SIXray: A Large-scale Security Inspection X-ray Benchmark for Prohibited Item Discovery in Overlapping Images

Caijing Miao†, Lingxi Xie‡, Fang Wan†, Chi Su‡, Hongye Liu‡, Jianbin Jiao†, Qixiang Ye†§∗
†University of Chinese Academy of Sciences  ‡The Johns Hopkins University  ♯Kingsoft Cloud
♮Noah’s Ark Lab, Huawei Inc.  §Peng Cheng Laboratory
{miaocaijing16, wanfang13}@mails.ucas.ac.cn, 198808xc@gmail.com
{suchi, liuhongye}@kingsoft.com, {jiaojb, qxye}@ucas.ac.cn

Abstract

In this paper, we present a large-scale dataset and establish a baseline for prohibited item discovery in Security Inspection X-ray images. Our dataset, named SIXray, consists of 1,059,231 X-ray images, in which 6 classes of 8,929 prohibited items are manually annotated. It raises a brand new challenge of overlapping image data, meanwhile shares the same properties with existing datasets, including complex yet meaningless contexts and class imbalance. We propose an approach named class-balanced hierarchical refinement (CHR) to deal with these difficulties. CHR assumes that each input image is sampled from a mixture distribution, and that deep networks require an iterative process to infer image contents accurately. To accelerate, we insert reversed connections to different network backbones, delivering high-level visual cues to assist mid-level features. In addition, a class-balanced loss function is designed to maximally alleviate the noise introduced by easy negative samples. We evaluate CHR on SIXray with different ratios of positive/negative samples. Compared to the baselines, CHR enjoys a better ability of discriminating objects especially using mid-level features, which offers the possibility of using a weakly-supervised approach towards accurate object localization. In particular, the advantage of CHR is more significant in the scenarios with fewer positive training samples, which demonstrates its potential application in real-world security inspection.†

1. Introduction

Security inspection has been playing a critical role in protecting public space from safety threatening such as terrorism. With the growth of population in large cities and crowd density in public transportation hubs, it becomes more and more important to fast, automatically and accurately recognize prohibited items in X-ray scanned images. Recent years, the rapid development of deep learning [19] in particular convolutional neural networks has brought an evolution to image processing and visual understanding, including discovering and recognizing objects in X-ray images [23] [24] [27]. Different from natural images and other X-ray scans [36], security inspection often deals with a baggage or suitcase where objects are randomly stacked and heavily overlapped with each other. Therefore, in the scanned images, the objects of interest may be mixed with arbitrary and meaningless clutters and thus can be ignored even by human inspectors, Fig. 1.

To provide a public benchmark for research in this field, in this paper, we present a dataset named Security Inspection X-ray (SIXray), which is 100 times larger than the existing largest image collection for prohibited item discovery, i.e., the baggage group in the GDXray dataset [25]. SIXray contains more than one million X-ray images in which only less than 1% images have positive labels (i.e., prohibited items are annotated). It mimics a similar testing environment to the real-world scenarios where inspectors often aim
at recognizing prohibited items appearing in a very low frequency (e.g., 1 in 1,000). Unlike GDxray which only contains grayscale images in simple backgrounds, our dataset is much more challenging. Although a color-X-ray scanner assigns various colors to different materials, objects in the containers often suffer a considerable variety in scale, viewpoint, and style, yet a prohibited item may be mixed and overlapped with arbitrary numbers and types of safe items, as shown in Fig. 1.

We formulate this problem into an optimization task which, provided a dataset \( D = \{(x_n, y^*_n)\}_{n=1}^N \), aims at minimizing the expected loss function between ground-truth and prediction \( y^*_n = f(x_n; \theta)^2 \). Here \( x_n \) denotes image data and \( y^*_n \) is a \( C \)-dimensional vector with each index indicating whether a specific class is present in \( x_n \). Based on this framework, we point out a clear difference between natural images and X-ray images. A natural image \( x_n \) can always be divided into regions and each of them contains only one class \( c_n \), thus can be sampled from a distribution \( P(x | c_n) \). However, an X-ray image is often composed of a set of overlapping images which, provided a multi-class label \( y^*_n \) (\( C \) dimensions), can be formulated using a mixture distribution \( x_n = \sum_c y^*_n c \cdot x_{n,c} \) where \( x_{n,c} \) is sampled from a hidden distribution \( P(x | c) \), as shown in Fig. 3.

