# Learning to Match Anchors for Visual Object Detection

Xiaosong Zhang, *Student Member, IEEE*, Fang Wan, *Student Member, IEEE*, Chang Liu, *Student Member, IEEE*, Xiangyang Ji, *Member, IEEE*, and Qixiang Ye, *Senior Member, IEEE*

**Abstract**—Modern CNN-based object detectors assign anchors for ground-truth objects under the restriction of object-anchor Intersection-over-Union (IoU). In this study, we propose a learning-to-match (LTM) method to break IoU restriction, allowing objects to match anchors in a flexible manner. LTM updates hand-crafted anchor assignment to "free" anchor matching by formulating detector training in the Maximum Likelihood Estimation (MLE) framework. During the training phase, LTM is implemented by converting the detection likelihood to anchor matching loss functions which are plug-and-play. Minimizing the matching loss functions drives learning and selecting features which best explain a class of objects with respect to both classification and localization. LTM is extended from anchor-based detectors to anchor-free detectors, validating the general applicability of learnable object-feature matching mechanism for visual object detection. Experiments on MS COCO dataset demonstrate that LTM detectors consistently outperform counterpart detectors with significant margins. The last but not the least, LTM requires negligible computational cost in both training and inference phases as it does not involve any additional architecture or parameter. Code has been made publicly available.

**Index Terms**—Object Detection, Maximum Likelihood Estimation, Learning to Match, Anchor-Free Detector, Generalized Linear Model.

## 1 Introduction

Over the past few years we have witnessed the success of convolution neural network (CNN) for visual object detection [1]–[8]. To represent objects of various appearance, aspect ratios and poses with limited convolutional features, many CNN-based detectors leverage anchor boxes as reference points for object localization [1]–[7], [9], [10]. By assigning each object to a single anchor or multiple anchors at proper scales and aspect ratios, convolutional features are determined and two fundamental detection procedures, classification and localization, are carried out.

Anchor-based detectors leverage spatial alignment, $i.e.$, Intersection over Union (IoU) between objects and anchors, as the criterion for anchor assignment. Each assigned anchor independently supervise network learning for object prediction, based upon the assumption that the anchors spatially aligned with objects are always appropriate for classification and localization. In what follows, however, we argue that such an assumption is implausible and the spatial alignment should not be the sole criterion for anchor assignment.

On the one hand, for objects of acentric features, $e.g.$, slender objects, the most representative features are not close to their geometric centers. A spatially aligned anchor might correspond to less representative features, which deteriorate classification and localization performance. On the other hand, it is implausible to match objects with proper anchors/features using the IoU criterion when multiple objects come together. These issues arise from pre-defining single anchors for specific objects which then independently supervises network learning for object predictions. The open problem is how to flexibly match anchors/features with objects, which is the focus of this study.

In this paper, we propose a learning-to-match (LTM) approach for object detection[1], with the aim to update hand-crafted anchor assignment to learnable anchor/feature configuration, Fig. 1. CNN-based detector with learning-to-match mechanism, referred to as an LTM detector, optimizes the training procedure from three aspects. First, to achieve a high recall rate, the detector requires to guarantee that at least one anchor's prediction is close to an object's ground-truth. Second, to achieve high detection precision, the detector requires to classify anchors with poor localization, $i.e.$, large bounding-box regression error, into background. Third, anchors' predictions should be compatible with the non-maximum suppression (NMS) procedure, $i.e.$, the higher classification score is, the more accurate localization is. Otherwise, an anchor predicts accurate location but low classification score will be suppressed by the subsequent NMS procedure.

To fulfill these purposes, we propose to define an anchor bag for each object and perform object-anchor matching in a maximum likelihood estimation (MLE) framework [11], [12]. We connect the likelihood probabilities of anchors with that of anchor bags by introducing positive and negative anchor matching functions. Optimizing the matching functions drives maximizing the detection customized likelihood and selecting optimal anchors in a "soft" manner. Meanwhile, the anchors with large classification or localization error are classified as backgrounds. During training, the anchor matching functions are converted into a plug-

*Xiaosong Zhang, Chang Liu, and Qixiang Ye are with the School of Electronic, Electrical and Communication Engineering. Fang Wan is with the School of Computer Science and Technology, University of Chinese Academy of Sciences (UCAS), Beijing, China, 100049. Emails: zhangxiaosong18@mails.ucas.ac.cn, wanfang@ucas.ac.cn, liuchang615@mails.ucas.ac.cn, qxye@ucas.ac.cn. Xiangyang Ji is with the Department of Automation, Tsinghua University. Qixiang Ye is the corresponding author.*

1. Code is available at github.com/zhangxiaosong18/FreeAnchor

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2021.3050494, IEEE Transactions on Pattern Analysis and Machine Intelligence



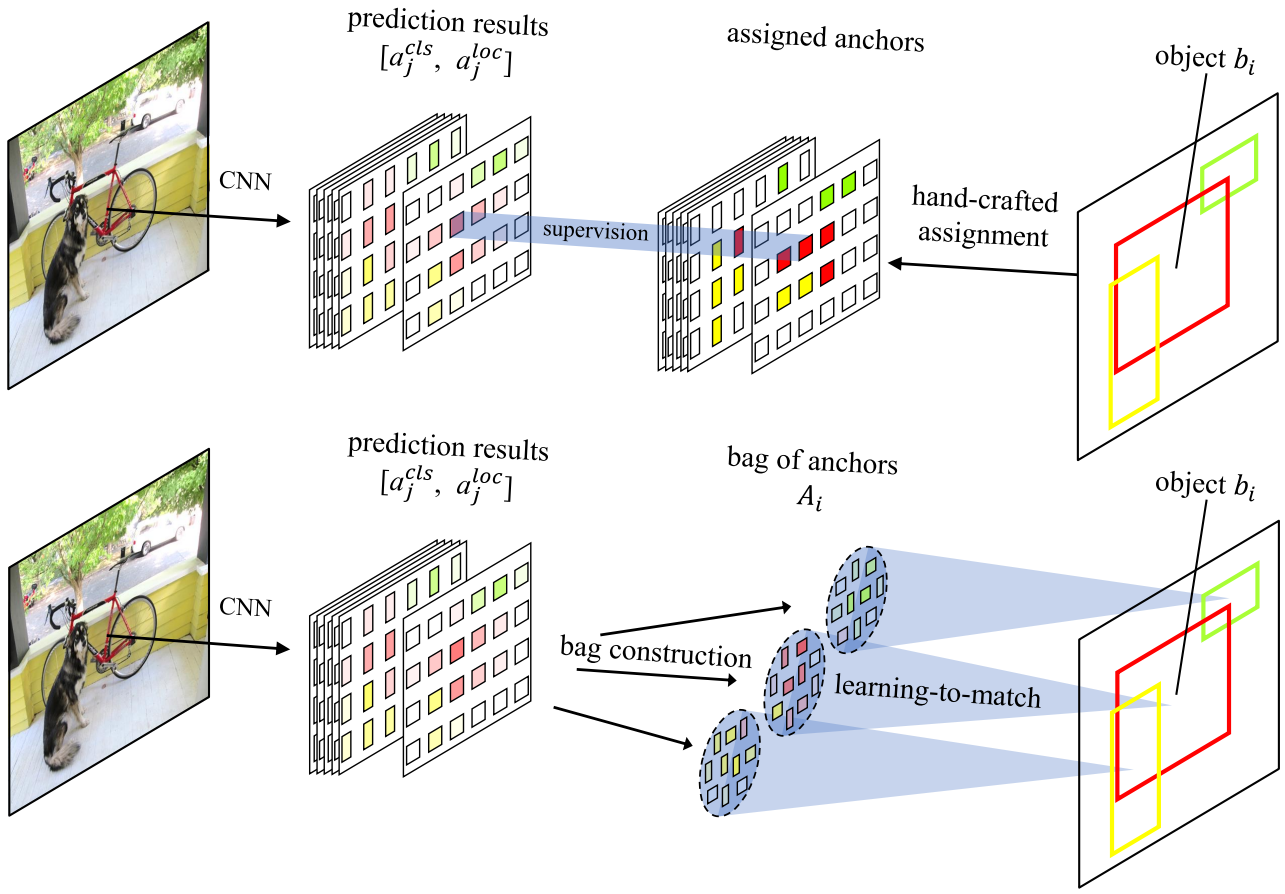



Fig. 1. Comparison of the hand-crafted anchor assignment (upper) with our learning-to-match method (lower). The former leverages Intersection over Union (IoU) between objects and anchors as the criterion for anchor assignment. Each assigned anchor independently supervises detector learning. In contrast, our approach allows each object flexibly matching positive/negative anchors from a "bag" of anchors by jointly evaluating object classification and object localization confidence. The anchor matching is performed in a "soft" manner. In the early training epochs, all anchors have similar matching confidence. In the final epoch, the confidence of positive anchors evolve to be large while those of negative anchors become small. Each box on the prediction result map indicates an anchor center point.

and-play loss function, which then drives detector training and object-anchor matching.

The learning-to-match method was first proposed in our NeurIPS 2019 paper [13] and is promoted theoretically and experimentally in this full version. The contributions of this paper include:

(1) We formulate detector training in a maximum likelihood estimation (MLE) framework, where detection customized likelihood is defined to unify two fundamental modules, classification and localization, of object detection. Maximizing the likelihood drives matching optimal anchors which guarantee the comparability of object classification and object localization with NMS.

(2) We propose a learning-to-match method and update hand-crafted anchor assignment to learnable anchor configuration. With differential $Mean\text{-}max$ matching functions and plug-and-play anchor matching loss, the learning-to-match method selects optimal positive anchors and mines hard negative anchors in a "soft" manner.

(3) We derive the $Mean\text{-}max$ matching functions using the generative linear model (GLM), which justifies detector optimization from the perspective of sufficient statistics.

(4) We achieve state-of-the-art detection performance on the commonly used COCO dataset, and validate the general applicability of the learning-to-match method on anchor-based and anchor-free detectors.

The remainder of this paper is summarized as follows. Related works are described in Section 2 and the proposed learning-to-match method and object detectors are presented in Section 3. Experimental results are described in Section 4. We conclude this paper in Section 5.

## 2 Related Work

According to recent survey papers [14]–[16], various taxonomies can be used to category the large amount of CNN-based object detectors, *e.g.*, one-stage [3] vs. two-stage [5], single-scale features [3] vs. Feature Pyramid Network (FPN) [6], and handcrafted network [7] vs. network architecture search [17]. In this paper, related works are reviewed from the perspective of anchors.

### 2.1 Anchor-based Method

A detection pipeline requires generating a set of bounding boxes along with their classification labels associated with objects in an image. However, it is not trivial for a CNN-based detector to directly predict an order-less set of arbitrary cardinals [18]. One widely-used workaround is to introduce anchors, each of which indicates a location on the convolutional feature map. With dense anchors configured, a divide-and-conquer process can be defined to match objects with convolutional features according to the IoU criterion, Fig. 1.

Anchor-based approach has been successfully demonstrated in modern detectors including Faster R-CNN [3], SSD [7], FPN [6], RetinaNet [5], DSSD [19], and YOLOv2 [20]. In these detectors, dense anchors require to be configured upon convolutional feature maps so that features can match object extent and bounding box regression can be well initialized. During detector training, anchors are assigned to objects or backgrounds by thresholding their IoUs with ground-truth bounding boxes [3].

IoU-Net [21] attempted selecting anchors by predicting the IoU between a detected bounding-box and a ground-truth box. By introducing IoU-guided NMS, it reduced suppression failures caused by misleading classification confidences. Cascade RPN [22] improved region-proposal quality and detection performance by addressing the limitation of the conventional RPN which heuristically defines anchors and aligns features to anchors. Instead of using multiple anchors with predefined scales and aspect ratios, Cascade RPN relied on a single anchor per location and performed multi-stage refinement. Each stage was progressively more stringent in defining positive samples by starting out with an anchor-free metric followed by anchor-based metrics. To align features with anchors throughout the stages, adaptive convolution was proposed to learn sampled features guided by anchors.

