Aircraft Detection by Deep Belief Nets


National Laboratory of Pattern Recognition
Institute of Automation, Chinese academy of Sciences
Beijing 100190, China

Email: {xueyun.chen, smxiang, liucl, chpang}@nlpr.ia.ac.cn

Abstract—Aircraft detection is a difficult task in high-resolution remote sensing images, due to the variable sizes, colors, orientations and complex backgrounds. In this paper, an effective aircraft detection method is proposed which exactly locates the object by outputting its geometric center, orientation, position. To reduce the influence of background, multi-images including gradient image and gray thresholding images of the object were input to a Deep Belief Net (DBN), which was pre-trained first to learn features and later fine-tuned by back-propagation to yield a robust detector. Experimental results show that DBNs can detect the tiny blurred aircrafts correctly in many difficult airport images, DBNs outperform the traditional Feature+Classifier methods in robustness and accuracy, and the multi-images help improve the detection precision of DBN than using only single-image.

Keywords—Remote Sensing; Object detection; Deep Belief Nets;

I. INTRODUCTION

Small object detection is very difficult in the large high-resolution remote sensing images for their complex environments. Although many works have been done [1-10], it remains unsolved to find an efficient location method and robust classifier for aircraft detection in complex environments.

In the literature, Cai et al. [4] showed the difficulty of segmenting aircraft from its background. They used anisotropic heat diffusion model to remove the aircraft shadow, but they cannot combine the blue aircraft-body with the white wings. Global thresholding has proven to be more efficient for aircraft detection [1], [3]. Because most aircrafts are white and brighter than their local backgrounds, its difficulty exists in determining the most suitable global threshold. Fig. 1 shows that the global thresholding method outperform methods based on gradient or edges. But no method (such as Otsu) can ensure the best threshold to segment aircraft properly from its background, especially for satellite images photographed in variable airports and various weathers. Local threshold methods such as the Maximally Stable Extremal Region [22] can supply multiple local thresholds. But it is still difficult to find the best local threshold, as a result, exists the danger that a large aircraft may miss its parts by different thresholding in adjacent regions.

Many features have been used for object detection from satellite images. Hsieh [1] used aircraft contour, Zernike moments, wavelet and SVM classifier to detect aircraft from satellite images. Yildiz and Polat [2] used Gabor+SVM. Liu et al. [3] proposed a coarse-to-fine shape modeling method based on edge computing (Sobel). Sun et al. [5] used the key-points and spatial sparse coding bag-of-words model to detect aircraft. Li et al. [6] used visual saliency computation and symmetry detection. Tien et al. [8] used cross-ratios to model curve data of aircraft contour. Xu and Duan [9] used artificial bee colony algorithm with edge potential function to seek aircraft targets. However, invariant moment, saliency and symmetry features, geometric contour, edge, shape and curve data are not suitable for tiny blurred aircraft with a small and unclear segmentation shape full of noise. It is well known that geometric shapes and edges are not stable under background and illumination changes (consider the examples of face recognition). SIFT, LBP [21], HOG [14], Gabor [20] are the most popular features used in object detection. SIFT is very similar to HOG, Histogram of oriented gradients(HOG) are rather stable because the gradient norm is very stable, but gradient orientation is not stable, so the orientation must be weighted by gradient norm, and the orientation bins cannot be too fine. Local Binary Patterns are the binary patterns of the pixels on local circle, thresholding by the center gray. LBP is a good texture descriptor, but its patterns are too fine, not stable to noise-disturbance and illumination change. So Multi-scales HOG and LBP are computed on overlapped blocks to enhance their stability. Gabor is actually a multi-scales and multi-directions gradient descriptor, which is used widely in saliency computation and object recognition. Grabner et al. [13] used boosting method based on Haar wavelets, HOG and LBP to detect vehicle. They segmented the image into streets, buildings, trees, etc, discarding vehicle detections that are not

Fig. 1. Locating aircrafts on the thresholding image is more easy than on the gradient images (such as canny, Sobel) of Los Angeles airport.
present on the streets. Kemhavi and Davis [12] used multi-scales HOG features computed on color maps to detect vehicle in the San Francisco images from google earth, showed HOG outperform SIFT. They got a detection rate about 70% on the challenging complex city environments.

