

# Object Recognition in X-ray testing Using Adaptive Sparse Representations

Domingo Mery, Erick Svec, Marco Arias

**Abstract**—In recent years, X-ray screening systems have been used to safeguard environments in which access control is of paramount importance. Security checkpoints have been placed at the entrances to many public places to detect prohibited items such as handguns and explosives. Human operators complete these tasks because automated recognition in baggage inspection is far from perfect. Research and development on X-ray testing is, however, ongoing into new approaches that can be used to aid human operators. This paper attempts to make a contribution to the field of object recognition by proposing a new approach called Adaptive Sparse Representation (XASR+). It consists of two stages: learning and testing. In the learning stage, for each object of training dataset, several patches are extracted from its X-ray images in order to construct representative dictionaries. A stop-list is used to remove very common words of the dictionaries. In the testing stage, test patches of the test image are extracted, and for each test patch a dictionary is built concatenating the ‘best’ representative dictionary of each object. Using this adapted dictionary, each test patch is classified following the Sparse Representation Classification (SRC) methodology. Finally, the test image is classified by patch voting. Thus, our approach is able to deal with less constrained conditions including some contrast variability, pose, intra-class variability, size of the image and focal distance. We tested the effectiveness of our method for the detection of four different objects. In our experiments, the recognition rate was more than 97% in each class, and more than 94% if the object is occluded less than 15%. Results show that XASR+ deals well with unconstrained conditions, outperforming various representative methods in the literature.

**Index Terms**—X-ray testing, computer vision, sparse representations, image analysis.



## 1 INTRODUCTION

**B**AGGAGE inspection using X-ray screening is a priority task that reduces the risk of crime, terrorist attacks and propagation of pests and diseases [28]. Security and safety screening with X-ray scanners has become an important process in public spaces and at border checkpoints [19]. However, inspection is a complex task because threat items are very difficult to detect when placed in closely packed bags, occluded by other objects, or rotated, thus presenting an unrecognizable view [2]. Manual detection of threat items by human inspectors is extremely demanding [22]. It is tedious because very few bags actually contain threat items, and it is stressful because the work of identifying a wide range of objects, shapes and substances (metals,

organic and inorganic substances) takes a great deal of concentration. In addition, human inspectors receive only minimal technological support. Furthermore, during rush hours, they have only a few seconds to decide whether or not a bag contains a threat item [1]. Since each operator must screen many bags, the likelihood of human error becomes considerable over a long period of time even with intensive training. The literature suggests that detection performance is only about 80–90% [14]. In baggage inspection, automated X-ray testing remains an open question due to: *i) loss of generality*, which means that approaches developed for one task may not transfer well to another; *ii) deficient detection accuracy*, which means that there is a fundamental tradeoff between false alarms and missed detections; *iii) limited robustness* given that requirements for the use of a method are often met for simple structures only; and *iv) low adaptiveness* in that it may be very difficult to accommodate an automated system to design modifications or different specimens.

---

• D. Mery is with the Department of Computer Science Department, Pontificia Universidad Católica, Vicuña Mackenna 4860, Santiago Chile.

• E-mail: dmery@ing.puc.cl URL: <http://dmery.ing.puc.cl>

There are some contributions in computer vision for X-ray testing such as applications on inspection of castings, welds, food, cargos and baggage screening [9]. For this research proposal, it is very interesting to review the advances in baggage screening that have taken place over the course of this decade. They can be summarized as follows: Some approaches attempt to recognize objects using a single view of mono-energy X-ray images (*e.g.*, the adapted implicit shape model based on visual codebooks [21]) and dual-energy X-ray images (*e.g.*, Gabor texture features [26], bag of words based on SURF features [25] and pseudo-color, texture, edge and shape features [29]). More complex approaches that deal with multiple X-ray images have been developed as well. In the case of mono-energy imaging, see for example the recognition of regular objects using data association in [10] and active vision [20] where a second-best view is estimated. In the case of dual-energy imaging, see the use of visual vocabularies and SVM classifiers in [6]. Progress also has been made in the area of computed tomography. For example, in order to improve the quality of CT images, metal artifact reduction and de-noising [16] techniques were suggested. Many methods based on 3D features for 3D object recognition have been developed (see, for example, RIFT and SIFT descriptors [4], 3D Visual Cortex Modeling 3D Zernike descriptors and histogram of shape index [8]). There are contributions using known recognition techniques (see, for example, bag of words [5] and random forest [17]). As we can see, the progress in automated baggage inspection is modest and still very limited compared to what is needed because X-ray screening systems are still being manipulated by human inspectors. Automated recognition in baggage inspection is far from being perfected given that the appearance of the object of interest can become extremely difficult due to problems of (self-)occlusion, noise, acquisition, clutter, etc.

We believe that algorithms based on sparse representations can be used for this general task because in many computer vision applications, under the assumption that natural images can be represented using sparse decomposition, state-of-the-art results have been significantly improved [24]. Thus, it is possible to cast the problem of recognition into a supervised recognition form with X-ray images and class labels (*e.g.*, objects to be recognized) using

learned features in a unsupervised way. In the sparse representation approach, a dictionary is built from the training X-ray images, and matching is done by reconstructing the test image using a sparse linear combination of the dictionary. Usually, the test image is assigned to the class with the minimal reconstruction error.

Reflecting on the problems confronting recognition of objects, we believe that there are some key ideas that should be present in new proposed solutions. First, it is clear that certain parts of the objects are not providing any information about the class to be recognized (for example occluded parts). For this reason, such parts should be detected and should not be considered by the recognition algorithm. Second, in recognizing any class, there are parts of the object that are more relevant than other parts (for example the sharp parts when recognizing sharp objects like knives). For this reason, relevant parts should be class-dependent, and could be found using unsupervised learning. Third, in the real-world environment, and given that X-ray images are not perfectly aligned and the distance between detector and objects can vary from capture to capture, analysis of fixed parts can lead to misclassification. For this reason, feature extraction should not be in fixed positions. Moreover, it would be possible to use a selection criterion that enables selection of the best regions. Fourth, an object that is present in a test image can be subdivided into ‘sub-objects’, for different parts (*e.g.*, in case of a handgun there are trigger, muzzle, grip, etc.). For this reason, when searching for images of the same class it would be helpful to search for image parts in all images of the training images instead of similar training images.