Despite the great progress, existing anchor-based detection methods remain restricted by the heuristics that spatially aligned anchors are compatible for both object classification and localization. For objects of acentric features, however, such heuristics is implausible and the detector could miss the optimal anchors/features. To overcome this disadvantage, we propose the learning-to-match method, and target at breaking the IoU restriction so that objects can flexibly match anchors under the principle of detection customized likelihood. Our work is related to the deformable convolutional network (DCN) [23], which augments the spatial sampling locations of feature with additional offsets and learns offsets from the target tasks. The essential difference lies in that we use a learning-to-match method instead of the spatial offset strategy.

### 2.2 Anchor Optimization Method

To facilitate object-anchor matching, researchers proposed to refine anchors [18], [24] or adaptively produce optimal anchors [25]. MetaAnchor [18] learned to predict anchors from the arbitrary customized prior boxes with a sub-net. GuidedAnchoring [25] leveraged semantic features to guide the prediction of anchors while replacing dense anchors with predicted anchors. Gaussian YOLO [26] introduced localization uncertainty which indicates the reliability of anchors/bounding-boxes. By using the predicted localization uncertainty during the detection process, Gaussian YOLO implemented online anchor/feature localization optimization.

Existing approaches have taken a step towards learnable anchor configuration. Nevertheless, to the best of our knowledge, there still lacks a systematic approach for anchor selection in the detector training procedure, which inhibits the simultaneous optimization of object classification and localization. For most of the methods, the anchors are evenly distributed within the image, so that each part in the image is considered with the same importance level. On the other hand, the objects in an image do not follow a uniform distribution, $i.e.$, there is a location imbalance issue [27]. In this paper, we define a detection customized likelihood, and aim to jointly optimize anchor matching and solve the location imbalance issue in a systematic way.

### 2.3 Anchor-free Method

To break the limitations brought by anchors, researchers attempted per-pixel prediction by modeling objects as feature points. Each feature point corresponding to a convolutional feature vector can be directly used for object classification and bounding-box regression.

EAST [28] used each deep pixel within the object bounding-box to learn detectors, while selecting the pixel of highest classification score for object localization. Fully convolutional one-stage object detector (FCOS) [29] attempted to solve object detection in a per-pixel prediction fashion, which is similar with semantic segmentation, leveraging pixel-level supervision and center-ness bounding box regression for object detection.

Considering that using all pixels within object extent for prediction significantly increases the computational cost, CornerNet [30] and CenterNet [8] used two corner points and a central point. To handle object appearance variation, a center-pooling operation was designed to align the most representative features to corner/center points. With the similar idea, ExtremePoint [31] detected four key-points (top-most, left-most, bottom-most, right-most) and one center point of objects using a keypoint estimation network. The five key-points were grouped into a bounding box if they are geometrically aligned. To improve adaptability, RepPoint [32] learned to automatically arrange keypoints in a manner which bounds the spatial extent of an object and indicates semantically significant local areas. It thus used limited key-points to sample a space of bounding boxes. FSAF [33] addressed hand-crafted feature selection and overlap-based anchor sampling by introducing online feature selection and dynamically assigning each instance to the most suitable feature level on the feature pyramid.

Anchor-free methods, which represent each object with a couple of feature points, have demonstrated greater potential using simpler pipelines. Nevertheless, most feature points are hand-crafted, which restricts the representative capacity of convolutional feature maps given limited spatial resolutions. In this study, we propose the learning-to-match method and apply it to anchor-free detectors. Our approach not only selects optimal features but also enhances the

representative capacity of convolutional features by using a bag of features which collaboratively supervise detector training.

## 3 METHODOLOGY

In this section, we first present the learning-to-match method in the maximum likelihood estimation (MLE) framework. We then introduce anchor-matching functions to implement the learning-to-match method for detector training. We finally implement anchor-based and anchor-free detectors based on the proposed learning-to-match method. The derivation about anchor matching functions is based on the generative linear model (GLM), which is included in the appendix.

### 3.1 Learning-to-Match Method

To formulate the learning-to-match method in the MLE framework, we first define detection customized likelihood to unify object classification and object localization.

#### 3.1.1 Detector Training as Maximum Likelihood Estimation

Given a training image $X$ containing $I$ objects, the ground-truth annotations are denoted as $B$ where the ground-truth box for $i\text{-}th$ object is denoted as $b_i \in B$. On the convolutional feature maps of $X$, a set of anchors $A$ are defined as reference points at multiple scales and aspect ratios. Each anchor corresponds to a feature vector across feature channels.

In most CNN-based detectors, the hand-crafted criterion based on IoU is used to assign anchors for each object. When the IoU between the ground-truth box $b_i$ and anchor $a_j$ is greater than a threshold, $b_i$ matches $a_j$. If multiple objects' IoUs are larger than the threshold, the object of the largest IoU will successfully match this anchor, which guarantees that each anchor matches a single object at most while an object can match multiple anchors. According to the IoU criterion, anchors in $A$ are categorised to multiple positive anchor bags $A_i \subseteq A_+$ and a negative anchor bag $A_-$, and $A = A_+ \cup A_-$. During inference, each anchor $a_j \in A_i$ predicts a classification confidence $a_j^{cls} \in [0, 1]$ by feeding the feature vector to a classification sub-network. The anchor also predicts an object's location, $a_j^{loc} = \{x, y, w, h\}$, by feeding its feature vector to a bounding box regression sub-network.

With anchor-object assignment, the classification loss[2] of a positive anchor is defined as

$$\mathcal{L}_{ij}^{cls}(\theta) = -\log\big(a_j^{cls}(\theta)\big), \quad for \quad a_j \in A_i,$$

and the classification loss of a negative anchor is defined as

$$\mathcal{L}_{j}^{cls}(\theta) = -\log\big(1 - a_j^{cls}(\theta)\big), \quad for \quad a_j \in A_-,$$

where $a_j^{cls}(\theta)$ denotes the classification confidence of anchor $a_j$ given network parameters $\theta$. The localization loss for a positive anchor is defined as

$$\mathcal{L}_{ij}^{loc}(\theta) = \ell_{reg}\big(a_j^{loc}(\theta), b_i\big),$$

where $\ell_{reg}(\cdot)$ denotes SmoothL1 regression loss [1] between location prediction $a_j^{loc}$ and ground-truth bounding box $b_i$.

Based on the loss defined, detector training is carried out by optimizing the following objective function, as

$$\min_{\theta} \mathcal{L}(\theta) = \sum_{i} \sum_{a_j \in A_i} \big(\mathcal{L}_{ij}^{cls}(\theta) + \beta \mathcal{L}_{ij}^{loc}(\theta)\big) + \sum_{a_j \in A_-} \mathcal{L}_{j}^{cls}(\theta), \tag{1}$$

where $\beta$ is a regularization factor balancing the importance of the regression loss. From the perspective of likelihood, the probability about positive anchor $a_j$ correctly predicting the $i\text{-}th$ object is defined as

$$\begin{aligned} \mathcal{P}_{ij}(\theta) &= \exp\big(-\mathcal{L}_{ij}^{cls}(\theta) - \beta \mathcal{L}_{ij}^{loc}(\theta)\big) \\ &= a_j^{cls}(\theta) \exp\big(-\beta \ell_{reg}(a_j^{loc}(\theta), b_i)\big), \end{aligned} \tag{2}$$

which unifies the classification confidence $a_j^{cls}(\theta)$ with the localization confidence $\exp\{-\beta \ell_{reg}(a_j^{loc}(\theta), b_i)\}$ of anchor $a_j$. Accordingly, minimizing detection loss $\mathcal{L}(\theta)$ defined by Eq. 1 for positive anchors is equal to maximizing a likelihood probability, as

$$\max_{\theta} \mathcal{P}(\theta) = \prod_{i} \prod_{a_j \in A_i} \mathcal{P}_{ij}(\theta). \tag{3}$$

#### 3.1.2 Detection Customized Likelihood

Eqs. 2 and 3 unify the classification and localization modules of positive anchors from the perspective of likelihood. However, the likelihood is based on hand-crafted anchors without learning-to-match. In what follows, we introduce detection customized likelihood in the MLE framework to update hand-crafted anchor assignment to learnable anchor/feature matching. The matching procedure operates like probabilistic multiple instance learning [12], where a positive bag is defined to contain at least one positive anchor and a number of negative anchors. During training, each object corresponding to a positive anchor bag learns to match positive/negative anchors from the bag by jointly evaluating object classification and object localization confidence. Correctly matched positive/negative anchors are true positive/negative anchors.

To achieve a high recall rate, for each object $b_i \in B$, it requires to guarantee that there exists at least one anchor $a_j \in A_i$ whose predictions ($a_j^{cls}$ and $a_j^{loc}$) are close to the ground-truth. This means that the probability that the anchor bag contains the true positive should be high. The true positive probability of anchor bag $A_i$ is defined as

$$\mathcal{P}_i^{tp}(\theta) = p(Y_i = 1 | A_i; \theta),$$

where $Y_i \in \{1, 0\}$ is a binary variable indicating whether anchor bag $A_i$ can predict object $b_i$ well or not. The likelihood of a matched anchor follows the definition in Eq. 2, as

$$\begin{aligned} p(y_{ij}; \theta) &= \mathcal{P}_{ij}(\theta) \\ &= a_j^{cls}(\theta) \exp\big(-\beta \ell_{reg}(a_j^{loc}(\theta), b_i)\big), \end{aligned} \tag{4}$$

where $y_{ij} \in \{1, 0\}$ is a binary variable indicating whether anchor $a_j$ can predict object $b_i$ well or not. $y_{ij} = 1$ means $a_j$ is a positive anchor and $y_{ij} = 0$ means $a_j$ is not a positive anchor.

To achieve high detection precision, detectors must classify the anchors of poor localization into negatives, which

2. The binary cross entropy loss is used to replace the Focal Loss [5] for formulation simplicity.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2021.3050494, IEEE Transactions on Pattern Analysis and Machine Intelligence



means that the true negative probability should be high. The true negative probability of anchor bag $A_i$ is defined as

$$\mathcal{P}_i^{tn}(\theta) = p(Y_i^- = 0|A_i;\theta),$$

where $Y_i^- \in \{1,0\}$ is defined to indicate that $A_i$ falsely localize object $b_i$. The likelihood that an anchor falsely localize the object is defined as

$$p(y_{ij}^-;\theta) = a_j^{cls}(\theta)(1 - \max_i \text{IoU}_{ij}), \forall i, \tag{5}$$

where $y_{ij}^- \in \{1,0\}$ is defined to indicate that $a_j$ falsely localize object $b_i$. $y_{ij}^- = 1$ indicates that $a_j$ is a negative anchor and $y_{ij}^- = 0$ indicates that $a_j$ is not a negative anchor. $\text{IoU}_{ij}$ denotes the Intersection over Union between the prediction of anchor $a_j$ and the ground-truth object $b_i$.

By maximizing the true positive probability $\mathcal{P}_i^{tp}(\theta)$ and true negative probability $\mathcal{P}_i^{tn}(\theta)$ defined above, Eq. 3 is materialized to a detection customized likelihood function, as

$$\begin{aligned}\mathcal{P}^{\mathcal{M}}(\theta) &= \prod_i \mathcal{P}_i^{tp}(\theta) \cdot \prod_i \mathcal{P}_i^{tn}(\theta) \\ &= \prod_i p(Y_i = 1|A_i;\theta) \cdot \prod_i p(Y_i^- = 0|A_i;\theta).\end{aligned} \tag{6}$$

By maximizing the likelihood defined by Eq. 6, we aim to simultaneously match anchors with objects and train an optimal detector.