In the case when only a small training set is available, using fixed features and traditional classifier is a reasonable way. But if you have thousands of or more samples for each class (such as aircraft detection), learning the intrinsic features from the training samples is more advisable way. Now, Deep Learning is fashionably used for learning the robust features from the raw data automatically. Deep Belief Network (DBNs) [15] are deep neural networks. After pre-training as the Restricted Boltzmann Machine (RBM)[16] layer by layer, DBNs are later fined tuned by back-propagation algorithm to become a classifier. Sarikaya and Hinton [19] used DBNs in natural language call routing, got the accuracy equal to the best of SVM, Boosting and Maximum Entropym even though it used an impoverished representation of the input. Mohamed et al. [18] used DBNs in automatic phone speech recognition.

We avoid seeking the best threshold by using multiple global thresholds. To locate the aircraft exactly, the orientation and length of the main-axis of the aircraft must be computed exactly. We propose a location method based on maximal projection height, which has a good anti-noise capability. Besides the orientation, it can also give the exact location and length of the main-axis (Fig. 3). To reduce the influence of background, we use multi-images, including a gray image and two local thresholding images as input data. We trained 6-layer DBNs as [17] to detect aircrafts on the satellite images from google earth. The experiments showed DBNs outperform the traditional method based on HOG, Wavelet, and Gabor.

II. DEEP BELIEF NETS

Deep Belief Nets (DBNs) are consisted by a visible input layer, several hidden layers, output layer. The visible layer input the image data, whose gray range has normalized into [0,1], the hidden layers are invisible, their state are binary values, activated by the sigmoid kernel function. The Restricted Boltzmann Machine (RBM) is the basic block of Deep Belief Networks (DBNs), it is trained by a learning algorithm called Contrastive Divergence (CD) [15-16], which uses the Gibbs sampling and the reconstruction error to train the weights of RBM. The energy function of RBM is defined by [17]:

\[ E(v, h) = \sum_{ij} v_i h_j W_{ij} - \sum_{i \in \text{pixels}} v_i c_i - \sum_{j \in \text{hidden layer}} h_j b_j \]  

where \( v_i \) is the pixel of the visible input layer, \( h_j \) is the node of the hidden layer, whose value must be 0 or 1, \( b_j \) and \( c_i \) are their biases, \( W_{ij} \) is the weights of RBM, its update formula is given by:

\[
\begin{align*}
\Delta W_{ij} &= -\varepsilon \frac{\partial E}{\partial W_{ij}} = \varepsilon (\langle v_i h_j \rangle_v - \langle v_i h_j \rangle_{\text{recon}}) \\
\Delta b_j &= -\varepsilon \frac{\partial E}{\partial b_j} = \varepsilon (h_j|_v - h_j|_{\text{recon}}) \\
\Delta c_i &= -\varepsilon \frac{\partial E}{\partial c_i} = \varepsilon (v_i|_v - v_i|_{\text{recon}})
\end{align*}
\]

\( \varepsilon \) is the LearnRate, \( \langle \rangle \) is the inner product, \( *|_v \) means * is get from visible input data, \( *|_{\text{recon}} \) denote the reconstruction value of *, \( *|_v \) are shown as the following:

\[
\begin{align*}
\{ v_i|_v &= v_i \\
\ h_j|_v &= \text{Pro}(h_j = 1) = \text{sign}(b_j + \sum_i v_i W_{ij}) \}
\end{align*}
\]

\( \text{Pro}(*) \) is the probability of *, \( \text{sign} \) is the standard sigmoid function. Because the states of the hidden layer are invisible binary value, we perform Gibbs sampling to estimate its states. We denote \( \text{rand\_value} = 1.0 \times \text{rand}\)\(]^{\text{RAND\_MAX}}\) \( \text{rand\_value} \) is a random value in [0,1]. \( \text{RAND\_MAX} \) is a constant of C language. We have:

\[
\text{Sample}(h_j) = \begin{cases} 1, & \text{if } \text{Pro}(h_j = 1) > \text{rand\_value} \\ 0, & \text{otherwise} \end{cases}
\]