We present an approach in the context of deep neural networks to deal with this complex scenario. The key idea is to combine two sources of information, namely, using mid-level features \( x_n \) (most often sampled from a mixture distribution) to determine high-level semantics \( y_n \), and reversely filtering irrelevant information out of \( x_n \) by referring to the information contained in \( y_n \). To this end, we formulate the high-level supervision signals into reversed network connections. To alleviate data imbalance, we introduce a loss-balancing term based on this hierarchy. This leads to the complete pipeline named class-balanced hierarchical refinement (CHR). With \( y_n \) being unobserved, an iterative process is required in optimization, which is computationally expensive in practice. To accelerate, we switch off iteration so that more training data are processed in a unit time period. In testing, CHR fuses visual information from different stages towards higher recognition accuracy, yet remains efficient in computation.

We evaluate CHR on the SIXray with different ratios of positive/negative samples\(^2\). CHR reports significantly higher classification performance over various baselines, i.e., different network backbones, demonstrating the effectiveness of using high-level cues to assist mid-level features. In addition, we verify the necessity of adding the class-balanced loss term as we observe more significant improvement on less balanced training data. Last but not least, we provide annotations of prohibited items at the bounding box level in the testing set, and apply the class activation mapping (CAM) algorithm \([39]\) as a baseline for weakly-supervised object localization.

The major contributions of this work are two-fold. (1) We provide a benchmark for future research in this challenging vision task. (2) We present an approach named CHR, which integrates multi-level visual cues and achieves class balance in the hierarchical structure.

2 Related Work

- X-ray Images and Benchmarks

X-ray images are captured by irradiating the objects with X-ray and rendering them with pseudo colors according to their spectral absorption rates. Therefore, in X-ray images, objects made of the same material are assigned with very similar colors, e.g., metals are often shown in blue while impenetrable objects are often shown in red. Besides, the most significant difference between X-ray and natural images lies in object overlapping, because X-ray is often applied in the scenarios that some objects may heavily occlude others, e.g., in a baggage, personal items are often stacked randomly. This property brings a new challenge to computer vision algorithms, while the traditional difficulties persist, e.g., scale and viewpoint variance, intra-class variance and inter-class similarity, etc., as widely observed in other object localization benchmarks like PascalVOC \([9]\) and MS-COCO \([21]\).

Researchers designed much work to deal with these difficulties and also approach the promising commercial value after them \([1]\)\(^{10}\) \([26]\)\(^{30}\) \([34]\). But unfortunately, very few X-ray datasets have been published for research purposes. A recently released benchmark, GDxray \([25]\), contains three major categories of prohibited items including gun, shuriken and razor blade. However, images in GDxray were provided with few background clutters as well as overlaps, thus, it becomes considerably easy to recognizing these images and/or detecting the objects within. In addition, the relatively small number of negative samples (images not containing prohibited items) ease the algorithm in both training and testing stages. ChestXray\(^8\) \([36]\) is a large-scale chest X-ray corpus for medical imaging analy-
sis. Different from our scenario, objects in these images are rarely overlapping with each other.

- **X-ray Images and Benchmarks**

  The research field of object recognition has been dominated by deep learning approaches. With the availability of large-scale datasets [18] and powerful computational resources, researchers are able to design and optimize very deep neural networks [18] [31] [16] [4] [13] [14] to learn visual patterns in a hierarchical manner. In the scenario that each image may contain more than one objects, there are typically two types of localization methods. The first one worked on the image level which produces a score for each class indicating its presence or absence [39]. The second one instead worked on the object level, and produced a bounding box as well as a class label for each object individually [12] [11] [29] [22] [28]. The former type often encounters the issues of multi-object classification and training data imbalance [36], for which binary cross entropy (BCE) loss [5] as well as class-balancing techniques [36] [15] were explored. The second type, on the other hand, was often based on a pipeline that first extracts a number of proposals in the image [12] [11] [29], and then determines the class of each proposal.

  This paper studies image-level recognition, as per-object annotation is missing for training data, while our approach has the ability of object-level localization. This is related to the research in weakly-supervised object localization [3] [6] [33], or a series of work in localizing objects using top-down class activation [8] [7] [40]. There were also efforts about formulating the object localization in multiple instance learning frameworks where convolutional filters behave as detectors which activate regions of interest on the feature maps [3] [37] [33] [35].