Inspired by these key ideas, we propose a method for recognition of objects using X-ray images<sup>1</sup>. Three main contributions of our approach are: 1) A new general algorithm that is able to recognize regular objects: it has been evaluated in the recognition of four different objects. 2) A new representation for the classes to be recognized using patches: this is based on representative dictionaries learned for each class of the training images, which correspond to a rich collection of representations of selected relevant parts that are particular to a specific class. 3) A new representation for the test X-ray image:

1. A similar approach was developed by us for a biometric problem [12].

this is based on *i*) a discriminative criterion that selects the ‘best’ test patches from the test image and *ii*) and an ‘adaptive’ sparse representation of the selected patches computed from the ‘best’ representative dictionary of each class. Using these new representations, the proposed method (XASR+) can achieve high recognition performance under many complex conditions, as shown in our experiments. A preliminary version of this paper was presented in [11].

The rest of the paper is organized as follows. In Section 2, the proposed method is explained in further detail. In Section 3, the experiments and results are presented. Finally, in Section 4, concluding remarks are given.

## 2 PROPOSED METHOD

The proposed XASR+ method consists of two stages: learning and testing (see Fig. 1). In the learning stage, for each object of the training, several patches are extracted and described from their images in order to build representative dictionaries. In the testing stage, test patches of the test image are extracted and described, and for each test patch a

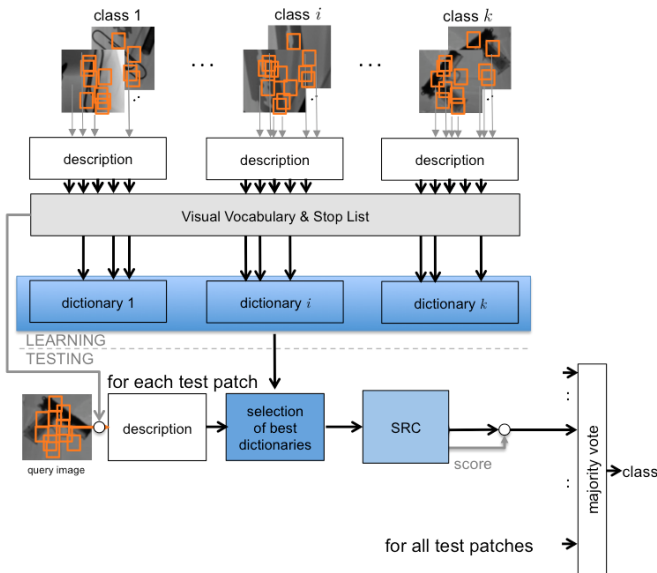


Fig. 1: Overview of the proposed method. The figure illustrates the recognition of three different objects. The shown classes are three: clips, razor blades and springs. There are two stages: Learning and Testing. The stop-list is used to filter out patches that are not discriminating for these classes. The stopped patches are not considered in the dictionaries of each class and in the testing stage.

dictionary is built concatenating the ‘best’ representative dictionary of each object. Using this adapted dictionary, each test patch is classified in accordance with the Sparse Representation Classification (SRC) methodology [27]. Afterwards, the patches are selected according to a discriminative criterion. Finally, the test image is classified by voting for the selected patches. Both stages will be explained in this section in further detail.

### 2.1 Model learning

In the training stage, a set of  $n$  object images of  $k$  objects is available, where  $\mathbf{I}_j^i$  denotes X-ray image  $j$  of object  $i$  (for  $i = 1 \dots k$  and  $j = 1 \dots n$ ) as illustrated in Fig. 2. In each image  $\mathbf{I}_j^i$ ,  $m$  patches  $\mathcal{P}_{jp}^i$  of size  $w \times w$  pixels (for  $p = 1 \dots m$ ) are extracted. They are centered in  $(x_{jp}^i, y_{jp}^i)$ . In this work, a patch  $\mathcal{P}$  is defined as vector:

$$\mathbf{p} = [\mathbf{z}; \alpha r] \in \mathcal{R}^{d+1} \quad (1)$$

where  $\mathbf{z} = f(\mathcal{P}) \in \mathcal{R}^d$  is a descriptor of patch  $\mathcal{P}$  (*i.e.*, a local descriptor of  $d$  elements extracted from the patch) that has been normalized to unit length (*i.e.*,  $\|\mathbf{z}\| = 1$ );  $r$  is the distance of the center of the patch  $(x_{jp}^i, y_{jp}^i)$  to the center of the image; and  $\alpha$  is a weighting factor between descriptor and location. Description  $\mathbf{z}$  must be rotation invariant because the orientation of the object can be anyone.

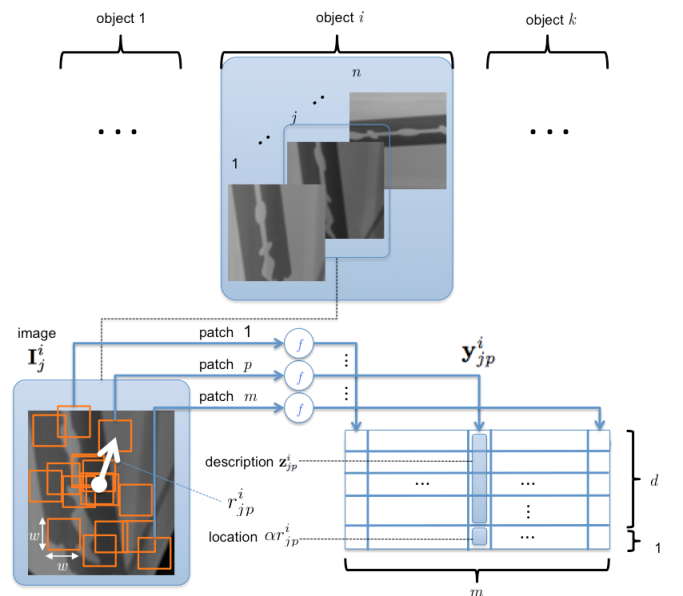


Fig. 2: Extraction and description of  $m$  patches of training image  $j$  of object  $i$ .

