FLEXIBLE 3D NEIGHBORHOOD CASCADE DEFORMABLE PART MODELS FOR OBJECT DETECTION

Hung Vu *Khoa Pho* *Bac Le*

VNU HCMC, University of Science, Ho Chi Minh City, Vietnam
Email: {vthung@fit, 1212185@student ,lhbac@fit}.hcmus.edu.vn

ABSTRACT

Cascade Deformable Part Models (DPMs) are cascade frameworks to speed up Deformable Part Models (DPMs), which are one of the state-of-the-art solutions for object detection. Its idea is to reject most non-object hypotheses from the early stages of detection process. By investigating the dependency between hypotheses over scales, we introduce a novel pruning method to accelerate Cascade DPM frameworks. Our scale pruning method includes two following strategies: a) rejecting the neighboring scales if the score at a certain scale is too low; b) keeping a few scales with the highest scores, instead of storing all scales per location. Next, we extend this pruning technique to the 3D neighborhood pruning and describe a novel approach to evaluate the root score efficiently without full computation as existing cascade DPM methods. Finally, look-up tables are introduced to work with flexible neighborhood whose volume varies over hypotheses. As a result, our cascade model is equipped with an efficient and aggressive pruning mechanism. Extensive experiments reveal that the proposed method is faster than the state-of-the-art methods for both problems of object detection and face detection.

Index Terms— Object Detection, Cascade, Deformable Part Model (DPM).

1. INTRODUCTION

Deformable Part Models (DPMs) [1] are one of the most popular solutions for multi-view object detection. DPMs describe the different views of an object via its components, each of which is made up of different parts. For detection process, the templates of these parts are matched against all positions and scales. Such huge search space is the main reason that slows down the performance of DPMs and limits its popularity in real applications. Some studies attempt to reduce the cost of feature extraction in DPMs, for example fast feature pyramid [2], low-cost channel features [3], HOG feature computation with lookup tables [4]. However, these methods do not resolve the primary bottleneck of DPM systems, that is the large number of cross-correlation operations between filter templates and the feature pyramid in the matching phase. Dubout and Fleuret [5] proposed to use FFT to speed up cross-correlation computations. Meanwhile, a Branch and Bound approach was proposed successfully for DPMs in [6]. Pedersoli et al. [7] was based on the matching results of root filters in low resolution images to reduce the search space of part filters in high resolution images. Cascade DPM methods [8, 4, 9] consider filter matching steps as cascade stages and use thresholds to discard non-object hypotheses as early as possible. In this work, we follow the approach of cascade DPMs because of its efficiency in speed [4, 10].

Cascade frameworks have been popularized in Computer Vision community by the seminal work of Viola and Jones [10] and other cascade systems [11, 12, 13, 14, 8, 9]. However, all of them usually evaluate hypotheses or sub-windows individually. Some notable recent efforts [15, 4] investigate the dependency between neighboring hypotheses, and hence aggressively improve the speed of the cascade frameworks. [15] introduced Crosstalk cascade that enables a detector to communicate the others in its neighborhood and stimulate or inhibit them. Yan et al. [4] proposed a 2D Neighborhood Aware cascade (NAC) to discard both low score negative hypotheses and non-maximal positive hypotheses. The results reported in [15, 4] point out a promising direction in the cascade research.

In this paper, we introduce a Flexible 3D Neighborhood cascade model for DPMs (Flex3DNB). The key contributions of this work are as follows:

a) **Scale Pruning Investigation:** We explore the capability of scale pruning in cascade model. Our idea is based on the observation that the co-occurrence of many different sized objects at the same location in an image is unlikely. It means that there usually exists one scale being associated with an object-like location. As a result, we can safely prune the other scales.

b) **3D Neighborhood Cascade:** By integrating NAC [4], we extend our scale pruning cascade to 3D neighborhood cascade. It is noted that although neighborhood with scale dimension is mentioned in [15], its usage is simple and limited.

c) **Flexible Neighborhood:** Unlike [4, 15] where neighborhood volumes are always fixed, we propose a mechanism to change the neighborhood size according to the scores of hypotheses. This can be done by discretizing the scores and

