# RINN-UOD: Randomly Initialised Neural Networks for Unsupervised Object Detection

**Project Category: Computer Vision**


**Samir Agarwala**
SUNet ID: samirag
Computer Science
Stanford University
samirag@stanford.edu

**Alexandre Hayderi**
SUNet ID: alexhay
Computer Science
Stanford University
alexhay@stanford.edu

**Melvin Orichi Socana**
SUNet ID: msorichi
Electrical Engineering
Stanford University
msorichi@stanford.edu


## 1 Introduction

When we look at the world around us, we can distinguish different objects, localize them in our environment, and recognize the object we are perceiving. In the past few years, computer vision models have had tremendous success in such object detection tasks. This success has mainly been concentrated in learning-based approaches [1, 2] which learn from image datasets and are supervised by ground-truth bounding boxes that are available during training. These methods are trained to detect particular classes of objects and are unable to find objects that they have not seen during training.

To build robust object detection methods that can generalize to diverse scenes that we may encounter in the wild, it is important to build detection methods that can discover objects that are unseen at training time. Unsupervised methods have shown promise in class-agnostic object detection. For instance, recent work in object detection [3] proposed an object detection algorithm that can use image features from a learned classification model to achieve high-quality detection results. Although their method does not require explicit object bounding-box supervision, it does use self-supervised features from pre-training while performing detection. This motivated us to consider: *is any kind of supervised or self-supervised pre-training required for this task?* Several papers [4, 5, 6] show that randomly initialized networks have the capacity to perform well in various tasks without learning. We are thus approaching the challenging problem of unsupervised object detection by leveraging this property of randomly initialized CNNs to locate regions of interest (RoI) for objects within each image and convert those RoIs into bounding boxes corresponding to detected objects.

## 2 Related Work

Our work on using randomly initialized CNNs for object detection builds upon past work in object detection and randomly initialized neural networks.

**Object Detection Methods.** Object detection methods can be broadly classified into traditional and learning-based methods. Traditional object detection methods often use handcrafted features that are extracted from images to find objects in images. This was done by using linear classifiers that were trained for specific detection tasks [7] and extracting feature descriptors such as SIFT [8] or HOG [9] and using them to detect objects [10]. Most learning-based work has focused on using learning-based models to detect objects which involves training convolutional neural network [11] and transformer models [12] on large datasets for detecting objects from a pre-specified set of categories. However, the number of different types of objects present in these datasets is limited as compared to objects that may be observed in the wild. In our work, we want to move away from using trained models as used in [3] and see if the structure of convolutional models allows us to get reasonable class-agnostic object detections without using any supervision.

**Randomly Initialized Neural Networks.** Learning-based methods have become immensely popular in computer vision tasks [12, 11, 13]. These methods train on large datasets and hope to learn a model that can generalize to unseen data, and have been very successful in the past few years. However, recent work has shown that the structure of convolutional models itself may allow us to approach several tasks even when we do not perform training and instead use randomized models. For instance, [6] found that randomly initialized networks could be used to find visual correspondences that can bootstrap training for point cloud registration. [4] showed that fixing most of the parameters of deep learning networks to random values often

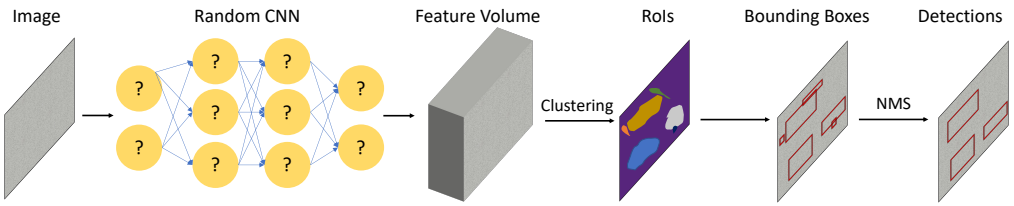


Figure 1: We propose an approach to detect objects in images using randomly initialized CNNs. We first pass the input image through a random CNN to extract a feature volume that should capture localized spatial information of images because of the structure of CNNs. We then cluster this feature volume to find regions of interest (RoIs) and convert these RoIs to bounding boxes. Finally, we post-process these bounding boxes using a non-max suppression (NMS) heuristic we developed to get our predicted object detection.

results in a similar performance to models that optimize all parameters. [5] showed that randomly initialized generator convolutional neural networks could be used for a variety of tasks such as image restoration and super-resolution by performing inference time optimization on degraded image inputs to recover the original image. Thus, through these works, we see that randomly initialized networks may serve as powerful tools in machine learning. In this paper, we examine how we can build upon past work in random networks by exploring their use in a fundamental computer vision task such as object detection.

