Determinantal Point Process as an alternative to NMS

Samik Some
samiks@iitk.ac.in
Dept. of CSE
Indian Institute of Technology, Kanpur
India

Mithun Das Gupta
migupta@microsoft.com
Microsoft
Hyderabad,
India

Vinay P. Namboodiri
vpn22@bath.ac.uk
Dept. of CS
University of Bath
United Kingdom

Abstract

We present a determinantal point process (DPP) inspired alternative to non-maximum suppression (NMS) which has become an integral step in all state-of-the-art object detection frameworks. DPPs have been shown to encourage diversity in subset selection problems [10]. We pose NMS as a subset selection problem and posit that directly incorporating DPP like framework can improve the overall performance of the object detection system. We propose an optimization problem which takes the same inputs as NMS, but introduces a novel sub-modularity based diverse subset selection functional. Our results strongly indicate that the modifications proposed in this paper can provide consistent improvements to state-of-the-art object detection pipelines.

1 Introduction

Object detection has gained a lot of momentum over the past few years, especially due to its application in a wide variety of fields such as autonomous driving, manufacturing industry, traffic and law enforcement [15] applications. The primary approaches for object detection can be loosely divided into a few dominant approaches, including sliding-window Deformable Parts Models [7, 38], region proposal with classification [8, 35], and location regression with deep learning [33, 34]. Almost all of the current day object detection frameworks follow a three step process, namely: (1) proposing a search space of windows, which has mostly converged to the output of a region proposal network (RPN), (2) scoring/ refining the window with a classifier/regressor, and (3) merging or discarding windows that might belong to the same object. This last stage is commonly referred to as "non-maximum suppression" (NMS) [7, 9, 12, 24, 29, 30].

NMS is a fairly simple test time post-processing routine. Maintaining parity with some of the published research in this area, we denote the basic NMS step as GreedyNMS [7, 13, 31] in this paper. The GreedyNMS algorithm, greedily selects high scoring detected windows and iteratively discards spatially close-by less confident neighbours with the assumption that

the neighbors are likely to cover the same object. Specifically, all the candidate windows are either selected or rejected based on the following procedure: first, the highest-scored window is marked as retained, and all those overlapping with it by more than some threshold (e.g. 30%) intersection-over-union (IoU) are marked as suppressed; then, the next highest-scored window neither retained nor suppressed is marked as retained, and again all others sufficiently-overlapping candidate windows are marked for rejection. This process is repeated until all windows are marked as either retained or suppressed. The retained windows then constitute the final set of detected proposals. Although GreedyNMS continues to be the method of choice due to its simplicity, it inherently suffers from significant conceptual shortcomings. GreedyNMS is based on the simple intuition that similar detection windows which are close in spatial sense, should be suppressed. It controls the influence span by a single threshold parameter which is chosen to keep the region of suppression not too wide, since a wide suppression would remove close-by high scoring detected windows that are likely to be false positives that hurt precision. If objects are indeed close to each other, such as persons in crowded scenes, then the windows detected close to each other should be counted as true positives, in which case suppression should be narrow to improve recall. Achieving both these targets with a single tuning parameter seems hard and indeed this inherent limitation is the biggest shortcoming of the GreedyNMS routine.

One of the seminal works in general object detection was the R-CNN model by Girshick et al. [9], which replaced the feature extraction and classifier pipeline by a neural network, resulting in almost two times performance gain on Pascal VOC. Another significant improvement was the F-RCNN model by Ren et al. [30], which absorbed the object proposal generation into the network, while YOLO [29] avoided proposals altogether, leading to both speed and quality improvements. A general trend towards end-to-end trainable object detection models has been the norm in recent times. NMS is one step in the object detection pipeline that is based on post-processing. Though a few works have tried to incorporate end-to-end trainable pipelines [13, 36], so far it is not widely accepted. We would like to retain the post-processing nature of NMS in order for our approach to be incorporated in any pipeline.

In this work, we propose a principled improvement of the core NMS step by incorporating a DPP cost function in it. This development leads to an overall improvement of the NMS step and can be incorporated to existing NMS implementation with minimal changes. The theoretical guarantees afforded by a DPP based cost function lets us bridge the aforementioned gaps in fundamental ways, namely:

- We improve the performance of NMS staying in the standard flow, wherein NMS still stays outside the main neural loop in state-of-the-art (SOTA) object detection implementations,
- The proposed system does not need any additional training as in [1, 13] or modification of standard cost functions as in [36].
- the proposed system works with the same inputs as NMS, namely proposal windows and their score, and introduces a new way to select diverse proposal subsets.

