DEEP MULTIVARIATE MIXTURE OF GAUSSIANS FOR OBJECT DETECTION UNDER OCCLUSION

Anonymous authors
Paper under double-blind review

ABSTRACT

In this paper, we consider the problem of detecting object under occlusion. Most object detectors formulate bounding box regression as a unimodal task (*i.e.*, regressing a single set of bounding box coordinates independently). However, we observe that the bounding box borders of an occluded object can have multiple plausible configurations. Also, the occluded bounding box borders have correlations with visible ones. Motivated by these two observations, we propose a deep multivariate mixture of Gaussians model for bounding box regression under occlusion. The mixture components potentially learn different configurations of an occluded part, and the covariances between variates help to learn the relationship between the occluded parts and the visible ones. Quantitatively, our model improves the AP of the baselines by **3.9%** and **1.2%** on CrowdHuman and MS-COCO respectively with almost no computational or memory overhead. Qualitatively, our model enjoys explainability since we can interpret the resulting bounding boxes via the covariance matrices and the mixture components.

1 INTRODUCTION

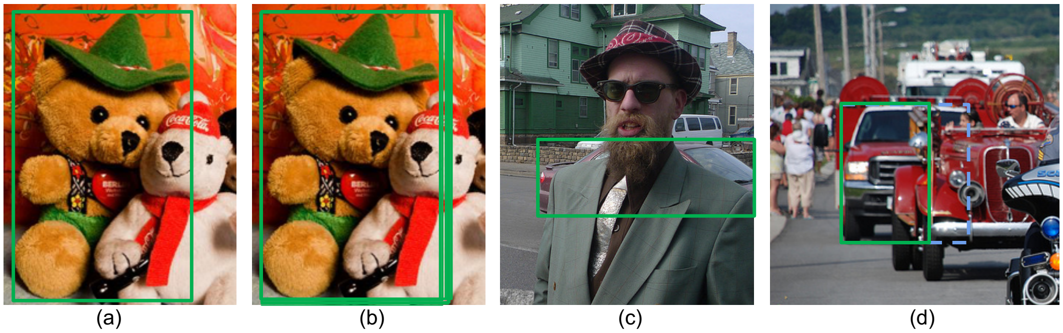


Figure 1: We observe that an occluded bounding box usually exhibits multiple modes in most detection datasets, no matter whether the ground truth annotation is visible box or full box: (a) visible bounding box annotation (b) full object bounding box labeled by different annotators (c) visible bounding box annotated accurately (d) visible bounding box annotated inaccurately

Object detectors based on deep convolutional neural networks (CNNs) are the backbone of many real-world applications like self-driving cars (Huval et al., 2015), robotics grasping (Calandra et al., 2018) and video surveillance (Joshi & Thakore, 2012). Most object detectors learn to detect an object in two folds (Ren et al., 2015): (1) categorization of the candidate bounding box (2) regress each coordinate of the candidate box towards the ground truth one independently.

Currently, there are two styles of bounding box annotation among the large-scale object detection datasets: (1) visible box that only contains visible parts (*e.g.*, MS-COCO (Lin et al., 2014) and PASCAL VOC (Everingham et al., 2010)) (2) full box that contains both visible and occluded parts (*e.g.*, CrowdHuman (Shao et al., 2018) and VehicleOcclusion (Wang et al., 2017)). For full box annotation, regressing a single set of bounding box coordinates works well for fully visible objects, since it is a unimodal problem. However, when an object is occluded, we observe that its occluded parts can have several plausible configurations (*e.g.*, Figure 1 (b)), which is a multimodal problem. Even for visible box annotation, an object sometimes still exhibits multiple modes due to inaccurate labeling (*e.g.*, Figure 1 (c) *vs*. (d)). We argue that an object detector robust to occlusion should learn a multimodal distribution with the capability of proposing more than one plausible hypothesis for the configuration of an occluded part.

Besides, we also observe that the bounding box coordinates have correlations by nature. Take Figure 1 (c) as an example, by knowing the position of the car's roof, we can easily infer the location of the left border even without looking at it. Therefore, an object detector robust to occlusion also needs to be capable of inferring the correlations between the occluded bounding box borders and the visible ones.

Motivated by these two observations, we propose a deep multivariate mixture of Gaussians model for object detection under occlusion. Concretely, instead of regressing a single set of bounding box coordinates, our model regresses several sets of coordinates, which are the means of the Gaussians. Moreover, we learn a covariance matrix for the coordinates of each Gaussian mixture component. These components are summed together as the prediction for the distribution of plausible bounding box configurations. At inference time, we choose the expectation of our model's distribution as the final predicted bounding box.