\( \text{Sample}(\ast) \) means the Gibbs sample of *. Now we reconstruct the visible layer and the hidden layer:

\[
\begin{align*}
\{ v_i|_{\text{recon}} &= \text{sign}(c_i + \sum_j W_{ij} \text{Sample}(h_j)) \\
\ h_j|_{\text{recon}} &= \text{sign}(b_j + \sum_i v_i |_{\text{recon}} W_{ij}) \}
\end{align*}
\]

The weights update formulas can be rewritten as:

\[
\begin{align*}
\Delta W_{ij} &= \varepsilon (v_i \text{sign}(b_j + \sum_i v_i W_{ij}) - v_i |_{\text{recon}} \text{sign}(b_j + \sum_i v_i |_{\text{recon}} W_{ij})) \\
\Delta b_j &= \varepsilon (\text{sign}(b_j + \sum_i v_i W_{ij}) - \text{sign}(b_j + \sum_i v_i |_{\text{recon}} W_{ij})) \\
\Delta c_i &= \varepsilon (v_i - \text{sign}(c_i + \sum_j W_{ij} \text{Sample}(h_j)))
\end{align*}
\]

The RBM must be trained properly when the reconstruction error diminishes to a small value. All weights of DBNs must be pre-trained layer-by-layer as the RBM training. After pre-training, the weights of DBNs are fine-tuned by the standard back-propagation algorithm and the steepest descent algorithm as the Multi-Layer Perceptron (MLP).

III. OUR APPROACH

Aircraft detection is a difficult problem. As shown in Fig 1, method based on gradient images is not suitable for complex environments. We segment aircrafts on multi-thresholding images, search objects by multi-scale sliding windows, compute the main-axis of objects. Then we send all location windows to DBNs for aircraft detection.

A. Image Thresholding

Suppose all aircrafts are white and brighter than their local backgrounds, for some colored aircrafts, we can pre-transform the image to highlight the colored aircrafts. We use multi-thresholds to separate the objects from their backgrounds as shown in Figure 2. For any aircraft, if it is brighter than its background, it always has a clear or blurred segmenting appearance in one of the thresholding images. Figure 2 shows even the aircrafts under strong sunshine are segmented clearly in one of the thresholding images.

B. Orientation Computing

Computing the orientation, position and length of the main-axis of the object is very important for exact location. We proposed a new method based on the maximal projection height, its process is shown in Algorithm 1 and Fig. 3.
C. Object Locating

In the multi-thresholding images, the sizes of the aircrafts in the airports vary in a wide range, we generate two scale sliding windows: 30x30,16x16, sliding them along horizontal and vertical directions by step 14 and 8 pixel respectively to cover the whole images. Then we use a strategy to locate the objects exactly as shown in Algorithm 2 and Figure 4:

Algorithm 2 Object Locating

Input: an initial sliding window \(W_p\) at position \(p = (x_0, y_0)\).

Output: The exact location window.
1: Compute the geometric center \(p_1 = (x_1, y_1)\) of \(W_p\), move the \(W_p\) to \((x_1, y_1)\), denote it as \(W_{p_1}\).
2: Enlarge the size of \(W_{p_1}\) twice, compute the new geometric center \(p_2 = (x_2, y_2)\) of the enlarged window.
3: Move \(W_{p_1}\) to \((x_2, y_2)\), denote it as \(W_{p_2}\).
4: Compute the main-axis of \(W_{p_2}\), rotate and move \(W_{p_2}\) to its main-axis orientation and position, change the window scale to the main-axis length.

At last, some repetitive windows are filtered by a small distance limit (5 pixel). Our aircraft database includes 25 images, 1300x950 size; 681 aircrafts at all. Our object locating method generated 25527 samples, 2890 positive samples, 1021.08 samples per image, 424 samples per aircraft in average, locating 97.4% aircrafts correctly. To achieve the same locating precision, the baseline sliding window method needs about \(2000^3\times950 = 2000^3\times950 = 25507\) samples per image. The search efficiency of our method is more than 20 times than the baseline sliding window method.