2 Related Work

Wan et al. [36], proposed to integrate the NMS cost function into the unified loss function of a joint optimization system which had a neural featurizer, a deformable parts model and an

NMS block. Since the NMS block was outside the neural loop, this implementation was similar to GreedyNMS, albeit with application dependent loss function. This work mentioned faster RCNN based models but did not use them and hence the baseline is considerably lower than the current day works. Hosang et al. [13], propose to absorb the entire NMS step into a neural network. The authors claim that the suppression width parameter can be better estimated by a neural net and hence it should be data dependent rather than an empirically chosen one. Even though this argument has merit, the adoption in state-of-the-art algorithms is still missing. Azadi et al. [1] propose a similar method, where they use DPP as an alternative to NMS. However, in their method DPP is implemented as a trainable layer and not as a simple plug and play module.

Informative subset selection problems arise in many applications where a small number of items must be chosen to represent or cover a much larger set; for instance, text summarization [22, 27], document and image search [19, 28, 37], sensor placement [11], viral marketing [16], and many others. Recently, probabilistic models extending determinantal point processes (DPPs) [4, 25] were proposed for several such problems [8, 18, 19]. DPP was first used to characterize the Pauli exclusion principle, which states that two identical particles cannot occupy the same quantum state simultaneously [25]. DPPs offer computationally attractive properties, including exact and efficient computation of marginals [25], sampling [14, 19], and (partial) parameter estimation [20]. DPP has emerged as a powerful method for selecting a diverse subset from a "ground set" of items [21].

2.1 Determinantal Point Processes

To define a determinantal point process (DPP) let us first consider the definition of a point process itself. A point process $\mathcal{P}$ on a ground set $\mathcal{Y}$ refers to a probability distribution on finite subsets of $\mathcal{Y}$. Let $\mathcal{Y}$ be a discrete set represented as $\mathcal{Y} = \{1, 2, \ldots, N\}$, then $\mathcal{P}$ defines a probability distribution on $2^{\mathcal{Y}}$, the powerset of $\mathcal{Y}$.

For $\mathcal{P}$ to be called a determinantal process, it should satisfy the following condition for all $A \subseteq \mathcal{Y}$:

$$\mathcal{P}(A \subseteq \mathbf{Y}) = \det(\mathbf{K}_A) \tag{1}$$

where, $\mathbf{Y}$ is a random subset drawn according to $\mathcal{P}$, $\mathbf{K}$ is a real, symmetric $N \times N$ matrix indexed by the elements of $\mathcal{Y}$, and $\mathbf{K}_A$ is the submatrix obtained from $\mathbf{K}$ when only the entries indexed by elements of A are considered. $\mathbf{K}$ is referred to as the marginal kernel.

The above definition of DPP defines $\mathcal{P}$ in terms of marginal probabilities using $\mathbf{K}$. There exists an alternative definition for a slightly restricted class of DPPs which allow us to model the probability of a subset directly. These are known as L-ensembles [3] and are much easier to work with practically. We define $\mathcal{P}$ using L-ensembles as follows:

$$\mathcal{P}_L(\mathbf{Y} = Y) \propto \det(\mathbf{L}_Y) \tag{2}$$

where, $\mathbf{Y}$ represents the random variable as earlier, $\mathbf{L}$ is a real, symmetric $N \times N$ matrix indexed by elements of $\mathcal{Y}$, and $\mathbf{L}_Y$ is similarly the submatrix of $\mathbf{L}$ indexed by elements of Y. To satisfy the fact that probability measures must always be positive, $\mathbf{L}$ has to be positive semidefinite (psd). The normalization constant for $\mathcal{P}$ can be obtained in closed form since

$$\sum_{Y \subseteq \mathcal{Y}} \det(\mathbf{L}_Y) = \det(\mathbf{L} + \mathbf{I}) \tag{3}$$

Thus, using L-ensembles we get a direct probability distribution on the subsets of $\mathcal{Y}$ as:

$$\mathcal{P}_L(\mathbf{Y} = Y) = \frac{\det(\mathbf{L}_Y)}{\det(\mathbf{L}+\mathbf{I})} \tag{4}$$

Exact MAP inference of DPP is a NP-hard problem [17]. However, approximation of the DPP formulation, notably,

$$f(S) = \log\det \mathbf{L}_Y \tag{5}$$

is a non-monotone submodular function [21], which has been the function of choice for most of the work in this domain [8, 10].