  In the context of object recognition in X-ray images, researchers realized that these images often contain fewer texture information, yet shape information stands out to be more discriminative. Therefore, in the era of bag-of-visual-word models [34] [2], the topic of designing effective and efficient handcrafted features is explored in depth [30] [26]. As deep learning becomes a standard tool of optimizing complex functions, researchers started to apply it to either extracting compact visual features for X-ray image representation [1] or fine-tuning a pre-trained model on X-ray images so that knowledge learned from natural images can be borrowed. This paper mainly focuses on the second approach.

3. **The SIXray Benchmark**

3.1. **Data Acquisition**

We collected a dataset named Security Inspection X-ray (SIXray), which contains a total of 1,059,231 X-ray images, and is more than 100 times larger than the only existing pub-
3.2. Dataset Properties

The SIXray dataset has several properties which bring difficulties to visual recognition. First, these images were mostly obtained from X-ray scans on personal luggage, e.g., bags or suitcases, in which objects are often randomly stacked. When these items passed an X-ray scan, the penetration property makes it possible to see even the occluded items in the image. This leads to the most important property of this dataset, which we call it overlapping. Second, prohibited items can appear in many different scales, viewpoints, styles and even subtypes, all of which cause considerable intra-class variation and increase the difficulty of recognition. Third, the images can be heavily cluttered yet it is almost impossible to assign all objects especially those non-prohibited ones with a clear class label. Thus, there is noise coming from an open set of objects, which makes it difficult to expect what appears in the background regions. Fourth and last, as mentioned above, the positive images (with at least one prohibited item) only occupy a small fraction of this dataset. Without a special treatment, it is easy for the training stage to bias towards the negative class, as simply guessing a negative label yields higher difficulty. Similarly, the very last layers (raw image pixels) are often largely impacted by small noise, in both cases.

Our goal is to learn a discriminative function \( y_n = f(x_n; \theta) \) to predict the image label. Since the object of interest may appear in various scales. In order to recognize and further detect it, a popular choice \([17]\) \([20]\) is to combine multi-stage visual information. Here we simply consider feature vectors extracted from \( D \) different layers, the \( l \)-th of which is denoted as \( x_n^{(l)} \). A regular solution to train a classifier beyond each layer, \( y_n^{(l)} = h^{(l)}(x_n^{(l)}; \xi^{(l)}) \), using the ground-truth signal \( y_n^{*} \) as supervision. In the testing stage, we fuse all \( y_n^{(l)} \) as the final output, i.e., \( y_n = \sum_{l=1}^{L} y_n^{(l)} \).

However, we note a significant weakness of this model, which comes from the penetration assumption, i.e., Eqn (1), applied to mid-level features\(^3\). This is to say, each \( x_n^{(l)} \) is the composition of sub-images sampled from different classes, including those items of no interest, and thus \( h^{(l)}(x_n^{(l)}; \xi^{(l)}) \) may be distracted. A reasonable idea is to refine \( x_n^{(l)} \) to get rid of these irrelevant information. This is achieved by a function \( g^{(l)}(x_n^{(l)}, y_n; \tau^{(l)}) \), which shares the same dimensionality with \( x_n^{(l)} \). Summarizing these contents yields the following optimization problem:

\[
\theta^*, \xi^*, \tau^* = \arg \min_{\theta \in \Theta, \xi, \tau} \mathbb{E}_{x_n \sim D} \sum_{l=1}^{L} \mathcal{L}_{n}^{(l)}, \quad \text{where} \quad (2)
\]

\[
\mathcal{L}_{n}^{(l)} = \mathcal{L}\left(y_n^{(l)}, h^{(l)}(\tilde{x}_n^{(l)}; \xi^{(l)})\right), \quad (3)
\]

\[
\tilde{x}_n^{(l)} = g^{(l)}(x_n^{(l)}, y_n; \tau^{(l)}), \quad \text{and} \quad (4)
\]

\[
y_n = \frac{1}{L} \sum_{l=1}^{L} h^{(l)}(\tilde{x}_n^{(l)}; \xi^{(l)}). \quad (5)
\]

Here \( \mathcal{L}\{\cdot, \cdot\} \) is a loss function which is discussed in details later. The above formulate define a recurrent model, in which \( y_n \) cannot be observed even in the training stage. The standard way of optimization involves iteration, in which we start with an \( x_n \) sampled from \( D \) and any \( y_n \) (in the training process, \( y_n \) always has \( C \) dimensions), the first \( C' \) dimensions are provided by ground-truth and other \( C - C' \)