## 3 Method

We will be exploring the use of randomly initialized CNNs for unsupervised object detection as shown in Figure 1. To this effect, we will: 1) extract visual features from input images using a randomly initialized CNN backbone, 2) find clusters of features that we consider as regions of interest (RoIs), 3) convert the RoIs to bounding boxes and 4) suppress unlikely bounding boxes.

### 3.1 Feature Extraction

We modify the architecture of the randomly initialized CNN backbone from [6] and use that as our feature extractor. Their CNN backbone consists of a standard convolutional network with two residual layers and performs no spatial downsampling, thus mapping each pixel to some vector in feature space. We modify this architecture to include max-pooling layers between the residual layers in their architecture to downsample features and get a coarser feature grid that is faster and less noisy to cluster. This gives a feature volume $F \in \mathcal{R}^{H/4 \times W/4 \times D}$ for an input image $I \in \mathcal{R}^{H \times W \times 3}$. We refer readers to [6] for details about the backbone.

### 3.2 Proposing Regions of Interest (RoIs)

To propose RoIs in the input images that correspond to areas that contain an object, we use standard clustering methods such as K-means [14] and agglomerative clustering [15] to cluster features into RoIs that correspond to an object. Thus, each cluster from the clustering algorithms represents a distinct object. Since the focus of our investigation is exploring the use of randomly initialized CNN for object detection, we are not proposing the use of a specific clustering method for finding RoIs but instead examine the use of K-means and hierarchical clustering methods in finding RoIs in our experiments.

When using K-means as our clustering method for finding RoIs, we set the number of clusters based on our validation set and also append a positional encoding representing the spatial location of each feature to feature volume $F$ so that the algorithm can incorporate some notion of position when clustering feature vectors in the volume. On the other hand, agglomerative clustering naturally finds the number of clusters in the data and does not require such a positional encoding. We only allow agglomerative clustering to form clusters with neighboring features in feature space at each step as we know that objects are contiguous in space and thus each part of an object which is represented by a feature needs to be adjacent to another feature that represents the same object.

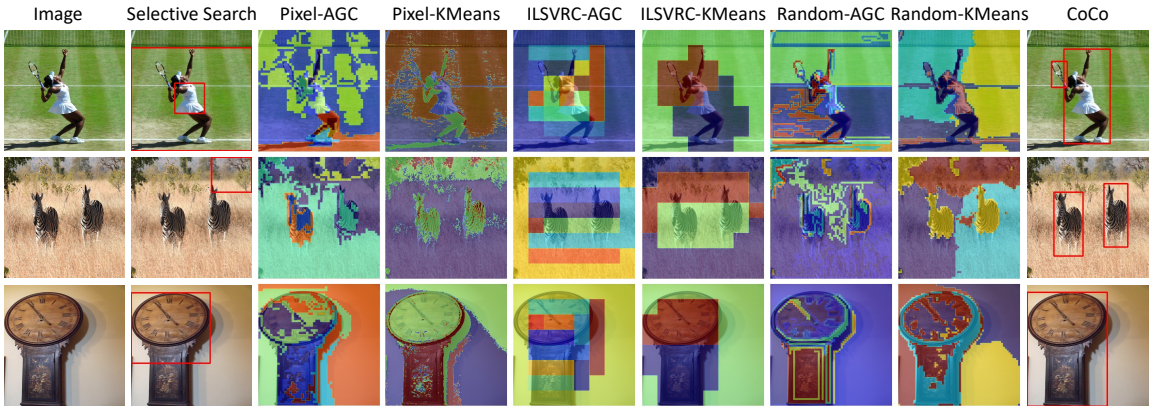


Figure 2: We compare RoIs from clustering in our proposed method and baselines [16, 17]. Random-KMeans does well in generating high-quality RoIs compared to baselines.

### 3.3 Finding Object Detections

We represent detected objects as bounding boxes in the input image and convert RoIs into bounding boxes using classical image processing operations. To do this, we approximate each RoI using a polygon and then find the best-fitting bounding box for the polygon. Since clustering methods may find RoIs that are not fully connected in pixel space, for simplicity we consider each disconnected component of an RoI as a separate detected object.