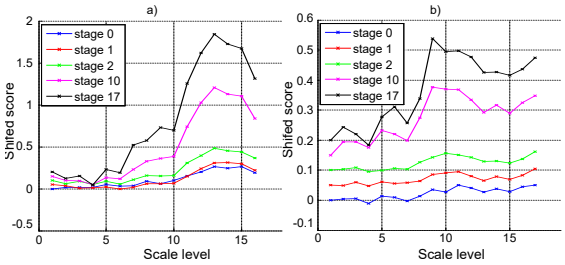

Fig. 1: The score functions of two positive hypotheses with respect to scales in a 18 stage cascade model (best view in color): a) A positive hypothesis with the same maximal scale level at all stages; b) A positive hypothesis has two potential maxima at the first stages but the lower maximum catches up in later stages to be the optimal scale, which agrees with the ground-truth object scale.

then using look-up table structures to search for the appropriate step sizes.

d) Root Score Pruning: Most DPM systems [1, 2, 6, 8] require full cross-correlation evaluation between root filters and feature maps. The acceleration techniques for root score computations only focus on the usage of low cost (low rank) root filters [4, 16] or low resolution images [7]. The proposed method allows us to prune hypotheses even in the root score evaluation step. Our technique is completely distinct from studies in the literature.

2. DPM AND CASCADE DPM

A DPM model is a mixture of m independent components each of which consists of a root filter w_0 and n part filters $\{w_t\}_{t=1}^{n}$, where w_t is the t-th part filter. Let l_0 and l_t be the locations of the root and the t-th part respectively. Suppose that an image I is represented as a feature pyramid $F = \{F_s\}_{s=1}^{N_s}$, where F_s is the feature map at the scale s of the image. Similarly, we write Γ_s and $\Gamma = \{\Gamma_s\}_{s=1}^{N_s}$ as the hypothesis set with respect to F_s and the whole feature pyramid respectively. The cumulative score of a hypothesis $\gamma \in \Gamma$ at the stage t is the appearance scores of the root and parts minus the deformation cost of the displacement:

$$g_t(\gamma) = w_0^\top \phi(l_0, I) + \sum_{i=1}^{t} w_i^\top \phi(l_i, I) - d_i^\top \theta(l_i, l_0) \quad (1)$$

where $\phi(l, I)$ is the feature vector extracted at the location l in I, d_t is the deformation coefficient of the t-th part filter and $\theta(l_i, l_0)$ denotes a quadratic deformation function .

Since there are a few positive hypotheses in an image, evaluating the scores at all locations for all parts is expensive and redundant. Cascade DPM [8] filters out unpromising hypotheses by using the cascade framework that contains many sequential stages with several thresholds at each stage. Each hypothesis travels through these stages and it can be rejected by any stage if its cumulative score is smaller than the stage thresholds. In Cascade DPMs, the PCA versions of these filters are usually prepended to the cascade pipeline to discard non-object regions with low computational cost. Furthermore, since scores should be completely computed before activating the threshold tests, there is no cascade stage associated with the first PCA-root filter and its scores are always computed densely in cascade-based DPM systems [8, 15, 4]. As a results, a Cascade DPM model with n part filters launches a cascade pipeline with $2n + 1$ stages.

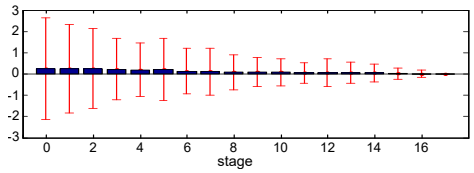

Fig. 2: The bar graph reveals the difference between the true scale level and the maximal scale at a certain stage in terms of mean and standard deviation.

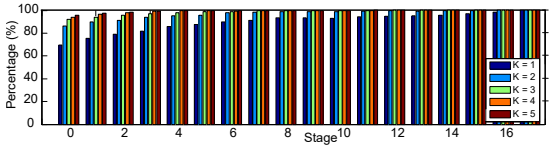

Fig. 3: The percentage of the true scale level in the candidate list at each stage is proportional to K (best view in color).