---

\(^3\)Eqn (1) fits mid-level features best, because low-level features (e.g., raw image pixels) are often largely impacted by small noise, in both case it is learning the class-conditional distribution \( P_c = \mathcal{P}(x|c) \) suffers a higher difficulty. Similarly, the very last layers (e.g., containing class-specific logits) are less likely to be additive as in Eqn (1).
dimensions can be randomly initialized.) We first compute $x_n^{(l)}$ for each $l$ accordingly, and use it to compute the first version of $y_n^{(l)} = h^{(l)}(x_n^{(l)}; \xi^{(l)})$. In each round, we compute $y_n$ and use it to compute $g^{(l)}(x_n^{(l)}, y_n; \tau^{(l)})$ so that $x_n^{(l)}$ is updated as $\tilde{x}_n^{(l)}$. Within this process, parameters $\xi^{(l)}$ and $\tau^{(l)}$ are updated accordingly with ground-truth $y_n^*$ and gradient back-propagation. This iteration continues until convergence or a maximal number of rounds is achieved.

### 4.2. Approximation with Hierarchical Refinement

In practice, however, the above formulation has two major drawbacks. The first one lies in the inaccuracy of generative models. We expect a model $g^{(l)}(\cdot)$ to eliminate the components in $x_n^{(l)}$ that correspond to the non-targeted classes in $y_n$. This is increasingly difficult especially when the $x_n^{(l)}$ is far from $y$. So, we assume that $x_n^{(l)}$ only receives supervision signals from $x_n^{(l+1)}$, which is much closer than $y$, while $x_n^{(l+1)}$ continues to receive information from $x_n^{(l+2)}$ and this process continues until $y_n$ is reached. In implementation, this implies that reversed connections only emerge between neighboring feature layers. Here an exception happens at the last feature layer, $x_n^{(L)}$, which is connected to $y_n$ via a classifier $h^{(L)}(\cdot)$. Since direct supervisions have already been provided by this classifier, we ignore the connection between $y_n$ and $x_n^{(L)}$ leaving a total of $L - 1$ connections between $x_n^{(l)}$ and $x_n^{(l+1)}$ for $l = 1, 2, \ldots, L - 1$. This is to say, $g^{(l)}(x_n^{(l)}, y_n; \xi^{(l)})$ is replaced by $g^{(l)}(x_n^{(l)}, x_n^{(l+1)}; \xi^{(l)})$. Nevertheless, $x_n^{(l)}$ can still obtain supervision signals from $y_n$ in an indirect manner, i.e., via a few intermediate steps. This is named the hierarchical refinement strategy.

Implementation details are illustrated in Fig. 4. We start with $x_n^{(L)} \equiv x_n^{(L)}$, the feature extracted from the top layer. It is concatenated with the feature at the previous stage, $x_n^{(L-1)}$, before which it is up-sampled if necessary. The concatenated feature is then fed into $g^{(L-1)}(x_n^{(L-1)}, x_n^{(L)}; \tau^{(L-1)})$ to produce $x_n^{(L-1)}$. This process continues until $\tilde{x}_n^{(l)}$ is obtained. $g^{(L-1)}(\cdot)$ means that up-sample $x_n^{(L)}$ and feed it into a function with $x_n^{(L-1)}$. Each $x_n^{(l)}$, $l = 1, 2, \ldots, L$, is sent into the corresponding classifier $h^{(l)}(\tilde{x}_n^{(l)}; \xi^{(l)})$ to obtain $y_n^{(l)}$. All $y_n^{(l)}$ are averaged into the final output and supervised by $y^*$.

The second drawback is the slowness of an iterative optimization. To accelerate, we switch off iteration so that each case $x_n \in D$ is forward-propagated and back-propagated only once, and the updated parameters $\theta, \xi^{(l)}$ and $\tau^{(l)}$ are directly applied to another case sampled from $D$. This can be understood as stochastic gradient descent on $D$. In practice, this allows us to sample more data in the same period of time, and thus improve training efficiency.