### 3.1.3 Anchor Matching Function

Eq. 6 defines the detection customized likelihood of anchor bags. However, it remains lacking a mechanism to bridge the likelihood of anchor bags with that of anchors. To fulfill this purpose, we propose the anchor matching functions, $\mathcal{M}_+$ and $\mathcal{M}_-$, as

$$p(Y_i = 1|A_i;\theta) = \mathcal{M}_+\big(p(y_i;\theta)\big), \tag{7}$$

and

$$p(Y_i^- = 0|A_i,\theta) = \mathcal{M}_-\big(p(y_i^-;\theta)\big), \tag{8}$$

where $p(y_i;\theta) = \{p(y_{ij};\theta); j = 1,\ldots,|A_i|\}$ defines a set of probabilities indicating whether the set of anchors in bag $A_i$ can predict object $b_i$ well or not. $p(y_i^-;\theta) = \{p(y_{ij}^-;\theta); j = 1,\ldots,|A_i|\}$ defines a set of probabilities indicating the set of anchors in bag $A_i$ falsely localize object $b_i$.

Based on the anchor matching functions, the detection customized likelihood in Eq. 6 is expanded as

$$\begin{aligned}\mathcal{P}^{\mathcal{M}}(\theta) &= \prod_i \mathcal{P}_i^{tp}(\theta) \cdot \prod_i \mathcal{P}_i^{tn}(\theta) \\ &= \prod_i p(Y_i = 1|A_i;\theta) \cdot \prod_i p(Y_i^- = 0|A_i;\theta) \\ &= \prod_i \mathcal{M}_+\big(p(y_i;\theta)\big) \cdot \mathcal{M}_-\big(p(y_i^-;\theta)\big).\end{aligned} \tag{9}$$

So far, the remaining problem is materializing the anchor matching functions $\mathcal{M}_+$ and $\mathcal{M}_-$. An off-the-shell approach is multiple instance learning [12], which can select (match) the instance (anchor) of largest likelihood from a bag of anchors, as

$$p(Y_i = 1;\theta) = p(\max_j y_{ij} = 1;\theta),$$

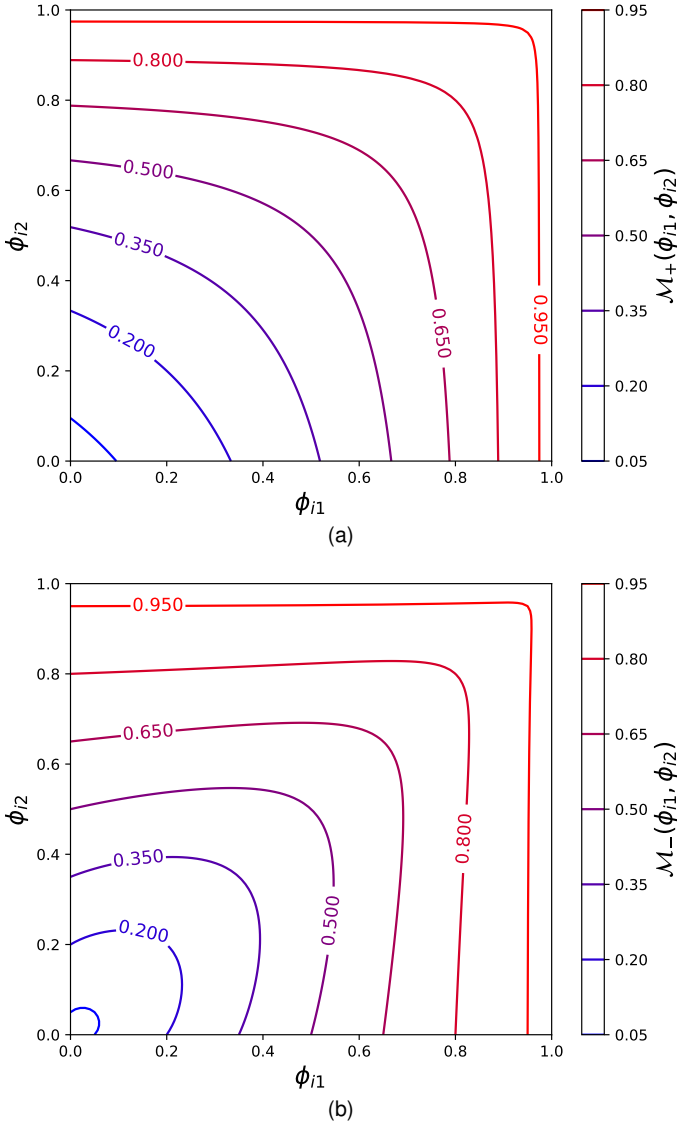



Fig. 2. Illustration of two-dimensional $Mean\text{-}max$ functions for positive anchor matching (a) and negative anchor matching (b). "Two-dimensional" means there are two anchors in an anchor bag. Each anchor corresponds a dimension. When the values of both dimensions are close to zero, the $Mean\text{-}max$ function approximates the $Mean$ function and the function value is determined by the two dimensions (anchors). When either value of the two dimensions is large, the $Mean\text{-}max$ function approximates the $max$ function and the function value is determined by the dimension of larger value.

where $p(Y_i = 1)$ denotes the likelihood of a positive anchor bag. It means that there exists at least one anchor which can correctly match the object ($y_{ij} = 1$).

Nevertheless, we validated that multiple instance learning is not applicable when training network parameters and selecting instances (anchors) at the same time. With multiple instance learning, a single anchor is selected from a bag of anchors to update the network parameter. At early training epochs, however, the confidence of each anchor is small for randomly initialized network parameters. Therefore, the anchor of the highest confidence may not be the best choice for detector training. To solve this problem, we propose the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2021.3050494, IEEE Transactions on Pattern Analysis and Machine Intelligence



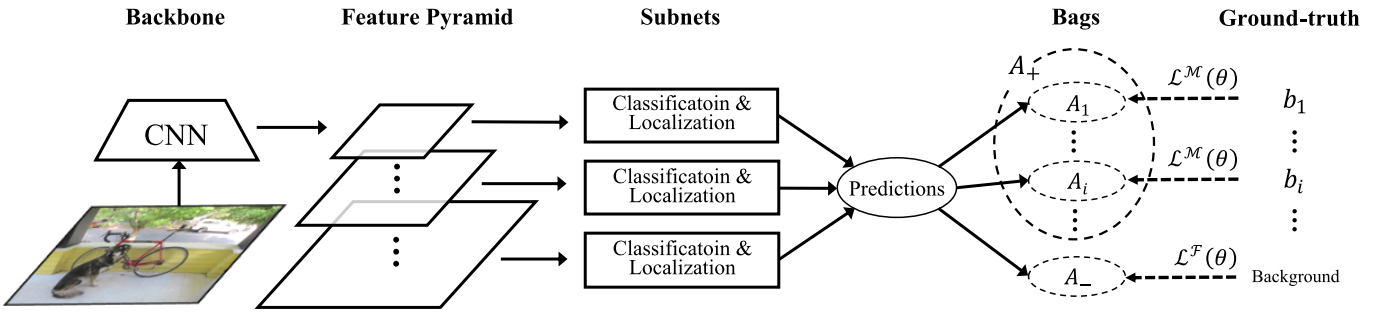



Fig. 3. Network architecture of the proposed LTM detector. The head of the detector consists of two subnets, one for object classification and other for object localization. During detector training, minimizing the anchor matching loss $\mathcal{L}^{\mathcal{M}}(\theta)$ drives matching positive/negative anchors within the positive anchor bag in a "soft" manner. Meanwhile, the Focal Loss $\mathcal{L}^{\mathcal{F}}(\theta)$ is applied on background (the negative bag) to prevent the vast number of easy negatives from overwhelming the detector. This architecture is also applicable to an anchor-free detector (LTM-AF) by simply replacing the anchors with pixels on the feature pyramid.

$Mean\text{-}max$ function[3], which matches anchors in a "soft" manner according to their likelihood, as

$$\mathcal{M}_+(\phi_i) = \frac{\sum_j \frac{\phi_{ij}}{1-\phi_{ij}}}{\sum_j \frac{1}{1-\phi_{ij}}}, \tag{10}$$

where $\phi_i = p(y_i;\theta)$ and $\phi_{ij} = p(y_{ij};\theta)$ respectively denote the likelihood set of anchors and the likelihood of the $j$-th anchor.

When training is insufficient, the $Mean\text{-}max$ function [13], Fig. 2, is close to the $Mean$ function, so that all anchors in a bag are used for training. When training proceeds, the confidence of some anchors increases and the $Mean\text{-}max$ function moves closer to the $max$ function. When sufficient training has taken place, a few optimal anchors in each anchor bag will have high likelihood to match the object.

Similarly, a negative matching function $\mathcal{M}_-$ is introduced to select negative anchors, as

$$\mathcal{M}_-(\phi_i) = \frac{\sum_j \frac{\phi_{ij}}{1-\phi_{ij}}}{\sum_j \frac{1}{(1-\phi_{ij})^2}}, \tag{11}$$

where $\phi_i = p(y_i^-;\theta)$ and $\phi_{ij} = p(y_{ij}^-;\theta)$ respectively denote the likelihood set of anchors and the likelihood of the $j$-th anchor. As shown in Fig. 2(a) and Fig. 2(b), $\mathcal{M}_-$ changes faster than $\mathcal{M}_+$ from $Mean$ to $max$. The reason lies in that $\mathcal{M}_+$ requires to fully consider all anchors using the $Mean$ function before it can match an optimal anchor using the $max$ function. In contrast, $\mathcal{M}_-$ targets at matching negative anchors from positive anchor bags. The faster change of $\mathcal{M}_-$ facilities earlier determining hard negative anchors.

The simultaneous usage of $\mathcal{M}_+$ and $\mathcal{M}_-$ contributes a bag splitting mechanism, which matches positive anchors/features while filtering out noisy anchors/features. Such a mechanism facilities matching representative anchors/features while mining very hard instances, as a complement to focal loss which handling the large mount of negative instances.

3. The derivation of $Mean\text{-}max$ function from the perspective of generalized linear model (GLM) is detailed in Appendix.

### 3.2 Anchor-based Detector

The proposed LTM detector is implemented upon the Feature Pyramid Network (FPN) [6] atop an ResNet [34] or ResNeXt [35] backbone, Fig. 3. Following FPN, feature layers from $P_3$ to $P_7$, each of which has $C = 256$ channels, are used for detection. The head of the LTM detector is made up of two subsets, one for object classification $i.e.$, predicting object category confidence, and other for object localization, $i.e.$, regressing object bounding boxes using anchor boxes as the reference locations.

**Anchor Bag Construction.** On the feature layers, anchors are generated following the setting of RetinaNet [5]. There are 9 anchors with three sizes $\{2^0; 2^{1/3}; 2^{2/3}\}$ and three aspect ratios $\{1:2; 1:1; 2:1\}$ for each pixel on each feature layer. Anchors on all feature layers cover the scale range from 32 to 813 pixels with respect to the input image. Based on the anchors defined, an anchor bag is constructed for each object. The positive anchor bag is defined according to the IoU between the anchors and the object ground truth but there is no IoU threshold used. The anchors not belonging to any positive bag are included to the negative bag $A_-$.

Specifically, we select top-$K$ anchors according to the IoU instead of using a predefined IoU threshold for the following two reasons: (1) Using top-$K$ anchors/features can guarantees consistent bag sizes. In contrast, using a predefined IoU threshold cannot produce the same number of anchors for bags (inconsistent bag sizes), which makes it difficult to implement loss vectorization; (2) It is not necessary to guarantee that all anchors in a bag are of high likelihood, because our approach can match postive anchors while depressing negative ones. In experiments, the number ($K$) of anchors in each positive anchor bag is significantly larger than that of hand-crafted assigned anchors, which increases the opportunity to match anchors of small IoU.