D. Training DBNs for aircraft detection

After object locating, all location windows were normalized to 41x33 size. Then we sent them to DBNs for feature extracting and classification. Fig. 5 shows the structure of our DBNs. The input of the visible layer contains three 41x33 images preprocessed from the location window: the first is the gradient image, which is computed by the maximal norm of the gradients in the RGB channel as [14]; the second is the thresholding image at the threshold where 50% pixels in the window have gray higher than it; the third is the image at the threshold where only 25% pixels have gray higher than it. We used a Gaussian smoother whose parameter \(\sigma = 2\) to erase the noise, the gray range of the images have been normalized to [0,1]. The dimension of the input layer is 41x33x3 = 4059. Our DBNs has one input layer, four hidden layers, one output layer. The number of the total weights is 4059 \(\times 800 + 800 \times 800 + 800 \times 400 + 400 \times 200 + 200 \times 2 = 4287600\). We pre-trained the weights of DBNs layer by layer, from the first hidden layer to the output layer, we set learrate=0.01, momentum=0.5, weightdecay=0.0001, minibatch=20. The right part of Fig. 5 shows the partial weights images. After pre-training, we used the label information, the back propagation and steepest descent algorithms to fine tune the weights, until the average training error dropped to a small value.

IV. EXPERIMENT

Our database contains 25 images, all collected from 20 airports in google-earth. The airports include many famous in-

![Image](image1.png)

Fig. 2. Constant multi-thresholds are suitable to different airports.

Algorithm 1 Main-axis-thresholds Computing

Input: a sliding window \(W_p\) at position \(p = (x_0, y_0)\), \(w_s=\)window scale, \(w=0.5 \times w_s, h=1.25 \times w_s\).

Output: The main-axis orientation, position, length.

\[
\begin{align*}
&\text{for } i = 0, 1, \ldots, 39 \text{ do} \\
&\quad \text{Rotate } W_p \text{ by angle} = i \times 4.5^\circ, \text{ denote the rotated window as } W_{p_i}. \\
&\quad \text{Compute } C_{p_i} = \text{the gray projection curve of } W_{p_i} \text{ to horizontal axis}, \ M_{p_i} = \text{maximal value of } C_{p_i}, \ X_{p_i} = \text{x-position of } M_{p_i}. \\
&\quad Y_{p_i} = \text{y-position of the geometric center of } W_{p_i}. \\
&\text{end for} \\
&\text{Compute } j = \text{arg max}_i \{ M_{p_i} : i = 0, \ldots, 39 \}. \\
&\text{Segment } R_{p_j} = \text{the rectangle region of } W_{p_j}, \text{ which is centered at } (X_{p_j}, Y_{p_j}), \text{ width} = h, \text{ height} = h. \\
&\text{Compute } C_{R_{p_j}} = \text{the gray projection curve of } R_{p_j} \text{ to vertical axis}. \\
&\text{The main-axis orientation} = j \times 4.5^\circ, \text{ x-position} = X_{p_j}, \text{ y-position} = Y_{p_j}, \text{ length} = \text{width of } C_{R_{p_j}}. \\
\end{align*}
\]

Our method, other three methods based on the minimal one or two order geometric center moment and the minimal including rectangle area [11] were tested by about 3000 positive samples of the aircraft database. The orientation precisions were 95.8%, 94.2%, 94.0%, 93.26% respectively, the average angle errors were 17.90, 21.90, 24.70, 30.40. Our method outperformed the other three methods obviously.

![Image](image2.png)

Fig. 3. Only the window rotated to the main-axis orientation has the steepest peak in its gray projection curve.
Fig. 4. The four steps of our multi-scales object locating process on clear and noisy thresholding images, it has good anti-noise capability.

Fig. 5. Structure of DBNs and the weight images we got after pre-training.

Fig. 6. Detection Rate Curves of Four methods.

Fig. 7. and Fig. 8 show the detection results on partial test images, owing to the multi-scales object localization method and our DBN detector, most aircrafts are detected repetitively for 2-4 times, including some tiny and very blurred aircrafts.

