

Keypoint Density based Region Proposal for object detection using rCNN

JT Turner
Knexus Research Corporation
National Harbor, MD
jt.turner@knexusresearch.com

Brendan Morris
University of Nevada, Las Vegas
Las Vegas, NV
brendan.morris@unlv.edu

Kalyan Gupta
Knexus Research Corporation
National Harbor, MD
kalyan.gupta@knexusresearch.com

David Aha
Naval Research Lab
Washington, DC
david.aha@nrl.navy.mil

Abstract

Recent changes to the topology of regional convolutional neural networks (rCNN) have allowed them to obtain near realtime speeds in image detection. We propose a method for region proposal alternate to selective search which is used in the current state of the art object detection [3] and introduce the fine grained image datasets. In a maritime surveillance setting, it maybe important to not only identify an object approaching your position but also know the type of vessel (e.g., a civilian fishing vessel or an enemy destroyer). Our region proposal technique Keypoint Density Region Proposal (KDRP) is able to achieve levels of performance that are not no worse than selective search at a very high level of significance, while only taking 49% of the time of selective search in the rCNN pipeline.

1. Introduction

As Convolutional Neural Networks (CNN) have evolved from the early work of Lecun [9] where they were used primarily in classification, to the point where they have been acheiving state of the art performance in object detection tasks [4]. The transition to them being applicable to detection took modifications in how they were trained and deployed, and although they were achieving high accuracy, it took around 13 seconds per image in the first generations of the rCNN.

Although detection accuracy is important in application, for practical tasks involving video, 13 seconds is unusable for real time; it would have to be run offline. An improvement to the rCNN a year later called fast r-CNN [4] did exactly what the name implied; sped up the detection pipeline. With these algorithmic improvements, the object detection pipeline could be performed in under 2 seconds. Although

this is an enormous improvement, it is still not quite real-time.

Fast r-CNN is the current state of the art results for the widely used pascal VOC of object detection. We elected not to test on this dataset because it is not finegrained, and we intend to show that the algorithm is applicable on the hard task of fine grained object detection. The more traditional approach to image classification tasks is keypoint descriptors and local feature descriptors [10], which are binned into histograms and compared to other keypoints to match similarly featurized objects. The work of Felzenszwalb [1] on deformable part models and detection of parts gave rise to specialized part models that operate by transfer of likely locations [5], which achieved great performance on the fine-grained Caltech UCSD bird dataset [15].

Our goal is to improve upon the current state of the art in fine grained visual object detection in such a way to reduce the time to less than one second; less than 50% the time of fast r-CNN. For this to be useful, our faster algorithm must perform no worse than fast r-CNN by accuracy.

2. Fast-rCNN Detection Pipeline

Here we lay out the proposed pipeline of using the Fast-rCNN system of Girshick, which is followed by our system.

1. Model Training
2. Region Proposal
3. Feature Extraction
4. Non Maximum Suppression
5. Hypothesis Selection

Unless otherwise stated, an *image* will be defined as an RGB pixel matrix between the sizes of 350 to 500 length or width from the CUB-200 bird dataset.

2.1. Model Training

Although this is not actually a part of the detection pipeline, it is necessary, and none of the subsequent steps have any impact on object detection if a model is not trained first. Training an rCNN is normally done using the open source library caffe [7]. The only way that our pipeline differs from the standard neural network training regiment is that we provide region coordinates at training time, and have an additional layer to display the coordinates of the region after convolution. This can all be implemented using caffe.

Training a network is still a time intensive process for rCNNs. Although it helps to start with an imagenet trained model and finetune from there, generally at least a week of finetuning is needed to produce state of the art performance. Because minimizing training time is not the primary concern of this study, we used selective search for the ROIs in the training of the rcnn. Future work will be done using KDRP, or a concatenation of both region proposal networks to see if it boosts performance from training.

2.2. Region Proposal

Region proposal is the first step of the object detection pipeline, and must be done adequately for object detection to work. Assuming we have an image I that is of dimensions $w \times h$, a region is a subset of that image r , with dimensions $w' \times h'$, where $0 < w' \leq w$, and $0 < h' \leq h$. The purpose of region proposal is to generate enough regions such that there is a high probability that one of the regions r in our set of all regions proposed R be an accurate region that bounds the ground truth object.