**Training and Inference.** For detector training, maximizing the detection customized likelihood defined by Eq. 9 simultaneously optimizes network parameters and matches anchors (determining $p(y_{ij};\theta)$ in Eq. 4 and $p(y_{ij}^-;\theta)$ in Eq. 5). For CNN-based detectors, maximizing the detection customized likelihood is implemented by minimizing the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2021.3050494, IEEE Transactions on Pattern Analysis and Machine Intelligence



anchor matching loss, as

$$\begin{aligned}\mathcal{L}^{\mathcal{M}}(\theta) &= -\log \mathcal{P}^{\mathcal{M}}(\theta) \\ &= -\sum_i \log \mathcal{M}_+\big(p(y_i;\theta)\big) - \sum_i \log \mathcal{M}_-\big(p(y_i^-;\theta)\big),\end{aligned} \tag{12}$$

by applying a $-\log(\cdot)$ function on Eq. 9. To balance the impact of positive and negative anchors, $\mathcal{L}^{\mathcal{M}}(\theta)$ is normalized by the numbers of objects ($|B|$), as

$$\begin{aligned}\mathcal{L}^{\mathcal{M}}(\theta) = &-\frac{1}{|B|}\sum_i \log \mathcal{M}_+\big(p(y_i;\theta)\big) \\ &-\frac{1}{|B|}\sum_i \log \mathcal{M}_-\big(p(y_i^-;\theta)\big),\end{aligned} \tag{13}$$

where $p(y_i;\theta) = \{p(y_{ij};\theta); j = 1,\ldots,|A_i|\}$ and $p(y_i^-;\theta) = \{p(y_{ij}^-;\theta); j = 1,\ldots,|A_i|\}$. $p(y_{ij};\theta)$ and $p(y_{ij}^-;\theta)$ are respectively calculated by Eq. 4 and Eq. 5. $\mathcal{M}_+(\cdot)$ and $\mathcal{M}_-(\cdot)$ are respectively defined by Eq. 10 and Eq. 11.

During detector training, we minimize the anchor matching loss defined by Eq. 13. Considering the extreme imbalance of foreground-background classes, the Focal Loss is adopted to prevent the vast number of easy negative anchors from overwhelming the detector, as

$$\mathcal{L}^{\mathcal{F}}(\theta) = -\frac{1}{N}\sum_{a_j \in A_-} a_j^{cls}(\theta)^{\gamma} \log\big(1 - a_j^{cls}(\theta)\big),$$

where $\gamma$ denotes the exponential parameter for the Focal Loss [5]. Accordingly, the final loss function for the LTM detector is defined by combining the Focal loss with anchor matching loss, as

$$\mathcal{L}(\theta) = \mathcal{L}^{\mathcal{M}}(\theta) + \mathcal{L}^{\mathcal{F}}(\theta), \tag{14}$$

where the anchor matching loss and Focal loss are empirically set to be of equal importance.

According to Eq. 14, the anchor matching procedure and detector training are fused and optimized using the stochastic gradient descent (SGD) algorithm in an end-to-end manner. The inference procedure of the LTM detector is exactly the same as RetinaNet, *i.e.*, we use the learned network parameters to predict classification scores and object bounding boxes, which are fed to the NMS procedure for object detection. As our detector does not involve any additional network architecture compared it with the baseline detector and the anchor matching loss is only applied in the training phase, the computational cost overhead in the inference phase is negligible.

### 3.3 Anchor-Free Detector

The proposed learning-to-match method is extended from the anchor-based detector to an anchor-free detector, termed LTM-AF. LTM-AF performs object detection in a per-pixel prediction fashion [29], without anchors involved. Following the way to construct anchor bags we construct positive point bags, where each point corresponds to a pixel on the convolutional feature map. However, we must realize that the pixels/points have no object extent information and how to assign them to objects of various sizes and aspect ratios is problematic.

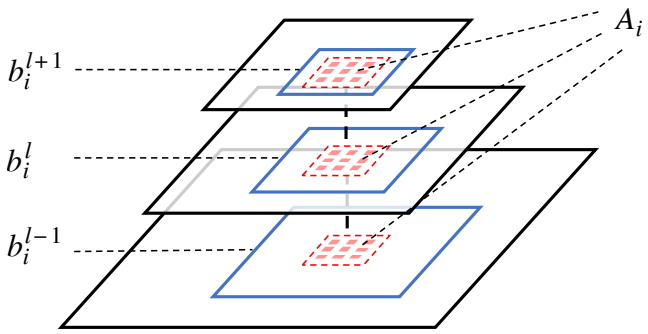



Fig. 4. Point bag construction for the proposed LTM-AF detector based on normalized distances. Each ground-truth object is normalized and scaled to feature layers using its bounding box center as the origin. The normalized distance is a block distance from the feature points to the normalized boxes. Top-$K$ points of smallest normalized distance (within the dashed box) are selected to construct the feature point bag.

To solve this problem, we propose a distance normalization strategy to assign feature points to objects and construct point bags. Denote an object bounding box as $b_i = (x_i, y_i, w_i, h_i)$ where $(x_i, y_i)$ is the central point and $(w_i, h_i)$ the width and height of $b_i$. Denote a feature point as $a_j = (x_j, y_j, l_j)$ where $(x_j, y_j)$ is the point coordinate and $l_j$ the index of feature layer on the feature pyramid. To match a ground-truth box $b_i$ with a point $a_j$ without extent, the normalized distances is first defined as $\max\big(\frac{|x_i - x_j|}{w_i \times 2^{l_j}}, \frac{|y_i - y_j|}{h_i \times 2^{l_j}}\big)$.

For each ground-truth object, we calculate and sort the normalized distances of points in bounding box $b_i$. We then select $K$ points of smallest normalized distances to construct a positive bag $A_i$, Fig. 4. This procedure guarantees that a similar number of feature points on each feature layer be included in a bag, which facilities point-object matching. When positive point bags are constructed, the representative feature points can be matched with each object during detector training.

When training an anchor-free detector, we use a similar loss defined for the anchor-based detector in Eq. 14 by simply replacing the regression loss function $\ell_{reg}(\cdot)$ with the GIoU Loss function [36]. We also experimentally search the hyper-parameter $K$ and regression weight $\beta$. In experiments, we validated that the anchor-free detector (LTM-AF) achieved comparable performance and speed with the anchor-based detector (LTM).

## 4 Experiments

The proposed learning-to-match method was applied to train anchor-based and anchor-free detectors. For each kind of detector, we first described experimental settings, then reported the performance and compared with state-of-the-arts. The effect of the proposed learning-to-match method was analyzed on the anchor-based detector.

### 4.1 Experimental Settings

Experiments were carried out on COCO 2017 [37], which contains ~118k images from 80 object categories for training (*train*). In the dataset, 5k images were used for validation (*val*) and ~20k for test (*test-dev*). Detectors were trained on

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2021.3050494, IEEE Transactions on Pattern Analysis and Machine Intelligence



TABLE 1
Effect of anchor matching on COCO $test\text{-}dev$ set. RetinaNet ($K$ anchors) denotes using top-$K$ anchors (without anchor matching) to supervise each object. LTM ($\mathcal{M}_+$) uses solely positive anchor matching while LTM uses both positive and negative anchor matching.

| Detector (Matching) | AP | $AP_{50}$ | $AP_{75}$ | $AP_S$ | $AP_M$ | $AP_L$ |
|---|---|---|---|---|---|---|
| RetinaNet (baseline) | 35.7 | 55.0 | 38.5 | 18.9 | 38.9 | 46.3 |
| RetinaNet ($K$ anchors) | 35.7 | 56.0 | 38.0 | 19.2 | 39.3 | 46.7 |
| LTM ($\mathcal{M}_+$) | 38.7 | 57.3 | 41.6 | 20.2 | 41.3 | **50.1** |
| LTM | **39.2** | **57.9** | **42.1** | **21.5** | **41.6** | 49.3 |

TABLE 2
Performance improvement under different object crowdedness.

| detector | AP sparse | AP medium | AP dense |
|---|---|---|---|
| RetinaNet (baseline) | 41.2 | 32.0 | 29.0 |
| LTM (ours) | 44.4 | 35.4 | 32.7 |
| Relative improvement | 7.8% | 10.6% | 12.8% |

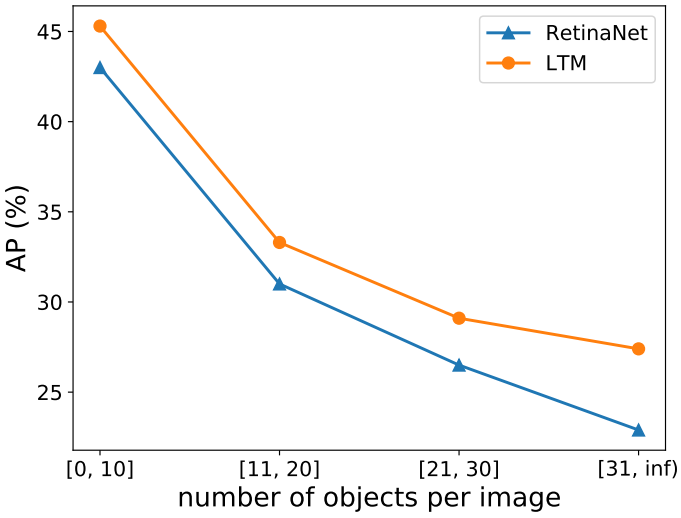



Fig. 5. Performance comparison on object crowdedness (number of objects per image) on COCO $val$ set. With higher crowdedness, LTM demonstrates larger advantage over the baseline method (RetinaNet).

the COCO training set and evaluated on the $val$ set. The detection performance was reported on the $test\text{-}dev$ set.

Following the settings in [3], [5], [6], [29], each input image was resized to have a shorter side of 800 pixels while the longer side less than 1333 pixels. We also followed previous settings to perform horizontal flipping on randomly selected images for data augmentation. When multi-scale training was performed, input images were jittered over scales {640, 672, 704, 736, 768, 800} at the shorter side. In addition to single-scale testing performance, we also reported the multi-scale testing performance by fusing the detection results on multi-scale images with sizes {400, 500, 600, 700, 900,1000, 1100, 1200}.

When training LTM detectors, we set the bias initialization to $b = -\log((1-\rho)/\rho)$ with $\rho = 0.02$ for the last convolutional layer of the classification subnet. With the initialization, training was carried out using the synchronized SGD over 8 GPUs with a total of 16 images per mini-batch (two images per GPU). Unless otherwise specified, all detectors were trained for 90k iterations with an initial learning rate 0.01, which was then respectively decreased by a magnitude at 60k and a magnitude again at 80k iterations. A weight decay of 0.0001 and a momentum of 0.9 were used. Following [29], group normalization [38] was employed for LTM-AF in classification and localization subnets.

### 4.2 Learning-to-match

As analyzed in Section 1, hand-crafted anchor assignment often fails when facing: (1) multiple objects in crowded scenes; and (2) slender objects with acentric features. The detectors based on the learning-to-match method can effectively alleviate these issues.

**$\mathcal{M}_+$ and $\mathcal{M}_-$.** In Table 1, positive anchor matching ($\mathcal{M}_+$) improved the detection AP by 3.0% (35.7% to 38.7%) which was a significant margin for the challenging object detection task. Using negative anchor matching ($\mathcal{M}_-$) further improved the performance by 0.5% (38.7% to 39.2%), demonstrating its effectiveness to filter out noisy anchors/features. The RetinaNet ($K$ anchors) was an improved baseline detector using top-$K$ anchors to supervise the detector without anchor matching. Unfortunately, there was no improvement over RetinaNet (baseline), which validated that the major improvement was from learning-to-match.

