ADAPTIVE CASCADE THRESHOLD LEARNING FROM NEGATIVE SAMPLES FOR DEFORMABLE PART MODELS

Khoa Pho *Hung Vu* *Bac Le*

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

ABSTRACT

A solution to deploy object detection systems to practical applications is to build cascade frameworks which do threshold comparisons in each stage to efficiently discard a large number of negative objects. For particular applications, these thresholds should be retrained for better effectiveness and the efficiency via training datasets. It means that we have to store labeled datasets permanently or collect huge data (for high-quality thresholds) whenever learning new thresholds. Both approaches are inconvenient and expensive in terms of memory and data collection cost. In this paper, we propose a novel threshold selection mechanism, named Adaptive Cascade Threshold Learning (ACTL), which learns thresholds directly from non-object regions in a single input image instead of object regions from large training data as other existing methods. As a result, we can completely remove the need of training data for cascade threshold learning. Experimental results on two problems of object detection and face detection confirm that our proposed method can obtain the same level of accuracy and speed as state-of-the-art cascade DPM methods but it has the benefit of no threshold training data.

Index Terms— Deformable Part Models, cascade models, threshold learning.

1. INTRODUCTION

Cascade model introduced by Viola was a huge breakthrough in computer vision [1] by deploying an efficient framework for real time applications. The idea behind the cascade framework is that, instead of spending much time on finding several objects out of thousands of irrelevant regions, it leverages on the low-cost thresholds to filter out as many non-object regions as possible. However, most research pays attention to improving other elements of the cascade system, such as feature representation [2, 3], convolution operations [4] and Non-Maximum Suppression (NMS) [5, 6]. A few studies put more efforts on investigating the good threshold set from the training data [1, 7, 8]. Besides manual threshold tuning [9], cascade threshold selection methods can be classified into two main approaches: learning both thresholds and detectors simultaneously and learning thresholds after the model training step.

In the first approach, thresholds are usually chosen by balancing the false positive rate and the detection rate [1] or minimizing the cost functions [10] or guaranteeing that the detection rate does not drop below a predefined detection rate [11]. The second approach discovers thresholds individually from the model training procedure. Bourdev et al. [12] reordered the learned classifiers of all stages and chose the maximum thresholds without sacrificing the detection rate. Another framework [13] chooses thresholds so that the detection rate is above the desired detection rate. Alternatively, Luo [8] proposed a greedy search on ROC curves to find the optimal thresholds. Some studies [7, 14] choose thresholds as the minimum of positive hypotheses in labeled training data.

In the aforementioned cascade frameworks, the models (stage classifiers [1, 11], neural networks [13] and DPM filters [7, 14]) are usually learned once while the cascade thresholds should be adjusted to specific applications via predefined detection rates [1, 11, 13] or false positive rates [1, 8] or global thresholds [7, 14]. For example, we prefer higher detection rates or global thresholds for applications that focus on high detection accuracy or conversely lower ones for better speed. Hence, labeled training datasets are required whenever cascade thresholds are retrained . Unfortunately, in practical scenario, the training datasets do not always accompany the learned models and we cannot reuse them to learn new thresholds. Meanwhile, another choice is to collect new data and annotate them but this solution is expensive and labor-intensive.

To overcome this issue, we propose an online threshold learning method without labeled training data. Our idea is that, instead of collecting numerous training images with just several object samples per image, we learn thresholds from countless negative samples in one image. As a result, given an input image, our method is able to automatically estimate thresholds and detect objects of interest on the fly. Specifically, our method scans over the image, collects high score negative hypotheses and uses them to update thresholds during the detection process. This idea can be generalized to many cascade systems but, in this work, we use Cascade DPM [7] to demonstrate our idea. The experimental results on both object detection and face detection problems show that our learning method provides comparable thresholds with ones of state-of-the-art cascade DPMs, in terms of accuracy and

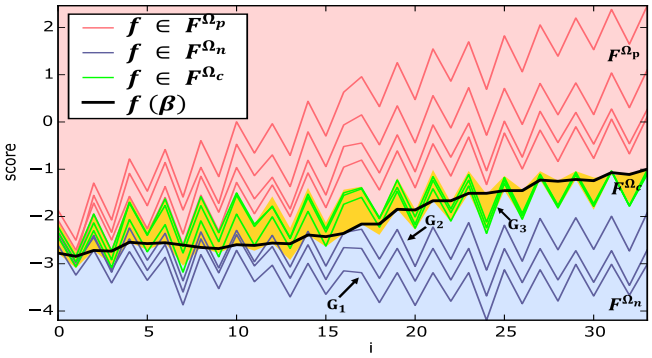

Fig. 1. Threshold function (black curve) and score functions in F^{Ω_p} (pink area), F^{Ω_n} (light blue area) and F^{Ω_c} (yellow area) are collected by human face cascade DPMs.

speed, but requires no labeled training data.