To demonstrate the generalizability of our proposed model, we conduct experiments on four datasets: CrowdHuman, MS-COCO, VehicleOcclusion, and PASCAL VOC 2007. Quantitatively, our model improves the AP (Average Precision of the baselines by **3.9%** and **1.2%** on CrowdHuman and MS-COCO respectively (Table 1 and Table 2). Qualitatively, our model enjoys explainability since the resulting bounding boxes can be interpreted using the covariance matrices and the Gaussian mixture components (Figure 5 and Figure 4). More importantly, our model is almost computation and memory free, since predicting the mixture components only requires a fully-connected layer, and we can discard the covariance matrices at inference time (Table 5).

2 Related Work

Object Detection: Deep convolutional neural networks were first introduced to object detection in R-CNN (Girshick et al., 2014) and Fast R-CNN (Girshick, 2015). Currently, there are mainly two types of object detectors: one-stage object detectors and two-stage object detectors. One-stage detectors like YOLO (Redmon et al., 2016), SSD (Liu et al., 2016) and RetinaNet (Lin et al., 2018) are fast in general. Two-stage detectors (Ren et al., 2015; He et al., 2017; Zhu et al., 2018; Singh et al., 2018) are accurate however sacrificing speed. In this paper, although we conduct experiments based on the Faster R-CNN heads of Faster R-CNN and Mask R-CNN, our method is not limited to two-stage detectors.

Object Detection Under Occlusion: Occlusion-aware R-CNN (Zhang et al., 2018b) proposes to divide pedestrian detection into five parts and predict the visibility scores respectively, which are integrated with the prior structure information of the human body into the network to handle occlusion. Zhang et al. (2018a) proposes an attention network with self or external guidance. These methods are specifically designed for pedestrian detection task. By contrast, our method is designed for general object detection.

Deep Voting (Zhang et al., 2018c) proposes to utilize spatial information between visual cues and semantic parts and also learn visual cues from the context outside an object. However, detecting semantic parts needs manual labels, which our approach does not require. Besides, our approach does not introduce additional computation during the inference (Table 5). Amodal instance segmentation (Li & Malik, 2016) considers the task of predicting the region encompassing both visible and occluded parts of an object. The authors propose to add synthetic occlusion to visible objects and retain their original masks, then employ a CNN to learn on the generated composite images, which resembles the VehicleOcclusion in our experiments. He et al. (2019) proposes bounding box regression with uncertainty, which is a degradation case of our model (Gaussian).

Datasets for Detection under Occlusion: Currently, there are three categories of annotation for an occluded object: (1) visible bounding box that contains the visible parts (2) full box that contains both visible and occluded parts of an object annotated by human (3) full box by synthesizing occluders on a visible object. MS-COCO, PASCAL VOC, ImageNet (Deng et al., 2009) and

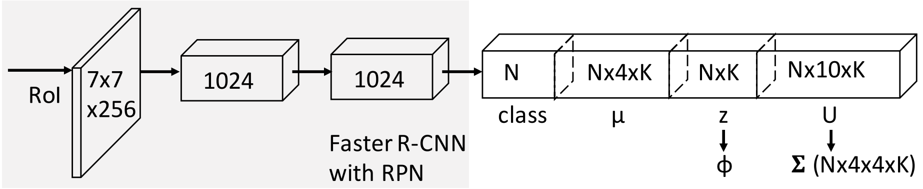


Figure 2: Faster R-CNN head architecture for our approach: We extended the existing Faster R-CNN head to predict the parameters of multivariate mixture of Gaussian $\boldsymbol{\mu}$, $\boldsymbol{\phi}$ and $\boldsymbol{\Sigma}$

Cityscapes (Cordts et al., 2016) fall into the first category. CrowdHuman and Semantic Amodal Segmentation dataset (Zhu et al., 2017) require the annotators to label the invisible parts. VehicleOcclusion instead synthesizes the occluders for visible objects. We conduct experiments on MS-COCO, PASCAL VOC 2007, CrowdHuman, and VehicleOcclusion, covering all these categories.

3 Approach

3.1 Formulation

We observe that when an object is partially occluded, the occluded bounding box border can usually be inferred to some extent by other visible parts of the object (*e.g.*, it is easy to infer the left border of the car given the car roof position in Figure 1 (c)). Besides, the occluded bounding box exhibits multiple modes. For example, the left arm of the teddy bear could have several possible configurations in Figure 1 (b). Motivated by these two observations, we propose to estimate the bounding box coordinates as a probability distribution during bounding box regression instead of a set of deterministic coordinates. Specifically, we propose to estimate a multivariate mixture of Gaussians distribution with a deep network. Multivariate Gaussian helps the case where bounding box borders have correlations, and a mixture of Gaussians helps the case where an occluded bounding box border exhibits multiple modes. Formally, we predict the distribution $p_\theta(\boldsymbol{x}|\boldsymbol{I})$ given the feature maps $\boldsymbol{I}$ of a region of interest (RoI). The distribution is parameterized by θ, which is a neural network (*e.g.*, Faster R-CNN head, Figure 2). The distribution has K components $\mathcal{N}(\boldsymbol{\mu_i}, \boldsymbol{\Sigma_i})$. Each component i has mean $\boldsymbol{\mu}_{i=1...K} = [x_1, y_1, x_2, y_2]^T$, which is the most probable bounding box coordinates relative to the RoI, estimated by the component:

$$\begin{aligned} p_\theta(\boldsymbol{x}|\boldsymbol{I}) &= \sum_{i=1}^{K} \phi_i \mathcal{N}(\boldsymbol{\mu_i}, \boldsymbol{\Sigma_i}) \text{ where } \sum_{i=1}^{K} \phi_i = 1 \text{ and } 0 \le \phi_i \le 1 \\ \mathcal{N}(\boldsymbol{\mu_i}, \boldsymbol{\Sigma_i}) &= \exp\Big(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu_i})^T \boldsymbol{\Sigma_i}^{-1} (\boldsymbol{x} - \boldsymbol{\mu_i})\Big) / \sqrt{(2\pi)^4 |\boldsymbol{\Sigma_i}|} \end{aligned} \tag{1}$$

where each $\mathcal{N}(\boldsymbol{\mu_i}, \boldsymbol{\Sigma_i})$ is a multivariate Gaussian distribution. ϕ_i is a mixture weight scalar for $\mathcal{N}(\boldsymbol{\mu_i}, \boldsymbol{\Sigma_i})$. $|\boldsymbol{\Sigma_i}|$ is the determinant of $\boldsymbol{\Sigma_i}$. $\boldsymbol{\Sigma}$ is the covariance matrix, which is a symmetric semi-positive definite matrix in general. To be able to compute the inverse $\boldsymbol{\Sigma}^{-1}$, we constrain the covariance matrix to be a symmetric positive definite matrix. In this case, the precision matrix $\boldsymbol{\Sigma}^{-1}$ is also a symmetric positive definite matrix. During training, the model estimates the precision matrix $\boldsymbol{\Sigma}^{-1}$ instead of the covariance matrix $\boldsymbol{\Sigma}$, so that we do not need to compute the inverse every time during training which we also find more stable in our experiments. To ensure the properties of the precision matrix $\boldsymbol{\Sigma}^{-1}$, we parameterize it using the Cholesky decomposition:

$$\begin{aligned} \boldsymbol{\Sigma}^{-1} &= \boldsymbol{U}^T \boldsymbol{U} \\ \boldsymbol{U} &= \begin{bmatrix} \exp(u_{11}) & u_{12} & u_{13} & u_{14} \\ & \exp(u_{22}) & u_{23} & u_{24} \\ & & \exp(u_{33}) & u_{34} \\ & & & \exp(u_{44}) \end{bmatrix} \\ |\boldsymbol{\Sigma}| &= \frac{1}{|\boldsymbol{\Sigma^{-1}}|} = \frac{1}{|\boldsymbol{U}^T\boldsymbol{U}|} = \frac{1}{|\boldsymbol{U}^T||\boldsymbol{U}|} = \frac{1}{|\boldsymbol{U}|^2} = \frac{1}{\exp(\sum_{i=1}^4 u_{ii})^2} \end{aligned} \tag{2}$$

where $\boldsymbol{U}$ is an upper triangular matrix with strictly positive diagonal entries, such that Cholesky decomposition is guaranteed to be unique.

We parameterize the mixture weights ϕ_i using Softmax, so that they range from 0 to 1 and sum to 1:

$$\phi_i = \exp(z_i) / \sum_{k=1}^{K} \exp(z_k) \tag{3}$$

z_i, u_{ii} and μ_i are outputs produced by a fully-connected layer on top of the final fully-connected layer `fc7` on the Faster R-CNN head. Take Faster R-CNN with RPN as an example, Figure 2 shows the architecture of our model. Since we only modify a small part of the architecture, our approach might also be applied to other object detectors than Faster R-CNN, like one-stage object detectors YOLO and RetinaNet.

Learning: Our model parameterizes the distribution over bounding boxes using a neural network which depends on RoI features. During training, we estimate the parameters θ with maximum likelihood estimation on a given dataset $\{\boldsymbol{I}_\ell,\ \boldsymbol{\mu}^*_\ell | \ell = 1, 2, ..., N\}$, where $\boldsymbol{\mu}^*_\ell$ represents the ground truth coordinates for RoI feature maps $\boldsymbol{I}_\ell$ and N is the number of observations:

$$\hat{\theta} = \arg\max_{\theta} \frac{1}{N} \sum_{\ell}^{N} \ln p_\theta(\boldsymbol{\mu}^*_\ell | \boldsymbol{I}_\ell) \tag{4}$$

In practice, N is the number of samples in a mini-batch. We use momentum stochastic gradient descent (SGD) to minimize the localization loss L_{loc} and the classification loss L_{cls}:

$$L = L_{cls} + \lambda L_{loc} \text{ where } L_{loc} = -\frac{1}{N} \sum_{\ell}^{N} \ln p_\theta(\boldsymbol{\mu}^*_\ell | \boldsymbol{I}_\ell) \tag{5}$$

Note that we use different parameters θ for different classes in practice. For simplicity, the formulation above only considers the regression problem for a single class.