**Object Crowdedness.** In Fig. 5, we compared the performance of the baseline method and that of LTM($\mathcal{M}_+$) in scenarios of various object crowdedness. As the number of objects in each image became larger, the advantage of LTM over RetinaNet became more significant. This demonstrated that our detector, with positive anchor matching, can select optimal anchors for objects in crowded scenes [39]. Table 2 included the relative performance improvements under different object crowdedness. Crowdedness is calculated as the largest IoU between an object with its nearest objects. According to the object crowdedness, the test images of MS COCO were divided into three groups: sparse, medium and dense. The range of crowdedness of the three groups is [0.0, 0.2], [0.2, 0.5], [0.5, 1.0], respectively. It can be seen that images of higher object crowdedness have larger relative improvements.

To further validate the effectiveness of our approach in crowded scenarios, we tested it on SKU110K [40], which is a dataset specified for densely packed objects in supermarkets. By simply plugging the LTM module to the RetinaNet baseline, we observed 5.6% (52.6% vs. 47.0%) performance improvement, which is a significant margin.

**Acentric Features**. From the confidence evolution of matched anchors in Fig. 6 one can see that LTM flexibly selected the most representative features to represent the object of interest. This endowed detectors the flexibility over objects of acentric features. In Fig. 7, the fitted lines clearly show that LTM has larger performance improvements over object categories of larger aspect ratios. In Fig. 8, one can see that the LTM detector significantly outperformed the RetinaNet baseline over the categories of slender objects. Particularly, for the categories "snowboard", "tie","keyboard", and "couch", LTM outperformed the baseline method up to 5-10% APs, which are large margins with respect to the challenging object detection task.

**Compatibility with NMS.** To qualitatively assess the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2021.3050494, IEEE Transactions on Pattern Analysis and Machine Intelligence



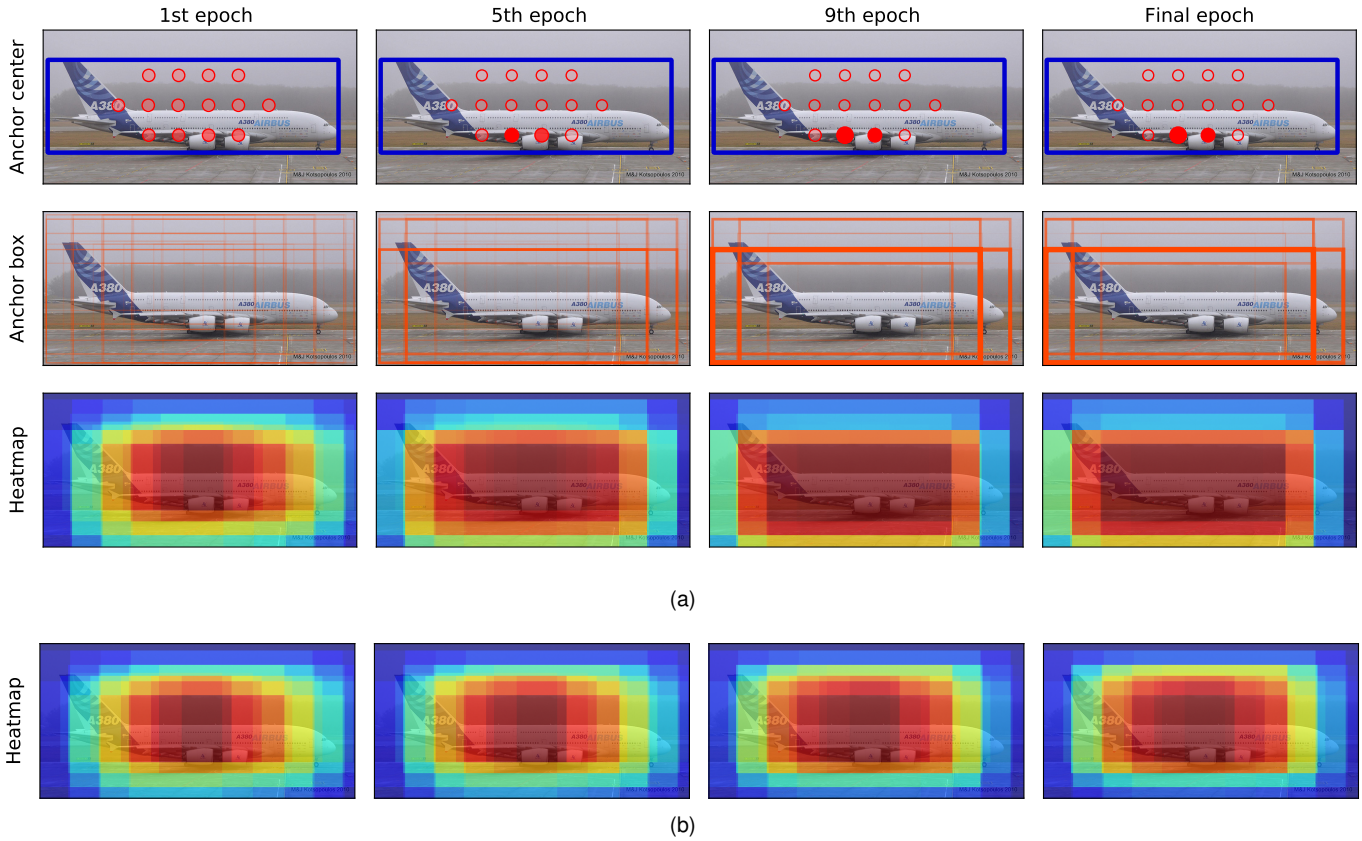



Fig. 6. Anchor confidence evolution when training the LTM detector (a). First row: dots denote anchor centers. Darker (redder) dots indicate higher confidence. Second row: Darker boxes indicate anchors of higher confidence. Third row: the heatmaps are calculated by accumulating anchor confidence. (b) Anchor confidence evolution when training the RetinaNet detector.

TABLE 3
The effect of IoU for anchor bag construction on COCO $val$ set. $S$ and $K$ mean randomly selecting $S$ anchors from the top-$K$ anchors with respect to their IoUs with objects.

| $S$ | $K$ | AP | $AP_{50}$ | $AP_{75}$ | $AP_S$ | $AP_M$ | $AP_L$ |
|---|---|---|---|---|---|---|---|
| 40 | 40 | **38.4** | **57.4** | **41.1** | **20.5** | **41.9** | 50.7 |
| 40 | 80 | 37.2 | 56.5 | 39.5 | 20.0 | 39.8 | **51.0** |
| 40 | 120 | 36.0 | 55.8 | 37.8 | 19.0 | 38.6 | 49.3 |
| 40 | 160 | 34.9 | 54.8 | 36.6 | 18.4 | 37.3 | 48.9 |

TABLE 4
Comparison of NMS recall (%) on COCO $val$ set.

| Detector | NR | $NR_{50}$ | $NR_{60}$ | $NR_{70}$ | $NR_{80}$ | $NR_{90}$ |
|---|---|---|---|---|---|---|
| RetinaNet [5] | 81.3 | 97.6 | 94.9 | 87.0 | 72.1 | 49.5 |
| LTM($\mathcal{M}_+$) [13] | 83.8 | **99.2** | **97.5** | 89.5 | 74.3 | 53.1 |
| LTM | **84.3** | 98.5 | 96.5 | **89.8** | **76.2** | **55.2** |

TABLE 5
Ablation study of hyper-parameters on COCO $val$ set. (a) Number of anchors ($K$) in each anchor bag. (b) Regularization factor $\beta$ for regression loss (Eq. 1).

(a)

| $K$ | AP | $AP_{50}$ | $AP_{75}$ |
|---|---|---|---|
| 10 | 37.3 | 56.2 | 39.5 |
| 20 | 38.1 | 56.8 | 40.9 |
| 30 | **38.4** | 57.5 | **41.2** |
| 40 | **38.4** | **57.6** | **41.2** |
| 50 | 38.2 | 57.2 | 41.0 |

(b)

| $\beta$ | AP | $AP_{50}$ | $AP_{75}$ |
|---|---|---|---|
| 0.3 | 37.9 | **57.8** | 40.2 |
| 0.4 | 38.1 | 57.5 | 40.7 |
| 0.5 | **38.4** | 57.6 | **41.2** |
| 0.6 | 38.2 | 57.0 | 40.7 |
| 0.7 | **38.4** | 57.0 | 40.9 |

compatibility of anchors' predictions with NMS, we defined the NMS Recall ($NR_\tau$), which denotes the ratio of the recall rates after and before NMS with the IoU threshold $\tau$. Higher $NR_\tau$ indicates smaller error of NMS, which causes accurate bounding box predictions be suppressed. Following the COCO-style AP metric [37], NR was defined as the averaged $NR_\tau$ where $\tau$ changes from 0.50 to 0.90 with an interval of 0.05. In Table 4, we compared RetinaNet with LTM in terms of their $NR_\tau$. LTM reported significantly higher $NR_\tau$ (84.3% vs. 81.3%), which means better compatibility with NMS. This validated that the detection customized likelihood (defined in Section 3.1) can guarantee joint optimization of classification and localization.

### 4.3 Ablation Studies

To quantitatively investigate the effect of the learning-to-match method, we conducted ablation studies on COCO $val$ set using ResNet-50 as the backbone network.

**Hyper-parameters.** Beyond the hyper-parameters from the baseline detector, our detector has two important hyper-parameters introduced. The first one is the anchor bag size $K$, which determines how many anchors/points are included in each anchor/point bag. As shown in Table 5a, The highest AP performance (38.4%) was achieved when $K = 30 \sim 40$ while marginal performance drops occurred

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2021.3050494, IEEE Transactions on Pattern Analysis and Machine Intelligence



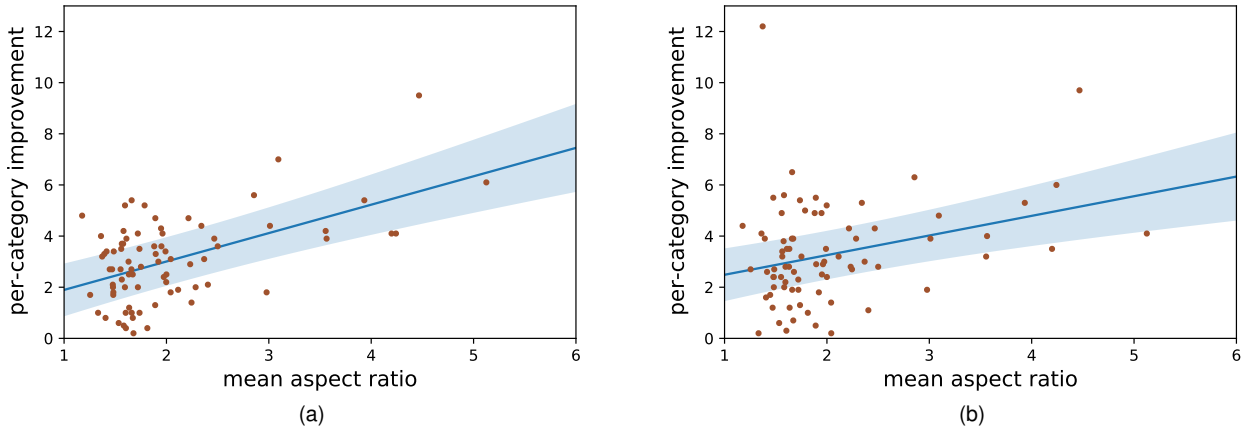



Fig. 7. Performance improvement with respect to object aspect ratios of LTM (a) and LTM-AF (b). Each point in the figure denotes an object category. The fitted lines clearly show that LTM has larger performance improvements over object categories of larger aspect ratios.

