Brand Product Identification with Deep Learning

Volha Leusha leusha@stanford.edu
Junhua Chen junhuac@stanford.edu
Trishia El Chemaly tchemaly@stanford.edu

Abstract

In this paper we introduce an image matching method that can be used in building prediction and recommendation models to help brands improve their ROI for performance marketing. The method is based on combining region selection and feature extraction techniques and employs CNN architecture as a backbone. We show that applying CNN architecture for feature extraction in ads results in superior performance over standard methods like SIFT, both in accuracy and inference time. Moreover, we employ multiple state-of-art object detection methods for region selection and prove that models pre-trained on existing datasets can only partially be used to successfully perform the task.

1. Contributors

Apoorva Dornadula, apoorva@viralspace.ai - provided dataset and counseling on the approach

2. Introduction

Deep learning methods have proven to be useful and have thus become very popular when it comes to improving digital advertisement. Aside from predicting and boosting ad performance, drawing correlations, improving contextual relevance, and targeting defined segments, deep learning can help brands assess and therefore improve their return on investment. In our work with Viralspace start-up, we target the latter task by building prediction and recommendation models. We aim to implement our solutions using new convolutional neural network methods.

3. Related Work

Image matching is a fundamental challenge in computer vision, used to address tasks such as object recognition, content-based image retrieval, camera calibration, and stereo correspondence. This problem can be defined as finding a measure of similarity between images and has been addressed using area based matching techniques or feature based matching techniques. Such techniques include cross correlation, least square region, simulated annealing, and others. Area based matching requires a numeric comparison of digital information in sub-areas of the images [1].
Feature detection is the process of finding particular image features at each local point of an image once the image abstraction has been calculated [2]. A critical point is that ideal features should be highly distinctive in order to be accurately matched with high probability. Scale Invariant Feature Transform (SIFT) is one of the most known feature detector. The features are created using a cascade filtering approach. In the first stage, the search, over all scales and image locations for potential interest points, is performed using difference-of-Gaussian function, followed by keypoints selection and orientation assignment. In the last step, keypoints descriptors are calculated using the local image gradient measurements. The feature creation proceeds by matching these features to a database of features (created for SKU images) using a fast nearest-neighbor algorithm. The details of this method are discussed in [16].
Speed up Robust Feature (SURF) is a faster approximation of SIFT that uses a blob detector based on the Hessian matrix to find key points and wavelet responses for orientation assignment and feature description. Features are compared only if they have same type of contrast based on the sign of the Laplacian [4, 5]. Oriented FAST and Rotated BRIEF (ORB) have been proposed as an efficient alternative for SIFT and SURF. It uses FAST to determine key points and Harris corner to find the top points [6].
Region proposal networks (RPNs) have emerged as an alternative to the traditional computationally expensive sliding window method [7]. These are fully-convolutional networks used to identify useful regions in an image by predicting object bounds and corresponding scores. They are trained to generate region proposals to be used by Fast R-CNN for detection [8]. Multiple ways of using deep networks for locating class-specific bounding boxes have been proposed [9, 10, 11, 12]. OverFeat [10] predicts box coordinates assuming a single object and using a fully-connected layer, or multiple class-specific objects using a convolutional layer. MultiBox [11, 12] generates region proposals using CNN applied on single or multiple large image crops. EdgeBoxes [13] is another famous approach using window scoring instead of segmentation. This method gen-

erates bounding boxes using edges that informatively represent an image. It is based on the hypothesis that the number of contours entirely inside a bounding box indicates the probability of an object being in the box. Candidate boxes are evaluated, and a ranked set of top-scoring proposals is returned based on an objectness score. Pre-trained Faster R-CNN model is another option for region proposal [8]. This model is composed of two blocks - region proposal network (RPN) and Faster R-CNN. RPN takes an image and outputs a set of region proposals using deep fully convolutional network. Four numbers for each of the object classes (encoding bounding-box positions) and objectness scores for each of the bounding box is generated. This regions are used as an input to the Fast R-CNN where object detection step is performed.
We suggest Edge Boxes and Faster-RCNN as main methods for region extraction.

4. Problem Statement

Given a certain ad as input to the model, we are required to identify products and product categories by referring to a catalog of around 160,000 SKU images. That is, we need to find matching SKU images that are present in the ad. Ad images are public facing ads run by a brand. The main challenge that we face is the lack of correlation between the input ads and the SKU images we have in store. This is because the data is not labeled. We therefore, approach this task with visual recognition techniques that allow us to identify matching SKU images.