Because our pipeline never infers bounding boxes through and calculations, and cannot modify the boundaries of the bounding box, *the region that contains the object must be generated or detection will fail*. In the first implementation of fast r- CNN [3], the region proposal method was selective search. In section 3 we will discuss this method, and how it varies from our new method KDRP.

2.3. Feature Extraction

Once regions have been proposed, we must use our trained model to featurize the regions. Featurization is done in the standard way of multiplying the image matrix through the trained model, the only difference being that there are two output layers from the fast r-CNN model. For a model trained on n classes, the output from the classification layer will be $n + 1$ probabilities, and for the bounding box coordinates of these classes, there will be $4n + 4$ numbers. The reason that we are adding one to the number of classes is that we allow a 'background' class. Each of the regions classification is the softmax of the $n + 1$ probabilities, and each class corresponds to a 4-tuple of coordinates.

The main algorithmic improvements that allow us to process regions so quickly is the region of interest pooling (ROIpooling) of [3] which is a simplified variant of the spatial pyramid pooling of [6]. The convolutional, pooling, and nonlinear transforms that occur in the network before are input size invariant (as long as they are larger than a minimum size such that the sliding window can be passed over the various layers of convolution). The computationally expensive convolutional phase needs to only happen once; in this implementation the ROIs that we are detecting are passed through the network at the same time as the image itself, and translated to the ROIpooling coordinates. Once the ROIpooling layer has been computed, the only computation that needs to be done multiple times are the multiplications between the fully connected layers, and dense matrix multiplications have been optimized to be extremely fast on GPUs.

Feature extraction was traditionally the bottleneck in using r-CNN's; that they are now able to be computed for an image faster than region proposals presents new opportunities for applications (such as real time object detection at 1 frame per second), but also puts a new burden on finding faster and more efficient region proposal methods.

2.4. Non Maximum Supression

Non Maximum Supression is a technique that has been used for r-CNNs since their genesis in 2013 [4]; they are an essential technique for dealing with an unknown number of objects in the image, and for making hypothesis selection faster. Non maximum suppression is very fast; although it is dependent on the number of regions, because it runs in $O(mcn^2)$ time (and in practice is much faster) where n is the number of regions, m is the number of images processed, and c is the number of classes it is almost an unnoticeable time component for detection. A description of non maximum suppression is given in [4], with an example figure 1.

2.5. Hypothesis Selection

The pipeline culminates naturally at hypothesis selection, which like non maximum suppression is a very fast process with respect to region proposal or featurization. Hypothesis selection can be undertaken in two different ways; in the first of which we tell the algorithm exactly how many objects to detect in the image (figure 2a shows a situation where we tell the hypothesis selection algorithm there is a single object in the image, when there is in fact 4). The second, more realistic scenario in which it is *unknown* at evaluation time how many objects if any are in the ground-truth of the image, and the hypothesis selection algorithm must determine which of its thousands of regions are valid detections using the confidence output from the r-CNN, as shown in figure 2b.



(a) Hypothesis selection forced to choose $top-k$, $k = 1$.

(b) Hypothesis selection uses all detections over probability threshold

Figure 2: Comparison of hypothesis selection implementing $top-k$ selection, or selecting all detected objects greater than a known probability threshold

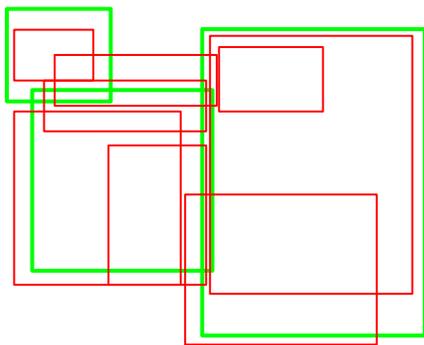


Figure 1: Non Maximum Suppression example; the three green regions were selected because they were the highest probability region in the area that did not overlap at a fraction greater than α with a higher probability region. The red regions are suppressed because the region they occupied was already occupied at a fraction greater than α by a higher probability region.

Every stage of the fast r-CNN pipeline is identical except for the region proposal between the selective search existing state of the art, and our proposed KDRP algorithm.