3 Method

We propose replacing GreedyNMS in detection pipelines with a DPP proposed in Eq. 5. Generally in a detection pipeline NMS is applied on final detections to filter them and keep only one detection per object. Faster RCNN, not only performs NMS on the final detections but also on the region proposals returned by the Region Proposal Network (RPN). We posit that the NMS after the RPN stage would gain with diversified selection, since its task is to retain all the informative regions. The second NMS which comes after the softmax stage just filters the boxes obtained for each class independently and hence does not gain with diversity preserving methods. Consequently, we replace the first stage NMS after the RPN layer in this work. As such it is here that we apply DPP. The basic idea is to use DPP to select or filter the proposals instead of NMS. Thus our ground set $\mathcal{Y}$ consists of the proposals returned by the RPN. GreedyNMS uses the box coordinates to compute an intersection over union metric and also the score provided by the RPN to filter the windows. We use the exact same two features for our method. To construct our L matrix we make use of 2 features.

- Scores for the proposals from the RPN (s_i)
- Intersection over union (IoU) of the proposals (IoU_{ij})

where $\{i \in \mathcal{Y}\}$. These features are then combined to form the $\mathbf{L}$ matrix given by,

$$\mathbf{L} = \alpha[e^{\mathbf{s}}e^{\mathbf{s}^T}] \odot \mathbf{IoU} \tag{6}$$

whose elements are written as follows:

$$L_{ij} = \alpha e^{s_i}\text{IoU}_{ij}e^{s_j} \tag{7}$$

where $\alpha > 1$ is a scaling constant provided to bias the selection process towards selecting larger subsets, and the values of $s_i \in (0,1) \forall i \in \mathcal{Y}$, $\mathbf{s}$ is a column vector with s_i as its i^{th} element, $e^{\mathbf{s}}$ represents the element-wise exponentiation of $\mathbf{s}$, $\mathbf{IoU}$ is a matrix composed of IoU_{ij}, and $\odot$ represents the Hadamard product of matrices. Note that the interaction of the two score s_i and s_j can be combined in many different ways. In this work we use the exponent function to bring it closer to the smooth maximum approximation, along with the large weighting constant α[1].

Lemma 1. $\mathbf{L} = \alpha[e^{\mathbf{s}}e^{\mathbf{s}^T}] \odot \mathbf{IoU}$ *is positive semidefinite.*

[1] https://en.wikipedia.org/wiki/Smooth_maximum

Proof. The constituents of the $\mathbf{L}$ matrix in the above manner can be proven to be individually positive semidefinite by the following three arguments. a) $e^{\mathbf{s}}e^{\mathbf{s}^T}$ is positive semidefinite since it is of the form $\mathbf{x}\mathbf{x}^T$, b) The $\mathbf{IoU}$ matrix, also known as the Jaccard similarity matrix, can be shown to be positive **definite** [2], and c) According to the Schur product theorem[2], the Hadamard product (elementwise multiplication product) of two positive semidefinite matrices is also positive semidefinite. Thus, the product $[e^{\mathbf{s}}e^{\mathbf{s}^T}] \odot \mathbf{IoU}$ is also positive semidefinite. □

The final probability of a selecting $Y \subseteq \mathcal{Y}$ can now be written as:

$$\mathcal{P}(\mathbf{Y} = Y) \propto \det(\alpha[e^{\mathbf{s}_Y} e^{\mathbf{s}_Y^T}] \odot \mathbf{IoU}_Y) = \alpha^{|Y|} \det([e^{\mathbf{s}_Y} e^{\mathbf{s}_Y^T}] \odot \mathbf{IoU}_Y) \tag{8}$$

Note that due to the determinant operation, the weighting term α is raised to the power $|Y|$, which is the size of the subset to be selected. Explicitly making the subset size influence the probability is important since the marginal gain decreases with increase in subset size. Hence, the weighting term acts as a counter to the diminishing marginal gain, which is due to the sub-modular nature of the objective function.

To obtain the set which maximizes the above probability we need to use some approximation technique. One choice is the simple greedy method. Before arriving at the final formulation we need the following lemmas.