TABLE I. FALSE ALARM RATES OF DBN (800,800,400,200,2)

<table>
<thead>
<tr>
<th>Input Data</th>
<th>85%</th>
<th>80%</th>
<th>75%</th>
<th>70%</th>
<th>65%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gray</td>
<td>57.92%</td>
<td>19.52%</td>
<td>13.15%</td>
<td>10.217%</td>
<td>7.843%</td>
</tr>
<tr>
<td>Gradient</td>
<td>55.26%</td>
<td>18.23%</td>
<td>12.73%</td>
<td>9.745%</td>
<td>7.382%</td>
</tr>
<tr>
<td>Gradient, 50%</td>
<td>49.64%</td>
<td>17.52%</td>
<td>11.87%</td>
<td>8.515%</td>
<td>6.180%</td>
</tr>
<tr>
<td>Gradient, 50%, 25%</td>
<td>37.92%</td>
<td>15.64%</td>
<td>10.19%</td>
<td>7.228%</td>
<td>5.248%</td>
</tr>
</tbody>
</table>

Table 1 shows that the gradient image has lower FAR than gray image. Input multi-images including gradient, 50% and 25% thresholding images get the lowest FAR. This is due to the fact that using multi-thresholding images can reduce the influence of the background. With the three images input, the total number of weights is 4287600, we trained the DBNs for more than two weeks.

TABLE II. FALSE ALARM RATES OF SIX METHODS

<table>
<thead>
<tr>
<th>Method</th>
<th>Detection Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>800,800,400,200,2</td>
<td>37.92% 15.64% 10.19% 7.228% 5.248%</td>
</tr>
<tr>
<td>500,500,400,200,2</td>
<td>47.82% 16.73% 11.98% 8.119% 6.119%</td>
</tr>
<tr>
<td>250,250,200,100,2</td>
<td>54.45% 17.72% 11.58% 9.208% 7.228%</td>
</tr>
<tr>
<td>HOG+SVM [1]</td>
<td>61.98% 24.15% 17.129% 11.08% 8.119%</td>
</tr>
<tr>
<td>Wavelet+SVM [1]</td>
<td>67.35% 26.72% 21.58% 12.50% 9.432%</td>
</tr>
<tr>
<td>Gabor+SVM [2]</td>
<td>64.80% 25.31% 19.20% 11.91% 8.378%</td>
</tr>
</tbody>
</table>

Table 2 shows FAR drops with the rising of the dimensions of DBNs. Here 800,800,400,200,2 means the DBN has 800, 800, 400 and 200 nodes in the first, second, third and fourth hidden layers respectively, two nodes in output layer. So is named for other DBNs. All DBNs outperform (HOG, Wavelet, Gabor)+SVM obviously. HOG feature is better than Gabor, and Gabor is better than Wavelet. The input is the multi-images of gradient, 50% and 25% thresholding images. We computed HOG features as [14], the Gaussian smoother parameter $\sigma = 2$, the derivative mask is [-1,0,1], spacial orientation bins is 9, using multi-scale blocks, dividing the detection window into 1x1+2x2+3x3+4x4=30 blocks. We computed Gabor feature as [20] in the 30 blocks , using 8 directions and 5 scales. We computed wavelet as [1], using the Daubechees4 basis, 3 scales. the SVM used a nonlinear rbf kernel. All other parameters were optimized.
V. Conclusion

Aircraft detection is a difficult problem. We proposed an object location method based on multi-thresholding method, which is suitable for white aircrafts but can be expanded to any colored aircrafts by pre-transforming the colored objects into the highlighted objects. Our method has a high location precision, with efficiency more than 20 times than the baseline sliding window approach. Deep Believe Net we used is a deep neural networks based on Restricted Boltzmann Machine. To train the DBN, we used multi-images including gradient, 50% and 25% thresholding images as the input of DBN, after pre-training, we fine tuned the DBN to be a robust classifier. Experiments showed the accuracy of the DBNs rose with its dimension increasing, DBNs outperformed the traditional Feature+Classifier methods with ease. The multi-images input excelled the performances of the single image input obviously.

ACKNOWLEDGMENT

This work was supported in part by the National Basic Research Program of China (973 Program) Grant 2012CB316300 and the Strategic Priority Research Program of the Chinese Academy of Sciences (Grant XDA06030300).

REFERENCES