CNN-based energy learning for MPP object detection in satellite images

Jules Mabon, Mathias Ortner, Josiane Zerubia

CNN-BASED ENERGY LEARNING FOR MPP OBJECT DETECTION IN SATELLITE IMAGES

J. Mabon[⋆] *M. Ortner*[†] *J. Zerubia*[⋆]

[⋆] Inria, Université Côte d'Azur, Sophia-Antipolis, France
[†] Airbus Defense and Space, Toulouse, France

ABSTRACT

This article presents a method combining marked point processes and convolutional neural networks in order to detect small objects in optical satellite images. In this setting, objects are scattered densely: the energy based formulation of a point process allows us to factor in priors to account for object interactions. Classical marked point process approaches use contrast measures to account for object location and shape, which prove limited in complex scenes. Instead, we use convolutional neural networks to learn energy terms that are more resilient to object and context visual diversity. Finally we present a procedure to learn the relative weights of prior and likelihood terms. We test our approach on remote sensing images and compare it to contrast based approaches. Code is available at https://github.com/Ayana-Inria/MPP_CNN_RS_object_detection.

***Index Terms*—** small object detection, remote sensing, marked point process, convolutional neural network

1. INTRODUCTION

Small object detection in optical satellite images is often a challenging task; limited spatial resolution makes the objects of interest only a few pixels large, leaving less visual information to use. Moreover objects such as vehicles are often scattered densely, introducing interactions between neighboring objects, and increasing the difficulty in separating instances.

Small object size in images with resolutions around 0.5m/px, makes raster to raster segmentation impractical. This issue furthers the need for vectorisation of the extracted information.

Numerous methods based on Convolutional Neural Networks (CNN) such as Faster R-CNN [1], YOLO [2] or RetinaNet [3] propose to detect objects in "natural" images, where objects are large and interactions between instances are scarce. Also, the size of detection boxes and their limited parametrisation, make these approaches less reliable in remote sensing applications.

On the other hand, Marked Point Process (MPP) based approaches rely on a stochastic geometry model. They jointly solve the detection and selection of objects while vectorizing the extracted information. These models also account for interaction models through priors. Such approaches classically use likelihood terms built upon contrast measures [4, 5, 6] which perform great in images where the objects of interest are clearly contrasted with respect to the background.

However, complex image backgrounds, varying illumination conditions as well as object diversity make the contrast measures less effective. While it is possible to design more elaborate measures, this proves tedious and increases computational cost.

In this article we solve this issue by combining CNN based energy terms with a MPP that models priors and object interactions. Our main contributions are as follows:

- we formulate the detection task as an energy minimisation problem, while introducing priors.
- we build likelihood measures on simple CNN models to replace contrast measures.
- we propose a method to learn the weights for the combination of energy terms on the data.

In section 2 we present the marked point process framework and the energy model that drives it in sections 3 and 4. We then detail in section 5 the training and inference procedures. Finally, in section 6 we evaluate this approach on satellite imagery and compare results with contrast based approaches, along with other methods such as [7].

2. POINT PROCESS FRAMEWORK

We consider the image space as $S \subset \mathbb{R}^2$. A configuration of points Y is a finite non-ordered set of elements of $S \times M$, with M the mark space. A mark can be any random variable from the radius of a circle to a discrete categorization of the object. In our case, an object $y \in Y$ is comprised of coordinates i, j

Thanks to BPI France (LiChiE contract) for funding, and to the OPAL infrastructure from Université Côte d'Azur for providing computational resources and support.

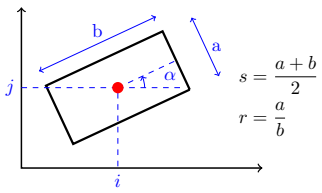


Fig. 1. Rectangle parametrization

in S, and three marks that describe a rectangle : size s, ratio r and angle α (see Fig. 1).

A configuration of points can be modeled as the realization of a non-uniform Marked Point Process (MPP); the density of which is defined by h relative to the uniform point process [8]. The model of selection and interaction of points derives from an energy U, through a non-normalized Gibbs density :

$$h(Y) \propto \exp(-U(Y)) \tag{1}$$

To infer the best fitting configuration of objects $\widehat{Y}$ from an image X, it needs an energy $U_{tot}(X, Y)$, the minimum of which is reached for the optimal configuration $Y = \widehat{Y}$.