Inference: During testing, we use the expectation of our mixture module as prediction:

$$\mathbb{E}\left[p_\theta(\boldsymbol{x}|\boldsymbol{I})\right] = \frac{1}{K} \sum_{i=1}^{K} \phi_i \boldsymbol{\mu_i} \tag{6}$$

Notice that the covariance matrix $\boldsymbol{\Sigma}_i$ is not involved in inference. In practice, we discard the neurons that produce the covariance matrix to speed up inference. In our experiments (Table 5), our model has almost the same inference latency and memory consumption as the baseline network.

3.2 Degradation Cases

Multivariate Gaussian: When the number of mixture components $K = 1$, our model degrades into a multivariate Gaussian model. And the localization loss can be rewritten as follow (for simplicity, we only illustrate the loss for a single sample ℓ):

$$\begin{aligned} L^\ell_{loc} &= \frac{(\boldsymbol{\mu}^* - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{\mu}^* - \boldsymbol{\mu})}{2} + \frac{\log|\boldsymbol{\Sigma}|}{2} + 2\ln 2\pi \\ &= \frac{(\boldsymbol{\mu}^* - \boldsymbol{\mu})^T \boldsymbol{U}^T \boldsymbol{U} (\boldsymbol{\mu}^* - \boldsymbol{\mu})}{2} - \sum_{i=1}^{4} u_{ii} + 2\ln 2\pi \end{aligned} \tag{7}$$

where $2\ln 2\pi$ is a constant which can be ignored during training. Multivariate Gaussian model is helpful under occlusion since the borders of a bounding box have correlations with each other inherently. For example, by looking at the location of a car's door, we can guess the location of its roof even if it is occluded.

method	AP	AP^{50}	AP^{75}	AP^{S}	AP^{M}	AP^{L}
baseline	36.7	68.3	35.2	23.4	37.2	40.4
soft-NMS (Bodla et al., 2017)	38.4	72.2	36.8	24.5	38.2	42.8
Gaussian	37.9	69.5	36.2	23.8	37.6	42.1
mixture of 8 Gaussian	39.0	71.0	37.6	25.2	39.0	43.0
multivariate Gaussian	39.6	71.6	38.4	25.6	39.5	43.8
multivariate mixture of 8 Gaussian	**40.6**	**73.1**	**39.6**	**26.8**	**40.8**	**44.4**

Table 1: Performance of our models on CrowdHuman on ResNet-50 FPN Faster R-CNN

method	AP	AP^{50}	AP^{75}	AP^{S}	AP^{M}	AP^{L}
baseline	38.6	**59.8**	42.1	22.2	41.5	50.8
Gaussian	39.0	58.5	42.0	21.8	41.8	51.4
mixture of 8 Gaussian	39.2	58.5	42.4	22.4	41.7	52.6
multivariate Gaussian	39.3	58.8	42.4	22.4	42.1	52.2
multivariate mixture of 8 Gaussian	**39.8**	58.9	**42.9**	**22.5**	**42.2**	**53.0**

Table 2: Performance of our models on MS-COCO on ResNet-50 FPN Mask R-CNN

Mixture of Gaussians: When the covariance matrix is constrained to be a diagonal matrix, our model becomes a mixture of Gaussians model with independent variables:

$$L_{loc}^{\ell} = -\ln \sum_{i=1}^{K} \phi_i \sum_{j=1}^{4} \frac{\exp\Big(-(\boldsymbol{\mu}_{ij}^{*} - \boldsymbol{\mu}_{ij})^2/2(\boldsymbol{U_i})_{jj}^2\Big)}{\sqrt{2\pi}(\boldsymbol{U_i})_{jj}} \tag{8}$$

where $(\boldsymbol{U_i})_{jj}$ is the jth diagonal element of the matrix $\boldsymbol{U_i}$. Multimodality is helpful under occlusion because an occluded object usually has multiple modes.