Lemma 2. *The principle sub-matrices of a psd matrix are also psd.*

According to this lemma any principle submatrix of $\mathbf{L}$ indexed by the set Y is also positive semidefinite. Hence, $\mathbf{L} \succeq 0$ leads to all subsets $\mathbf{L}_Y \succeq 0$.

Lemma 3. $\log\det \mathbf{L}_Y$ *for a psd matrix* $\mathbf{L}_Y$ *is submodular.*

Proof. Submodularity of DPPs can be established by the geometrical argument as shown in [21]. □

Connecting all the lemmas, we can claim that all principal submatrices of $\mathbf{L} \succeq 0$ are themselves $\mathbf{L}_Y \succeq 0$. Finally, invoking Lemma. 3 and extending it to the current setting, we can maximize $\log\det \mathbf{L}_Y$ to obtain the approximate MAP set. As such the final formulation for DPP based NMS is given by:

$$\arg\max_Y \quad \log\det \mathbf{L}_Y = \log\det(\alpha[e^{\mathbf{s}_Y} e^{\mathbf{s}_Y^T}] \odot \mathbf{IoU}_Y) \tag{9}$$

We employ a greedy algorithm to maximize this cost function, where at every iteration we add the element which has the highest marginal gain with respect to the currently selected set. While greedy algorithms are not optimal in general, for monotone sub-modular problems they have well-defined approximation bounds [21]. Our final algorithm is given as follows:

We utilise a heap-based implementation to speed up the algorithm as proposed by Minoux [26]. The additional check for positivity of the marginal gain in the greedy algorithm, ensures that the value of our currently selected set always increases at every iteration.

[2] https://en.wikipedia.org/wiki/Schur_product_theorem

Input: RPN proposals $\mathcal{Y}$, RPN scores $\mathbf{s}$, parameter α, maximum boxes k
Output: Filtered proposals Y
Compute **IoU** matrix using $\mathcal{Y}$
$\mathbf{L} \leftarrow \alpha[e^{\mathbf{s}} e^{\mathbf{s}^T}] \odot \mathbf{IoU}$
$Y \leftarrow \text{Greedy}(\mathcal{Y}, k)$
return Y

Function Greedy($\mathcal{Y}, k$):
$X \leftarrow \mathcal{Y}, Y \leftarrow \emptyset$
while $|Y| < k$ **do**
  $e \leftarrow \max_{i \in X} f(Y \cup i) - f(Y)$ Eq.9
  if $f(Y \cup e) - f(Y) \leq 0$ **then**
    return Y
  end if
  $Y \leftarrow Y \cup e$
  $X \leftarrow X \setminus e$
end while
return Y

4 Experiments and Results

In this section we provide details about the experiments performed and discuss the various results obtained. We work with a standard PyTorch[3] version of faster-RCNN[4] and use VGG-16 as the backbone network. We maintain all the default settings to make the experiments as reproducible as possible. All our experiments are subsequently based on replacing the NMS module after the RPN block, with our own proposed method. We perform experiments on MS-COCO [23] and PASCAL VOC 2007 [5] datasets. In all cases we train the network for 6 epochs on the default training splits, which are mentioned in the respective dataset subsections. During training we do not use DPP. We replace the NMS module with DPP during test time. We believe that the merit of existing GreedyNMS is its simplicity and the fact that it does not need to be tuned much for any experiment. Consequently, we propose a similar setting where the default parameter configuration works well for most applications. We evaluate a few variants of our model to understand the different modes of its operation and then converge onto one model with default parameter recommendation. The models in the experiments are as follows:

- gNMS_x: This is the standard Greedy NMS algorithm with a maximum of $x = \{300, 400\}$ selected windows. Note that gNMS_{300} is the default setting in most SOTA object detection pipelines with GreedyNMS.
- DPP_x^{α}: This refers to DPP with bias factor $\alpha = 5$, (Eq.7), with a maximum of $x = \{300, 400\}$ selected boxes.