3. ENERGY MODEL

Given an image X and a configuration of points Y, the total energy is defined, as the sum for each point $y \in Y$, of the combination of energy terms listed in Table 1.

Energy term	**Notation**	**Eq. #**
position likelihood	$U_p(X, y)$	2
mark likelihood	$U_m(X, y)$	3
size prior	$U_s(y)$	4
overlap prior	$U_o(y, \mathcal{N}_y)$	5
alignment prior	$U_a(y, \mathcal{N}_y)$	6

Table 1.

The likelihood energy terms U_p and U_m measure the conformity of the point y regarding the image X. The prior terms U_s, U_o, U_a measure the coherence of the point configuration itself considering the known properties of the studied objects. The latter depends on the point $y \in Y$ and its neighborhood in Y, $\mathcal{N}_y = \{\tilde{y} \in Y | \tilde{y} \neq y, \|y - \tilde{y}\| < d_{\max}\}$.

In this section we detail each energy term. The combination of these energy terms into a total energy $U_{tot}(X, Y)$ is discussed in section 4.

3.1. CNN based likelihood measurement

To circumvent the limitations of contrast measures as likelihood energy terms, we devise U_p and U_m sampled from energy maps inferred with CNN models.

3.1.1. Position likelihood term

A first Unet [9][1] infers a probability map of object centers in order to extract $U_p(X, y)$, the energy relative to the position of the object.

The localisation of object centers is similar to a keypoint detection task [10]. However, these keypoints are often so close that a heatmap inference approach – *ie.* learning a center probability map dilated by a Gaussian filter (see Fig. 2b) – creates connectivity between objects that makes them harder to separate. Moreover, reducing the variance on the Gaussian around centers, reinforces the unbalance in labels, and contradicts the imprecision of annotations (ground truth object locations are often noisy).

Thus, we learn this center map through a proxy task of learning a vector field. The model is trained to infer for an image X of size (h, w), a vector field $\widehat{V}$ of size $(h, w, 2)$, which unitary vectors point towards the closest object center (Fig. 2c). By computing the divergence of the vector field, object centers become negative divergence locations, and separations between instances are mapped to positive values (Fig. 2d and 2f).

The position energy is then :

$$U_p(X, y) = -\sigma\left(b + a.\text{div}(\widehat{V})_{y_i, y_j}\right) \tag{2}$$

where σ is the sigmoid function and $\text{div}(\widehat{V})_{y_i, y_j}$ is the value of the divergence of $\widehat{V}$ inferred from the image X, at the location of y. Parameters a and b are estimated while training the neural network.