5. Dataset

The dataset handed to us by Viralspace consists of (i) 3000 ad images, and (ii) an excel sheet with multiple product SKUs and links to download 160,000 corresponding images. The SKUs in the datasheet are categorized such that the categories are useful to understand the content of the provided dataset, but is not applicable for clustering the ads, namely an ad could possibly be assigned to multiple categories simultaneously. For example, looking at the third ad image in Figure 1, it is not a straightforward task to categorize the paintings on the wall as - 'fine arts' or 'wall decor'. Moreover, the difference between some SKU categories such as 'greeting cards', 'holidays', 'baby and kinds' is vague. This drives us to fail to categorize the ad images to single categories. Examples of SKU images that we have are postcards, paintings, business cards, and others. This differentiates the dataset we are tackling from other typical and well known vision datasets. However, a lot of postcards have objects on them that can be found in MSCOCO and ImageNet labels. Such objects are cats, dogs, humans, trees, and others. It is therefore potentially useful to reapply existing models.

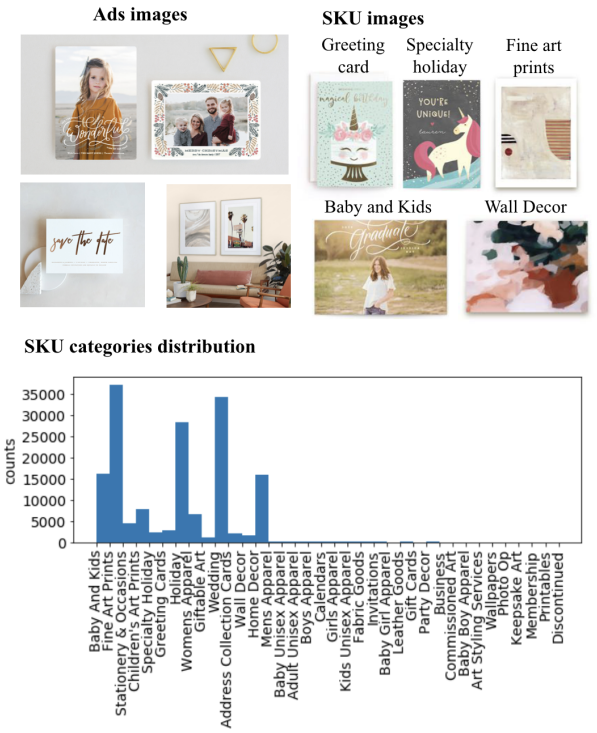

Figure 1. Dataset Description

As usual with real word data, the dataset required extensive cleaning and pre-processing. A lot of ads images are duplicated, but have different sizes or formats. As the latter creates natural data augmentation, we decided to use all images, unless they are 100% identical. Additionally, 25,000 of the SKU images are not downloadable and consequently there is no guarantee that we will find corresponding SKU match for every ad image we have.
The final dataset we are using contains around 135,000 SKU images and 2000 ad images. As was previously mentioned, the dataset is not labeled and cannot be manually labeled due to the large amount of SKUs. This tasks would require going through 135,000 pictures to label a single ad image with the corresponding SKU image.
Main Conclusions
1. As there is no guarantee that all ad images have matching SKUs, choosing the evaluation metric presents a challenge.
2. Manual labeling is not feasible - this task falls into the topic of unsupervised learning, i.e. the task is similar to creating automated labels.
3. As there are 160,000 products, thus 160,000 labels, our approach cannot be a simple image classification task while assigning these labels.
4. The provided dataset mainly consists of postcards, paintings, and business cards. However, the images include objects that can be found in MSCOCO, Image Net and Visual

Genome labels.

6. Technical Approach

This project falls under the computer vision task of unsupervised image matching. In this kind of tasks, where manual labelling is not possible, feature extraction techniques are often applied. The features extracted from an image are compared with a test image stored in the database. The image that matches best is retrieved as the output result [15].