TABLE 6
Comparison of training time, detection speed and AP on COCO $test\text{-}dev$ set. Multi-scale training is not used.

| Backbone | Detector | Training time | FPS | AP |
|---|---|---|---|---|
| ResNet-50 | RetinaNet [5] | 5.02h | 9.8 | 35.7 |
| | LTM | 5.27h | 9.1 | **39.2** |
| | LTM-AF | 4.89h | 9.8 | 39.1 |
| ResNet-101 | RetinaNet [5] | 6.96h | 8.0 | 37.8 |
| | LTM | 7.26h | 7.9 | **41.1** |
| | LTM-AF | 6.75h | 8.5 | 41.0 |

given $K$ = 10, 20 or 50. This indicated that our approach is not sensitive to the number of anchors/points used for anchor construction. As long as a bag covers the optimal anchor, the learning-to-match process can select the optimal anchor despite of noise anchors.

The second hyper-parameter is the regression factor $\beta$, which is defined in Eq. 1 to balance the importance of object classification and object localization. In Table 5b, we reported the performance given different values of $\beta$. The highest AP (38.4%) was achieved at $\beta$ = 0.5, whereas $AP_{50}$ and $AP_{75}$ reached the highest values at 0.3 and 0.5, respectively. This indicated the positive correlation between the regression accuracy and the regression factor.

**Bag Construction.** Top-$K$ anchors were used for bag construction according their spatial alignment (IoU) with the objects. To verify the effect of spatial alignment, we tested the detection performance under different $K$ values, Table 3. Smaller $K$ implies anchors of better spatial alignment. We randomly select $S = 40$ anchors from the top-$K$ anchors for bag construction. Experiments show that smaller $K$ corresponds to higher performance, validating that spatial alignment (IoU) remains an important factor for object-feature correspondence.

**Anchor-free Detector.** As a general learning-to-match method, LTM was applied on the anchor-free detector which performs object detection in a per-pixel prediction fashion [29]. Following the way to construct anchor bags, we construct feature point bags, where each pixel is a feature point. Experimentally, the number of feature points in each bag was set to $K = 40$ and the regularization factor for bounding box regression was set to $\beta = 5.5$. In Table 6, it can be seen that the LTM-AF detector achieved comparable performance with the LTM detector.

**Consistency.** Our proposed detectors consistently improved the detection performance on deeper or shallower backbone networks in both anchor-based and anchor-free frameworks. With ResNet-50 and ResNet-101, LTM and LTM-AF respectively outperformed the baseline method by 3.5%, 3.4%, 3.3%, and 3.2%, Table 6, indicating the consistent effectiveness of our proposed approach.

**Efficiency.** The LTM module requires negligible computational cost as no additional architecture or parameter introduced during the training and inference phases. During the training phase, it solely introduced two additional anchor matching losses compared with the baseline RetinaNet method. The learning-to-match method starts by using a bag of anchors/features to train the detector while ends by matching anchors of high confidence. In Table 6, the training time and inference speed were compared. All detectors were trained using 8× Tesla V100 GPUs and tested using a single GTX 1080Ti GPU with CUDA 10. It can be seen that LTM achieved significant performance gains with negligible computational cost. The LTM-AF detector achieved comparable performance gains and a higher detection speed.

### 4.4 Performance

In Table 7, the proposed LTM and LTM-AF detectors were evaluated on advanced backbone networks by training 2× iterations (180K) and using scale jitter. The experiments were carried out on COCO $test\text{-}dev$ set and compared with state-of-the-art two-stage and one-stage detectors.

The LTM detector achieved state-of-the-art performance, outperforming most one-stage and two-state detectors including FasterRCNN+FPN, Libra RCNN, IoU-Net, and TridentNet. With a ResNeXt-101 backbone, it reported new state-of-the-art detection performance 44.9%. The performance further improved to 46.3% when multi-scale testing was used. It outperformed the state-of-the-art CenterNet and TridentNet detectors using HourGlass-104 backbone.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2021.3050494, IEEE Transactions on Pattern Analysis and Machine Intelligence



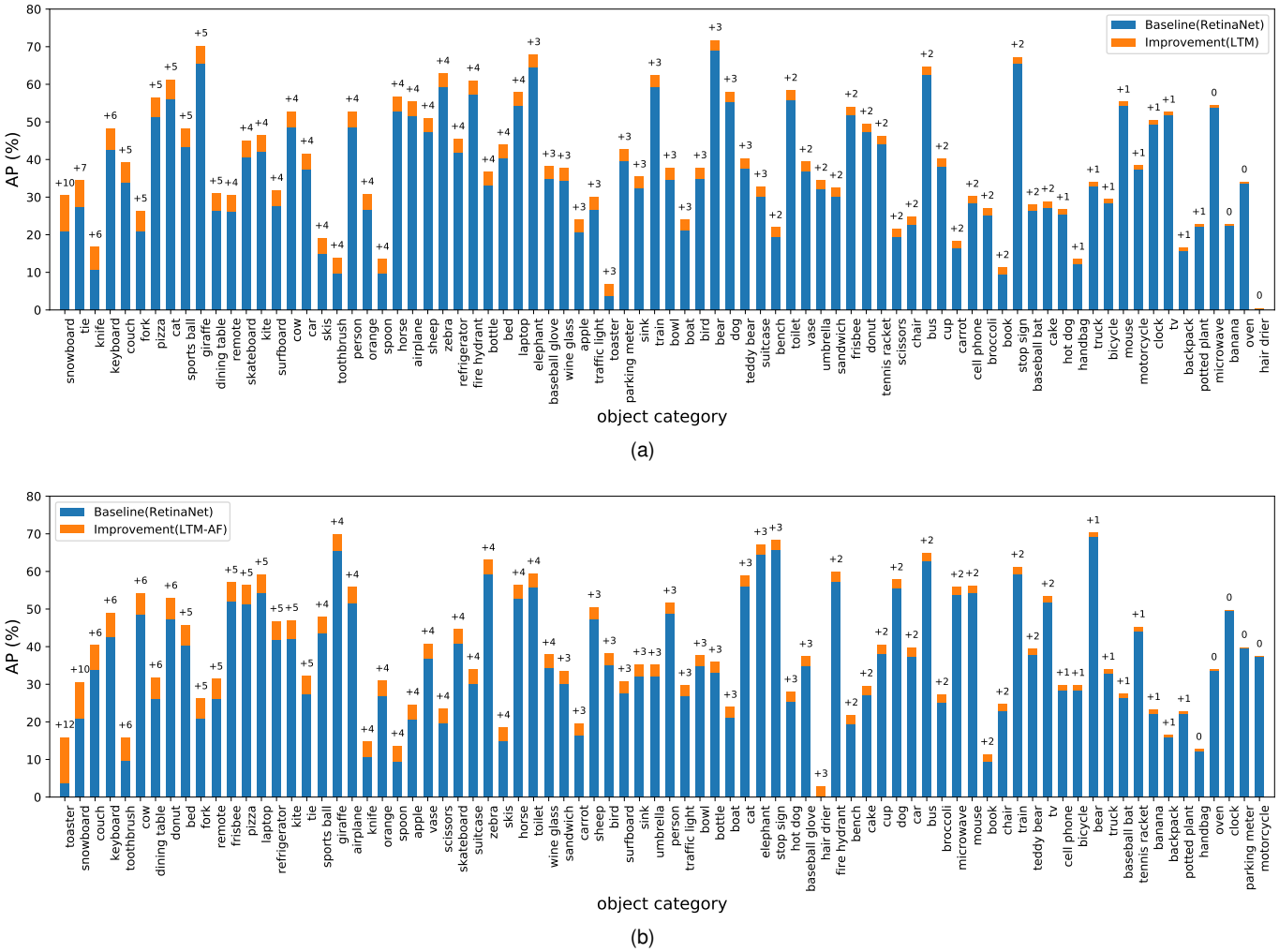



Fig. 8. Category-wised performance and improvement of LTM (a) and LTM-AF (b) on the COCO $test\text{-}dev$ set using ResNet-50. For the categories about slender objects, $e.g.$, "snowboard", "toaster", "tie" and "keyboard", LTM and LTM-AF significantly improved the baseline method (RetinaNet).

CenterNet used a backbone network with significantly more network parameters than that of our backbone network (210.1 M vs. 96.9 M). In that case, the detection speed of LTM (8.2 FPS) is more than two times higher than that of CenterNet (3.3 FPS). With the same ResNet101 backbone, LTM outperformed the Cascade R-CNN detector which used multi-stage prediction correction.

The LTM-AF detector also achieved state-of-the-art performance. With the ResNet-101 backbone, it respectively outperformed recent anchor-free detectors FCOS and FSAF by 2.5% (43.4% vs. 40.9%) and 1.9% (43.4% vs. 41.5%) which are significant margins. When considering the $AP_{75}$ metric with ResNet-101, LTM-AF respectively outperformed the FCOS and FSAF by 1.6%(46.6% vs. 45.0%) and 2.6%(46.6% vs. 44.0%). For $AP_L$ it respectively outperformed the FCOS and FSAF by 3.3%(54.9% vs. 51.6%) and 3.6%(54.9% vs. 51.3%). With all the experiments in Table 7, it is concluded that LTM-AF reported new state-of-the-art (43.4% with ResNet-101 and 44.9% with ResNeXt-101) for anchor-free detectors, as well as filling the gap between anchor-based and anchor-free detectors.

Comparisons of detection results in Fig. 9 show that LTM and LTM-AF detected more slender objects and objects of partial occlusion. The most representative features of these objects bias from their geometric centers, which challenged the IoU-based anchor assignment but can be well handled by the learning-to-match mechanism.

## 5 Conclusion

We proposed the elegant and effective learning-to-match (LTM) approach for visual object detection. LTM updated the hand-crafted anchor assignment to "free" object-anchor correspondence by formulating detector training as a maximum likelihood estimation (MLE) framework. In the framework, we proposed positive and negative anchor matching functions, based on which we improved the performance of object detection, in striking contrast with the baseline detector. We extended the learning-to-match method from anchor-based detectors to anchor-free detectors and closed the performance gap between the anchor-based and anchor-free detectors. We further provided theoretical analysis about the anchor-matching functions from a perspective of generative linear models. The learning-to-match method and anchor-matching mechanism provide fresh insights for object representation and object localization problems in the deep learning framework.