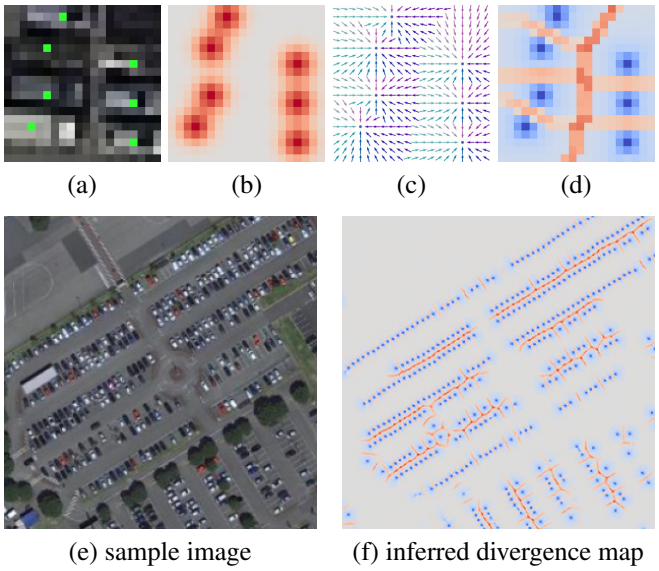
3.1.2. Mark likelihood term

The energy $U_m(X, y)$ associated to the marks of a point y (s, r, α, defining the shape of a rectangle) is computed as follows: for each mark m, we discretize its value range into N_m intervals. From the Unet, for a given image X of size (h, w), we extract a tensor $\widehat{M}^m$ of size (h, w, N_m). We use $\text{Softmax}(\widehat{M}^m_{y_i, y_j})$ as discrete approximation of the likelihood of the values of mark m at the position of y. Thus for each y and mark $m \in \{s, r, \alpha\}$, given a value y_m, and $\text{index}(y_m)$ the index of the interval containing y_m: $\text{Softmax}(\widehat{M}^m_{y_i, y_j})_{\text{index}(y_m)}$ is trained to estimate the probability $P(y_m | X, y_i, y_j)$.

While a regression model would output a single mark value per pixel, our classification-like approach allows to assign a probability to all possible values of a mark at a given pixel, allowing to inform on ambiguous situations with multimodal estimated likelihood densities.

$$U_m(X, y) = \sum_{m \in \{s, r, \alpha\}} -\text{Softmax}(\widehat{M}^m_{y_i, y_j})_{\text{index}(y_m)} \tag{3}$$

[1] as described in [9], but with a limited depth (3 pooling operations instead of 4) and 32 channels in the first convolutional layer.



(a) (b) (c) (d)

(e) sample image (f) inferred divergence map

Fig. 2. Closeup image with centers overlay (a), centers heatmap (b), vector field (c) and divergence (d) (red$>$ 0, blue$<$ 0)

3.2. Priors on configurations

We use various prior terms to regularize point configurations, given the expected properties of the studied objects:

3.2.1. Size prior

Enforces size limits on rectangles. This helps avoiding zero area objects. With a_0 and a_1 the minimum and maximum area hyperparameters:

$$U_s(y) = \max\{a_0 - \text{area}(y), \text{area}(y) - a_1, 0\} \quad (4)$$

3.2.2. Overlap prior

Penalizes the overlapping of objects [4] :

$$U_o(y, \mathcal{N}_y) = \max_{\tilde{y} \in \mathcal{N}_y} \left\{ \frac{\text{area}(\tilde{y} \cap y)}{\min\{\text{area}(\tilde{y}), \text{area}(y)\}} \right\} \quad (5)$$

3.2.3. Alignment prior

Rewards configurations where objects are aligned (*eg.* cars often drive and park aligned), with y_α the angle of y relative to the image horizontal axis (see Fig. 1):

$$U_s(y, \mathcal{N}_y) = \min_{\tilde{y} \in \mathcal{N}_y} \{-|\cos(|y_\alpha - \tilde{y}_\alpha|)|\} \quad (6)$$

4. COMBINING ENERGY TERMS

In order to compute the total energy $U_{tot}(X, Y)$ we combine for each point all energy terms defined previously (Table 1). We formalize this energy combination as a function f parametrised by θ, with $U_{X,y}$ the vector of energies defined in Table 1:

$$U_{tot}(X, Y, \theta) = \sum_{y \in Y} f_\theta(U_{X,y}) \quad (7)$$

4.1. Weighted-sum combination

In classical point process applications, such as [4, 5, 6], the energies are combined as a weighted sum. This weighted sum can be written as follows [2]:

$$f_\theta(U_{X,y}) = \theta_p U_p + \mathbb{1}_{U_p<0}.(\theta_m U_m + \theta_s U_s + \theta_o U_o + \theta_a U_a) \quad (8)$$

The indicator term $\mathbb{1}_{U_p<0}$ forces to omit any prior or mark contribution when the point is not well positioned. Weights $\theta_p, \ldots, \theta_a$ are usually set manually. This approach also requires finding a threshold for every energy term so that valid points have a negative energy contribution, and non-valid ones have positive energy contributions: *ie.* one would have to set a threshold t_p for $U_p(X, y) = \sigma(b + a.\text{div}(\widehat{V})_{y_i, y_j}) - t_p$ such as valid points y have $U_p(X, y) < 0$ (see [6]).