### 4.3. Class-Balanced Loss

Here we study the impact of the loss function, i.e., Eqn (3), in the training process. In this specific problem, i.e., prohibited item discovery, there are much fewer positive training samples (at least one prohibited item is labeled) than negative ones. This makes regular loss functions such as the Euclidean loss $L(y^*, y_n) = \|y^* - y_n\|^2$. Instead, we prefer $L(l, y_n; \xi^{(l)})$ to be class-balanced because it greatly reduces the influence of positive training samples.
We evaluate both image-level classification mean Average Precision and object-level localization accuracy, for the second goal we manually labeled all prohibited items with bounding-box in the testing images. For image classification, we apply the evaluation metric in the PascalVOC image classification task [9], which works on each class individually – all testing images are ranked by the confidence of containing the specified object, and the mean average precision (mAP) is computed. For object localization, we follow [38] to compute the accuracy of pointing localization. A hit is counted if the pixel of the maximum response falls within one of the ground-truth bounding-boxes of the specified object, otherwise a missed is counted. Thus, each class has a localization accuracy computed by \( \frac{\text{#Hits}}{\text{#Hits + #Misses}} \). For both tasks, we also report the overall performance which is the average over all five classes.

We investigate five popular backbones, including ResNets [13] with 34, 50 and 101 layers, Inception-v3 [32], and DenseNet with 121 layers. We follow the conventions to setup all these networks, and CHR is applied to each of them using \( L = 3 \) – three pooling layers with different spatial resolutions (e.g., in ResNets, \( 28 \times 28, 14 \times 14 \), and \( 7 \times 7 \)) are used as features. It is of course possible to increase \( L \) by adding more features, yet in practice we find \( L = 3 \) is sufficient to provide complementary information.

### 5.2. Classification: Quantitative Results

We first investigate the overall (averaged over five classes) image classification results which are summarized in Tab. 2. CHR achieves consistent mean Average Precision gain beyond all network backbones as well as in all different subsets, i.e., SIXray10, SIXray100 and SIXray1000. We observe that CHR works better in deeper networks, which is also observed in experiments, e.g., on top of Inception-v3 and DenseNet, the absolute improvement over SIXRay1000 is 8.22% and 9.08%, respectively.

We next observe five types of objects individually. The benefit brought by CHR is different from class to class. Take the DenseNet as an example. When it is aimed at finding \textit{gun}, classification performance is not boosted in all subsets, while we observe significant gains over all the other classes, especially for \textit{scissors}, the accuracy is improved by an impressive amount of 30%. We can see in Table 1 that the training samples of \textit{scissors} is the least among all five prohibited items, for which reason the baseline suffers significant bias in the training stage. CHR, by introducing hierarchical signals for supervision, largely alleviates this bias.

Finally, we study the issue of data imbalance over different subsets. Recall that the ratio of negative over positive images is 10, 100 and 1000, respectively. From Fig. 5, we can see that the performance gain goes up with data imbalance, which, as analyzed in Sec. 5.4, comes from our special treatment towards class balancing.
5.3. Localization: Quantitative Results

To verify that CHR is not over-tuned to image classification, we attach the class activation map (CAM) [39], an weakly supervised approach for object localization, on top of the features extracted at different stages. CAM produces one heatmap for each class individually, and on each of these maps. We first rescale the maps to the original image size. If the maximal response across scales falls within one of the ground truth bounding boxes of the specified object, the predicted location is considered a valid localization.

Tab. 3 summarizes localization results. CHR based on DenseNet outperforms DenseNet by 5.61% (50.31% vs 44.70%) for SIXray100 and 9.26% (43.87% vs 34.61%) for SIXray1000. In particular, for the *wrench* class in

![Figure 5](image_url)

**Figure 5.** The overall accuracy gain of CHR becomes more significant in the subsets with larger negative-positive ratios.