3. EARLY PREDICTION OF OPTIMAL SCALE

Two objects, possibly at different scales, rarely occur at the same location in an image. Even if this happens, only the one with the highest score can survive after the Non-Maximum Suppression (NMS) step. In other words, there is usually one scale linked to every location in final detections. Therefore, it is superfluous to calculate the intensive scores at non-maximal scales. Unfortunately, the optimal scales with the highest scores are only determined until reaching the last stage. In what follows, we aim to find out the possibility of predicting the optimal scales from early stages. Early prediction benefits us by saving our time from computing numerous incorrect scales and hence speeds up the whole framework.

We conduct an experiment by collecting 1000 positive hypotheses randomly from 20 object classes in the PASCAL VOC 2007 [17] training dataset. Then, we investigate the score functions of hypotheses with respect to scale levels at different cascade stages (Fig. 1). Interestingly, although score functions change a lot over stages, their maxima are nearly consistent. Fig. 2 describes the average discrepancy between the maximal scale levels at each stage compared to the true object scales of 1000 hypotheses. As expected, the difference is small since the mean values are extremely close to 0. The high standard deviation at the beginning stages is due to the similar values of score functions in these stages, where high scores at a few filters are not enough to distinguish the objects from the background. This consistency of scale maxima over

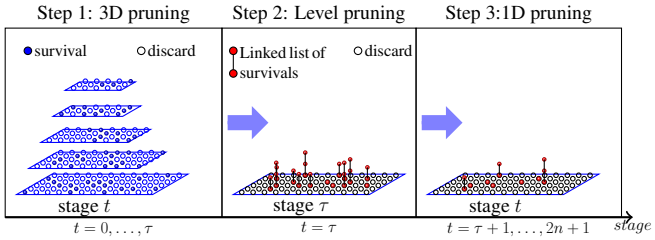

Fig. 4: The overview of Flex3DNB Cascade DPM.

stages is a strong evidence for predicting scale levels containing objects. Consequently, by defining a cut-off stage τ, we can select the highest score scale and prune all other scales.

Although we can lessen the errors by choosing a later stage τ, we expect to improve the errors at the first few stages to take advantage of early scale pruning. To that end, we propose to store a candidate list of K highest scores rather than only the top one (wherein $K = 1$). Fig. 3 points out the percentage of the true scale levels in the candidate list of 1000 positive hypotheses at different values of K over stages. Obviously, larger K enhances the prediction quality dramatically, for example over 91% for $K \geq 2$ compared to 78.9% ($K = 1$) if $t = 2$.

This experiment shows that by choosing the K highest scales, it is possible to predict the optimal scales, and therefore prune a lot of non-maximal scales from the early cascade stage τ with high accuracy. Our proposed cascade method, which integrates this scale pruning ability, will be introduced in the next session.

4. PROPOSED METHOD

Our cascade model follows a 3-step strategy including: 3D neighbor pruning over the first τ stages, a scale pruning step and finally a typical 1D pruning process. The overview of our method is given in Fig. 4.

4.1. Flexible 3D Neighbor Pruning

Neighbor pruning is a technique of removing a batch of hypotheses under a pre-defined condition such as the score of the central hypothesis below a threshold. While most studies only consider planar neighborhood, our work expands this term to the third dimension of scales.

The 3D neighbor pruning step performs in the first τ stages of cascade pipeline (including the PCA-root filter), where τ is the predefined scale pruning stage. At the stage $t = 0, \ldots, \tau$, we scan over all alive hypotheses $\gamma \in \Gamma$ and compute their scores. If the scores do not satisfy given thresholds, we prune the hypotheses and even their neighbors. The pruning techniques that are integrated into our model include individual pruning thresholds and our proposed 3D neighbor pruning thresholds.

Individual Pruning Thresholds: These techniques use thresholds to prune a single hypothesis without making any effect on other hypotheses. We have three conventional opportunities to prune hypotheses via hypothesis thresholds α_t^1 and deformation thresholds α_t^2 [1] and semi-positive threshold α_t^3 [4]. The hypothesis thresholds provide a mechanism to get rid of individual low score hypotheses while the deformation thresholds avoid an exhaustive search for optimal part configuration. By contrast, the semi-positive thresholds prevent from passing too many positive hypotheses of the same object through the cascade. According to [1, 4], these thresholds are learned automatically by investigating the labeled object hypotheses in training set.