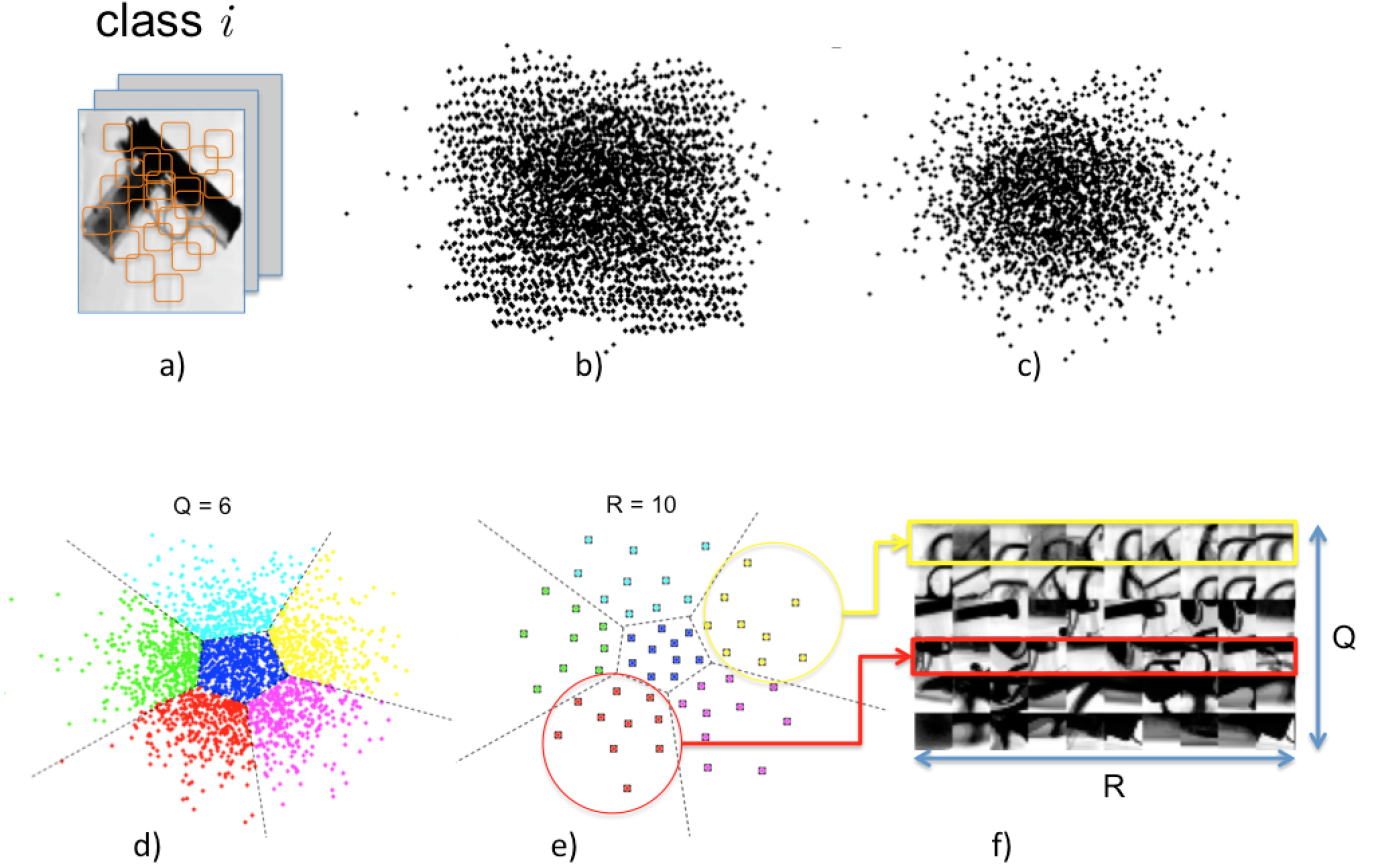


Fig. 3: Representation of object  $i$ : a) Patches of all X-ray images of class  $i$  are extracted. b) Set  $\mathbf{Z}_i$  contains the representation of each patch  $\mathbf{z}_{jp}^i \in \mathcal{R}^{(d+1)}$  (see black points). c) Set  $\mathbf{Y}_i$  contains the points of  $\mathbf{Z}_i$  that do not belong to the stop list. d) Set of points  $\mathbf{Y}^i$  is clustered in  $Q$  parent clusters which points are arranged in array  $\mathbf{Y}_q^i$  with centroid  $\mathbf{c}_q^i$ . e) Each parent cluster  $\mathbf{Y}_q^i$  is clustered in  $R$  child clusters which centroids  $\{\mathbf{c}_{qr}^i\}_{r=1}^R$  are arranged in array  $\mathbf{A}_q^i$ . f) Visualization of the patches of the dictionary. In this example  $\mathbf{Y}^i$  has 2,400 points,  $Q = 6$  (parent clusters) and  $R = 10$  (child clusters).

In order to eliminate non-discriminative patches, a *stop-list* is computed from a *visual vocabulary*. The visual vocabulary is built using all descriptors  $\mathbf{Z} = \{\mathbf{z}_{jp}^i\} \in \mathcal{R}^{d \times knm}$ , for  $i = 1 \dots k$ , for  $j = 1 \dots n$  and for  $p = 1 \dots m$ . Array  $\mathbf{Z}$  is clustered using a k-means algorithm in  $N_v$  clusters. Thus, a visual vocabulary containing  $N_v$  visual words is obtained. In order to construct the stop-list, the *term frequency* ‘ $t_f$ ’ is computed:  $t_f(d, v)$  is defined as the number of occurrences of word  $v$  in document  $d$ , for  $d = 1 \dots K$ ,  $v = 1 \dots N_v$ . In our case, a document corresponds to an X-ray image, and  $K = kn$  is the number of classes in the training dataset. Afterwards, the *document frequency* ‘ $d_f$ ’ is computed:  $d_f(v) = \sum_d \{t_f(d, v) > 0\}$ , i.e., the number of images in the training dataset that contain a word  $v$ , for  $v = 1 \dots N_v$ . The stop-list is built using words with highest and smallest  $d_f$  values:

On one hand, visual words with highest  $d_f$  values are not discriminative because they occur in almost all images. On the other hand, visual words with smallest  $d_f$  are so unusual that they correspond in most of the cases to noise. Usually, the top 5% and bottom 10% are stopped [23]. Those patches of  $\mathbf{Z}$  that belong to the stopped clusters are not considered in the following steps of our algorithm. The filtered patches are represented by  $\mathbf{Y}$ , and  $\mathbf{Y}_i$  corresponds to the filtered patches of object  $i$  as shown in Fig. 3b.