Gaussian: When the number of mixture components $K = 1$ and the covariance is constrained to be a diagonal matrix, it becomes a simple Gaussian model where different variables are independent:

$$L_{loc}^{\ell} = \sum_{j=1}^{4} (\boldsymbol{U})_{jj}^2 \frac{(\boldsymbol{\mu}_j^{*} - \boldsymbol{\mu}_j)^2}{2} - \ln(\boldsymbol{U})_{jj} + \frac{\ln 2\pi}{2} \tag{9}$$

We argue that this simple model helps detection in most cases. Here $(\boldsymbol{U})_{jj}$ behaves like a balancing term. When the bounding box regression is inaccurate (large $(\boldsymbol{\mu}_j^{*} - \boldsymbol{\mu}_j)^2/2$), the variance $1/(\boldsymbol{U})_{jj}^2$ tends to be larger. Therefore smaller gradient will be provided to bounding box regression $(\boldsymbol{U})_{jj}^2(\boldsymbol{\mu}_j^{*} - \boldsymbol{\mu}_j)^2/2$ in this case, which might help training the network (Table 1 and Table 2). If bounding box regression is perfect, $\boldsymbol{U}$ tend to infinity (*i.e.*, the variance should be close 0). However, regression is not that accurate in practice, $\boldsymbol{U}$ will be punished for being too large.

Euclidean Loss: When all the diagonal elements $(U)_{jj}$ are one $(u_{jj} = 0)$, our model degenerates to the standard euclidean loss:

$$L_{loc}^{\ell} = \sum_{j=1}^{4} \frac{(\boldsymbol{\mu}_j^{*} - \boldsymbol{\mu}_j)^2}{2} + \frac{\ln 2\pi}{2} \tag{10}$$

4 Experiments

We initialize the weights of $\boldsymbol{\mu}_i$, z_i and u_{ii} layers (Figure 2) using random Gaussian initialization with standard deviations 0.0001 and biases 0, -1 and 0 respectively. So that at the start of training, bounding box coordinate $\boldsymbol{\mu}_i$ is at an unbiased position, U_i is an identity matrix and ϕ_i treats each mixture component equally. Our model can be trained end-to-end. Unless specified, we follow settings in Detectron (Girshick et al., 2018) and those original papers.

To demonstrate the generalizability of our method, we conduct experiments on four datasets:

method	mAP
baseline	60.4
Gaussian	61.9
mixture of 8 Gaussian	62.0
multivariate Gaussian	62.3
multivariate mixture of 8 Gaussian	**62.4**

Table 3: Self-comparison on VOC 2007 with VGG-CNN-M-1024 Faster R-CNN

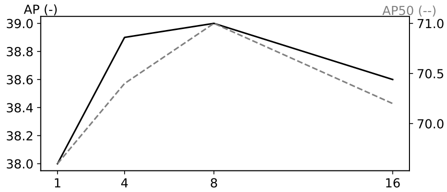


Figure 3: AP and AP50 when varying the number of mixture components.

CrowdHuman (Shao et al., 2018) is a large, rich-annotated and highly diverse dataset for better evaluation of detectors in crowd scenarios. Its training and validation sets contain a total of 470k human instances, and around 22.6 persons every image under various kinds of occlusions. The annotations for occluded bounding boxes are full boxes (Figure 1 (b)) instead of visible boxes (Figure 1 (a)). The experiments are in Table 1.

VehicleOcclusion is a synthetic dataset designed for object detection under occlusion (Wang et al., 2017). Same as above, the annotations are full boxes. The occlusion annotations are more accurate since the occluders (occluding objects) are randomly placed on the annotated visible object. It contains six types of vehicles and occluded instances at various difficulty levels. Specifically, it consists of four occlusion levels: No occlusion (0%), L1 ($20\% \sim 40\%$), L2 ($40\% \sim 60\%$), L3 ($60\% \sim 80\%$). The percentages are computed by pixels. At level L1, L2 and L3, there are two, three, and four occluders placed on the object, respectively (Table 4).

MS-COCO (Lin et al., 2014) is a large-scale object detection dataset containing 80 object categories, $330k$ images ($> 200k$ labeled) and 1.5 million object instances. Compared with the two datasets above, MS-COCO has fewer occlusion cases. For example, the IoU (intersection over union) between overlapped human bounding boxes in MS-COCO are less than 0.7 (Shao et al., 2018). We use `train2017` for training and `val2017` for testing (Table 2). Different from above, the annotations are visible boxes.

PASCAL VOC 2007 has 9,963 images and 20 classes in total, containing 24,640 annotated objects (Everingham et al.). Similar with MS-COCO, this dataset has less occlusion cases than the first two datasets. We use `voc_2007_train` and `voc_2007_val` for training and `voc_2007_test` for testing (Table 3). The annotations are visible boxes.

4.1 Ablation Study

Number of Mixture Components: Shown in Figure 3, we test our mixture of Gaussians model by varying the number of mixture components. The baseline is ResNet-50 FPN Faster R-CNN (He et al., 2016; Lin et al., 2017) on CrowdHuman. As the number of components increases from 1, 4 to 8, we observe consistent performance improvement. The mixture of eight Gaussians model (Eq. 8) outperforms Gaussian model (Eq. 9) by **1%** AP. However, the performance goes down when there are more than 16 components. This might be because the objects in the dataset might not have as many as 16 modes when occluded. Besides, the more components we have, the higher the chance of over-fitting. Unless specified, we use eight components for the mixture of Gaussians model.