### 3.4 Non-Max Suppression (NMS)

In this step, we propose a heuristic for NMS that suppresses unlikely bounding boxes that are proposed by our method. Our heuristic ignores bounding boxes that are of the same size as the image, prioritizes the discovery of larger objects, and excludes bounding boxes that have a reasonable overlap measured by Intersection-over-Union (IoU) with a bounding box of greater area i.e. $IoU > 0.1$. We also exclude bounding boxes that are unlikely to be observed in the wild and are artifacts of noisy clustering such as bounding boxes that have an area of 1 squared pixel or extremely thin bounding boxes in this step.

**Implementation Details.** Our input images are resized to $256 \times 256$. We normalize the images using mean normalization before running feature extraction. For agglomerative clustering, we apply a Gaussian filter before clustering to reduce aliasing as per library documentation [18]. We choose hyperparameters for our approach based on the results of a validation set of 50 samples from the CoCo 2017 validation set [19]. In our approach, we thus decided to run K-means with 4 clusters and agglomerative clustering with a distance threshold of $1.5\sigma$ where $\sigma$ is the standard deviation of pair-wise distance values seen during clustering. We also weigh the positional encoding features by a factor of $\gamma = 0.03$ before clustering using K-means.

## 4 Experiments

We ran experiments to evaluate the effectiveness of our proposed approach compared to several baselines. In this section, we provide 1) a qualitative and quantitative evaluation of object detections and 2) an ablation study examining the impact of K-means clustering hyperparameters on object detection quality.

**Dataset.** We use a random subset of 497 images from the CoCo 2017 validation set [19] as our test set for evaluating our method and baselines. We also note that our test set has no overlap with our validation set of 50 images that was used for hyperparameter optimization.

**Baselines.** We compare our method against several baselines. The first type of baselines we compare against are methods that cluster pixels directly such as K-means and agglomerative clustering (AGC). To show how randomly initialized networks compare against learned features, we also replace our backbone with a ResNet-18 backbone trained on ILSVRC classification [17] and see how using these features compares to using random

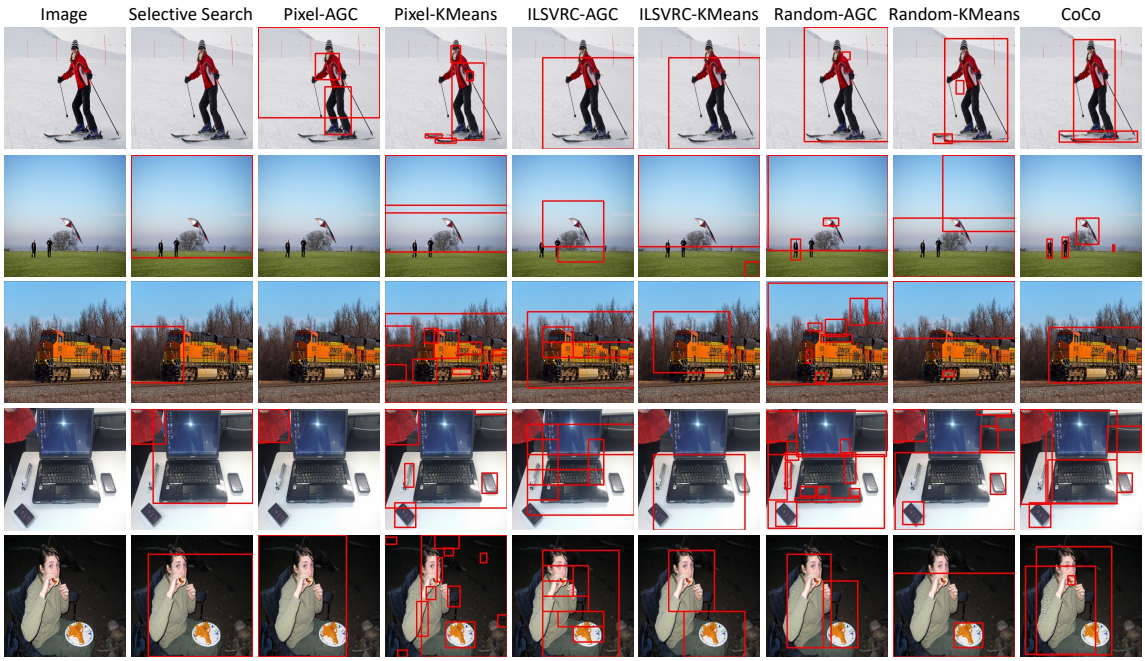


Figure 3: We see that our random CNN based object detector can often find high-quality object detections in a variety of scenarios compared to using features trained on ILSVRC [17], pixel-based clustering and selective search [16].

features. We optimize hyperparameters for clustering-based baselines similar to our random CNN-based approach. Lastly, we compare against selective search [16] which groups segments from a graph-based method to find bounding boxes and has been used to generate region proposals in [11, 13]. For fair comparison, we also run our NMS heuristic on all baselines before performing evaluation.