The description  $\mathbf{Y}^i$  of object  $i$  is clustered using k-means algorithm in  $Q$  clusters that will be referred to as *parent clusters* (Fig. 3c):

$$\mathbf{c}_q^i = \text{kmeans}(\mathbf{Y}^i, Q) \quad (2)$$

for  $q = 1 \dots Q$ , where  $\mathbf{c}_q^i \in \mathcal{R}^{(d+1)}$  is the centroid of parent cluster  $q$  of object  $i$ . We define  $\mathbf{Y}_q^i$  as the

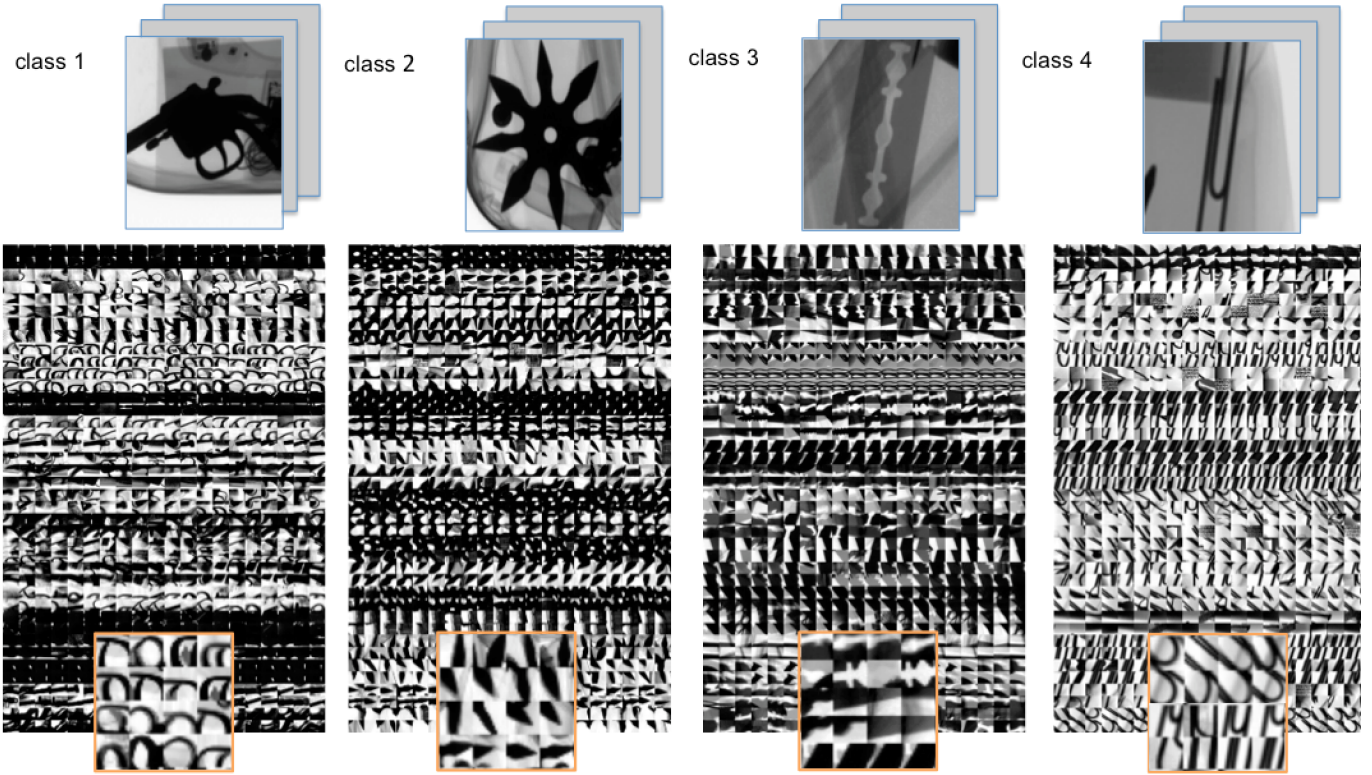


Fig. 5: Example of four dictionaries ( $Q = 32$ ,  $R = 20$ ): guns, shuriken, razor blades and clips. A zoom of each dictionary is shown at the bottom.

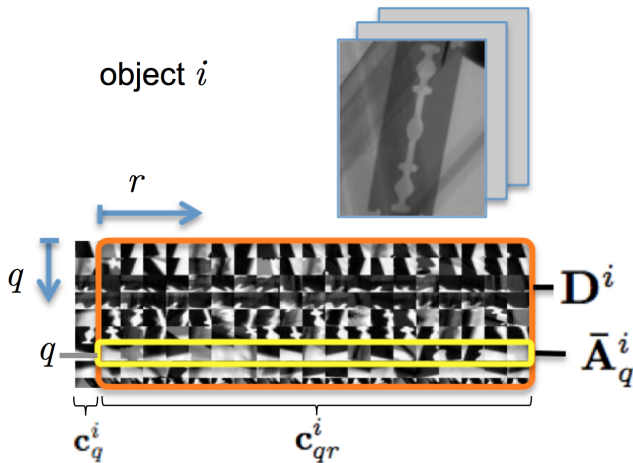


Fig. 4: Representative dictionaries of object  $i$  for  $Q = 32$  (only for  $q = 1 \dots 7$  is shown) and  $R = 20$ . Left column shows the centroids  $c_q^i$  of parent clusters. Right columns (orange rectangle called  $D^i$ ) shows the centroids  $c_{qr}^i$  of child clusters.  $\bar{A}_q^i$  is row  $q$  of  $D^i$ , i.e., the centroids of child clusters of parent cluster  $q$ .

array with all samples  $y_{jp}^i$  that belong to the parent cluster with centroid  $c_q^i$ . In order to select a reduced number of samples, each parent cluster is clustered

again in  $R$  child clusters (Fig. 3d):

$$c_{qr}^i = \text{kmeans}(Y_q^i, R) \quad (3)$$

for  $r = 1 \dots R$ , where  $c_{qr}^i \in \mathcal{R}^{(d+1)}$  is the centroid of child cluster  $r$  of parent cluster  $q$  of object  $i$ . All centroids of child clusters of object  $i$  are arranged in an array  $D^i$ , and specifically for parent cluster  $q$  are arranged in a matrix:

$$\bar{A}_q^i = [c_{q1}^i \dots c_{qr}^i \dots c_{qR}^i]^T \in \mathcal{R}^{(d+1) \times R} \quad (4)$$

Thus, this arrange contains  $R$  representative samples of parent cluster  $q$  of object  $i$  as illustrated in Fig. 4. The set of all centroids of child clusters of object  $i$  ( $D^i$ ), represents  $Q$  representative dictionaries with  $R$  descriptions  $\{c_{qr}^i\}$  for  $q = 1 \dots Q$ ,  $r = 1 \dots R$ .