For all of the above models the number of input proposals (the ones returned by the RPN) are limited to a maximum of $|\mathcal{Y}| = 6000$ windows. We present comparison against the previous works which are most similar to our in spirit. **Neural-NMS** represents the deep network based NMS proposed by Hosang et al. [13]. They train their own deep network to replace Greedy NMS and plug it in after the detection step of Faster RCNN. This is a deviation from the generic way of using NMS, where it is plugged after the RPN but before the detection stage. **MP-NMS** refers to the message passing based NMS algorithm proposed by Rothe et al. [31]. We also compare against the end to end integration of convolution network, deformable parts model and NMS into one unified pipeline, proposed by Wan et al. [36]. Though this method, denoted as **CN-DPM-NMS**, does not use F-RCNN like network, but

[3] https://pytorch.org/
[4] https://github.com/jwyang/faster-rcnn.pytorch

Model	$AP_{0.5}$	$AP_{0.5}^{0.95}$
$gNMS_{300}$	47.7	27.3
$gNMS_{400}$	48.0	27.4
DPP_{300}^{5}	47.8	27.4
DPP_{400}^{5}	**48.1**	**27.5**
Neural-NMS [13]	-	24.3
LDDP [1]	32.2	15.5

Table 1: NMS vs DPP experiments on MS COCO (All Classes)

Model	$AP_{0.5}$	$AP_{0.5}^{0.95}$
$gNMS_{300}$	69.8	40.2
$gNMS_{400}$	70.0	40.3
DPP_{300}^{5}	69.9	40.3
DPP_{400}^{5}	**70.2**	**40.6**
Neural-NMS [13]	67.3	36.9

Table 2: NMS vs DPP experiments on MS COCO (Persons)

the results can still work as a baseline comparison. Finally, **LDDP** refers to the pipeline proposed by Azadi et al. [1] where they use a trainable DPP layer as an alternative to NMS. All experiments were performed on a system with a i7-6850k CPU, a GTX 1080 Ti GPU and 64GB RAM. We implement DPP in C++ using the Eigen3 framework and run it on the CPU. When compared to a basic C++ CPU implementation of NMS we get comparable runtime upto approximately 100 selections for which NMS takes about 0.3s/image whereas DPP takes about 0.5s/image. The runtime of DPP however scales significantly with the number of selected proposals since the complexity involved is approximately $O(k^4)$, where k is the number of proposals selected.

4.1 MS-COCO

For the MS-COCO dataset the model was trained on the training and valminusminival data splits and was tested on the minival split. In the results $AP_{0.5}$ represents average precision (AP) calculated considering 50% overlap with ground truth. $AP_{0.5}^{0.95}$ represents AP averaged over multiple overlap thresholds ranging from 50% to 95% in steps of 5%. The results for multi-class classification are shown in Table. 1. Results for MS-COCO person detection class has been reported by several authors and hence we also report it separately in Table. 2.

4.2 PASCAL VOC

For PASCAL VOC 2007 we perform several experiments. We start off by evaluating Greedy NMS vs several variants of DPP over each class individually. For these experiments Faster-RCNN was trained on the training and validation sets and tested on the test set for PASCAL VOC 2007. For assigning proposed bounding boxes to ground truth detections PASCAL VOC considers overlaps greater than 50% to be correct detections. This evaluation criteria is denoted as $AP_{0.5}$. Table. 3 shows the results of class wise performance. Average performance across all classes along with comparative methods are shown in Table. 4.

4.3 Varying the maximum window and scaling parameters

We perform more experiments to identify the core strengths of the proposed algorithm. The maximum number of windows returned by the algorithm is a parameter, which has a direct implication on the run-time of the algorithm. As such, the minimum value at which acceptable results are obtained needs to be selected. Keeping $\alpha = 5$, we run the algorithm with different values of $k \in \{100, 200, 300, 400\}$. The results are shown in Fig. 1. Note that, for the setting $k = 200$, our algorithm already beats $gNMS_{300}$ and is almost at par with $gNMS_{400}$.

Model	aeroplane	bicycle	bird	boat	bottle	bus	car	cat	chair	cow
gNMS$_{300}$	67.66	77.39	67.15	54.36	54.43	**78.40**	**85.52**	85.67	48.45	**79.78**
gNMS$_{400}$	68.18	77.96	**67.89**	54.61	54.72	78.17	85.50	**85.98**	**48.61**	79.73
DPP$^{5}_{300}$	69.93	77.22	65.75	54.79	55.43	78.25	85.05	82.40	47.93	80.36
DPP$^{5}_{400}$	**69.94**	**78.51**	65.42	**55.13**	**55.49**	77.90	85.29	83.53	48.01	78.20
Model	diningtable	dog	horse	motorbike	person	pottedplant	sheep	sofa	train	tvmonitor
gNMS$_{300}$	61.50	78.89	82.14	75.61	77.26	40.65	70.42	**63.77**	74.94	72.19
gNMS$_{400}$	61.10	78.59	82.32	75.39	77.23	40.96	70.16	**63.77**	**75.28**	71.76
DPP$^{5}_{300}$	63.35	80.97	**83.10**	75.67	**77.60**	42.22	**71.45**	63.71	74.89	72.61
DPP$^{5}_{400}$	**63.91**	**81.26**	83.06	**76.17**	77.54	**42.63**	70.34	63.49	75.11	**72.49**