**Evaluation Metrics.** We perform a quantitative evaluation of our bounding boxes using average precision (AP) and average recall (AR). Different from evaluation in CoCo [19], we perform class-agnostic evaluation where we do not consider categories while performing evaluation. Instead, we compute the AP and AR metrics by comparing all the predicted bounding boxes to the ground-truth bounding boxes across all categories that are present in CoCo. For AR computation, we consider a maximum of 100 detections when performing evaluation. We emphasize here that CoCo has a limited subset of objects in its dataset and thus evaluation using CoCo is not ideal as the compared methods may detect objects that are not included in the dataset. For example, CoCo does not consider the plate in the last row in Figure 3 to be an object. However, using such a labeled dataset is the most practical way to evaluate a class-agnostic object detector without having to densely label a new dataset. We also perform AP and AR evaluation over two IoU ranges, i.e. $[0.3, 0.95]$ and $[0.5, 0.95]$, to get more insight into the quality of bounding boxes being predicted by the compared methods.

### 4.1 Evaluation of Object Detection

**Qualitative Results.** We present qualitative results of RoIs and detections for our method and all baselines in Figures 2 and 3 . From Figure 2, we see that Random-KMeans seems to find better RoIs compared to other methods which often find noisy clusters or miss object instances. Figure 3 shows that this trend continues for the object detections, and random KMeans and AGC often find high-quality clusters. The pixel-based methods find a lot of false positives while clusters from the trained baseline are not as well aligned with objects.

**Quantitative Results.** Quantitative results are presented in Table 1. The random CNN based object detector performs competitively on average recall to Pixel-KMeans, the best-performing baseline on this metric. However, we find that the precision of the detector is low. Based on qualitative comparisons, we argue that this is an artifact of the CoCo dataset which only has 80 labeled classes [19] while our detector and baselines might detect objects not in the dataset resulting in a low AP. From qualitative results in Figure 3, we see that our approach often finds better detections than baselines.

| | $AP_{[0.3:0.95]} \uparrow$ | $AP_{[0.5:0.95]} \uparrow$ | $AR_{[0.3:0.95]} \uparrow$ | $AR_{[0.5:0.95]} \uparrow$ |
|---|---|---|---|---|
| Selective Search [16] | **1.27** | **0.57** | 3.30 | 2.01 |
| Pixel-KMeans | 0.51 | 0.18 | **7.00** | **4.01** |
| Pixel-AGC | 0.64 | 0.26 | 3.25 | 1.84 |
| ILSVRC-KMeans [17] | **1.27** | 0.35 | 3.10 | 1.23 |
| ILSVRC-AGC [17] | 0.15 | 0.01 | 3.92 | 1.14 |
| Random-KMeans | 0.60 | 0.17 | 5.03 | 2.57 |
| Random-AGC | 0.45 | 0.09 | 4.98 | 2.40 |

Table 1: We compare detection using random CNNs to selective search, baselines that directly cluster pixel space, and baselines that use pre-trained features from ILSVRC [17] classification on average precision and average recall using IoU thresholds in the range of $[0.3, 0.95]$ and $[0.5, 0.95]$

.

### 4.2 Impact of K-means Hyperparameters on Object Detection Quality

| | $AP_{[0.3:0.95]} \uparrow$ | $AP_{[0.5:0.95]} \uparrow$ | $AR_{[0.3:0.95]} \uparrow$ | $AR_{[0.5:0.95]} \uparrow$ |
|---|---|---|---|---|
| Pixel-KMeans-NoPos | 0.49 | 0.17 | 7.03 | 4.04 |
| Pixel-KMeans-GT | 0.50 | 0.14 | **7.85** | **4.19** |
| Pixel-KMeans | 0.51 | 0.18 | 7.00 | 4.01 |
| ILSVRC-KMeans-GT [17] | 0.68 | 0.09 | 4.38 | 1.51 |
| ILSVRC-KMeans [17] | **1.27** | **0.35** | 3.10 | 1.23 |
| Random-KMeans-NoPos | 0.46 | 0.14 | 5.84 | 2.98 |
| Random-KMeans-GT | 0.49 | 0.10 | 5.53 | 2.36 |
| Random-KMeans | 0.60 | 0.17 | 5.03 | 2.57 |

Table 2: We perform an ablation study involving K-means which we use while generating RoI proposals. We compare *-NoPos* and *-GT* in this ablation where *-NoPos* refers to using no positional encoding during clustering and *-GT* refers to using the GT number of clusters for each image during RoI clustering. We do not report *-NoPos* for the trained ILSVRC baseline since it does not consider positional information.