## 2.2 Testing

In the testing stage, the task is to determine the identity of the test image  $I^t$  given the model learned in the previous section. From the test image,  $s$  selected test patches  $\mathcal{P}_p^t$  of size  $w \times w$  pixels are extracted and described using function  $f$  of equation

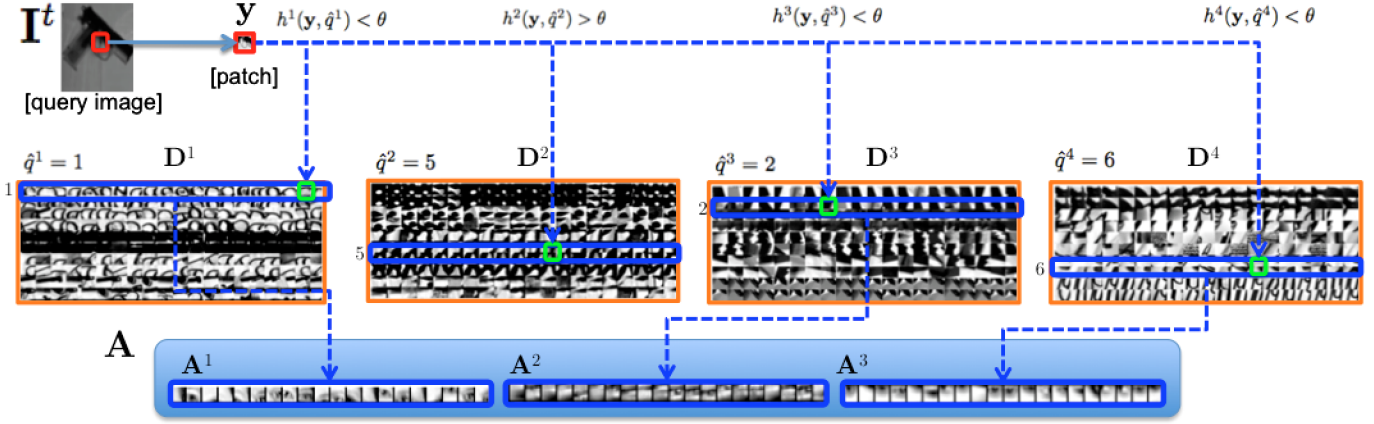


Fig. 6: Adaptive dictionary  $\mathbf{A}$  of patch  $y$ . In this example there are  $k = 4$  objects in the training dataset. For this patch only  $k' = 3$  objects are selected. Dictionary  $\mathbf{A}$  is built from those objects by selecting all child clusters (of a parent cluster -see blue rectangles-) which has a child cluster with the smallest distance to the patch (see green squares). In this example, object 2 does not have child clusters that are similar enough, *i.e.*,  $h^2(y, \hat{q}^2) > \theta$ .

(1) as  $\mathbf{y}_p^t = f(\mathcal{P}_p^t)$  (for  $p = 1 \dots s$ ). The selection criterion of a test patch will be explained later in this section.

For each selected test patch with description  $\mathbf{y} = \mathbf{y}_p^t$ , a distance to each parent cluster  $q$  of each object  $i$  of the training dataset is measured:

$$h^i(\mathbf{y}, q) = \text{distance}(\mathbf{y}, \bar{\mathbf{A}}_q^i). \quad (5)$$

We tested with several distance metrics. The best performance, however, was obtained by:

$$h^i(\mathbf{y}, q) = \min_r \|\mathbf{y} - \mathbf{c}_{qr}^i\| \quad \text{for } r = 1 \dots R, \quad (6)$$

which is the smallest Euclidean distance to centroids of child clusters of parent cluster  $q$  as illustrated in Fig. 6. For  $\mathbf{y}$  and  $\mathbf{c}_{qr}^i$  normalized to unit  $\ell_2$  norm, the following distance can be used based on (6):

$$h^i(\mathbf{y}, q) = \min_r (1 - \langle \mathbf{y}, \mathbf{c}_{qr}^i \rangle) \quad \text{for } r = 1 \dots R, \quad (7)$$

where the term  $\langle \bullet \rangle$  corresponds to the scalar product that provides a similarity (cosine of angle) between vectors  $\mathbf{y}$  and  $\mathbf{c}_{qr}^i$ . The parent cluster that has the minimal distance is searched:

$$\hat{q}^i = \underset{q}{\operatorname{argmin}} h^i(\mathbf{y}, q), \quad (8)$$

which minimal distance is  $h^i(\mathbf{y}, \hat{q}^i)$ .

For patch  $y$ , we select those training objects that have a minimal distance less than a threshold  $\theta$  in order to ensure a similarity between the test patch and representative object patches. If  $k'$  objects fulfill the condition  $h^i(\mathbf{y}, \hat{q}^i) < \theta$  for  $i = 1 \dots k'$ , with

$k' \leq k$ , we can build a new index  $v_{i'}$  that indicates the index of the  $i'$ -th selected object for  $i' = 1 \dots k'$ . For instance in a training dataset with  $k = 4$  objects, if  $k' = 3$  objects are selected (*e.g.*, objects 1, 3 and 4), then the indices are  $v_1 = 1$ ,  $v_2 = 3$  and  $v_3 = 4$  as illustrated in Fig. 6. The selected object  $i'$  for patch  $y$  has its dictionary  $\mathbf{D}^{v_{i'}}$ , and the corresponding parent cluster is  $u_{i'} = \hat{q}^{v_{i'}}$ , in which child clusters are stored in row  $u_{i'}$  of  $\mathbf{D}^{v_{i'}}$ , *i.e.*, in  $\mathbf{A}^{i'} := \bar{\mathbf{A}}_{u_{i'}}^{v_{i'}}$ .