Mixture of Gaussian *vs*. Multivariate Gaussian: Shown in Table 1 and 2, we compare the degradation cases of our complete model (Eq. 1): Gaussian (Eq. 9), mixture of Gaussians (Eq. 8) and multivariate Gaussian (Eq. 7) on CrowdHuman and MS-COCO. For CrowdHuman, we use ResNet-50 FPN Faster R-CNN as the baseline. For MS-COCO, we use ResNet-50 FPN Mask R-CNN.

On CrowdHuman which has a lot of crowded scenes, our model greatly improves the baseline. Gaussian improves the baseline by **1.2%** AP. A mixture of eight Gaussians improves **2.3%** AP, and multivariate Gaussians improves **2.9%** AP. The complete model improves the performance by **3.9%** AP. The improvements indicate all these assumptions are helpful under heavy occlusion. Gaussian helps training the regression network by learning to decrease the gradients for high variance cases. Multivariate Gaussian helps to learn the correlations between an occluded border and the visible borders. Mixture of Gaussians helps to learn a multimodal model for the occluded cases which have

	no occ.	L1	L2	L3
baseline	73.6	48.3	35.0	23.0
DeepVoting	72.0	53.7	42.6	31.6
DeepVoting+	74.0	58.0	46.9	35.2
Ours	**74.4**	**62.1**	**50.9**	**38.4**

Table 4: Comparison with a state-of-the-art occlusion-aware detector on VehicleOcclusion. The metric is mAP. occ.: occlusion

method	# params	FPS
baseline	91M	11.1
Gaussian	91M	11.1
multivariate Gaussian	91M	11.1
mixture of 8 Gaussian	93M	10.2
multivariate mixture of 8	93M	10.2

Table 5: Model size and FPS comparison with ResNet-50 FPN Mask R-CNN on a single GPU

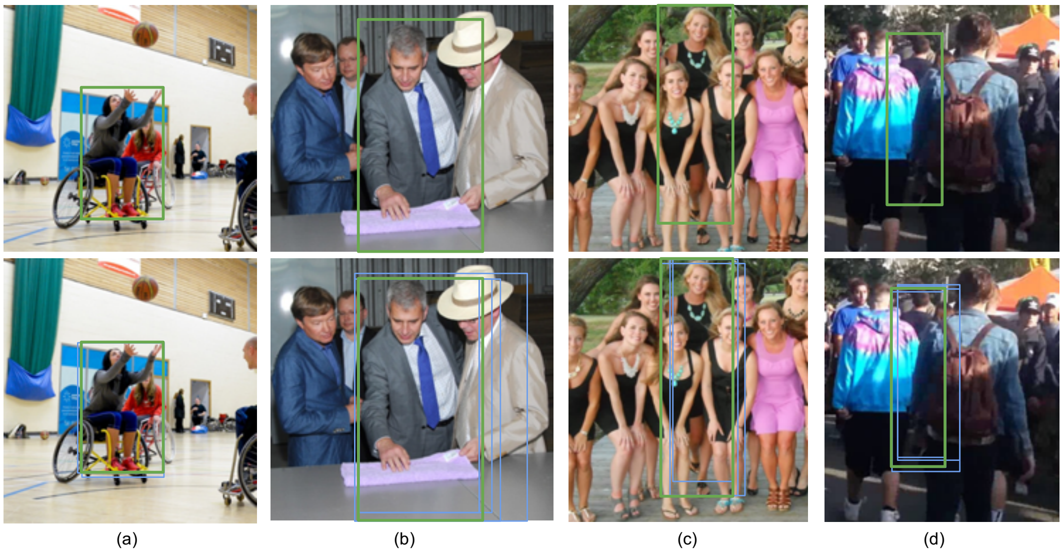


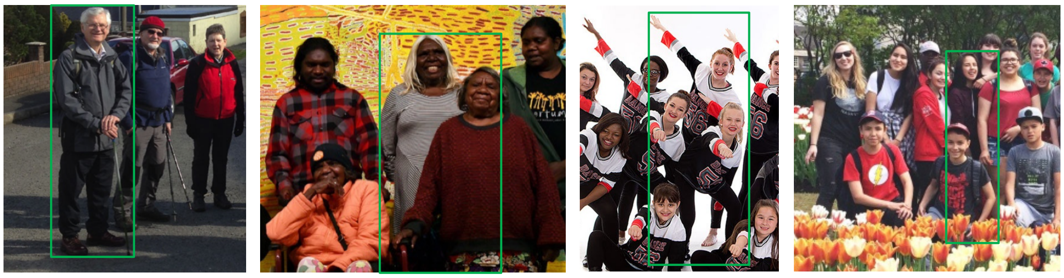
Figure 4: Mixture of Gaussians predictions. First row: baseline Faster R-CNN. Second row: mixture of four Gaussians. Blue boxes are the mixture components. Green boxes are the final predictions. (a) not occluded (b) left arm is occluded (c) both arms are occluded (d) heavily occluded.

multiple modes. Soft-NMS (Bodla et al., 2017) modifies classification scoring, while our approach improves localization. Though it achieves comparable performance (1.7% AP improvement), it can be applied together with our method. With soft-NMS, the AP of mixture of 8 Gaussian, multivariate Gaussian and the complete model further improves **1.7%**, **1.5%** and **1.5%** respectively.