We use standard clustering methods such as K-means and agglomerative clustering to find RoIs in our approach. In Table 2, we examine the impact of K-means clustering hyperparameters such as the number of clusters and the use of a positional encoding on detection results since this would help us learn how improving RoI quality might contribute to improvements in detection quality. We see that providing GT number of clusters tends to increase recall at the cost of precision. This may occur because knowing the number of objects might allow the clustering methods to find the CoCo objects correctly, but at the same time, the possibility of having disconnected RoIs during clustering can reduce precision. We also see that not having a positional encoding seems to reduce precision which might be because it does not incentivize points in an RoI to be connected and thus there might be more spurious detections resulting in a lower AP.

## 5 Conclusion and Next Steps

We propose an approach that performs class-agnostic object detection using features from random CNNs. Our approach has promising qualitative and quantitative results on our test set and shows the possible effectiveness of using random neural networks in tasks that have predominantly been approached using learning-based methods. We also emphasize the challenges we faced in evaluating our unsupervised method since existing datasets are often not densely labeled with all objects in the image which makes it challenging to accurately interpret metrics calculated on datasets such as CoCo [19].

Unsupervised object detection is an exciting area of research and progress in such approaches would allow us to develop more robust object detectors. Some areas for future work include experimenting with the detection method from [3] with random features, working on clustering methods specific to randomized features, and developing better NMS heuristics for suppressing spurious detections that might arise during inference.

**Contributions.** All of our group members have contributed reasonably to the project. Samir worked on coming up with the project idea and proposal, developing the random CNN based approach and NMS heuristics, performing evaluation, and running the experiments. He also contributed significantly to writing the project proposals and reports. Alexandre handled most of the clustering tasks by testing and comparing the effectiveness of several clustering methods, then fine-tuning and running validation. He also worked on post-processing, handling bounding box computations, plotting and the development of the non-max suppression heuristic. Tangentially, he also worked on refactoring the codebase. Melvin contributed to dataset setup, developing the evaluation code and interfacing the bounding-box predictions with the CoCo API for results. He also worked on developing the scripts for formatting the images and developing the NMS heuristic.

## References


[1] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 91–99, Cambridge, MA, USA, 2015. MIT Press.

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

[3] Huy V Vo, Patrick Pérez, and Jean Ponce. Toward unsupervised, multi-object discovery in large-scale image collections. In *European Conference on Computer Vision*, pages 779–795. Springer, 2020.

[4] Amir Rosenfeld and John K Tsotsos. Intriguing properties of randomly weighted networks: Generalizing while learning next to nothing. In *2019 16th Conference on Computer and Robot Vision (CRV)*, pages 9–16. IEEE, 2019.

[5] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9446–9454, 2018.

[6] Mohamed El Banani and Justin Johnson. Bootstrap Your Own Correspondences. In *ICCV*, 2021.

[7] Pedro Felzenszwalb, David McAllester, and Deva Ramanan. A discriminatively trained, multiscale, deformable part model. In *2008 IEEE conference on computer vision and pattern recognition*, pages 1–8. Ieee, 2008.

[8] G Lowe. Sift-the scale invariant feature transform. *Int. J*, 2(91-110):2, 2004.

[9] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 886–893. Ieee, 2005.

[10] Xiaoyu Wang, Ming Yang, Shenghuo Zhu, and Yuanqing Lin. Regionlets for generic object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 17–24, 2013.

[11] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[12] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

[13] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[14] S. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.

[15] Joe H. Ward Jr. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58(301):236–244, 1963.

[16] J.R.R. Uijlings, K.E.A. van de Sande, T. Gevers, and A.W.M. Smeulders. Selective search for object recognition. *International Journal of Computer Vision*, 2013.

[17] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[18] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[19] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 740–755, Cham, 2014. Springer International Publishing.