Therefore, a dictionary for patch  $y$  is built using the best representative patches as follows (see Fig. 6):

$$\mathbf{A}(\mathbf{y}) = [ \mathbf{A}^1 \dots \mathbf{A}^{i'} \dots \mathbf{A}^{k'} ] \in \mathcal{R}^{(d+1) \times Rk'} \quad (9)$$

With this adaptive dictionary  $\mathbf{A}$ , built for patch  $y$ , we can use *Sparse Representation Classification* (SRC) methodology [27]. That is, we look for a sparse representation of  $y$  using the  $\ell_1$ -minimization approach:

$$\hat{\mathbf{x}} = \underset{\mathbf{x}}{\operatorname{argmin}} \|\mathbf{x}\|_1 \quad \text{object to } \mathbf{A}\mathbf{x} = \mathbf{y} \quad (10)$$

The residuals are calculated for the reconstruction for the selected objects  $i' = 1 \dots k'$ :

$$r_{i'}(\mathbf{y}) = \|\mathbf{y} - \mathbf{A}\delta_{i'}(\hat{\mathbf{x}})\| \quad (11)$$

where  $\delta_{i'}(\hat{\mathbf{x}})$  is a vector of the same size of  $\hat{\mathbf{x}}$  whose only nonzero entries are the entries in  $\hat{\mathbf{x}}$  corresponding to class  $v(i') = v_{i'}$ . Thus, the class

of selected test patch  $\mathbf{y}$  will be the class that has the minimal residual, that is it will be

$$\hat{i}(\mathbf{y}) = v(\hat{i}') \quad (12)$$

where  $\hat{i}' = \operatorname{argmin}_{i'} r_{i'}(\mathbf{y})$ .

Finally, the identity of the test object will be the majority vote of the classes assigned to the  $s$  selected test patches  $\mathbf{y}_p^t$ , for  $p = 1 \dots s$ :

$$\operatorname{identity}(\mathbf{I}^t) = \operatorname{mode}(\hat{i}(\mathbf{y}_1^t), \dots, \hat{i}(\mathbf{y}_p^t), \dots, \hat{i}(\mathbf{y}_s^t)) \quad (13)$$

The selection of  $s$  patches of test image is as follows:

*i)* From test image  $\mathbf{I}^t$ ,  $m$  patches are extracted and described using (??):  $\mathbf{y}_j^t$ , for  $j = 1 \dots m$ , with  $m \geq s$ .

*ii)* Each patch  $\mathbf{y}_j^t$  is represented by  $\hat{\mathbf{x}}_j^t$  using the mentioned adaptive sparse representation according to (10).

*iii)* The *sparsity concentration index* (SCI) of each patch is computed in order to evaluate how spread are its sparse coefficients [27]. SCI is defined by

$$S_j := \operatorname{SCI}(\mathbf{y}_j^t) = \frac{k \max(\|\delta_{i'}(\hat{\mathbf{x}}_j^t)\|_1) / \|\hat{\mathbf{x}}_j^t\|_1 - 1}{k - 1} \quad (14)$$

If a patch is discriminative enough it is expected that its SCI is large. Note that we use  $k$  instead of  $k'$  because the concentration of the coefficients related to  $k$  classes must be measured.

*iv)* Array  $\{S\}_{j=1}^m$  is sorted in a descended way.

*v)* The first  $s$  patches in this sorted list in which SCI values are greater than a  $\tau$  threshold are then selected. If only  $s'$  patches are selected, with  $s' < s$ , then the majority vote decision in (13) will be taken with the first  $s'$  patches.

### 3 EXPERIMENTS

Our method was tested in the recognition of five classes ( $k = 5$ ) in baggage screening: 1) handguns, 2) *shuriken* (ninja stars), 3) clips, 4) razor blades and 5) background (see some samples in Fig. 7). In our experiments, there are 100 X-ray images per class. All images were resized to  $128 \times 128$  pixels. We defined the following experimental protocol based on leave-one-object-out strategy: from each class, we choose randomly 50 images for training and one for testing. The test accuracy ( $\eta_i$ ) for this test  $i$  is

defined as the ratio  $c_i/k$ , where  $c_i$  is the number of correctly classified samples and  $k$  is the number of samples. In order to obtain a better confidence level in the estimation of the accuracy, the mentioned test was repeated 100 times (for  $i = 1 \dots 100$ ) by randomly selecting new 51 images per class in each test (50 for training and 1 for testing). Thus, the reported accuracy ( $\bar{\eta}$  in Table 1) in all of our experiments is the average calculated over the 100 tests, *i.e.*,  $\bar{\eta} = \frac{1}{100} \sum_i \eta_i$ . The code for the MATLAB implementation is available on our webpage<sup>2</sup>. The X-ray images belong to GDXray database [13].

We tested the proposed methods using three different descriptors:

- 1) LBP<sub>8,1</sub><sup>*r*</sup>, *i.e.*, Local Binary Pattern rotation-invariant with 8 samples and radius 1 [18]. That yields a 36-bin descriptor ( $d = 36$ ). The patches were extracted randomly. The size of the patch was  $24 \times 24$  pixels ( $w = 24$ ).
- 2) SIFT descriptor of  $d = 128$  elements extracted in the detected SIFT keypoints [7].
- 3) A concatenation of both descriptors obtaining a descriptor of  $d = 128 + 36 = 164$  elements, where the LBP features are extracted from a patch centered in the location of the SIFT key-point with a size of  $24 \times 24$  pixels ( $w = 24$ ).

We call these methods XASR<sub>+1</sub>, XASR<sub>+2</sub> and XASR<sub>+3</sub> respectively.

In order to evaluate the robustness against occlusion, we corrupted the test images with a square of random gray value of size  $a \times a$  pixels located randomly, for  $a = 15, 30, 50, 70$  (see example in Table 1). The obtained result is given in first three rows of Table 1 (see XASR+'s rows). We observe that XASR<sub>+3</sub> (that used both LBP and SIFT features) had the best accuracy. It achieved more than 95% in each class when there is no occlusion, and around 85% if the object is occluded less than 30%.

In order to evaluate the effectiveness of the stop-list, we repeated the same experiment without considering this step. The results are given in the XASR's rows of Table 1 (see XASR<sub>1</sub>, XASR<sub>2</sub> and XASR<sub>3</sub> for LBP, SIFT and the concatenation LBP and SIFT). We observe that the use of a stop-list can increase the accuracy significantly.