4.2. Logistic combination

We introduce a formulation inspired by logistic models, denoting θ_w the weights vector and θ_b a scalar:

$$f_\theta(U_{X,y}) = 2\sigma\left(\theta_w \cdot U_{X,y} + \theta_b\right) - 1 \quad (9)$$

This formulation allows for the following :

- the sigmoid σ maps valid points to an energy close to -1 and non-valid one close to 1, thus keeping energy per point bounded.
- removing the indicator term from equation 8 makes f_θ continuous and allows to learn θ more consistently (see subsection 5.2).
- introducing a bias term θ_b, allows to learn all sub-energies threshold terms at once, subtracting the need to fit thresholds on individual energies.

5. TRAINING AND INFERENCE

Our model runs as follows:

1. learning underlying CNN models for likelihood terms U_p and U_m on the training set.
2. learning $\widehat{\theta}$ for the energy combination model f_θ, given the sub-energies and the training set (see Section 5.2).

[2] for better readability we shorten $U_p(X, y)$ as U_p, as for $U_m, \ldots, U_a$

3. at inference, for an image X, computing vector field $\widehat{V}$ and map $\widehat{M}$. Since U_p and U_m only depend on y, $\widehat{V}$ and $\widehat{M}$, we write U_{tot} as a function of $\widehat{V}$, $\widehat{M}$, Y and $\widehat{\theta}$. This way we don't have to recompute $\widehat{V}$ and $\widehat{M}$ for every Y we want to compute U_{tot} for.

4. then inference of $\widehat{Y}$ through the simulation of the point process derived from U_{tot} (see subsection 5.3).

5.1. Learning likelihood terms

5.1.1. Position

The position energy model is trained to minimize the following cost function:

$$L_{pos}(X,Y) = \text{MSE}(\widehat{V}, V) + \text{BCE}(M, \sigma(a.\text{div}(\widehat{V}) + b)) \tag{10}$$

where V is the vector field built from the ground truth configuration Y, M a heatmap of object centers (binary center map derived from Y, passed through a Gaussian filter with variance 0.6). MSE and BCE are respectively the Mean Square Error and the Binary Cross Entropy.

5.1.2. Marks

The energy model on marks is trained to minimize the following cost function for every mark m (s, r and α):

$$L_m(X,Y) = \frac{1}{|P|} \sum_{p \in P} \text{CE}(\text{Softmax}(\widehat{M}_p^m), \text{Softmax}(M_p^m)) \tag{11}$$

where P is the set of pixels in X, and CE the Cross Entropy between the estimated distribution $\text{Softmax}(\widehat{M}_p^m)$ and the ground truth $\text{Softmax}(M_p^m)$ for every pixel.

Both models are trained on $2/3$ of the dataset, from which we sample smaller patches ($2/3$ around objects of interest, $1/3$ uniformally).

5.2. Learning $\widehat{\theta}$, the energy combination model

To learn the function f_θ that combines the sub-energies, we proceed to enforce an ordering of configuration energies: given a ground truth configuration Y for an image X, we perturbate Y to get $\widetilde{Y}$ with a perturbation kernel Q (by removing, adding, translating, rotating or scaling points randomly). Thus the point configuration $\widetilde{Y}$ is less fitting to X than Y; this means $U_{tot}(X,Y,\theta) < U_{tot}(X,\widetilde{Y},\theta)$.

The works in [5, 11] set a number of these constraints, then solve for the parameters with linear programming. This can lead to over-constrained problems because of imperfect ground truth or setting too many constraints.

We instead formulate these constraints as a loss to minimize over all examples, allowing to consider more samples without any over-constraint issue:

$$L_\theta(X,Y) = U_{tot}(X,Y,\theta) - U_{tot}(X,\widetilde{Y},\theta) \tag{12}$$

Algorithm 1 shows the learning procedure for the energy model from equation 9; gradient descent is performed on parameters θ_w and θ_b, with Q a perturbation kernel introduced in section 5.3.

