model inception net v3 [6] is chosen due to its good performance in practice.

Clustering algorithm: K-Means with K ranging from 2 to 40
Results: algorithm returned clusters w/ similar seal images especially for K values between 5 and 15. The clusters were not clean and the algorithm especially struggeled with different motives sharing similar details e.g. eagle wings and mantling of full coat of arms (Helmdecke eines Vollwappens).

Clustering algorithm: DBSCAN with epsilon (0.027 - 0.9) and min_samples (2-10)
Results: algorithm returned one large cluster and many noise points. When increasing epsilon, the number of clusters increases, however, the majority of seal images remain in one cluster whereas other clusters do not extend beyond the size of 30 data points.

2.2.2 Finetune pre-trained model to extract features from seal data and cluster accordingly [Inception Net]

The model inception net v3 is chosen to have comparability of clustering performance before and after finetuning.

Training: model training with RandomRotations(0°-360°) to make the model rotation invariant. Pre-training already made the model translation (moving of object) invariant. No validation set is used and accuracy is used to measure the trainings effect. The model was trained with 4 classes: eagle, crowned seal, random seal which does not contain eagle or crowned seal, malfunctioning hough transformation (no seal is contained). Varying batch sizes are used (5-16) and several number of epochs (50 - 200) have been assessed.

Clustering algorithm: K-Means with K ranging from 2 to 40
Results: algorithm returned clusters w/ similar seal images especially for K values between 5 and 15. The clusters were not clean and the algorithm especially struggeled with different motives sharing similar details e.g. eagle wings and "Helmdecke of Vollwappen". The clustering improved compared to the not finetuned net, however, the cluster quality is not yet satisfactory.

Clustering algorithm: DBSCAN with epsilon (0.027 - 0.9) and min_samples (2-10)
Results: algorithm returned clusters w/ similar seal images especially for low epsilon values. When the model was trained with more epochs, then the extracted features were closer to each other, however the clustering performance did not increase. Interestingly, when epsilon was chosen small enough, it was possible to extract somewhat clean clusters of seals containing coats of arms, both crowned and full ones. All remaining seal images would be clustered as noise points. The performance increased incomparison to the pre-trained net, however, it is not yet satisfactory.

2.2.3 Finetune pre-trained model to extract features from seal data and cluster accordingly [Dense Net]

The simpler Densenet [3] is chosen to assess whether it can better grasp the seal images contents.

Training: same as Inception net.

Clustering algorithm: K-Means with K ranging from 2 to 40
Results: algorithm returned clusters w/ similar seal images especially for K values between 5 and 15. The clusters were quite clean for crowned wappen and clusters could be extracted. Relatively early, already starting from a K-value of 15, some cluster were of size 1. The clustering algorithm captured not only the content of the seal, but also its conservation status. The algorithm struggles to create somewhat clean clusters of eagles even after 200 epochs of training. Lastly, the algorithm generated clusters of medieval seals which all contained a circumscription. Since this cluster was too general, it did not add value. The clustering performance improved greatly compared to the inception net

Clustering algorithm: DBSCAN with epsilon (0.027 - 0.9) and min_samples (2-10)
Results: the clustering performance hardly increased compared to Inception net. Density-based clustering of seal features does not seem produce meaningful results.

2.3 Next steps

The clustering performance is assumed to improve further when trained with more classes. Furthermore, an iterative approach is proposed: all images clustered correctly are included into further rounds of training. Important: all training classes should contain the same number of images.

3 Relevant literature

3.1 Deep clustering for unsupervised learning of visual features [1]

Authors: Caron, Bojanowski, Joulin, Douze [Facebook]
Model name: DeepCluster
Clustering Algorithm: K-Means
Neuronal network architecture: AlexNet, VGG (better performance)
Advantages: end-to-end trainable, scales well to large amount of data

Linear classification (discriminative):

- Avoid trivial solution [empty clusters]: reassign centroid to the location of a random other centroid and add a small, random perturbation. Reassigning data points among both centroids.
- Avoid trivial solution [parametrization]: weight input to loss function by inverse of cluster size.
- Over-segmentation improves clustering performance
- Benefits from long training time
- State of the art for classification, recognition and segmentation [2018]
- Robust to image class distribution changes (unbalanced classes)
- Robust to changes of the network architecture
- Evaluations metric: normalized Mutual Information (NMI)

Input augmentation:

- Sobel filter: removes color and increases image contrast
- Center crop
- RandomHorizontalFlip
- RandomResizedCrop

Training:

- W/o drop-out
- Constant step size / learning rate
- l2 penalization
- Momentum 0.9
- Mini-batch of 256 images
- Clustering: features are PCA reduced to 256 dimensions, whitening transformed, and are l2-normalized
- Hyperparameter selection based on object classification task
- 500 epochs, with clustering in each epoch
- Pre-training: Image Net
- Trick: Batch normalization

3.1.1 Reconstruction of results of Deep Cluster research paper using original code and data

Virtual enviroment: Python 3.6, SciPy, scikit-learn, PyTorch, CUDA 10.2, Faiss [Facebook clustering package]
Data: Image Net 2012 [4] (study links to torchvision, where this version of ImageNet has previously been available [2]). Data set for task 3 (Fine-grained classification of dogs) [5] used for training and testing
Number of classes: 120
Number of images: 20.580 files
Class distribution: imbalanced (~148 to 252 images per class, checked dev kit)
Number of Clusters: 480 (# of classes *4); according to paper it should have been 1200 (# of classes * 10). However, this produces an error: WARNING clustering 20580 points to 1200 centroids: please provide at least 46800 training points.
Results: reconstruction not possible due to technical constraints when using the faiss package. Versions described in original paper not accessible.

References

[1] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 132–149, 2018.

[2] FacebookResearch. Deep clustering for unsupervised learning of visual features.

[3] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[4] ImageNet. Imagenet large scale visual recognition challenge 2012 (ilsvrc2012). `https://image-net.org/challenges/LSVRC/2012/2012-downloads.php`.

[5] ImageNet. Results: Imagenet large scale visual recognition challenge 2012 (ilsvrc2012). `https://image-net.org/challenges/LSVRC/2012/results.html`.

[6] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.