In addition, we compared our method with four known methods that can be used in object recognition: *i)* SIFT [7], *ii)* sparse representation classifica-

2. See <http://dmery.ing.puc.cl/index.php/material/>

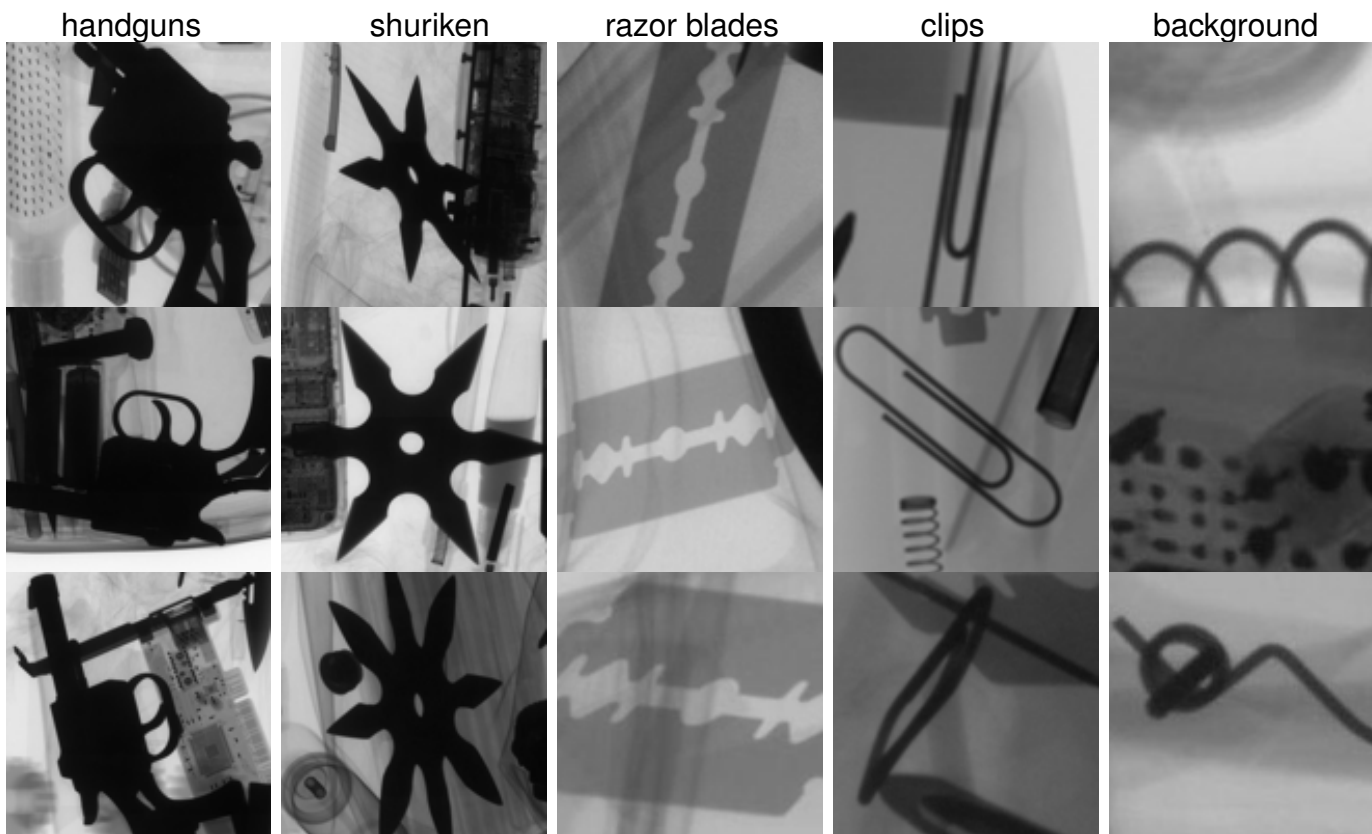


Fig. 7: Example of three images per class used in our experiments. The five classes are: handguns, shuriken, razor blades, clips and background.

tion (SRC) [27] with SIFT descriptors, *iii*) efficient visual search based on an information retrieval approach (Vgoogle) [23], and *iv*) bag of words [3] using KNN (BoW-KNN) and random forest (BoW-RF) [15] with SIFT descriptors. We coded these methods according to the specifications given by the authors in their papers. The parameters were set so as to obtain the best performance. The results are summarized in the corresponding rows of Table 1. Results show that XASR+ deals well with unconstrained conditions in every experiment, achieving a high recognition performance in many conditions and obtaining similar or better performance in comparison with other representative methods in the literature.






The time computing depends on the size of the dictionary that is proportional to the number of classes to be detected. In our experiments with 5 classes the computational time is about 0.2s per testing image (testing stage) on a Mac Mini Server OS X 10.10.1, processor 2.6 GHz Intel Core i7 with 4 cores and memory of 16GB RAM 1600 MHz DDR3.

## 4 CONCLUSIONS

In this paper, we have presented XASR+, an algorithm that is able to recognize objects automatically in cases with less constrained conditions including some contrast variability, pose, intra-class variability, size of the image and focal distance. We tested the effectiveness of our method for the detection of four different objects: razor blades, *shuriken* (ninja stars) handguns and clips. In our experiments, the recognition rate was more than 95% in every class. The robustness of our algorithm is due to three reasons: *i*) the dictionaries learned for each class in the learning stage corresponded to a rich collection of representations of relevant parts which were selected and clustered; *ii*) the testing stage was based on *adaptive sparse representations* of several patches using the dictionaries estimated in the previous stage which provided the best match with the patches, and *iii*) a visual vocabulary and a stop-list used to reject non-discriminative patches in both learning and testing stage.



TABLE 1: Accuracy [%] of each method ( $\bar{\eta}$ ). An example of the occlusion in the class handgun is illustrated at the top.

					
Occlusion →	0 (0%)	15×15 (1.4%)	30×30 (5.5%)	50×50 (15.3%)	70×70 (29.9%)
Method ↓					
XASR <sub>+1</sub>	97.0	96.5	95.0	89.5	82.3
XASR <sub>+2</sub>	92.8	93.2	89.6	79.6	66.8
XASR <sub>+3</sub>	<b>98.8</b>	<b>98.4</b>	<b>97.2</b>	<b>94.4</b>	<b>84.8</b>
XASR <sub>1</sub>	92.0	92.0	85.5	31.5	20.5
XASR <sub>2</sub>	90.8	84.8	87.6	82.0	66.0
XASR <sub>3</sub>	98.0	97.2	95.2	92.8	78.8
SIFT	91.0	87.6	84.2	78.4	64.6
SRC	94.8	89.4	85.8	81.0	70.6
Vgoogle	87.2	83.6	82.8	70.4	54.6
BoW-KNN	88.6	84.4	82.6	73.8	55.0
BoW-RF	84.4	75.2	73.6	61.0	38.2

## ACKNOWLEDGMENTS

This work was supported in part by Fondecyt Grant No. 1130934 from CONICYT, Chile.