On MS-COCO, the bounding box annotations are visible boxes instead of full boxes used in CrowdHuman. Gaussian still works here which improves the baseline by **0.4%** AP, since there are variances in the dataset caused by inaccurate annotation (*e.g.*, Figure 1 (d)). Gaussian helps to reduce the gradients for these ambiguous cases. A mixture of eight Gaussians improves **0.6%** AP, and multivariate Gaussians improves **0.7%** AP. The complete model improves the performance by **1.2%** AP. The improvements are noticeable, however less significant than on CrowdHuman. On the one hand, there are fewer occluded instances in MS-COCO, multimodality and covariances might not as helpful as in CrowdHuman. On the other hand, predicting full boxes require guessing the invisible parts where multimodality and covariances are more useful.

We further conduct experiments on PASCAL VOC 2007, shown in Table 3. VGG-CNN-M-1024 Faster R-CNN (Simonyan & Zisserman, 2014) is the baseline. Similar to MS-COCO, the bounding box annotations are visible boxes instead of full boxes used in CrowdHuman. We observe that Gaussian improve the mAP (mean Average Precision) by **1.5%**. The complete model improves the mAP by **2.0%**. Multimodality and multivariate Gaussian do not substantially improve the performance. These observations coincide with the observations on MS-COCO.

Comparison with State-of-the-art: Shown in Table 4, we compare multivariate mixture of eight Gaussians model to DeepVoting Zhang et al. (2018c) on VehicleOcclusion. Similar to CrowdHuman, the bounding box annotations are full boxes. The baseline is VGG-16 Faster R-CNN.



	x_1	y_1	x_2	y_2
x_1	0.009	0	0	0
y_1	0	0.009	0	0
x_2	0	0	0.001	0
y_2	0	0	0	0.001

(a)

	x_1	y_1	x_2	y_2
x_1	0.070	-0.006	-0.01	0
y_1	-0.006	0.045	0.093	0
x_2	-0.01	0.093	1.503	0
y_2	0	0	0	0

(b)

	x_1	y_1	x_2	y_2
x_1	1.080	-0.207	2.006	0.146
y_1	-0.207	0.193	-0.474	-0.038
x_2	2.006	-0.474	5.434	0.380
y_2	0.146	-0.038	0.380	0.119

(c)

	x_1	y_1	x_2	y_2
x_1	2.376	0.211	0.949	0.337
y_1	0.211	1.734	-0.205	0.121
x_2	0.949	-0.205	1.139	0.578
y_2	0.337	0.121	0.578	1.324

(d)

Figure 5: Multivariate Gaussian predictions and corresponding covariances on CrowdHuman.

Our multivariate mixture of eight Gaussians model outperforms DeepVoting by a large margin at different occlusion levels. Without occlusion, our model also helps to learn a better detector, coinciding the experiments above. We argue that our model considers multiple modes of an object and the correlations between each border of a bounding box, which helps detection under occlusion.

Model Size and Inference Speed: We measure the inference speed of our models using ResNet-50 FPN Mask R-CNN with a TITAN Xp, CUDA 10.1 and cuDNN 7.5.0 on MS-COCO `val2017`. Shown in Table 5, Gaussian (Eq. 9) and multivariate Gaussian (Eq. 7) neither slow down the inference nor increase the number of parameters, since we can discard the covariance $\boldsymbol{\Sigma}$ at inference time (Section 3.1). The complete model, multivariate mixture of eight Gaussians (Eq. 1), only increases **2M** parameters and sacrifices **0.9 FPS** on GPU. Our models outperform the baselines by large margins (Table 1, 2 and 4), while requires almost no additional computation and memory.

Note that we measure the inference latency on MS-COCO where there are 80 classes, such that the number of parameters for $\boldsymbol{\mu}$ is $1024 \times 80 \times K$ (1024 is the number of output channels of `fc7`, Figure 2). On CrowdHuman where there is only one class (human), the number of parameters for $\boldsymbol{\mu}$ is only $1024 \times K$, which will consume even fewer computation and memory resources.