2. DPM AND CASCADE DPM

The DPM is a mixture of N_{comp} components each of which represents the same object at a particular pose or viewpoint via a set of filters including a root filter w_0 and n part filters $w_1, ..., w_n$. A location to place a filter is a pair of a position and a scale in the feature pyramid of an image I. Let Ω be the space of all possible object hypotheses. An object hypothesis $\gamma \in \Omega$ is specified as $\{l_0, l_1, ..., l_n\}$, where l_0 and l_t are the locations of the root and the t-th part filter. The cumulative score of γ at the filter t-th is defined as the score of root filter plus the sum over the first t parts minus their deformation cost:

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

where $\phi_a(l, I)$ extracts the feature vector at location l, d_i is the deformation cost weight associated with the i-th part filter, and $\phi_d(l, l_0)$ is the quadratic function of the deformation. The score of the hypothesis γ is its cumulative score at the final stage $s(\gamma) = s_n(\gamma)$.

During the detection phase, the system scans over all locations in Ω to find objects of interest. Let $\Delta_t(l)$ specify all possible positions of l_t whose root filter is located at l. The optimal part location l_t is inferred by maximizing the part appearance score minus the deformation cost. The detector considers a region γ as an object if $s(\gamma) \geq T$.

Due to the large number of hypotheses in Ω, verifying all of them in the DPM models above is extremely expensive. Cascade DPM [7] aims to improve this process by filtering out negative hypotheses as many as possible via low-cost threshold comparisons. Cascade DPM is a sequence of cascade stages each of which consists two thresholds (b_t, b'_t). Such threshold pairs and the global thresholds form the $2n+1$ dimensional threshold vector $\boldsymbol{\beta} = (b_1, b'_1, \ldots, b_n, b'_n, T)$. We define $\boldsymbol{v}^\gamma$, whose elements $\boldsymbol{v}^\gamma_{2i} = s_i(\gamma)$ and $\boldsymbol{v}^\gamma_{2i+1} = s_i - d^\top_{i+1}\phi_d(l_{i+1}, l_0)$, as the score vector of γ. If there is any element $\boldsymbol{v}^\gamma_i < \boldsymbol{\beta}_i$, γ will be assigned as a non-object hypothesis and eliminated. Moreover, each filter in Cascade DPM has its own low-dimensional PCA filter to speed up the elimination process. Cascade DPM usually places PCA filters first to efficiently prune negative hypotheses, and then original filters to carefully search for objects.

3. OBJECT-LIKE NEGATIVE HYPOTHESES

The threshold learning step in most existing cascade methods strongly depends on the availability of positive hypotheses in the labeled training data. By removing this dependence, we expect to eliminate this step. In this section, we conduct experiments to verify the possibility to estimate thresholds from negative hypotheses in images.

Given a learned threshold vector $\boldsymbol{\beta}$, we can separate Ω into two subsets: a positive hypothesis set $\Omega_p = \{\gamma \in \Omega | \forall i, \boldsymbol{v}^\gamma_i \geq \boldsymbol{\beta}_i\}$ and a negative hypothesis set $\Omega_n = \{\gamma \in \Omega | \exists i, \boldsymbol{v}^\gamma_i < \boldsymbol{\beta}_i\}$. We define an object-like negative hypothesis set $\Omega_c \subset \Omega_n$ including hypotheses γ_c that meet two conditions: a) their intermediate scores $s_i(\gamma_c) = \boldsymbol{v}^{\gamma_c}_{2i}$ pass all stage thresholds $\boldsymbol{\beta}_{2i}$ but b) its final score $s_n(\gamma_c)$ fails the global threshold T. Let $f(\boldsymbol{v}^\gamma)$ describe a score function which is formed by connecting two adjacent elements in $\boldsymbol{v}^\gamma$ and similarly $f(\boldsymbol{\beta})$ is the threshold function of $\boldsymbol{\beta}$. Some examples of these functions are demonstrated in Fig. 1. By defining $F^A = \{f(\boldsymbol{v}^\gamma) | \gamma \in A\}$, we obtain the score function sets $F^{\Omega_p}, F^{\Omega_n}$ and F^{Ω_c} of the hypothesis sets. According to the definition of Ω_c, its score function set F^{Ω_c} reveals two following properties.