## REFERENCES

- [1] Garrick Blalock, Vrinda Kadiyali, and Daniel H Simon. The Impact of Post-9/11 Airport Security Measures on the Demand for Air Travel. *The Journal of Law and Economics*, 50(4):731–755, November 2007.
- [2] Anton Bolting, Tobias Halbherr, and Adrian Schwaninger. How image based factors and human factors contribute to threat detection performance in X-ray aviation security screening. In Andreas Holzinger, editor, *HCI and Usability for Education and Work*, volume 5298 of *Lecture Notes in Computer Science*, pages 419–438. Springer Berlin Heidelberg, 2008.
- [3] Gabriella Csurka, Christopher Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. In *European Conference on Computer Vision (ECCV 2004)*, page 327?334, 2004.
- [4] Greg Flitton, Toby P Breckon, and Najla Megherbi. A comparison of 3D interest point descriptors with application to airport baggage object detection in complex CT imagery. *Pattern Recognition*, 46(9):2420–2436, September 2013.
- [5] Greg Flitton, André Mouton, and Toby P Breckon. Object classification in 3D baggage security computed tomography imagery using visual codebooks. *Pattern Recognition*, 48(8):2489–2499, August 2015.
- [6] T Franzel, U Schmidt, and S Roth. Object Detection in Multi-view X-Ray Images. *Pattern Recognition*, 2012.
- [7] D Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110, November 2004.
- [8] N Megherbi, Jiwan Han, T P Breckon, and G T Flitton. A comparison of classification approaches for threat detection in CT based baggage screening. In *Image Processing (ICIP), 2012 19th IEEE International Conference on*, pages 3109–3112. IEEE, 2012.
- [9] D Mery. *Computer Vision for X-Ray Testing*. Springer, 2015.
- [10] D Mery. Inspection of Complex Objects Using Multiple-X-Ray Views. *IEEE/ASME Transactions on Mechatronics*, 20(1):338–347, 2015.
- [11] D. Mery, E. Svec, and M. Arias. Object recognition in baggage inspection using adaptive sparse representations of X-ray images. In *Proceedings of the Pacific Rim Symposium on Image and Video Technology (PSIVT 2015)*, 2015.
- [12] Domingo Mery and Kevin Bowyer. Automatic facial attribute analysis via adaptive sparse representation of random patches. *Pattern Recognition Letters*, 68:260–269, 2015.
- [13] Domingo Mery, Vladimir Riffo, Uwe Zscherpel, German Mondragón, Iván Lillo, Irene Zuccar, Hans Lobel, and Miguel Carrasco. GDxray: The database of X-ray images for non-destructive testing. *Journal of Nondestructive Evaluation*, 34(4):1–12, 2015.
- [14] S. Michel, S.M. Koller, J.C. de Ruyter, R. Moerland, M. Hogervorst, and A. Schwaninger. Computer-based training increases efficiency in X-ray image interpretation by aviation security screeners. In *Security Technology, 2007 41st Annual IEEE International Carnahan Conference on*, pages 201–206, Oct 2007.
- [15] Frank Moosmann, Bill Triggs, and Frederic Jurie. Fast discriminative visual codebooks using randomized clustering forests. In *Twentieth Annual Conference on Neural Information Processing Systems (NIPS’06)*, pages 985–992. MIT Press, 2007.
- [16] A Mouton, G T Flitton, and S Bizot. An evaluation of image denoising techniques applied to CT baggage screening imagery. In *IEEE International Conference on Industrial Technology (ICIT 2013)*. IEEE, 2013.
- [17] André Mouton and Toby P Breckon. Materials-based 3D segmentation of unknown objects from dual-energy computed tomography imagery in baggage security screening. *Pattern Recognition*, 48(6):1961–1978, June 2015.
- [18] Timo Ojala, Matti Pietikäinen, and Topi Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, 2002.
- [19] European Parliament. Aviation security with a special fo-

- cus on security scanners. *European Parliament Resolution (2010/2154(INI))*, pages 1–10, October 2012.
- [20] V Rizzo and D Mery. Active X-ray testing of complex objects. *Insight-Non-Destructive Testing and Condition Monitoring*, 54(1):28–35, 2012.
- [21] V. Rizzo and D. Mery. Automated detection of threat objects using Adapted Implicit Shape Model. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 46(4):472–482, 2016.
- [22] Adrian Schwaninger, Anton Bolting, Tobias Halbherr, Shaun Helman, Andrew Belyavin, and Lawrence Hay. The impact of image based factors and training on threat detection performance in X-ray screening. In *Proceedings of the 3rd International Conference on Research in Air Transportation, ICRAT 2008*, pages 317–324, 2008.
- [23] J Sivic and A Zisserman. Efficient Visual Search of Videos Cast as Text Retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(4):591–606, 2009.
- [24] I. Tosic and P. Frossard. Dictionary learning. *Signal Processing Magazine, IEEE*, 28(2):27–38, 2011.
- [25] D Turcsany, A Mouton, and T P Breckon. Improving feature-based object recognition for X-ray baggage security screening using primed visualwords. In *IEEE International Conference on Industrial Technology (ICIT 2013)*, pages 1140–1145, 2013.
- [26] I Uroukov and R Speller. A preliminary approach to intelligent x-ray imaging for baggage inspection at airports. *Signal Processing Research*, 2015.
- [27] John Wright, Allen Y Yang, Arvind Ganesh, Shankar S Sastry, and Yi Ma. Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2):210–227, 2009.
- [28] G. Zentai. X-ray imaging for homeland security. *IEEE International Workshop on Imaging Systems and Techniques (IST 2008)*, pages 1–6, Sept. 2008.
- [29] N Zhang and J Zhu. A study of X-ray machine image local semantic features extraction model based on bag-of-words for airport security. *International Journal on Smart Sensing and Intelligent Systems*, 8(1):45–64, 2015.