Table 3: NMS vs DPP experiments on PASCAL VOC 2007 (Classwise)

Model	AP$_{0.5}$
gNMS$_{300}$	69.81
gNMS$_{400}$	69.90
DPP$^{5}_{300}$	70.13
DPP$^{5}_{400}$	**70.17**
MP-NMS [32]	56.14
CN-DPM-NMS [36]	46.50
LDDP [1]	62.21

Table 4: Average performance on PASCAL VOC 2007

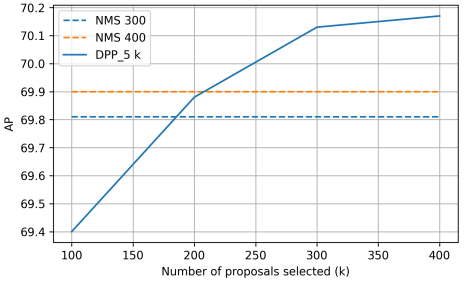

Figure 1: Comparison of varying the maximum window parameter k in our algorithm.

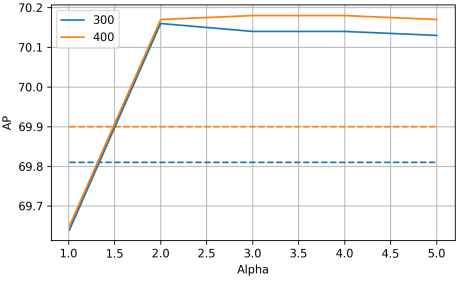

Figure 2: Comparison of varying the scaling parameter α in our algorithm. The horizontal dotted lines denote GreedyNMS.

This is the key contribution of introducing diversified window selection in the NMS algorithm, wherein, a diverse set of lesser number of proposal windows ($k = 200$) outperform a larger set of proposal windows ($k = 300$) selected by GreedyNMS.

Similarly, we also perform experiments to observe the effect of the scaling parameter α on the detection performance. We test different values of $\alpha \in \{1.01, 2, 3, 4, 5\}$ while keeping the maximum number of windows k fixed. The results are shown in Fig. 2. The proposed method beats GreedyNMS for $\alpha > 1.5$ for both 300 and 400 region proposal selections.

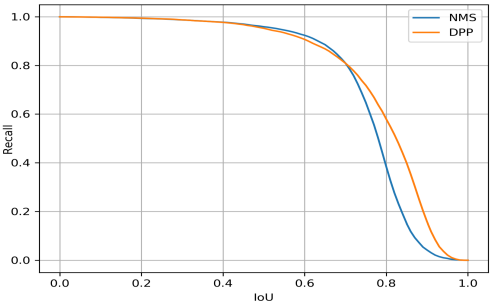

Figure 3: IoU vs Recall plot for gNMS_{400} and DPP^{5}_{400}

4.4 IoU vs Recall

In [32] the authors propose evaluating the recall rate of detections at different IoU thresholds to measure how well fitting the selected bounding boxes are. We perform a similar evaluation, where we plot the recall with respect to the ground truth boxes against varying IoU thresholds (Fig. 3). As NMS/DPP is applied on the RPN proposals in Faster-RCNN, we directly consider these proposals before any bounding box regression for this experiment. The IoU threshold determines whether a predicted bounding box is matched to a ground truth object or not. The AUC scores for the two curves are 0.7575 for Greedy NMS and 0.7869 for the DPP based method. In addition to having higher AUC we also note that the DPP based method becomes especially better when more precise bounding boxes are required ($IoU > 0.7$). This indicates that DPP chooses better fitting bounding boxes than Greedy NMS.