Algorithm 1 θ learning procedure

$\theta_w \leftarrow [1,\dots,1]^T, \quad \theta_b \leftarrow 0$
while not converged **do**
  $\widetilde{Y} \sim Q(Y \to \cdot)$
  $\Delta\theta \leftarrow \Delta_\theta(U_{tot}(X,Y,\theta) - U_{tot}(X,\widetilde{Y},\theta))$
  Update θ based on $\Delta\theta$ using Adam optimizer
end while

5.3. Inferring configurations: point process simulation

Once the total energy U_{tot} is fully defined by $\widehat{\theta}$, $\widehat{V}$ and $\widehat{M}$, we look for the configuration that minimizes this energy:

$$\widehat{Y} = \underset{Y \in S \times M}{\text{argmin}}\, U_{tot}(X,Y) \tag{13}$$

We simulate the point process through a Reversible Jump Monte Carlo Markov Chain (RJMCMC) [12]. This method extends the Metropolis Hastings algorithm allowing to explore a state space of varying dimension. The convergence is ensured as long as a uniform birth-death kernel in $S \times M$ is implemented. To speed up convergence [12] uses translation/rotation/scaling kernels. Moreover we add birth-death and translation/rotation/scaling kernels that propose points based on a non-uniform density [6], derived from the potential $U_p(X,\cdot) + U_m(X,\cdot)$. These densities are easy to sample from as they derive from pre-computed maps $\widehat{V}$ and $\widehat{T}$.

The modified configuration $\widehat{Y}$ is obtained by simulation of the MPP of density proportional to $\exp(-U_{tot}(X,\cdot)/T)$. We use simulated annealing for which the temperature T decreases geometrically (see Algorithm 2). Here Q is the perturbation kernel produced by the random choice of one of the above mentioned kernels.

6. EXPERIMENTAL RESULTS

6.1. Data: DOTA gsd50

Our goal is the detection of small objects in images from satellites such as Pléiades[3] or CO3D[4], which have a typical spatial resolution (or ground sampling distance) of 50cm/px. We use the DOTA [13] dataset, containing images and ground truth from various sources and resolutions. We set the spatial

[3]Constellation Pléiades [pleiades.cnes.fr]
[4]Constellation Optique 3D [intelligence-airbusds.com]

Algorithm 2 MPP inference procedure

$\widehat{V} \leftarrow$ Position-CNN(X)
$\widehat{M} \leftarrow$ Marks-CNN(X)
$Y \leftarrow \{\}, \quad T \leftarrow T_0$
while not converged **do**
  $Y' \sim Q(Y \to \cdot)$
  $r \leftarrow \min\left\{1, \frac{Q(Y' \to Y)}{Q(Y \to Y')} e^{\frac{U_{tot}(\widehat{V},\widehat{M},Y,\widehat{\theta}) - U_{tot}(\widehat{V},\widehat{M},Y',\widehat{\theta})}{T}}\right\}$
  With probability r: $Y \leftarrow Y'$
  $T \leftarrow \alpha T$
end while

resolution to 50cm/px by subsampling the images with higher resolutions. We only keep the *small-vehicle* class as we are only interested in ground vehicles detection for this application.

6.2. MPP based on contrast mesures

We also build MPP models using some contrast measures found in literature [6, 14]. The framework remains similar but we replace U_p and U_m with a single contrast energy U_c. In Table 2 we denote μ_y σ_y^2 the empirical mean and variance of pixels inside shape y, μ_s σ_s^2 mean and variance on shape contour s, n_y normals of contour s, and ∇X the image gradient.

First, to assert the viability of such measures for these data, we compute U_c for ground truth rectangles and for randomly scattered rectangles. Suited measures should allow to classify easily between ground truth points and random points. We plot precision-recall curves for several of those contrast measures (see Table 2) on one sample image of DOTA and in synthetic high-contrast data (see Fig 3). We compare with the proposed detection energy given by $U_p + U_m$ and show that it outperforms contrast based measures. This is to be expected as those measures were designed for usecases where objects of interest are well contrasted with respect to their background.