Flexible 3D Neighborhood and Multiple thresholds: Our idea of 3D neighborhood pruning is that we discard a hypothesis γ and its neighbors in 3 dimensional feature pyramid if its score is below a threshold. When an hypothesis has a low score, it is likely that the scores of its neighbors are low also. As a result, we can speed up the detection process by getting rid of all hypotheses in the neighborhood of γ. We choose the neighborhood volume as a square bipyramid for both accuracy and computational efficiency.

The neighborhood is denoted $\mathcal{N}(\gamma, \rho, \xi)$, where γ and 2ρ are the center and the side of the square base, and ξ is the height of each pyramid. The planar step ρ and the scale step ξ should be monotonically decreasing functions with respect to the score $g_t(\gamma)$. In other words, if a hypothesis has significantly low score, we are totally confident in choosing large steps.

This could be done by associating each pair (ρ, ξ) at the t-th stage with a threshold $\beta_t^{\rho,\xi} = \min_{\gamma \in X} g_t(\mathcal{N}(\gamma, \rho, \xi))$, where X is the object hypothesis training set. As a result, given $g_t(\gamma)$, we compute the largest values of ρ and ξ such that $g_t(\gamma) \leq \beta^{\rho,\xi}$ and then we can prune all hypotheses in $\mathcal{N}(\gamma, \rho, \xi)$. This approach faces two issues: a) countless thresholds should be learned because of infinite combinations of ρ and ξ; b) many threshold comparisons slow down the detection process.

We overcome these problems by dealing with step parameters separately and using look-up tables to search for the optimal neighborhood volume with the computational complexity of $\mathcal{O}(1)$. Firstly, we collect $B_t^1(\rho) = \min_{\gamma \in X} g_t(\mathcal{N}(\gamma, \rho, 0))$ for any $\rho \in \mathbb{N}$ from training data. Note that $B_t^1(\rho) \geq B_t^1(\rho + 1)$ and $B_t^1(0) = \alpha_t^1$. Next, given $g_t(\gamma)$, we obtain the planar step as $\rho = \text{argmin}_\rho(B_t^1(\rho) - g_t(\gamma))$ with constraint of $B_t^1(\rho) \geq g_t(\gamma)$. However, we can accelerate the search by discretizing the range of $[0, \alpha_t^1 - b_t^1]$, where $b_t^1 = \min_\rho B_t^1(\rho)$, into L bins and building up a look-up table LUT_t^1 where each bin is linked to the appropriate optimal ρ. Finally, ρ with respect to $g_t(\gamma)$ is achieved as $\text{LUT}_t^1\left(L \times (\alpha_t^1 - g_t(\gamma)) / (\alpha_t^1 - b_t^1)\right)$. Similarly, we construct a look-up tables LUT_t^2 for the scale step ξ through $B_t^2(\xi) = min_{\gamma \in X} g_t(\mathcal{N}(\gamma, 0, \xi))$.

In the first step of 3D neighbor pruning, we deploy two thresholds α_t^2 and α_t^3 and our flexible neighborhood thresholds via LUT_t^1 and LUT_t^2. Both α_t^2 and α_t^3 are just triggered

from the second stage since both depend on the availability of neighbors' scores, which are computed in the previous stage. By contrast, our proposed thresholds just require the hypothesis itself to make a pruning decision and they are the only ones that can work directly with root filters. It is also notable that we do not use the hypothesis threshold α_t^1 because it is a special case of 3D neighbor threshold, where $\rho = \xi = 0$.

4.2. Scale pruning

The scale pruning step is the implementation of our idea of optimal scale prediction and it is carried out as soon as the stage τ ends. The feature pyramid is collapsed into the feature plane at the scale 0 by mapping each survival hypothesis $\gamma \in \Gamma$ to the corresponding location in F_0 to create a candidate map D. If there are many survivors assigned to the same location, we store K hypotheses with the highest scores and reject the others. From the programming view, D is a sparse 2-dimensional array of pointers, each of which is a linked-list with no more than K nodes. Due to the mapping from coarse feature maps F_s to the finest feature map F_0, these hypotheses do not always fall exactly into one location, but in a local region. For this reason, we adopt NMS_K to discard $\gamma \in D$ if it is not in the top-K of the best hypotheses in its local region. The scale pruning step not only reduces a large number of unimportant hypotheses but also creates an efficient and compact data structure that benefits the next step.