3. KDRP

The current standard for region proposal is *selective search* [14], an image segmentation method that takes uses multiple image scales, and 8 opponent colorspace to generate regions. This is time intensive, and creates several thousand regions, but it is important that there are so many regions generated, because if the region containing the ground truth object with an intersection over union above a pre-set threshold is not generated, than the object cannot possibly be detected.

Our algorithm for generating regions is known as KDRP, and has a 2 phase approach to generate any arbitrary k regions of an image. Pseudocode for KDRP is shown in algorithm 1. SIFT features correspond to areas of a strong

change of the gradient of the image; these areas have been known to be where distinguishing points of the image exist [10]. Our theory is that by capturing more regions that have a high density of these descriptive keypoints, we will have at least one region that is an accurate bound of the object we are trying to detect.

Algorithm 1 Keypoint Density Region Proposal algorithm.

- 1: **procedure** KDRP($image, regions_needed$) $\triangleright image$ is the pixel matrix, $regions_needed$ is an integer
 - 2: Generate SIFT-like keypoints on $image$
 - 3: Slide windows over $image$ to get μ, σ of keypoint density
 - 4: **while** $len(len_output) < regions_needed$ **do**
 - 5: Randomly generate region r in image
 - 6: Calculate percentile p sample of r with μ, σ
 - 7: **if** Binomial trial(p) successful **then**
 - 8: Append r to $output$
 - 9: **return** $output$
-

1. **Generate Keypoints-** Standard keypoint detection algorithms are used to find and plot points of interest on the image. We performed cross validation using permutations of 8 keypoint methods on the UEC food dataset, and CUB bird dataset. The greatest performance came from *Shi Tomasi* features, *ORB* features, and *STAR* features. This differs from the existing method of segmentation and nearest neighbor search of color segments, under the assumption that regions will contain strong corners and edges of the image, which is needed for recognition and detection.
2. **Region Hypothesis-** A square window is slid over the image at a uniform stride, and density of key points from Stage 1 is sampled. Once we know the mean and standard deviation of the region density, we begin stochastically generating regions (not of fixed width to height ratios), and examine how the density of this random region compares to the baseline just established.



Figure 3: KDRP example of Max Scherzer batting. Red keypoints are *ORB* features, green keypoints are *STAR* features. Only 5% of the regions are shown for increased visibility.

For a region in the n^{th} density percentile, we keep this image for detection following a binomial sampling with an $n\%$ chance of success. The reason for this sampling is that it has been shown before to reduce bias of algorithms [13]. This differs from traditional rCNN with selective search in the time required to produce the equivalent number of regions.

As can be seen from algorithm 1, a major advantage of KDRP is that the exact number of regions to be used can be selected. Selective Search implementation in matlab (as used in [4] [3]) has a *fast* and *slow* mode, but is unable to generate more regions than the *slow* mode generates.

We theorize that KDRP’s simplicity of generating keypoint descriptors (which was been shown to be fast [10]), and then randomly generating regions to be much faster than converting an image into multiple color spaces, and performing a greedy nearest neighbor search as done in selective search. An example of regions generated with KDRP is seen in figure 3.

4. Experiments

4.1. Hypothesis

In this experiment, there are two different results that we wish to measure to get a comprehensive evaluation of selective search against. The first is accuracy of detection; to be successful we would like to show using a paired t-test there is not a statistically significant difference between detection with KDRP, and detection with selective search. The other metric we are testing is execution time of the pipeline; we

Table 1: Description of datasets used

Dataset	Train/Test	Num Classes	Instances per image
UEC-100 [11]	10205/2966	100	variable, [0, n]
CUB-200 [15]	5994/5794	200	1

anticipate the KDRP detection pipeline to take about 50% less time than the selective search detection pipeline.

4.2. Datasets

We use two datasets in our experiment; the UEC-100 food dataset, and the CUB-200 bird dataset. Since both of these datasets include bounding box coordinates, they are both suitable for detection. The characteristics of the datasets are given in table 1.

The two datasets both offer unique challenges for detection, beyond them both being fine grained. The CUB dataset is notoriously difficult, especially when not using the additional metadata annotations such as beak and wing location. The UEC dataset on the other hand has a variable number of groundtruth objects in the in the image, so we cannot hard code the algorithm to search for a certain number of items. The way that this is handled was discussed in section 2.5.