Measure	U_c **equation**	**AP**$_D$	**AP**$_s$
Ours	$U_p(X,y) + U_m(X,y)$	0.97	0.96
T-test[6]	$\frac{\lvert\mu_y - \mu_s\rvert}{\sqrt{\frac{\sigma_y^2}{n_y} + \frac{\sigma_s^2}{n_s}}}$	0.40	0.99
Gradient[14]	$\int n_y(t) \cdot \frac{\nabla X(s(t))}{\sqrt{\lvert\nabla X(s(t))\rvert^2 + \epsilon}} dt$	0.48	0.99

Table 2. Various contrast measures from literature and Average Precision (AP) computed on the DOTA image (AP$_D$) and the synthetic image (AP$_s$) from Fig. 3

6.3. MPP with CNN based energies

Fig. 4 shows results on image samples for our method and various others: we compare the proposed CNN and MPP combination with models based on contrast or image gradient

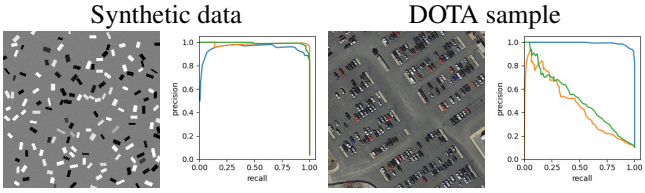


Fig. 3. Precision-recall plots for different likelihood measures in real and synthetic images. Blue: ours, orange: T-test, green: gradient (see Table 2)

measures [6, 14]. We also compare to a full deep learning method BBA-Vectors [7]. Detection metrics are shown in table 3.

We perform detection using the weighted-sum f_θ with manual weights (see 4.1) and denote it MPP+CNN*. The model using the logistic f_θ and learned weights (see 4.2) is denoted MPP+CNN†. For BBA-Vectors [7] detection threshold is set at $\text{argmax}_{s\in[0,1]}$ F1(s) [5].

Our method performs better than contrast based MPP that miss closely packed cars or detect high contrast elements as positives; the over-detection of irrelevant objects leads to low precision and F1 scores. Those methods would require further adaptations to function properly on the current dataset.

The proposed method matches state of the art fully deep learning approach [7] in terms of metrics, while having more visually coherent inferred configurations: the priors included in the model regularize inferred configurations as shown in Fig.4.

Method	F1	Precision	Recall
MPP+T-test[6]	0.03	0.01	0.59
MPP+Gradient[14]	0.09	0.05	0.73
BBA-Vec.[7]	0.77	0.76	0.78
MPP+CNN*	**0.83**	**0.77**	0.90
MPP+CNN†	0.81	0.72	**0.91**

Table 3. Detection metrics for several models (IOU threshold: 0.25).

7. CONCLUSION AND PERSPECTIVES

In this paper we solve the shortcomings of contrast mesures in visually complex images by replacing those with CNN learned measures, while maintaining the MPP formulation thus allowing regularization over the output using priors. Additionally we show how energy weights can be learned over the training data in a more robust manner than [5]. Finally, as the use of priors in our model incrementally improves detection results compared to state of the art, we aim at applying our method to image sequences where object dynamics require stronger priors.