5 Qualitative Results

We show a few qualitative results in Fig. 4 using similar parameter settings as used for all the previous results. We select images from the MS-COCO validation set and plot the region boundaries found by the two competing methods, namely gNMS_{400} and DPP^{5}_{400}. It is interesting to observe that DPP based selection works well when there is large overlap between two correct detections. DPP was able to remove some extraneous windows, such as the extra person detection for the tennis player blue cluster in Fig. 4. Similarly, it selects only meaningful windows for the collection of people in the bottom right image in the blue cluster. For images with very simple / few detections, both the methods perform at par. A few examples where NMS still performs better are shown in the green cluster in Fig. 4.

6 Conclusion and Future Work

We propose a novel integration of DPP based diverse set selection technique into the NMS paradigm. We formulate a principled cost function which uses the same two features which the traditional NMS routines use, and show that this formulation can be driven to improve on NMS accuracy by carefully selecting the bias parameter α which promotes larger subsets. The comparative results against Greedy NMS as well as other recent methods prove that the proposed method is working at par or superior than most other methods.

Figure 4: Qualitative results (best viewed in color). Blue boxes are produced by our method DPP^5_{400}, green boxes are produced by gNMS_{400}. Blue dotted cluster represents results where DPP^5_{400} performs better than gNMS_{400}. Brown cluster represents similar performance. Green cluster represents cases where gNMS_{400} seems to perform better, although the person detection is still superior for DPP^5_{400}.

References

[1] Samaneh Azadi, Jiashi Feng, and Trevor Darrell. Learning detection with diverse proposals. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[2] Mathieu Bouchard, Anne-Laure Jousselme, and Pierre-Emmanuel Doré. A proof for the positive definiteness of the jaccard index matrix. *International Journal of Approximate Reasoning*, 54(5):615–626, 2013.

[3] Victor-Emmanuel Brunel, Ankur Moitra, Philippe Rigollet, and John Urschel. Rates of estimation for determinantal point processes. In Satyen Kale and Ohad Shamir, editors, *Proceedings of the 2017 Conference on Learning Theory*, volume 65 of *Proceedings of Machine Learning Research*, pages 343–345, Amsterdam, Netherlands, 07–10 Jul 2017. PMLR.

[4] D. J. Daley and D. Vere-Jones. *An introduction to the theory of point processes. Vol. I.* Probability and its Applications (New York). Springer-Verlag, New York, second edition, 2003. Elementary theory and methods.

[5] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, Jun 2010.

[6] Moran Feldman, Amin Karbasi, and Ehsan Kazemi. Do less, get more: Streaming submodular maximization with subsampling. In *Advances in Neural Information Processing Systems*, pages 732–742, 2018.

[7] Pedro F. Felzenszwalb, Ross B. Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(9):1627–1645, September 2010. ISSN 0162-8828. doi: 10.1109/TPAMI.2009.167. URL http://dx.doi.org/10.1109/TPAMI.2009.167.

[8] Jennifer Gillenwater, Alex Kulesza, and Ben Taskar. Discovering diverse and salient threads in document collections. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 710–720, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

[9] Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *CoRR*, abs/1311.2524, 2013. URL http://dblp.uni-trier.de/db/journals/corr/corr1311.html#GirshickDDM13.

[10] Boqing Gong, Wei Lun Chao, Kristen L Grauman, and Fei Sha. Diverse sequential subset selection for supervised video summarization. *Advances in Neural Information Processing Systems*, 3:2069–2077, 2014.

[11] Carlos Guestrin, Andreas Krause, and Ajit Paul Singh. Near-optimal sensor placements in gaussian processes. In *Proceedings of the 22Nd International Conference on Machine Learning*, ICML '05, pages 265–272, New York, NY, USA, 2005. ACM.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *ECCV (4)*, volume 9908 of *Lecture Notes in Computer Science*, pages 630–645. Springer, 2016.

[13] Jan Hosang, Rodrigo Benenson, and Bernt Schiele. Learning non-maximum suppression. In *CVPR*, 2017.

[14] J. Ben Hough, Manjunath Krishnapur, Yuval Peres, and BÃ¡lint VirÃ¡g. Determinantal processes and independence. *Probab. Surveys*, 3:206–229, 2006. doi: 10.1214/154957806000000078. URL https://doi.org/10.1214/154957806000000078.

[15] Haroon Idrees, Mubarak Shah, and Ray Surette. Enhancing camera surveillance using computer vision: a research note. *Policing*, 41:292–307, 04 2018. doi: 10.1108/PIJPSM-11-2016-0158.