<table>
<thead>
<tr>
<th>Method</th>
<th>Gun</th>
<th>Knife</th>
<th>Wrench</th>
<th>Pliers</th>
<th>Scissors</th>
<th>mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResNet34 [13]</td>
<td>89.71</td>
<td>83.06</td>
<td>72.05</td>
<td>85.46</td>
<td>78.75</td>
<td>56.42</td>
</tr>
<tr>
<td>ResNet34+CHR</td>
<td>87.16</td>
<td>81.96</td>
<td>73.55</td>
<td>87.17</td>
<td>77.70</td>
<td>60.46</td>
</tr>
<tr>
<td>ResNet50 [13]</td>
<td>90.64</td>
<td>84.75</td>
<td>74.19</td>
<td>87.82</td>
<td>77.92</td>
<td>59.82</td>
</tr>
<tr>
<td>ResNet50+CHR</td>
<td>87.55</td>
<td>82.64</td>
<td>73.43</td>
<td>86.38</td>
<td>79.60</td>
<td>61.32</td>
</tr>
<tr>
<td>ResNet101 [13]</td>
<td>87.65</td>
<td>82.83</td>
<td>76.04</td>
<td>84.26</td>
<td>76.16</td>
<td>63.53</td>
</tr>
<tr>
<td>ResNet101+CHR</td>
<td>85.45</td>
<td>81.25</td>
<td>75.85</td>
<td>87.41</td>
<td>77.53</td>
<td>63.80</td>
</tr>
<tr>
<td>Inception-v3</td>
<td>90.05</td>
<td>81.18</td>
<td>76.52</td>
<td>83.80</td>
<td>77.28</td>
<td>56.31</td>
</tr>
<tr>
<td>Inception-v3+CHR</td>
<td>88.90</td>
<td>79.22</td>
<td>76.91</td>
<td>87.23</td>
<td>73.48</td>
<td>61.29</td>
</tr>
<tr>
<td>DenseNet [18]</td>
<td>87.36</td>
<td>83.23</td>
<td>75.00</td>
<td>87.71</td>
<td>77.24</td>
<td>66.55</td>
</tr>
<tr>
<td>DenseNet+CHR</td>
<td>87.05</td>
<td>82.06</td>
<td>74.87</td>
<td>85.89</td>
<td>78.75</td>
<td>71.23</td>
</tr>
</tbody>
</table>

Table 2. Classification mean average precision (%) on the subsets of SIXray (each cell, left to right: SIXray10, SIXray100, SIXray1000).

<table>
<thead>
<tr>
<th>Method</th>
<th>Gun</th>
<th>Knife</th>
<th>Wrench</th>
<th>Pliers</th>
<th>Scissors</th>
<th>mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResNet34 [13]</td>
<td>71.60</td>
<td>70.02</td>
<td>45.93</td>
<td>51.28</td>
<td>55.38</td>
<td>38.97</td>
</tr>
<tr>
<td>ResNet34+CHR</td>
<td>75.02</td>
<td>70.19</td>
<td>45.47</td>
<td>55.38</td>
<td>63.18</td>
<td>26.15</td>
</tr>
<tr>
<td>ResNet50 [13]</td>
<td>63.89</td>
<td>47.51</td>
<td>42.32</td>
<td>57.44</td>
<td>52.82</td>
<td>48.72</td>
</tr>
<tr>
<td>ResNet50+CHR</td>
<td>68.83</td>
<td>57.72</td>
<td>60.67</td>
<td>58.46</td>
<td>49.23</td>
<td>37.44</td>
</tr>
<tr>
<td>ResNet101 [13]</td>
<td>73.75</td>
<td>75.15</td>
<td>70.44</td>
<td>65.15</td>
<td>64.10</td>
<td>60.00</td>
</tr>
<tr>
<td>ResNet101+CHR</td>
<td>80.86</td>
<td>79.32</td>
<td>79.03</td>
<td>73.85</td>
<td>69.23</td>
<td>61.54</td>
</tr>
<tr>
<td>Inception-v3</td>
<td>79.94</td>
<td>64.81</td>
<td>71.16</td>
<td>75.38</td>
<td>63.64</td>
<td>52.31</td>
</tr>
<tr>
<td>Inception-v3+CHR</td>
<td>75.70</td>
<td>67.20</td>
<td>74.41</td>
<td>74.36</td>
<td>63.08</td>
<td>44.54</td>
</tr>
<tr>
<td>DenseNet [13]</td>
<td>74.38</td>
<td>71.00</td>
<td>58.05</td>
<td>71.28</td>
<td>62.05</td>
<td>56.92</td>
</tr>
<tr>
<td>DenseNet+CHR</td>
<td>79.01</td>
<td>78.40</td>
<td>76.78</td>
<td>76.92</td>
<td>62.56</td>
<td>57.95</td>
</tr>
</tbody>
</table>

Table 3. Localization accuracy (%) on the subsets of SIXray (each cell, left to right: SIXray10, SIXray100, SIXray1000).