Baseline: As was mentioned in background section a very famous algorithm to extract features from an image is SIFT (Scale-invariant feature transform). As our baseline model we use SIFT available in opencv. The assumption behind using this method is that it should provide robust results - as products on ad images are subjected only to affine transformations, illumination distortions, and noise, meaning distortions that SIFT is invariant to. Even though SIFT has proven to provide relatively robust matching of same objects in different scenes, it requires a licence to use in commercial purposes. Moreover, SIFT creates descriptors of different sizes and is computationally heavy during the matching procedure.
Recently, image feature representation with deep learning is gaining popularity. It proved to handle challenges and complexities of image description and provides stronger feature representation than traditional methods [18], [19]. In such methods, the entire image is typically fed into the pre-trained or fine-tuned network to extract features. We use this method as our second benchmark. Namely, a pre-trained on MSCOCO ResNet-50 [20] is applied on full ads images as a feature extraction network. It provides a reasonable trade off between acceptable computational complexity and accuracy. However, using full ad images to extract ad features is not optimal for our dataset. In that case, ad features may not accurately reflect matching SKU features, since SKU images hold a specific product, while ad images can have multiple products and various backgrounds.

Main Method: In [21] a novel region-wise feature extraction technique using deep CNN has been proposed to solve a similar problem in another domain such as remote sensing images. The main idea of the method is to extract regions from images that contain useful information and only then feed it to pre-trained CNN to extract features. In the paper, authors use edge-boxes algorithm [13] to generate bounding boxes around the object. We use this method as one of region proposal techniques. However, the topic of object proposal in images is well researched and there are other strategies to investigate. This includes Attention Mask [22] and Faster-RCNN for region proposal [8]. In our project, we adapt Faster-RCNN method for region proposal stage as alternative to edge-boxes algorithm. The method is based on using pre-trained model on an object detection task to extract bounding boxes around potential regions of interest. The model architecture is presented in Figure 2. By default Pytorch used MSCOCO dataset to pre-train Faster-RCNN. This dataset is not optimal for our task as it has only 91 labels and does not have such labels as 'picture', 'painting', 'book', etc. Therefore, for our application it is better to use RCNN pre-trained on Visual Genome [23]. The dataset has more than 80000 categories with the opportunity to filter out categories of interest, namely: 'board', 'card', 'frame', 'logo', 'photo', 'picture', 'sign', 'sticker', 'painting', 'paper', 'page', etc.

Artificial Oracle:
We introduce artificial oracle, a strategy that shows potential accuracy of our main method, in the case where regions are adequately extracted from ads. We manually create bounding boxes around potential SKUs in ad images and use them to crop proposed regions. The difference between manually extracted regions and those extracted by Faster-RCNN is shown in the example in Figure 5.

Metric
As previously mentioned, the dataset is not labeled and not all ad images have an existing corresponding SKU image. As it would be uncertain whether the match was not found due to wrong prediction or due to absent SKU, the absolute accuracy does not reflect real accuracy. The calculated top5 accuracy is a better candidate for relative metric between algorithms rather than the standard absolute metric. Moreover, manually checking matches for every 3000 ad images would be time expensive. Therefore, the sub-sample of randomly chosen images is used across all algorithms.

7. Experiments and Results

For experiments a sub-sample of 30 randomly chosen ads images is created. This number is limited by SIFT inference time and required manual judgement for corresponding matches and regions during evaluation. When using CPU, it takes around one hour to find corresponding matching products for each ad.
For all algorithms, SKU features are pre-extracted and saved which saves significant time during inference. More information on running time for 64Gb memory and 2.4 GHz Intel Core i9 is presented in Table 1 together with standard top 5 accuracy metric. We emphasize that accuracy results can only be treated as relative across algorithms, but not as absolute accuracy for this dataset.
Experiments are performed for the 5 different strategies described in section 5 Technical Approach: SIFT, Resnet, combination of Faster-RCNN and Resnet pre-trained on