Property 1: F^{Ω_c} is close to and almost above $f(\boldsymbol{\beta})$ (according to the definition, $\boldsymbol{v}^\gamma_{2i} \geq \boldsymbol{\beta}_{2i}$ for $0 \leq i \leq n$ tends to make $f \in F^{\Omega_c}$ above $f(\boldsymbol{\beta})$ but $\boldsymbol{v}^\gamma_{2n} < \boldsymbol{\beta}_{2n} = T$ pins the tails of f and $f(\boldsymbol{\beta})$ around T).

Property 2: Ω_c contains negative hypotheses γ_c with highest score functions (since most $f \in F^{\Omega_n}$ are below $f(\boldsymbol{\beta})$, $f(\boldsymbol{v}^{\gamma_c})$ is above majority of $f \in F^{\Omega_n}$).

To verify such properties, we collect all hypotheses for Ω_p, Ω_n and Ω_c from 2000 random images for each PASCAL VOC 2007 object class and FDDB face class. For each class, we visualize $F^{\Omega_p}, F^{\Omega_n}$ and F^{Ω_c} and discover their relationship. Fig. 1 is a demonstration of these function sets for the face object. In this figure, the black curve indicates $f(\boldsymbol{\beta})$ which is the lower bound of the positive set F^{Ω_p} (pink area). Meanwhile, F^{Ω_c} is represented by the yellow region and F^{Ω_n} consists of the yellow and the light blue. We observe that if Ω_c exists, both properties hold.

An explanation for our interest in the object-like negative set Ω_c and its properties is that a) Property 1 says that F^{Ω_c} is close to $f(\boldsymbol{\beta})$ and we have more opportunities to choose a good negative hypothesis $\gamma_c \in \Omega_c$ so that $f(\boldsymbol{v}^{\gamma_c}) \approx f(\boldsymbol{\beta})$; whilst b) Property 2 suggests us an online algorithm to reach Ω_c from any $\gamma \in \Omega_n$ by climbing up in F^{Ω_n}. Therefore, the object-like negative set provides us a useful mechanism to approximate the cascade thresholds using the score functions of negative hypotheses.

By investigating Ω_n and Ω_c deeply, we find out that their score functions can be classified into three different groups according to their trend. They are the groups of decreasing functions (G_1), score functions fluctuating around the horizontal lines (G_2) and functions with increasing trend (G_3). Since both $f(\boldsymbol{v}^{\gamma_c})$, where $\gamma_c \in \Omega_c$, and $f(\boldsymbol{\beta})$ end at points around T, the proximity between $f(\boldsymbol{v}^{\gamma_c})$ and $f(\boldsymbol{\beta})$ is specified by the similarity of their function forms. Recall that in Cascade DPM, $\boldsymbol{\beta}$ is selected as the minimal positive hypothesis in the training set. This implies an increasing trend in $f(\boldsymbol{v}^{\gamma_c})$ should be preferred for accurately approximating $f(\boldsymbol{\beta})$. Fig. 1 illustrates some examples of three function groups and the increasing score functions in Ω_c which are close to the true threshold function.

Our observations on Ω_c and its properties above not only suggest us an idea to learn thresholds from negative hypotheses but also provide us a guideline to design an efficient online threshold learning system.

4. PROPOSED METHOD

In this section, we describe a method to find out the approximate thresholds $\tilde{\boldsymbol{\beta}}$ on the fly. Specifically, after DPM models are learned on training data, we deploy them immediately to detect objects in images. The remarkable feature of our proposed framework is that the thresholds $\tilde{\boldsymbol{\beta}}$ are computed during this detection phase. This characteristic is completely distinct from other cascade methods that usually require a training step to estimate the cascade thresholds $\boldsymbol{\beta}$. Our idea is that, starting with extremely low initial thresholds $\tilde{\boldsymbol{\beta}}$, we use arriving negative hypotheses to raise $\tilde{\boldsymbol{\beta}}$ until convergence to Ω_c. To be more precise, we firstly initialize all elements of $\tilde{\boldsymbol{\beta}}$ to be negative infinity except for the last one $\tilde{\boldsymbol{\beta}}_{2n} = T$. During detection, we use $\tilde{\boldsymbol{\beta}}$ as the cascade thresholds to prune hypotheses. If there exists a hypothesis γ which satisfies all $\tilde{\boldsymbol{\beta}}_i$ but fails at the end $s_{2n}(\gamma) < T$, its score vector is used to update stage thresholds $\tilde{\boldsymbol{\beta}}_i$, where $i = 0, \ldots, 2n-1$, as follows:

$$\tilde{\boldsymbol{\beta}}_i = \eta_i \boldsymbol{v}_i^{\gamma} + (1 - \eta_i)\tilde{\boldsymbol{\beta}}_i \tag{2}$$

$$\eta_i = \alpha + (1 - \alpha)\frac{i}{2n-1} \tag{3}$$

where η_i is the update coefficient of $\tilde{\boldsymbol{\beta}}_i$ and it varies according to its index i and the global slope $\alpha = (\boldsymbol{v}_{2n}^{\gamma} - \boldsymbol{v}_0^{\gamma})/2n$ of the score function $f(\boldsymbol{v}^{\gamma})$. Eq. 2 implies that γ affects $\tilde{\boldsymbol{\beta}}_i$ via the coefficient η_i whose strength is specified by α and i in Eq. 3. Basically, we encourage $f(\boldsymbol{v}^{\gamma}) \in G_3$ with the large positive slope to contribute more to the current thresholds and deactivate the effect of G_1 and G_2 by assigning $\alpha = 0$ for any $\alpha \leq 0$. Furthermore, to produce the increasing trend for new $\tilde{\boldsymbol{\beta}}$, we force a threshold in $\tilde{\boldsymbol{\beta}}$ to produce a higher coefficient than its previous thresholds via the fraction $i/(2n-1)$. However, when the score function rockets unexpectedly ($\alpha \geq 1$), Eq. 3 lessens coefficients with respect to i to prevent the thresholds from becoming too steep.

Intuitively, the update equation (Eq. 2) is the realization of Property 2 in Sec. 3 where we iteratively elevate the current thresholds to approach Ω_c. The update condition $\tilde{\boldsymbol{\beta}}_{2n} < T$ guarantees that $\tilde{\boldsymbol{\beta}}$ cannot move so far to F^{Ω_p} but stops around T, where Ω_c lies on. Meanwhile, the coefficients (Eq. 3) control the slope of $\tilde{\boldsymbol{\beta}}$ by both updating it with increasing functions and adjusting its slope directly. As a result, $f(\tilde{\boldsymbol{\beta}})$ is an increasing function and is a good approximation of the true cascade thresholds $\boldsymbol{\beta}$ as described in Sec. 3.

Our aforementioned method still works properly in the case where Ω_c does not exist. If Cascade DPM components are trained on a small number of positive images, $f(\boldsymbol{\beta})$ (as the minimal positive hypothesis) can not touch F^{Ω_n}. This creates a gap between F^{Ω_p} and F^{Ω_n}, and results in the non-existence of Ω_c. In this case, our method is unable to reach Ω_c but just stops at one of the highest functions $f \in F^{\Omega_n}$ that is still good enough to discard most negative hypotheses.

5. EXPERIMENTS

We evaluate our threshold learning method for both face detection in AFW dataset [15] and object detection in PASCAL VOC 2007 [16]. All experiments ran in a single thread on a Core i7-4790 3.6 GHz CPU, 20 GB RAM. We compare our method with Cascade DPM [7] and Neighbor Awareness Cascade (NAC) [14]. All versions of our ACTL method were implemented (Matlab code [1]) using DPM release 5 [17]. We train face models with $N_{comp} = 12$ components, $n = 8$ parts on FDDB [18] and $T = -0.5$ and object models with $N_{comp} = 6$, $n = 8$ and $T = -1$ on PASCAL VOC 2007 training set [16].

5.1. Online threshold learning

In this experiment, we evaluate our first version of ACTL, named onl-ACTL, as described in Sec. 4 in the online scenario where thresholds are learned automatically during detection phase. Despite losing rich information source of positive hypotheses, our onl-ACTL still achieves MAP of 31% versus 31.39% of NAC and 32.69% of Cascade DPM for object detection (Table 1) whilst the difference in accuracy is negligible ($< 0.1\%$) for AFW dataset (Table 2). Tables 1 and 2 also show the detection time of all methods. The onl-ACTL runs 1.66 and 1.13 times faster than Cascade DPM in PASCAL VOC and AFW respectively. The improvement in speed is because our the adaptive thresholds $\tilde{\boldsymbol{\beta}}$ are higher than the true thresholds $\boldsymbol{\beta}$, and therefore they can prune more negative hypotheses. Both Cascade DPM and our onl-ACTL can not outperform NAC in speed because NAC is equipped with 2 more thresholds per cascade stage in order to prune hypotheses in a neighborhood. However, these two thresholds can

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

Table 1. MAP (%) and detection time (second) in PASCAL VOC 2007.

MAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	**mean**
Cascade	33.48	**59.70**	**10.16**	**14.74**	**26.69**	**50.56**	52.23	**21.75**	**19.98**	**23.91**	26.26	12.95	**56.44**	**47.21**	**42.77**	13.43	**20.27**	**34.93**	**45.16**	**41.20**	**32.69**
NAC	**33.98**	58.69	9.71	12.31	25.10	48.43	**54.28**	19.43	17.95	23.43	22.77	11.69	55.19	46.39	40.51	11.89	19.11	31.12	44.89	40.99	31.39
onl-ACTL	32.41	59.36	9.27	9.09	23.30	46.60	52.69	21.12	13.98	23.62	26.31	13.10	56.22	47.17	41.19	13.54	10.51	33.77	45.09	41.19	31.00
trn-ACTL	32.10	59.66	9.27	9.09	24.92	44.94	52.43	21.43	12.79	23.33	**26.37**	**13.16**	56.27	47.03	41.21	**13.84**	13.38	34.39	45.14	40.87	31.08
Det Time	**aero**	**bike**	**bird**	**boat**	**bottle**	**bus**	**car**	**cat**	**chair**	**cow**	**table**	**dog**	**horse**	**mbike**	**person**	**plant**	**sheep**	**sofa**	**train**	**tv**	**mean**
Cascade	0.68	0.33	0.90	0.78	0.93	0.36	0.61	0.58	1.11	0.41	0.44	0.68	0.36	0.38	0.65	1.01	0.54	0.38	0.27	0.64	0.60
NAC	0.32	0.29	0.32	0.38	0.27	0.32	0.30	**0.27**	0.38	**0.26**	**0.25**	**0.34**	**0.26**	**0.23**	**0.31**	0.34	0.31	**0.27**	**0.21**	0.34	0.30
onl-ACTL	0.44	0.39	0.33	0.26	0.30	0.34	0.33	0.37	0.25	0.33	0.50	0.46	0.42	0.45	0.47	0.28	0.20	0.38	0.40	0.29	0.36
trn-ACTL	**0.30**	**0.28**	**0.24**	**0.21**	**0.26**	**0.26**	**0.20**	0.28	**0.21**	0.27	0.36	**0.34**	0.28	0.31	0.37	**0.23**	**0.19**	0.29	0.25	**0.20**	**0.27**

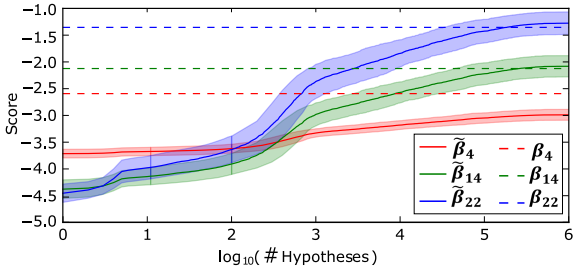

Fig. 2. Updated threshold values with respect to $\log_{10}(\#\text{hypotheses})$. The skeletons and the width of the colored strips are means and standard deviations of the average thresholds.

also be learned by our techniques and it is potential to extend our framework to NAC cascade method. These experiment results show that threshold approximation with negative hypotheses produces the same performance as conventional threshold learning but takes more advantage of no training step and no annotation effort.

Table 2. MAP (%) and detection time (second) in AFW.

	Cascade	NAC	onl-ACTL	trn-ACTL
MAP	80.03	**80.11**	80.01	80.04
Time	4.53	**3.20**	4.02	4.66

To understand comprehensively the threshold updating process, we do an experiment in order to observe the change of threshold values during detection. Given an image, we run an onl-ACTL face detector and store the current threshold values with respect to the number of processed hypotheses. Then we take the average of these thresholds over 205 images of AFW dataset and draw the graph that illustrates the updated thresholds in average for every arriving hypothesis. Fig. 2 compares our adaptive thresholds $\tilde{\boldsymbol{\beta}}_i$ and the true thresholds $\boldsymbol{\beta}_i$ at $i = 4, 14$ and 22. It can be seen that $\tilde{\boldsymbol{\beta}}$ converges to a threshold vector that is close to $\boldsymbol{\beta}$ after 10^6 hypotheses. The early adaptive thresholds usually show under-approximations of the true thresholds due to their small update coefficients. Nevertheless, this does not harm the accuracy much since alive negative hypotheses are still discarded correctly by the later thresholds. Therefore, these results confirm the quality of our approximate thresholds.