![Figure 6](image_url)

**Figure 6.** Examples of object localization based on DenseNet, Which shows CHR is effective in complex background and overlapping images. (best viewed in color).
5.4. Ablation Studies

In this part, we investigate the impact brought by hierarchical refinement and class-balanced loss, respectively. All experiments are performed on three subsets of SIXray, which have different ratios of negative-positive samples. Results are summarized in Tab. 4.

First, we study the performance of hierarchical refinement, namely, the reversed connections in the network. It can be seen that the top-down refinement (ResNet34+HR) improves the classification and localization accuracies by about 1% and 6.52% on SIXray100, and 3.15% and 2.13% on SIXray1000. We note that ResNet34+HR outperforms ResNet34+H, the direct way incorporating hierarchical information, because the ResNet34+HR allows low-level features to be refined with high-level semantic cues.

Second, we study the impact of different loss functions. ResNet34+CH, with the designed class-balanced loss, improves classification and localization accuracies by 1.00% and 3.77% on SIXray100, and 3.09% and 3.44% on SIXray1000. By combining hierarchical refinement with the class-balance loss, ResNet34+CHR further improves the classification and localization accuracies by +2.37% and +7.51% on SIXray100, and +5.48% and +5.11% on SIXray1000, over the baseline. This shows the usefulness of class balance in imbalanced scenarios.

CHR achieves accuracy gain with a relatively small amount of extra computation. For example, ResNet34 requires 7.68ms to process each testing image and ResNet34-CHR requires 8.28ms, both tested on a Tesla V100 GPU. This is to say, CHR requires 7.81% extra time.

5.5. ILSVRC2012 Classification

Last but not least, we evaluate CHR on ILSVRC2012, a large-scale image classification dataset. This is to observe how CHR generalizes to natural image data, provided that it achieves significant accuracy gain on overlapping image data. ILSVRC2012 is a popular subset of the ImageNet databased, which has 1.000 classes and each of them contains a well-defined concept in WordNet. There are 1.3M training images and 50K validation images, both of which are roughly uniformly distributed over all classes.

We follow the standard training and testing pipelines, including the policies of model initialization, data augmentation, learning rate decay, etc. Since ILSVRC2012 is not an imbalanced dataset, we switch off the weight terms in the loss function which was designed for this purpose. The top-1 error of CHR based on ResNet18 is 27.01% [13], which slightly lower than the baseline by 0.87% (27.01% vs 27.88%). Besides, the top-1 and top-5 errors of CHR based on ResNet50 [13] are 22.00% and 6.22%, which are lower than the baseline by 0.85% (22.00% vs 22.85%) and 0.49% (6.22% vs 6.71%), respectively. This slight but consistent accuracy gain delivers two-fold messages. The reversed connections in our approach which carries high-level supervision to mid-level features do not conflict with natural images – although it aligns with overlapping image data much better. Given that the additional computational costs are almost negligible, it is worth investigating its extension in the natural image domains.

6. Conclusions

In this paper, we investigate prohibited item discovery in X-ray scanned images, which is a promising application in industry yet remains fewer studied in computer vision. To facilitate research in this field, we present SIXray, a large-scale dataset consisting of more than one million X-ray images, all of which were captured in real-world scenarios and therefore covered complicated scenarios. We manually annotated six types and more than 20,000 prohibited items, which is at least 100 times larger than all existing datasets. In methodology, we formulate X-ray images as the overlap of several sub-images, therefore sampled from a mixture distribution. Motivated by filtering irrelevant information, we present an algorithm to refine mid-level features in a hierarchical and iterative manner. In practice, we switch off iteration to optimize the network weights in an approximate but efficient manner. A novel loss function is also built upon the hierarchical architecture to deal with heavy data imbalance between positive and negative classes. Beyond a few popular network backbones, our approach produces consistent gain in both classification and localization accuracy, establishing a strong baseline for the proposed task.

The future research mainly lies in two directions. First, the formulation of overlapping images from the penetration assumption is not accurate in many aspects – we look forward to more effective approaches based on a better physical model. Second, the connection between overlapping images and natural images, e.g., object occlusion, remains unclear – studying this topic may imply some ways of extending these approaches to a wider range of applications.

Acknowledgments. This work was supported by the NSFC grant 61836012, 61771447, and 61671427, and Beijing Municipal Science and Technology Commission grant Z181100008918014.
References


[35] Fang Wan, Pengxu Wei, Zhenjun Han, Jianbin Jiao, and Qixiang Ye. Min-entropy latent model for weakly supervised object detection. TPAMI, 2019. 3