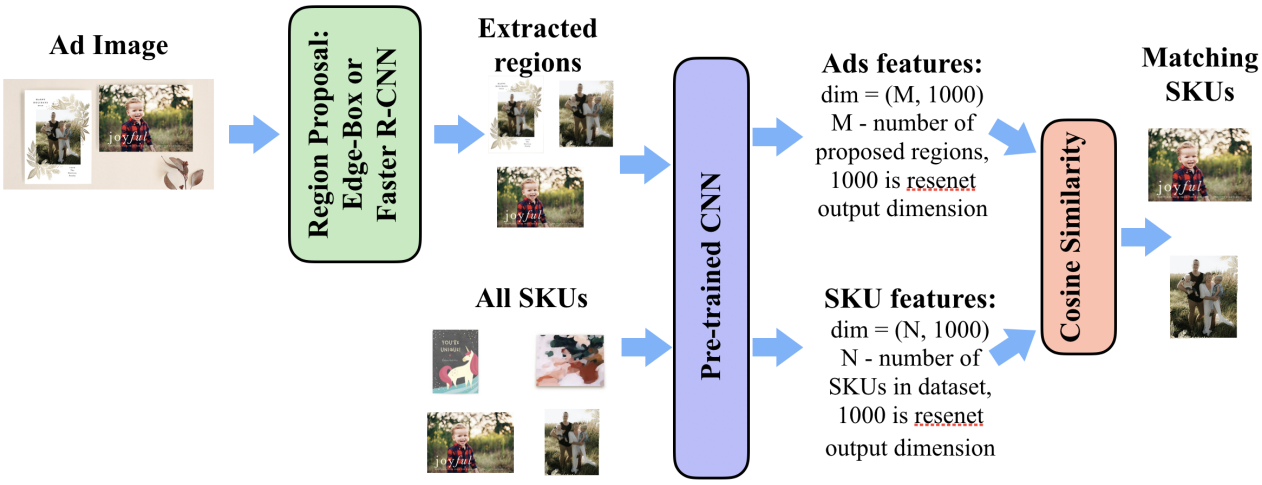

Figure 2. Main method architecture

MSCOCO, combination of Faster-RCNN and Resnet pre-trained on Visual Genome, and oracle, and presented in Table 1. Moreover, for Oracle we tested different length of feature embeddings (1000 and 2048) showing that longer embeddings provide better performance for a large amount of SKUs.

Example of successful and failed region selection for SIFT and one of CNN methods (Faster R-CNN pre-trained on Visual Genome) is presented in Figure 4.

Analysis:

1. From the Table 1, it can be concluded that SIFT is slow during inference (more than 1h for a single ad image and 135000 SKUs), even though it is faster in feature extraction step. This result is expected as SIFT creates different amount of descriptors for each image and pluralization of computations is vague. SIFT is not usable for this application as inference time is critical.
2. The second baseline method based on feature extraction from full images is fast in inference step (due to parallel nature of CNN computations) but is not able to find matches and provides only 6% accuracy. The reason is that features are created from full ads images and have a lot of noise coming from unnecessary background information. Consequently, this method is also ruled out.
3. The Faster-RCNN method is relatively fast, but pre-trained on MSCOCO has relatively low accuracy of 23.3 % - less than half accuracy of the Oracle. It is not surprising as regions extracted by pre-trained on MSCOCO model do not match SKU images (see Figure 5). Fine tuning of model hyper-parameters without adding labels does not help to solve this issue, as MSCOCO has labels like trees, humans, animals, etc., rather than pictures and cards.

Table 1: Performance comparison across algorithms

Region Selection Algorith	Feature Extraction Algorithm	Size of feature embeddings	Top-5 Accuracy, %	Inference time (1 ad image vs all SKUs)	Feature extraction time (all SKUs)
SIFT		n/a	22	>1h	0.5h
Full images	pre-trained Resnet-50	2048	6	1s	4.5h
MSCOCO Faster-RCNN		2048	23.3	10s	
Visual Genome Faster-RCNN		2048	46.7	10s	
Edge-Box		2048	33.3	13s	
Oracle		1000	50	8s	
		2048	60	8s	

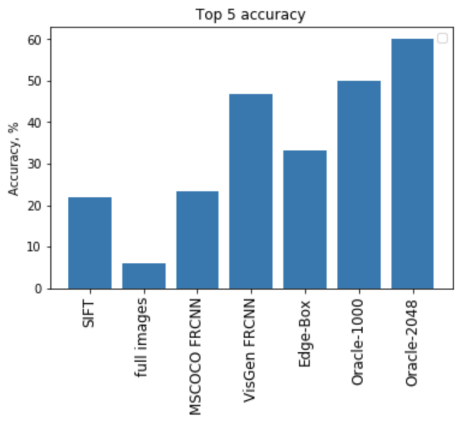

Figure 3. Performance comparison across algorithms

One solution to this problem is to fine-tune the model on our dataset with manually created bounding boxes around potential SKUs and add customized labels for new useful classes or use another pre-trained model or region extraction technique.