4.3. 1D Pruning

For every stage $t > \tau$, we scan over D and remove any hypothesis that does not meet the conditions of the hypothesis threshold α_t^1 and the deformation threshold α_t^2. At the end of the pipeline, all survival hypotheses that pass the global threshold T are considered as final detection results. We also apply a conventional NMS procedure at the end of the process similar to other object detection systems. However, since the number of final detection is just a few, the computational cost of this NMS is insignificant in our system.

5. EXPERIMENT

To evaluate the performance of our proposal, we conduct experiments on object detection and face detection. We ran all experiments on an Intel Core i7 2.6 GHz desktop with 20 GB memory. Our proposed method implementation (Matlab code[1]) is based on the available DPM release 5 [18].

Object detection: The object detection experiment is conducted on PASCAL VOC 2007 [17] which is taken as a test bed for performance comparison. We choose DPM [1], Cascade DPM [8] and Neighborhood Awareness Cascade DPM (NAC) [4] as the main competitors of our Flex3DNB Cascade DPM. For both DPM and Cascade DPM, we use the available source codes and follow the default settings in the DPM release 5 package. Meanwhile, NAC was introduced in [4] as well as two other speed-up techniques of Low Rank Root Filter and Look-up Table HOG. However, since these two techniques are not related to cascade pruning principle, we only re-implement the NAC for fair comparison. We train all methods with the same configuration of $m = 6$ components and $n = 8$ parts per component. We set the global threshold $T = -1$ for all 20 object classes. For Flex3DNB, according to Sec. 3, we choose $\tau = 2$ and $K = 2$ and use the same 5×5 local regions for semi-positive thresholds as [4]. Table 1 reveals the mean Average Precision [17] results and performance time of all methods in average. Flex3DNB achieves comparable quality of 29.55% against 32.85% of the original DPM while it is 6, 3.2 and 1.6 times faster than DPM, Cascade DPM and NAC respectively.

Table 1: Mean AP and detection time in PASCAL VOC 2007 and AFW datasets.

Object Detection			**DPM**	**Cascade**	**NAC**	**Flex3DNB**
mAP (%)			**32.85**	32.69	31.39	29.30
Detection Time (second)			1.14	0.60	0.30	**0.19**
Face Detection	**TSPM**	**EDEL**	**DPM**	**Cascade**	**NAC**	**Flex3DNB**
mAP (%)	**81.38**	80.84	80.02	80.03	80.11	80.58
Detection Time (second)	42.26	23.29	14.98	4.53	3.20	**2.02**

Face detection: DPMs have been recently applied to face detection and shows the capacity for successfully localizing faces in various poses and facial changes, e.g. TSPM [19]. In this experiment, we compare our proposed method with DPM, Cascade DPM, NAC and two DPM variants for face detection including TSPM [19] and EDEL [20]. For TSPM and EDEL, we use the face model provided by the authors in [19, 20]. Meanwhile, the other methods are trained on FDDB face dataset [21]. We set $m = 12$, $n = 8$ and $T = -0.3$ so that the settings of these methods are equivalent to TSPM and EDEL models. Table 1 reports the performance of all methods on AFW dataset [19]. It can be seen that our Flex3DNB produces a competitive detection quality (80.31% versus 81.38% of the state-of-the-art TSPM) but more efficiency in performance time (around 2 seconds per image).

6. CONCLUSION

In this paper, we introduced the Flexible 3D Neighborhood Cascade DPM (Flex3DNB) that is based on two novel strategies of the 3D neighbor pruning and the scale pruning to aggressively reject a lot of hypotheses during detection. The proposed method runs faster than the state-of-the-art cascade DPM frameworks without sacrificing too much accuracy.

7. ACKNOWLEDGEMENT

This research is funded by Vietnam National Foundation for Science and Technology Development (NAFOSTED).