[5] F1(s): F-score for a given detection threshold s

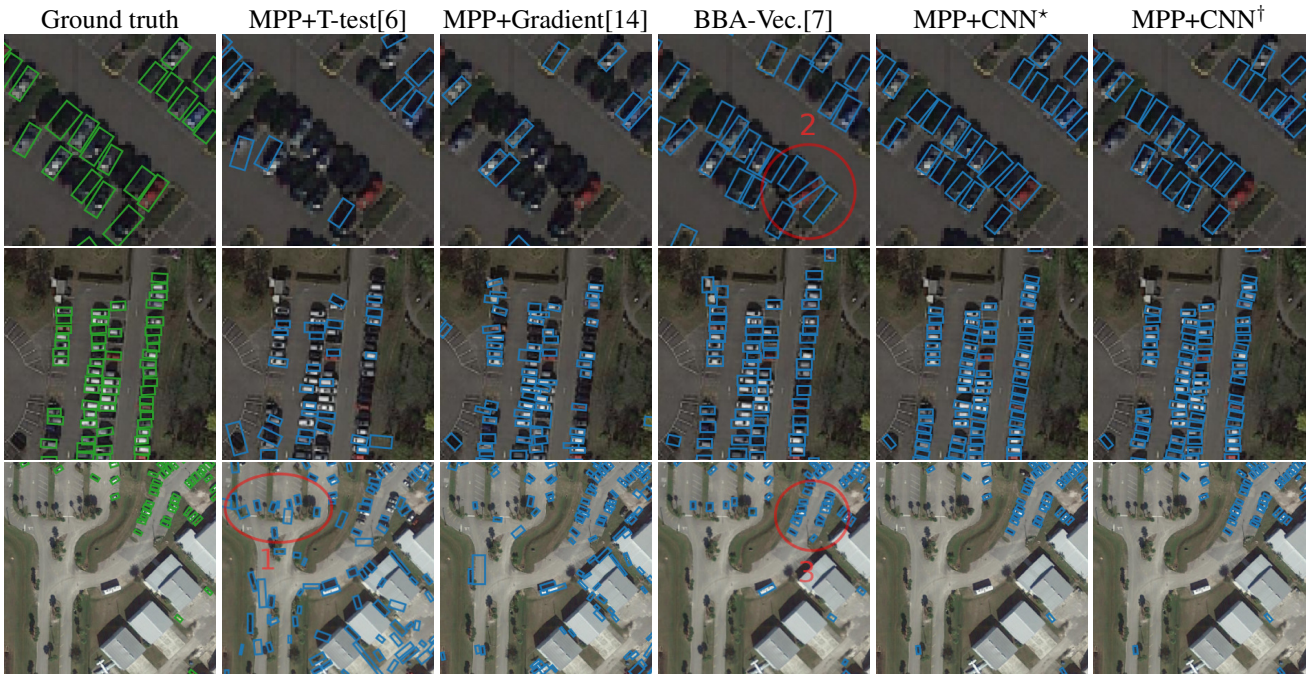


Fig. 4. Inference results for various models. For example 1: false positives, 2-3: misaligned objects

8. REFERENCES

[1] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, 2015.

[2] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *Proc. CVPR*, 2016.

[3] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar, "Focal loss for dense object detection," in *Proc. ICCV*, 2017.

[4] Xavier Descombes, "Multiple objects detection in biological images using a marked point process framework," *Methods*, vol. 115, pp. 2–8, 2017.

[5] Paula Craciun, Mathias Ortner, and Josiane Zerubia, "Joint detection and tracking of moving objects using spatio-temporal marked point processes," in *Proc. IEEE Winter Conf. on Applications of Computer Vision*, 2015.

[6] Caroline Lacoste, Xavier Descombes, and Josiane Zerubia, "Point processes for unsupervised line network extraction in remote sensing," *IEEE TPAMI*, vol. 27, no. 10, pp. 1568–1579, 2005.

[7] Jingru Yi, Pengxiang Wu, Bo Liu, Qiaoying Huang, Hui Qu, and Dimitris Metaxas, "Oriented object detection in aerial images with box boundary-aware vectors," in *Proc. IEEE Winter Conf. on Applications of Computer Vision*, 2021.

[8] Marie-Colette Van Lieshout, *Markov Point Processes and Their Applications*, Imperial College Press, 2000.

[9] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in *Proc. MICCAI*, 2015.

[10] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh, "Realtime multi-person 2D pose estimation using part affinity fields," in *Proc. CVPR*, 2017.

[11] Qian Yu and Gérard Medioni, "Multiple-target tracking by spatiotemporal Monte Carlo Markov chain data association," *IEEE TPAMI*, vol. 31, no. 12, pp. 2196–2210, 2008.

[12] Peter J Green, "Reversible jump Markov chain Monte Carlo computation and Bayesian model determination," *Biometrika*, vol. 82, no. 4, pp. 711–732, 1995.

[13] Gui-Song Xia, Xiang Bai, Jian Ding, Zhen Zhu, Serge Belongie, Jiebo Luo, Mihai Datcu, Marcello Pelillo, and Liangpei Zhang, "DOTA: A large-scale dataset for object detection in aerial images," in *Proc. CVPR*, 2018.

[14] Maria S Kulikova, Ian H Jermyn, Xavier Descombes, Elena Zhizhina, and Josiane Zerubia, "Extraction of arbitrarily-shaped objects using stochastic multiple birth-and-death dynamics and active contours," in *Proc. Computational Imaging VIII*. SPIE, 2010.