[16] David Kempe, Jon Kleinberg, and Éva Tardos. Influential nodes in a diffusion model for social networks. volume 3580, pages 1127–1138, 07 2005. doi: 10.1007/11523468_91.

[17] Chun-Wa Ko, Jon Lee, and Maurice Queyranne. An exact algorithm for maximum entropy sampling. *Oper. Res.*, 43(4):684–691, August 1995. ISSN 0030-364X. doi: 10.1287/opre.43.4.684. URL http://dx.doi.org/10.1287/opre.43.4.684.

[18] Alex Kulesza and Ben Taskar. Structured determinantal point processes. In J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 1171–1179. Curran Associates, Inc., 2010. URL http://papers.nips.cc/paper/3969-structured-determinantal-point-processes.pdf.

[19] Alex Kulesza and Ben Taskar. k-dpps: Fixed-size determinantal point processes. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1193–1200, 01 2011.

[20] Alex Kulesza and Ben Taskar. Learning determinantal point processes. In *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence*, UAI'11, pages 419–427, Arlington, Virginia, United States, 2011. AUAI Press.

[21] Alex Kulesza and Ben Taskar. Determinantal point processes for machine learning. *Foundations and Trends in Machine Learning*, 5(23):123–286, 2012. ISSN 1935-8237. doi: 10.1561/2200000044. URL http://dx.doi.org/10.1561/2200000044.

[22] Hui Lin and Jeff Bilmes. Multi-document summarization via budgeted maximization of submodular functions. In *In Proc. NAACL/HLT*, 2010.

[23] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollar, and Larry Zitnick. Microsoft coco: Common objects in context. In *ECCV*. European Conference on Computer Vision, September 2014.

[24] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. Ssd: Single shot multibox detector, 2015. URL http://arxiv.org/abs/1512.02325.

[25] Odile Macchi. The coincidence approach to stochastic point processes. *Advances in Applied Probability*, 7(1):83–122, 1975. doi: 10.2307/1425855.

[26] Michel Minoux. Accelerated greedy algorithms for maximizing submodular set functions. In J. Stoer, editor, *Optimization Techniques*, pages 234–243. Springer Berlin Heidelberg, 1978.

[27] Ani Nenkova, Lucy Vanderwende, and Kathleen McKeown. A compositional context sensitive multi-document summarizer: exploring the factors that influence summarization. In *Proceedings of SIGIR 2006*, January 2006.

[28] Filip Radlinski, Robert Kleinberg, and Thorsten Joachims. Learning diverse rankings with multi-armed bandits. In *Proceedings of the International Conference on Machine Learning (ICML)*, January 2008.

[29] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection, 2015. URL http://arxiv.org/abs/1506.02640.

[30] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 91–99. Curran Associates, Inc., 2015.

[31] Rasmus Rothe, Matthieu Guillaumin, and Luc Van Gool. Non-maximum suppression for object detection by passing messages between windows. In *ACCV*, 2014.

[32] Rasmus Rothe, Matthieu Guillaumin, and Luc Van Gool. Non-maximum suppression for object detection by passing messages between windows. In Daniel Cremers, Ian Reid, Hideo Saito, and Ming-Hsuan Yang, editors, *Computer Vision – ACCV 2014*, pages 290–306, Cham, 2015. Springer International Publishing.

[33] Pierre Sermanet, David Eigen, Xiang Zhang, Michael Mathieu, Robert Fergus, and Yann LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. 2nd international conference on learning representations, iclr 2014. 1 2014. 2nd International Conference on Learning Representations, ICLR 2014 ; Conference date: 14-04-2014 Through 16-04-2014.

[34] Christian Szegedy, Alexander Toshev, Dumitru Erhan, and Google Inc. Deep neural networks for object detection. In *Advances in neural information processing systems*, 2013.

[35] 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*, 104(2):154–171, 2013.

[36] Li Wan, David Eigen, and Robert Fergus. End-to-end integration of a convolutional network, deformable parts model and non-maximum suppression. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015*, volume 07-12-June-2015, pages 851–859. IEEE Computer Society, 10 2015.

[37] Yisong Yue and Thorsten Joachims. Predicting diverse subsets using structural svms. In *ICML*, 2008.

[38] Long Zhu, Yuanhao Chen, Alan L. Yuille, and William T. Freeman. Latent hierarchical structural learning for object detection. *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1062–1069, 2010.