TABLE 7
Performance comparison with state-of-the-art detectors on MS COCO $test\text{-}dev$ dataset. LTM($\mathcal{M}_+$) uses only positive anchor matching while LTM uses both positive and negative anchor matching. 'multi-scale' denotes input images are jittered over scales and results are fused when testing.

| Method | Backbone | Anchor Free | FPS | AP | $AP_{50}$ | $AP_{75}$ | $AP_S$ | $AP_M$ | $AP_L$ |
|---|---|---|---|---|---|---|---|---|---|
| **Multi-stage detectors** | | | | | | | | | |
| Faster R-CNN+++ [34] | ResNet-101 | | | 34.9 | 55.7 | 37.4 | 15.6 | 38.7 | 50.9 |
| Faster R-CNN w FPN [3] | ResNet-101 | | 9.9 | 36.2 | 59.1 | 39.0 | 18.2 | 39.0 | 48.2 |
| GA-Faster-RCNN [25] | ResNet-50 | Y | 9.4 | 39.8 | 59.2 | 43.5 | 21.8 | 42.6 | 50.7 |
| IoU-Net [21] | ResNet-101 | | | 40.6 | 59.0 | - | - | - | - |
| RPDet [32] | ResNet-ResNet101 | Y | 10.0 | 41.0 | 62.9 | 44.3 | 23.6 | 44.1 | 51.7 |
| Libra R-CNN [41] | ResNet-101 | | 9.5 | 41.1 | 62.1 | 44.7 | 23.4 | 43.7 | 52.5 |
| Libra R-CNN [41] | ResNeXt-101-64x4d | | 5.6 | 43.0 | 64.0 | 47.0 | 25.3 | 45.6 | 54.6 |
| Cascade R-CNN [42] | ResNet-101 | | 8.0 | 42.8 | 62.1 | 46.3 | 23.7 | 45.5 | 55.2 |
| TridentNet [43] | ResNet-101 | | 2.7 | 42.7 | 63.6 | 46.5 | 23.9 | 46.6 | 56.6 |
| **Single-stage detectors** | | | | | | | | | |
| RetinaNet [5] | ResNet-101 | | 8.0 | 39.1 | 59.1 | 42.3 | 21.8 | 42.7 | 50.2 |
| GA-RetinaNet [25] | ResNet-50 | Y | 10.8 | 37.1 | 56.9 | 40.0 | 20.1 | 40.1 | 48.0 |
| FCOS [29] | ResNet-101 | Y | 9.3 | 41.5 | 60.7 | 45.0 | 24.4 | 44.8 | 51.6 |
| FCOS [29] | ResNeXt-101-64x4d | Y | 5.4 | 44.7 | 64.1 | 48.4 | 27.6 | 47.5 | 55.6 |
| FSAF [44] | ResNet-101 | Y | 7.1 | 40.9 | 61.5 | 44.0 | 24.0 | 44.2 | 51.3 |
| FSAF [44] | ResNeXt-101-64x4d | Y | 4.2 | 42.9 | 63.8 | 46.3 | 26.6 | 46.2 | 52.7 |
| CornerNet [30] | HourGlass-104 | Y | 3.1 | 40.5 | 56.5 | 43.1 | 19.4 | 42.7 | 53.9 |
| CenterNet [8] | HourGlass-104 | Y | 3.3 | 44.9 | 62.4 | 48.1 | 25.6 | 47.4 | **57.4** |
| **Ours** | | | | | | | | | |
| LTM($\mathcal{M}_+$) [13] | ResNet-101 | | 8.0 | 43.1 | 62.2 | 46.4 | 24.5 | 46.1 | 54.8 |
| LTM | ResNet-101 | | 8.2 | 43.8 | 62.7 | 47.1 | 25.1 | 46.6 | 55.2 |
| LTM | ResNeXt-101-64x4d | | 5.1 | 44.9 | 64.7 | 48.3 | 26.9 | 47.8 | 55.8 |
| LTM(multi-scale) | ResNeXt-101-64x4d | | 1.6 | **46.3** | 65.9 | **50.3** | **30.5** | 48.9 | **57.4** |
| LTM-AF | ResNet-101 | Y | 8.3 | 43.4 | 63.7 | 46.6 | 24.3 | 46.9 | 54.9 |
| LTM-AF | ResNeXt-101-64x4d | Y | 5.2 | 44.9 | 65.5 | 48.1 | 26.1 | 48.4 | 56.7 |
| LTM-AF(multi-scale) | ResNeXt-101-64x4d | Y | 1.7 | **46.3** | **66.7** | 50.1 | 30.1 | **49.0** | 57.3 |

## APPENDIX: DERIVATION OF MATCHING FUNCTION

We derive the anchor matching functions, Eqs. 10 and 11, by reformulating anchor matching as latent variable learning and converting the likelihood calculating with a generalized linear model (GLM).

**Latent Variable Model.** Given the labels of anchor bags, the anchor matching problem (Section 3.1.3) can be formulated as a latent variable model, where a latent variable $Z_i \in \{1, 2, \cdots, j, \cdots\}$ is defined to indicate the instantaneously matched anchors in bag $A_i$. The solution for the latent variable should guarantee $\sum_j p(Z_i|A_i;\theta) = 1$ and $p(Y_i, Z_i = 1|A_i;\theta) = p(y_{ij} = Y_i|A_i;\theta)$. In that case the likelihood of anchor bags is approximated by that of anchors, as

$$p(Y_i|A_i;\theta) = \sum_{Z_i} p(y_{ij} = Y_i|A_i;\theta)p(Z_i|A_i;\theta), \tag{15}$$

which defines an anchor matching model in terms of bag likelihood, anchor likelihood, and the latent variable.

The latent variable $Z_i$ in Eq. 15 requires to be simultaneously learned with network parameter $\theta$. This procedure can be formulated as an Expectation-Maximization (EM) algorithm consist of E-steps and M-steps is employed, as

(E-step:) For each bag $A_i$, we define the distribution for $Z_i$ as

$$Q(Z_i) = \frac{p(Y_i, Z_i|A_i;\theta)}{\sum_{Z_i} p(Y_i, Z_i|A_i;\theta)}.$$

(M-step:) With all bags, we maximize the likelihood as

$$\theta = \arg\max_\theta \sum_i \sum_{Z_i} Q(Z_i) \log p(Y_i, Z_i|A_i;\theta),$$

where the likelihood for the bag $A_i$ can be converted to

$$\mathcal{P}_i(\theta) = \prod_{Z_i} p(Y_i, Z_i|A_i;\theta)^{Q(Z_i)}. \tag{16}$$

**Generalized Linear Model (GLM).** GLM defines a family of models used to solve the problem that given input features and model parameters. The learning objective follows an exponential family distribution parameterized with $\eta$ [45]. In CNNs, the commonly used activation function, $i.e.$, the sigmod function, is a GLM, as

$$\phi = 1/(1 + e^{-\eta}), \tag{17}$$

where $\phi$ denotes likelihood output by the detector. In what follows, we show that the anchor matching functions defined in Eqs. 10 and 11 can be derived based on Eq. 17.

We define the likelihood of an anchor bag as

$$\phi_j = p(Y_i, Z_i = j|A_i;\theta), \tag{18}$$

and

$$q(y) = [Q(Z_i = 1), \cdots, Q(Z_i = j), \cdots]^T$$

as the target distribution, Eq. 16, is converted to

$$p(y; \phi_1, \cdots, \phi_j, \cdots) = \prod_j \phi_j^{q(y)_j},$$

which defines a multinomial distribution in the exponential family, as

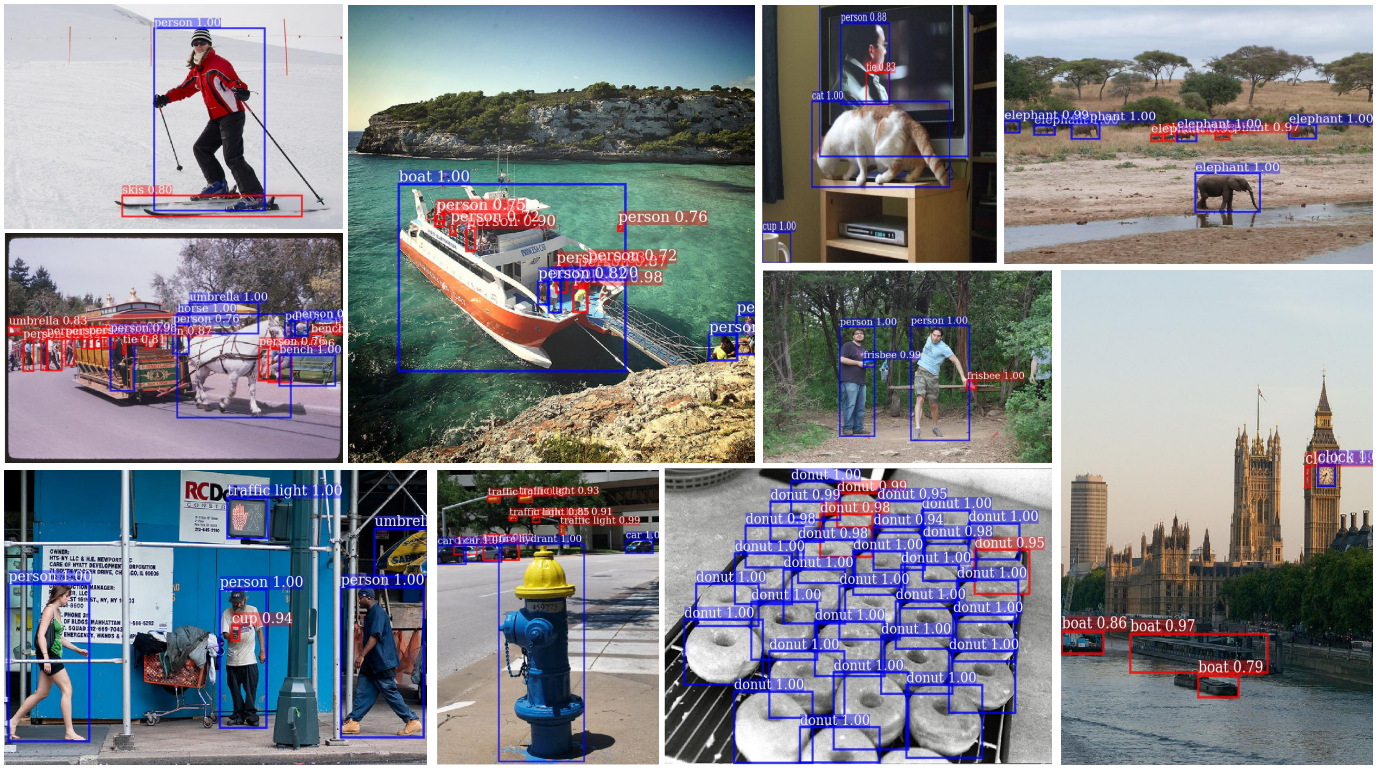
$$\begin{aligned} p(y; \phi_1, \cdots, \phi_j, \cdots) &= \exp\left([\log\phi_1 \cdots \log\phi_j \cdots]^T q(y)\right) \\ &= e^{\eta_0} \exp\left(\eta^T q(y)\right), \end{aligned}$$

by defining $\eta_0 + \eta_j = \log\phi_j$ and according to Eq. 18, we have

$$\phi_j = p(Y_i, Z_i = j|A_i;\theta) = e^{\eta_0} e^{\eta_j}. \tag{19}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2021.3050494, IEEE Transactions on Pattern Analysis and Machine Intelligence





(a)

(b)

Fig. 9. Examples of object detection results on COCO *val* set. (a) Comparison of detection results by RetinaNet (blue boxes) and LTM (red and blue boxes). (b) Comparison of detection results by RetinaNet (blue boxes) and LTM-AF (red and blue boxes). A score threshold of 0.7 is used to display the detection results on images. It can be seen that LTM and LTM-AF detected more slender objects and objects of partial occlusion, particularly when multiple object come together. It also can be seen that LTM and LTM-AF achieved similar detection results, which show that the performance gap between anchor-based and anchor-free detectors are largely closed. (Best viewed in color and with zoom)

**Anchor Matching Functions.** According to Eq. 17, the likelihood of a positive anchor bag is

$$p(y_{ij} = Y_i | A_i; \theta) = \frac{1}{1 + e^{-\eta_{ij}}}. \tag{20}$$

According to Eq. 19 and Eq. 20, we have

$$p(Z_i = j|A_i;\theta) = \frac{p(Y_i, Z_i = j|A_i;\theta)}{p(y_{ij} = Y_i|A_i;\theta)} = \frac{e^{\eta_0} e^{\eta_{ij}}}{\frac{1}{1+e^{-\eta_{ij}}}}.$$

According to the latent variable definition, the value of $\eta_0$ must satisfy $\sum_j p(Z_i = j|A_i;\theta) = 1$. Thus we have

$$\sum_j \frac{e^{\eta_0} e^{\eta_{ij}}}{\frac{1}{1+e^{-\eta_{ij}}}} = 1,$$

and

$$e^{\eta_0} = \frac{1}{\sum_j (1 + e^{\eta_{ij}})}.$$

When anchor $j$ in anchor bag $A_i$ is matched, we have $\eta_j = \eta_{ij}$. Upon substitution of $\eta_0$ in Eq. 19 and calculating $e^{\eta_{ij}} = \frac{\phi_{ij}}{1-\phi_{ij}}$ according to Eq. 17, we have

$$\begin{aligned} p(Y_i = 1, Z_i = j|A_i;\theta) &= \frac{e^{\eta_{ij}}}{\sum_j (1 + e^{\eta_{ij}})} \\ &= \frac{\frac{\phi_{ij}}{1-\phi_{ij}}}{\sum_j \frac{1}{1-\phi_{ij}}}. \end{aligned} \tag{21}$$