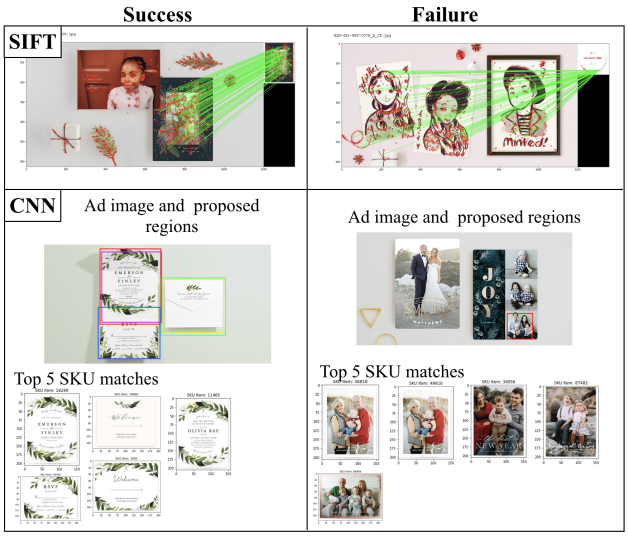

Figure 4. Examples of successful and failed matching

Figure 5. Extracted regions of interest for main model and oracle

4. The Faster-RCNN method method pre-trained on Visual Genome performs significantly better than pre-trained on MSCOCO and almost reaches performance of Oracle method. This result is expected as extracted by the model regions are close to optimal. To achieve better result, fine-tuning of model hyper-parameters was done. For more details about optimal model hyper parameters refer to Appendix.

5. The Edge-box model is fast and produces relatively good top-5 accuracy of 33.3%. The reason behind lower accuracy than pre-trained on Visual Genome Faster R-CNN is that the model uses edges for region proposals, and thus it fails to distinguish edges representing postcards boundaries from edges representing boundaries of other objects. Even after fine-tuning of hyper-parameters we observed that region extraction is not always optimal, and there is opportunity for further refinement. Moreover, the model performance is unstable as improving regions for one image can lead to worse regions for other images. On the Figure 5 results for a model with best found hyper-parameters is presented. it can be seen that even with best parameters important regions are often missed. To improve result further fine-tuning can be done or potentially another edge detection technique can be used.

7. As can be observed from Table 1 while comparing Oracle with 1000 features to Oracle with 2048 features, increasing size of feature embeddings helps models to perform better. This is not surprising. Due to high number of SKUs, the variability of the dataset is high and more dimensions to express this variability proves to be useful.

7. The accuracy and inference time of the oracle method is superior. This result is expected as for oracle we manually constructed ideal image regions, and does not spend additional inference time to calculate features for bad regions. The oracle performance shows the potential of using CNN features in the case where region extraction step is performed well. Achieved 60% accuracy does not represent absolute accuracy. Most probably the model was not able to achieve 100% accuracy as products in ads images are simply missing from SKU images. Moreover, the Oracle model was able to find all the matches that other models found combined.

8. From the Figure 4, we can see that even when CNN fails to find exact matching SKU it still finds look-alike SKUs, which is not the case with SIFT. SIFT often matches multiple keypoints on the ad image to a single keypoint on the SKU image which looks nothing alike as it tries to find a nearest neighbor even if the neighbor is far. Consequently, the number of matching points is not directly linked to the similarity between two images and matches can contain both good (inliers) and bad (outliers) matches. One way to address this is to only consider the number of inliers as a measure of similarity. More detailed information can be found in [17].

8. Conclusions

CNN architecture for feature extraction in advertising images results in superior performance over SIFT in accuracy and inference time. Edge-Box algorithm for region

extraction provides better performance than pure images (33.3% vs 6%) and Faster-RCNN pre-trained on MSCOCO (33.3% vs 23.3%). As model relies on edge detection, it proved to be unstable across images and difficult to tune.
The Faster-RCNN pre-trained on Visual Genome achieves relatively high performance and almost reaches accuracy achieved on manually constructed regions (46.7% vs 60%). However, the gap between performance shows that models pre-trained on existing datasets can only partially be used to successfully perform the task and require attentive fine-tuning of hyper-parameters. Therefore, there is a potential to improve performance by manually labeling a sample of ads images with optimal bounding boxes and use it to fine-tune the model weights instead of hyper-parameters.

Suggestions for further research:

We currently use the Faster-RCNN and Edge-box methods for region proposal in order to restrict our search for features in meaningful regions. The gap between Oracle and these methods proves that there is an opportunity to further improve region definition, either by (i) labeling our ad images and fine tuning Fast R-CNN model weights or (ii) investigating other region proposal methods. One approach is a proposed region-wise deep feature extraction framework [21] where regions that may contain relevant information are extracted and fed into a pre-trained convolutional neural network model to extract regional deep features.