4.3. Experimental Design

4.3.1 Algorithmic Differences

Our goal is to keep as much of the pipeline the same as possible, to get a direct comparison between selective search and KDRP. Looking at the complete detection pipeline in section 2, the only difference is the region proposal step. This of course effects the features generated, the non-maximum suppression, and the hypothesis selection, but that is being measured by the overall accuracy and time.

For training, we fine tuned the pre-existing imagenet model VGG16 [12], only modifying the output layers of the classification and bounding box regression steps. No other layers were modified (fully connected or convolutional). The finetuning was over the course of 5,000,000 iterations with a base learning rate of .01, decreasing by a factor of 10 every 500,000 iterations. Momentum term was set to .9, and weight decay was set to .0005. The training proposed regions were generated by selective search for both the selective search and KDRP pipelines.

Using the matlab code of [14], selective search was used to generate region proposals for selective search. The number of regions per image was variable, as it is dependent on color features of the image, although it averaged to around 2,000 per image. The number of regions generated with KDRP was set to 2,250; this was the maximum number of regions we found that we could use without the total time exceeding our predetermined threshold of 1 second per image. The study of the effect of using less regions per image

warranted investigation, and is in the ablation studies section 4.5.

Featurization has no tunable parameters, non maximum suppression was applied for high probability windows with an overlap exceeding 30%, as per [4], and the threshold for selecting a region for hypothesis selection when there is an unknown amount of objects in the ground truth was set to $p > .88$. This value was determined through cross validation.

4.3.2 Evaluation Metrics

There are two hypothesis that we wish to test; time and accuracy. For time we would like to show that the mean time to run the full KDRP detection pipeline on an image is less than 50% the selective search detection pipeline. Assuming that the mean pipeline detection time of selective search is μ_{SS} , and the mean pipeline detection time of KDRP is μ_{KDRP} , then our hypothesis are

$$H_0 : \frac{1}{2} \times (\mu_{SS}) < \mu_{KDRP}, \quad (1a)$$

$$H_a : \frac{1}{2} \times (\mu_{SS}) \geq \mu_{KDRP}. \quad (1b)$$

For accuracy, we consider an object to be correctly detected if the intersection over union of the groundtruth and bounding boxes is greater than 50%. Correctly identified groundtruths are True Positives (TP), groundtruths not covered by at least 50% of the detection box, or of the wrong class are False Negatives (FN), and bounding boxes that do not correctly detect exactly 1 object are False Positives (FP). Our average accuracy is given as $\mu_x = \frac{TP}{TP+FP+FN}$ for method x , making our hypothesis

$$H_0 : \mu_{KDRP} < \mu_{SS}, \quad (2a)$$

$$H_a : \mu_{KDRP} \geq \mu_{SS}. \quad (2b)$$

4.4. Results

During both of the experiments, time was kept, and the results are recorded in table 2. The numbers of interest are in bold; for the UEC food dataset, the KDRP detection pipeline executes in 48.9% the time that the selective search pipeline does, and for the CUB dataset, the KDRP pipeline executes in 48.4% the time that the selective search pipeline does. Because the execution time of KDRP for both datasets is less than 50% of selective search, we reject the null hypothesis 1a in favor of the alternate 1b. With detection time of less than one second, we would be able to do detection realtime in a 1 frame per second surveillance video.

Although time is a critical component of real time detection, it is worthless without the ability to predict with

Table 2: Timed results for KDRP pipeline versus SS pipeline

Dataset	Task	KDRP time	SS time	KDRP ÷ SS
UEC	RP	.603 s	1.67 s	.361
	FE	.346 s	.265 s	1.305
	NS	.004 s	.003 s	1.333
	HS	.005 s	.006 s	.833
	Total	.96 s	1.96 s	.489
CUB	RP	.605 s	1.72 s	.351
	FE	.354 s	.267 s	1.32
	NS	.005 s	.005 s	1.00
	HS	.004 s	.005 s	.80
	Total	0.968	1.997	.484

Table 3: Accuracy, and p values comparing likelihood of generating function differences.

Dataset	KDRP accuracy	SS accuracy	n	p-value
CUB (k=1)	67.94	68.26	5794	.758
CUB	66.24	65.18	5794	.00009
UEC	68.03	68.19	2966	.546

accuracy; a trivial algorithm that always guessed a fixed location could process thousands of images per second. For comparison of accuracy we used a paired t-test, comparing the accuracy on each individual image using both KDRP and selective search for both corpora, the results are shown in table 3.