According to Eq. 20, the likelihood of a negative anchor bag is

$$p(y_{ij} = Y_i|A_i;\theta) = 1 - \frac{1}{1 + e^{-\eta_{ij}}} = \frac{1}{1 + e^{\eta_{ij}}}. \tag{22}$$

Following the derivation above, we can conclude

$$\begin{aligned} p(Y_i = 0, Z_i = j|A_i;\theta) &= \frac{e^{\eta_{ij}}}{\sum_j e^{\eta_{ij}} (1 + e^{\eta_{ij}})} \\ &= \frac{\frac{\phi_{ij}}{1-\phi_{ij}}}{\sum_j \frac{\phi_{ij}}{(1-\phi_{ij})^2}}. \end{aligned} \tag{23}$$

By summing on $j$, we derive the anchor matching function from Eq. 21 and Eq. 23, as

$$\begin{aligned} p(Y_i = 1|A_i;\theta) &= \sum_j p(Y_i = 1, Z_i = j|A_i;\theta) \\ &= \frac{\sum_j \frac{\phi_{ij}}{1-\phi_{ij}}}{\sum_j \frac{1}{1-\phi_{ij}}}, \end{aligned} \tag{24}$$

and

$$\begin{aligned} p(Y_i = 0|A_i;\theta) &= \sum_j p(Y_i = 0, Z_i = j|A_i;\theta) \\ &= \frac{\sum_j \frac{\phi_{ij}}{1-\phi_{ij}}}{\sum_j \frac{\phi_{ij}}{(1-\phi_{ij})^2}}. \end{aligned} \tag{25}$$

Eq. 24 and Eq. 25 are exactly same as Eq. 10 and Eq. 11. As GLMs [45], the anchor matching functions convert the likelihood of an anchor bag as a combination of a set of observed anchors. In this way, the learning procedure comprehensively considers all the anchors in each positive bag by summarizing theirs likelihood to pursue the optimal anchors and network parameters. This facilities learning more representative features for object detection while prevent getting stuck into local solutions from the perspective of sufficient statistics.

## ACKNOWLEDGMENTS

The authors would like to express their sincere appreciation to the editors and the reviewers for their constructive comments. This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 61836012 and 61771447 and Post Doctoral Innovative Talent Support Program of China under Grant 119103S304.

## REFERENCES

[1] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2014, pp. 580–587.

[2] R. B. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2015, pp. 1440–1448.

[3] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," in *Adv. in Neural Inf. Process. Syst. (NeurIPS)*, 2015, pp. 91–99.

[4] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 779–788.

[5] T. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2017, pp. 2999–3007.

[6] T. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017, pp. 936–944.

[7] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C. Fu, and A. C. Berg, "SSD: single shot multibox detector," in *Proc. Europ. Conf. Comput. Vis. (ECCV)*, 2016, pp. 21–37.

[8] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, "Centernet: Object detection with keypoint triplets," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019.

[9] F. Wan, P. Wei, Z. Han, J. Jiao, and Q. Ye, "Min-entropy latent model for weakly supervised object detection," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 41, no. 10, pp. 2395–2409, 2019.

[10] F. Wan, C. Liu, W. Ke, X. Ji, J. Jiao, and Q. Ye, "C-mil: Continuation multiple instance learning for weakly supervised object detection," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019, pp. 2199–2208.

[11] R. A. Fisher, "On the mathematical foundations of theoretical statistics," *Philosophical Transactions of the Royal Society of London. Series A*, vol. 222, no. 594-604, pp. 309–368, 1922.

[12] O. Maron and T. Lozano-Pérez, "A framework for multiple-instance learning," in *Adv. in Neural Inf. Process. Syst. (NeurIPS)*, 1997, pp. 570–576.

[13] X. Zhang, F. Wan, C. Liu, R. Ji, and Q. Ye, "FreeAnchor: Learning to match anchors for visual object detection," in *Adv. in Neural Inf. Process. Syst. (NeurIPS)*, 2019.

[14] L. Liu, W. Ouyang, XiaogangWang, P. Fieguth, J. Chen, X. Liu, and M. Pietikainen, "Deep learning for generic object detection: A survey," *Int. J. Comput. Vis.*, vol. 128, no. 2, pp. 261–318, 2020.

[15] S. Agarwal, J. O. du Terrail, and F. Jurie, "Recent advances in object detection in the age of deep convolutional neural networks," *arXiv:1809.03193*, 2018.

[16] Z. Zou, Z. Shi, Y. Guo, and J. Ye, "Object detection in 20 years: A survey," *arXiv:1905.05055*, 2019.

[17] G. Ghiasi, T. Lin, and Q. V. Le, "NAS-FPN: learning scalable feature pyramid architecture for object detection," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019, pp. 7036–7045.

[18] T. Yang, X. Zhang, Z. Li, W. Zhang, and J. Sun, "Metaanchor: Learning to detect objects with customized anchors," in *Adv. in Neural Inf. Process. Syst. (NeurIPS)*, 2018, pp. 320–330.

[19] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg, "Dssd: Deconvolutional single shot detector," *arXiv:1701.06659*, 2017.

[20] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017, pp. 6517–6525.

[21] B. Jiang, R. Luo, J. Mao, T. Xiao, and Y. Jiang, "Acquisition of localization confidence for accurate object detection," in *Proc. Europ. Conf. Comput. Vis. (ECCV)*, 2018, pp. 784–799.

[22] T. Vu, H. Jang, T. X. Pham, and C. Yoo, "Cascade rpn: Delving into high-quality region proposal network with adaptive convolution," in *Adv. in Neural Inf. Process. Syst. (NeurIPS)*, 2019, pp. 1430–1440.

[23] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 764–773.

[24] S. Zhang, L. Wen, X. Bian, Z. Lei, and S. Z. Li, "Single-shot refinement neural network for object detection," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2018, pp. 4203–4212.

[25] J. Wang, K. Chen, S. Yang, C. C. Loy, and D. Lin, "Region proposal by guided anchoring," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019, pp. 2965–2974.

[26] J. Choi, D. Chun, H. Kim, and H.-J. Lee, "Gaussian yolov3: An accurate and fast object detector using localization uncertainty for autonomous driving," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2019, pp. 502–511.

[27] K. Oksuz, B. C. Cam, S. Kalkan, and E. Akbas, *IEEE Trans. Pattern Anal. Mach. Intell.*

[28] X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, and J. Liang, "EAST: an efficient and accurate scene text detector," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017, pp. 2642–2651.

[29] Z. Tian, C. Shen, H. Chen, and T. He, "Fcos: Fully convolutional one-stage object detection," *arXiv:1904.01355*, 2019.

[30] H. Law and J. Deng, "Cornernet: Detecting objects as paired keypoints," in *Proc. Europ. Conf. Comput. Vis. (ECCV)*, 2018, pp. 765–781.

[31] X. Zhou, J. Zhuo, and P. Krähenbühl, "Bottom-up object detection by grouping extreme and center points," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019, pp. 850–859.

[32] Z. Yang, S. Liu, H. Hu, L. Wang, and S. Lin, "Reppoints: Point set representation for object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2019, pp. 502–511.

[33] C. Zhu, Y. He, and M. Savvides, "Feature selective anchor-free module for single-shot object detection," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019, pp. 840–849.

[34] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 770–778.

[35] S. Xie, R. Girshick, P. Dollar, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017, pp. 1492–1500.

[36] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019, pp. 658–666.

[37] T.-Y. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Proc. Europ. Conf. Comput. Vis. (ECCV)*, 2014, pp. 740–755.

[38] Y. Wu and K. He, "Group normalization," in *Proc. Europ. Conf. Comput. Vis. (ECCV)*, 2018, pp. 3–19.

[39] S. Shao, Z. Zhao, B. Li, T. Xiao, G. Yu, X. Zhang, and J. Sun, "Crowdhuman: A benchmark for detecting human in a crowd," *CoRR*, vol. abs/1805.00123, 2018.

[40] E. Goldman, R. Herzig, A. Eisenschtat, J. Goldberger, and T. Hassner, "Precise detection in densely packed scenes," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019, pp. 5227–5236.

[41] J. Pang, K. Chen, J. Shi, H. Feng, W. Ouyang, and D. Lin, "Libra R-CNN: towards balanced learning for object detection," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019, pp. 821–830.

[42] Z. Cai and N. Vasconcelos, "Cascade r-cnn: Delving into high quality object detection," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2018, pp. 6154–6162.

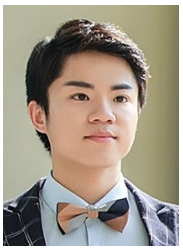
[43] Y. Li, Y. Chen, N. Wang, and Z. Zhang, "Scale-aware trident networks for object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2019, pp. 6053–6062.

[44] C. Zhu, Y. He, and M. Savvides, "Feature selective anchor-free module for single-shot object detection," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019, pp. 840–849.

[45] P. McCullagh and J. A. Nelder, *Generalized Linear Models*. Springer, 1989.

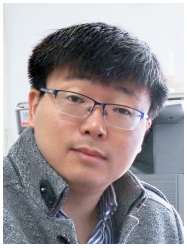

**Xiaosong Zhang** received the B.S. degree from Harbin Institute of Technology (HIT), Weihai, Shandong, China, in 2018. Since 2018, he has been a Ph.D student in the School of Electronic, Electrical and Communication Engineering, University of Chinese Academy of Sciences, Beijing, China. His research interests include computer vision and machine learning, specifically for visual object detection and representation learning.

**Fang Wan** received the B.S. degree from Wuhan University, Wuhan, China, in 2013. Since 2013, he has been a Ph.D student in the School of Electronic, Electrical and Communication Engineering, University of Chinese Academy of Sciences, Beijing, China. His research interests include computer vision and machine learning, specifically for weakly supervised learning and visual object detection. He has published more than 10 papers in refereed conferences and journals including IEEE CVPR, ICCV, NeurIPS, and PAMI, and received President Award of Chinese Academy of Sciences.



**Chang Liu** received the B.S. degree from Jilin University, Jilin, China, in 2012. Since 2015, he has been a Ph.D student in the School of Electronic, Electrical and Communication Engineering, University of Chinese Academy of Sciences, Beijing, China. His research interests include computer vision and machine learning, specifically for neural architecture design and visual object detection.

**Xiangyang Ji** (M'10) received the B.S. degree in materials science and the M.S. degree in computer science from the Harbin Institute of Technology, Harbin, China, in 1999 and 2001, respectively, and the Ph.D. degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China. He joined Tsinghua University, Beijing, in 2008, where he is currently a Professor with the Department of Automation, School of Information Science and Technology. He has authored over 100 referred conference and journal papers. His current research interests include signal processing, image/video compressing, and intelligent imaging.



**Qixiang Ye** (M'10-SM'15) received the B.S. and M.S. degrees from Harbin Institute of Technology, China, in 1999 and 2001, respectively, and the Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences in 2006. He has been a professor with the University of Chinese Academy of Sciences (UCAS) since 2015, and was a visiting assistant professor with the University of Maryland, College Park until 2013. His research interests include visual object detection and machine learning. He has published more than 100 papers in refereed conferences and journals including IEEE CVPR, ICCV, ECCV, NeurIPS, and PAMI, and received the Sony Outstanding Paper Award. He is a senior member of IEEE.