We also measure the average number of threshold updates in AFW dataset and observe that most updates occur early on and the update cost is insignificant with only 3.35, 41.79 and 94.47 updates for 10, 10^4 and 10^6 hypotheses respectively.

5.2. Learning with negative samples

The onl-ACTL resets $\tilde{\boldsymbol{\beta}}$ and learns new ones for each input image. Another scenario is to train ACTL on negative training images once and use these learned thresholds to detect objects in all images. To that end, we firstly collect 20 random images from the Internet. Note that it does not matter if the training images are polluted with some positive objects. For each category, we run onl-ACTL to learn thresholds for each training image and use the average of these thresholds as the final thresholds. The average thresholds are plugged into Cascade DPM and tested in PASCAL VOC and AFW (Tables 1 and 2. Interestingly, this ACTL setting, named trained-ACTL, offers an excellent alternative to Cascade DPM with equivalent MAP and detection time. However, its primary advantage is that it reduces the cost of training data collection and data annotations for threshold learning from thousands of labeled images (e.g. PASCAL VOC) to a few images (e.g. 20 images) without the need for labeling data.

6. CONCLUSION

Thresholds play an important role in controlling the efficiency and effectiveness of most cascade frameworks. These systems evaluate the thresholds using a large dataset of object images. This paper proposes a novel approach to learn thresholds from non-object regions to remove the dependence on positive training data. We introduce two scenarios of online threshold learning during detection and offline threshold learning with several negative images. Experiments show that our thresholds produce the same performance and speed as other state-of-the-art cascade DPM frameworks but more flexible in training settings.

7. ACKNOWLEDGEMENT

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

8. REFERENCES

[1] Paul Viola and Michael Jones, "Rapid object detection using a boosted cascade of simple features," in *CVPR*. IEEE, 2001, vol. 1, pp. I–511.

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

[3] Rainer Lienhart and Jochen Maydt, "An extended set of haar-like features for rapid object detection," in *ICIP*. IEEE, 2002, vol. 1, pp. I–900.

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

[5] Alexander Neubeck and Luc Van Gool, "Efficient non-maximum suppression," in *ICPR*. IEEE, 2006, vol. 3, pp. 850–855.

[6] Matthew B Blaschko, Juho Kannala, and Esa Rahtu, "Non maximal suppression in cascaded ranking models," in *Scandinavian Conference on Image Analysis*. Springer, 2013, pp. 408–419.

[7] Pedro F Felzenszwalb, Ross B Girshick, and David McAllester, "Cascade object detection with deformable part models," in *CVPR*. IEEE, 2010, pp. 2241–2248.

[8] Huitao Luo, "Optimization design of cascaded classifiers," in *CVPR*. IEEE, 2005, vol. 1, pp. 480–485.

[9] Hongwei Qin, Junjie Yan, Xiu Li, and Xiaolin Hu, "Joint training of cascaded cnn for face detection," in *CVPR*, 2016, pp. 3456–3465.

[10] S Charles Brubaker, Jianxin Wu, Jie Sun, Matthew D Mullin, and James M Rehg, "On the design of cascades of boosted ensembles for face detection," *IJCV*, vol. 77, no. 1-3, pp. 65–86, 2008.

[11] Rong Xiao, Huaiyi Zhu, He Sun, and Xiaoou Tang, "Dynamic cascades for face detection," in *ICCV*. IEEE, 2007, pp. 1–8.

[12] Lubomir Bourdev and Jonathan Brandt, "Robust object detection via soft cascade," in *CVPR*. IEEE, 2005, vol. 2, pp. 236–243.

[13] 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*, 2016, pp. 2129–2137.

[14] Junjie Yan, Zhen Lei, Longyin Wen, and Stan Z Li, "The fastest deformable part model for object detection," in *CVPR*, 2014, pp. 2497–2504.

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

[16] Mark Everingham and John Winn, "The pascal visual object classes challenge 2007 (voc2007) development kit," *University of Leeds, Tech. Rep*, 2007.

[17] 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.

[18] 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.