Based on the p-values calculated from the paired t-test, we can reject the null hypothesis 2a, and accept the alternate 2b. The only sufficiently large/small p-value to say that there is a noticeable difference in the algorithms is the CUB dataset when we do not force hypothesis selection to select exactly 1 detection, and the KDRP pipeline outperformed selective search by over 1%. By our results on the two datasets of table 3, we can say that KDRP is no worse accurate than selective search for accuracy of detections.

4.5. Ablation Studies

Instead of just assuming that we need as many regions as possible that are able to run in less than one second is necessary for such performance, we wish to decrease the number of regions and see how the performance increases or decreases correlated with decreased regions. The results of ablation are shown in figure 4.

Figure 4 suggests a time/accuracy trade-off is available for the KDRP algorithm; when we are willing to spend a long amount of time is no less accurate than the selective search algorithm (alternate hypothesis from equation 2b).

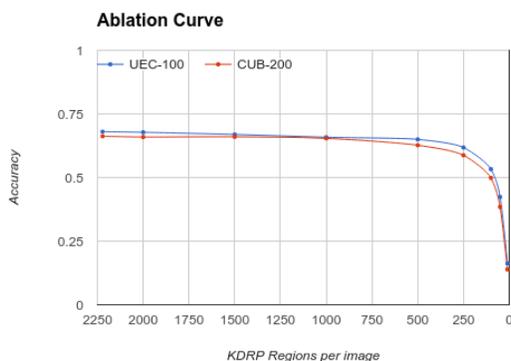


Figure 4: Effect of less regions per image on detection accuracy for CUB-200 and UEC-100 datasets

5. Conclusion

Our goal was to construct a pipeline that would be able to place accurate bounding boxes around objects of interest in a dataset in less than one second, so that it could be used in a low frame per second surveillance camera setting. The initial r-CNN models made it unable to do this, because convolution was such a time consuming process, but with the fast-rCNN of Girshick [3] this became possible. KDRP in this setting is an improvement on the selective search algorithm because it is not worse in terms of performance, but is customizable with the number of regions, and can generate more regions than selective search in less than half the time.

Future work involves improving upon KDRP further, by potentially adding different region aspect ratios or scales upon acceptance; the nature of the fast r-CNN network makes additional convolutions take very slightly more time, and the more regions that we have, the more likely we are to succeed in detection.

References

- [1] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(9):1627–1645, 2010.
- [2] Z. Ge, C. McCool, C. Sanderson, and P. I. Corke. Modelling local deep convolutional neural network features to improve fine-grained image classification. *CoRR*, abs/1502.07802, 2015.
- [3] R. Girshick. Fast r-cnn. *arXiv preprint arXiv:1504.08083*, 2015.
- [4] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.
- [5] C. Goering, E. Rodner, A. Freytag, and J. Denzler. Nonparametric part transfer for fine-grained recognition. In *Proceed-*

- ings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2489–2496, June 2014.
- [6] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *CoRR*, abs/1406.4729, 2014.
- [7] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. B. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *CoRR*, abs/1408.5093, 2014.
- [8] Y. Kawano and K. Yanai. Food image recognition with deep convolutional features. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication, UbiComp '14 Adjunct*, pages 589–593, New York, NY, USA, 2014. ACM.
- [9] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- [10] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.
- [11] Y. Matsuda, H. Hoashi, and K. Yanai. Recognition of multiple-food images by detecting candidate regions. In *Proceedings of International Conference on Multimedia and Expo (ICME)*, pages 25–30. IEEE, 2012.
- [12] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.
- [13] J. Turner, A. Page, T. Mohsenin, and T. Oates. Deep belief networks used on high resolution multichannel electroencephalography data for seizure detection. In *2014 AAAI Spring Symposium Series*, 2014.
- [14] J. Uijlings, K. van de Sande, T. Gevers, and A. Smeulders. Selective search for object recognition. *International Journal of Computer Vision*, 2013.
- [15] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The caltech-ucsd birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.
- [16] K. Yanai, T. Kaneko, and Y. Kawano. Real-time photo mining from the twitter stream: Event photo discovery and food photo detection. In *Multimedia (ISM), 2014 IEEE International Symposium on*, pages 295–302, Dec 2014.