4.2 Qualitative Results

Figure 4 shows the visualization of our mixture of Gaussian prediction results on CrowdHuman. When the object is not occluded, our model usually only exhibits a single mode. In Figure 4 (a), the predictions of the mixture components for the athlete are almost the same. When the object is occluded, the occluded bounding box border usually exhibits multiple modes. For example, the left arm of the man can have several reasonable poses in Figure 4 (b).

Figure 5 shows the visualization of our multivariate Gaussian prediction results on CrowdHuman. When the object is not occluded, like in Figure 5 (a), most terms in the covariance matrix are usually almost zeros. When a border of the object is occluded, like in Figure 5 (b), the variance term for that border tends to be very high. Sometimes our model learns the covariance between bounding box borders. For example, in Figure 5 (c), x_1 and x_2 has a positive correlation, which suggests if the left border moves right, the right border might also move right. When the object is heavily occluded, most of its variance terms are usually very high, shown in Figure 5 (d).

5 Conclusion

We propose a multivariate mixture of Gaussians model for object detection under occlusion. Quantitatively, it demonstrates consistent improvements over the baselines among MS-COCO, PASCAL VOC 2007, CrowdHuman, and VehicleOcclusion. Qualitatively, our model enjoys explainability as the detection results can be diagnosed via the covariance matrices and the mixture components.

REFERENCES

Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S Davis. Soft-nms – improving object detection with one line of code. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pp. 5562–5570. IEEE, 2017.

Roberto Calandra, Andrew Owens, Dinesh Jayaraman, Justin Lin, Wenzhen Yuan, Jitendra Malik, Edward H Adelson, and Sergey Levine. More than a feeling: Learning to grasp and regrasp using vision and touch. *IEEE Robotics and Automation Letters*, 3(4):3300–3307, 2018.

Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 248–255. IEEE, 2009.

M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.

Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2): 303–338, 2010.

Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1440–1448, 2015.

Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587, 2014.

Ross Girshick, Ilija Radosavovic, Georgia Gkioxari, Piotr Dollár, and Kaiming He. Detectron. `https://github.com/facebookresearch/detectron`, 2018.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pp. 2980–2988. IEEE, 2017.

Yihui He, Chenchen Zhu, Jianren Wang, Marios Savvides, and Xiangyu Zhang. Bounding box regression with uncertainty for accurate object detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Brody Huval, Tao Wang, Sameep Tandon, Jeff Kiske, Will Song, Joel Pazhayampallil, Mykhaylo Andriluka, Pranav Rajpurkar, Toki Migimatsu, Royce Cheng-Yue, et al. An empirical evaluation of deep learning on highway driving. *arXiv preprint arXiv:1504.01716*, 2015.

Kinjal A Joshi and Darshak G Thakore. A survey on moving object detection and tracking in video surveillance system. *International Journal of Soft Computing and Engineering*, 2(3):44–48, 2012.

Ke Li and Jitendra Malik. Amodal instance segmentation. In *European Conference on Computer Vision*, pp. 677–693. Springer, 2016.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, Cham, 2014.

Tsung-Yi Lin, Piotr Dollár, Ross B Girshick, Kaiming He, Bharath Hariharan, and Serge J Belongie. Feature pyramid networks for object detection. In *CVPR*, pp. 3, 2017.

Tsung-Yi Lin, Priyal Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *IEEE transactions on pattern analysis and machine intelligence*, 2018.

Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pp. 21–37. Springer, 2016.

Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788, 2016.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pp. 91–99, 2015.

Shuai Shao, Zijian Zhao, Boxun Li, Tete Xiao, Gang Yu, Xiangyu Zhang, and Jian Sun. Crowdhuman: A benchmark for detecting human in a crowd. *arXiv preprint arXiv:1805.00123*, 2018.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Bharat Singh, Mahyar Najibi, and Larry S Davis. Sniper: Efficient multi-scale training. In *Advances in Neural Information Processing Systems*, pp. 9310–9320, 2018.

Jianyu Wang, Cihang Xie, Zhishuai Zhang, Jun Zhu, Lingxi Xie, and Alan Yuille. Detecting semantic parts on partially occluded objects. *arXiv preprint arXiv:1707.07819*, 2017.

Shanshan Zhang, Jian Yang, and Bernt Schiele. Occluded pedestrian detection through guided attention in cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6995–7003, 2018a.

Shifeng Zhang, Longyin Wen, Xiao Bian, Zhen Lei, and Stan Z Li. Occlusion-aware r-cnn: detecting pedestrians in a crowd. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 637–653, 2018b.

Zhishuai Zhang, Cihang Xie, Jianyu Wang, Lingxi Xie, and Alan L Yuille. Deepvoting: A robust and explainable deep network for semantic part detection under partial occlusion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1372–1380, 2018c.

Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. *arXiv preprint arXiv:1811.11168*, 2018.

Yan Zhu, Yuandong Tian, Dimitris Mexatas, and Piotr Dollár. Semantic amodal segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.