[1] https://sites.google.com/site/hungthanhvu1986

8. REFERENCES

[1] Pedro F. Felzenszwalb, Ross B. Girshick, David McAllester, and Deva Ramanan, "Object detection with discriminatively trained part-based models," *PAMI*, vol. 32, no. 9, pp. 1627–1645, 2010.

[2] Piotr Dollár, Ron Appel, Serge J. Belongie, and Pietro Perona, "Fast feature pyramids for object detection," *PAMI*, vol. 36, no. 8, pp. 1532–1545, 2014.

[3] Piotr Dollár, Zhuowen Tu, Pietro Perona, and Serge Belongie, "Integral channel features," in *BMVC*, 2009, pp. 91.1–91.11.

[4] Junjie Yan, Zhen Lei, Longyin Wen, and Stan Z. Li, "The fastest Deformable Part Model for object detection," in *CVPR*, 2014, pp. 2497–2504.

[5] Charles Dubout and François Fleuret, "Exact acceleration of linear object detectors," in *ECCV*, 2012, pp. 301–311.

[6] Iasonas Kokkinos, "Rapid deformable object detection using dual-tree branch-and-bound," in *NIPS*, 2011, pp. 2681–2689.

[7] Marco Pedersoli, Andrea Vedaldi, Jordi Gonzàlez, and Xavier Roca, "A coarse-to-fine approach for fast deformable object detection," *Pattern Recognition*, vol. 48, no. 5, pp. 1844–1853, May 2015.

[8] Pedro F. Felzenszwalb, Ross B. Girshick, and David A. McAllester, "Cascade object detection with Deformable Part Models," in *CVPR*, 2010, pp. 2241–2248.

[9] Tianfu Wu and Song-Chun Zhu, "Learning near-optimal cost-sensitive decision policy for object detection," *PAMI*, vol. 37, no. 5, pp. 1013–1027, 2015.

[10] Paul Viola and Michael J. Jones, "Robust real-time face detection," *IJCV*, vol. 57, no. 2, pp. 137–154, May 2004.

[11] Dong Chen, Gang Hua, Fang Wen, and Jian Sun, "Supervised transformer network for efficient face detection," in *ECCV*, October 11-14 2016, pp. 122–138.

[12] Shuzhe Wu, Meina Kan, Zhenliang He, Shiguang Shan, and Xilin Chen, "Funnel-structured cascade for multi-view face detection with alignment-awareness," *Neurocomputing*, vol. 221, pp. 138 – 145, 2017.

[13] Hakan Cevikalp and Bill Triggs, "Visual object detection using cascades of binary and one-class classifiers," *IJCV*, pp. 1–16, 2017.

[14] Fan Yang, Wongun Choi, and Yuanqing Lin, "Exploit all the layers: Fast and accurate CNN object detector with scale dependent pooling and cascaded rejection classifiers," in *CVPR*, June 2016.

[15] Piotr Dollár, Ron Appel, and Wolf Kienzle, "Crosstalk cascades for frame-rate pedestrian detection," in *ECCV*, 2012, pp. 645–659.

[16] Hamed Pirsiavash and Deva Ramanan, "Steerable part models," in *CVPR*, 2012, pp. 3226–3233.

[17] Mark Everingham, Luc Gool, Christopher K. Williams, John Winn, and Andrew Zisserman, "The Pascal Visual Object Classes (VOC) Challenge," *IJCV*, vol. 88, no. 2, pp. 303–338, June 2010.

[18] R. B. Girshick, P. F. Felzenszwalb, and D. McAllester, "Discriminatively trained Deformable Part Models, release 5," `http://people.cs.uchicago.edu/~rbg/latent-release5/`.

[19] Deva Ramanan, "Face detection, pose estimation, and landmark localization in the wild," in *CVPR*, 2012, pp. 2879–2886.

[20] Hung Thanh Vu, Mai Vuong Minh Nhat, and Bac Le, "An efficient model for simultaneous face detection, pose estimation and landmark localisation," in *2015 Seventh International Conference on Knowledge and Systems Engineering (KSE)*, 2015, pp. 13–18.

[21] Vidit Jain and Erik Learned-Miller, "FDDB: A benchmark for face detection in unconstrained settings," Tech. Rep. UM-CS-2010-009, University of Massachusetts, Amherst, 2010.