References

[1] Jyoti Joglekar, Shirish S. Gedam *Area based image matching methods—A survey.* International Journal of Emerging Technology and Advanced Engineering (ISSN 2250-2459, Volume 2, Issue 1, January 2012)

[2] Ebrahim Karami, Siva Prasad, Mohamed Shehata *Image matching using SIFT, SURF, BRIEF and ORB: performance comparison for distorted images.* arXiv:1710.02726

[3] David G Lowe *Distinctive Image Features from Scale-Invariant Keypoints* International Journal of Computer Vision (volume 50, Issue 2, 2004, pp.91-110)

[4] Yan Ke, R. Sukthankar *PCA-SIFT: a more distinctive representation for local image descriptors* Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. (CVPR 2004)

[5] Herbert Bay, Andreas Ess, Tinne Tuytelaars, LucVan Gool *Speeded-Up Robust Features (SURF)* Computer Vision and Image Understanding (Volume 110, Issue 3, June 2008, Pages 346-359)

[6] Ethan Rublee, Vincent Rabaud, Kurt Konolige, Gary Bradski *ORB: An efficient alternative to SIFT or SURF* IEEE International Conference on Computer Vision, 2011.

[7] Bidyut B. Chaudhuri, Masaki Nakagawa, Pritee Khanna *Proceedings of Third International Conference on Computer Vision Image Processing (CVIP 2018)*

[8] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks* arXiv:1506.01497v3

[9] Christian Szegedy, Alexander Toshev, Dumitru Erhan *Deep Neural Networks for Object Detection* Advances in Neural Information Processing Systems 26 (NIPS 2013)

[10] Pierre Sermanet, David Eigen, Xiang Zhang, Michael Mathieu, Rob Fergus, Yann LeCun *OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks* International Conference on Learning Representations (ICLR 2014)

[11] Dumitru Erhan, Christian Szegedy, Alexander Toshev, Dragomir Anguelov *Scalable object detection using deep neural networks.* Conference on Computer Vision and Pattern Recognition (CVPR 2014)

[12] Christian Szegedy, Scott Reed, Dumitru Erhan, Dragomir Anguelov, Sergey Ioffe *Scalable, high-quality object detection.* arXiv:1412.1441

[13] Zitnick, C.; Dollar, *Edge boxes: Locating object proposals from edge* In Proceedings of the 13th European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 391–405.

[14] *https://github.com/shilrley6/Faster-R-CNN-with-model-pretrained-on-Visual-Genome*

[15] Preeti Mandle, Vibha Shaligram *A Survey: Image Matching* International Journal of Digital Application & Contemporary Research Website: www.ijdacr.com (Volume 4, Issue 1, August 2015)

[16] David G. Lowe *Distinctive Image Features from Scale-Invariant Keypoints* International Journal of Computer Vision, 60, 2 (2004), pp. 91-110

[17] Augereau, O., Journet, N., Domenger, J. P. *Semi-structured document image matching and recognition* https://hal.archives-ouvertes.fr/hal-00755748

[18] Dutta, R.; Das, A.; Aryal, J. *Big data integration shows Australian bush-fire frequency is increasing significantly* R. Soc. Open Sci. 2016, 3, 150241

[19] Li, Y.; Zhang, Y.; Huang, X.; Zhu, H.; Ma, J. *Large-scale remote sensing image retrieval by deep hashing neural networks* IEEE Trans. Geosci. Remote Sens. 2018, 56, 950–965

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun *Deep Residual Learning for Image Recognition* https://arxiv.org/abs/1512.03385

[21] Peng Li, Peng Ren, Xiaoyu Zhang, Qian Wang, Xiaobin Zhu, and Lei Wang *Region-Wise Deep Feature Representation for Remote Sensing Images* Remote Sensing, Volume 10, Issue 6, 2018

[22] Christian Wilms and Simone Frintrop *AttentionMask: Attentive, Efficient Object Proposal Generation Focusing on Small Objects* arXiv:1811.08728v1

[23] github.com/shilrley6/Faster-R-CNN-with-model-pretrained-on-Visual-Genome *Faster R-CNN with model pretrained on Visual Genome*

9. Appendix

1. Hyper-parameters for Faster-RCNN pre-trained of Visual Genome [23]

   Modified in demo.py:
   conf_thresh = 0.4
   MIN_BOXES = 10
   MAX_BOXES = 36
   keep_class_list = ['board', 'card', 'frame', 'logo', 'mirror', 'photo', 'picture', 'screen', 'sign', 'sticker', 'painting', 'paper', 'page', 'couple', 'writing', 'people']