NSH: Normality Sensitive Hashing for Anomaly Detection

Hirotaka Hachiya, Masakazu Matsugu
Canon Inc.
3-30-2 Shimomaruko, Ohta-ku, Tokyo 146-8501, Japan
{hachiya.hirotaka, matsugu.masakazu}@canon.co.jp

Abstract

Locality sensitive hashing (LSH) is a computationally efficient alternative to the distance based anomaly detection. The main advantages of LSH lie in constant detection time, low memory requirement, and simple implementation. However, since the metric of distance in LSHs does not consider the property of normal training data, a naive use of existing LSHs would not perform well. In this paper, we propose a new hashing scheme so that hash functions are selected dependently on the properties of the normal training data for reliable anomaly detection. The distance metric of the proposed method, called NSH (Normality Sensitive Hashing) is theoretically interpreted in terms of the region of normal training data and its effectiveness is demonstrated through experiments on real-world data. Our results are favorably comparable to state-of-the arts with the low-level features.

1. Introduction

Anomaly detection is a problem of detecting data that do not follow a normal behavior, and there are a wide variety of applications in computer vision such as a surveillance camera. In a general setting of anomaly detection problem, it is prohibitively expensive to collect training data at an abnormal situation, *e.g.*, criminal and accidental situations. This causes the difficulty of the anomaly detection that we need to deal with unseen types of anomaly not included in the training data. To overcome this difficulty, many of anomaly detection approaches define a *normal* model representing the *normal* training data collected at normal situations, and detect a test instance which does not follow this model as an anomaly [6].

Along this line, many types of anomaly detection approaches have been proposed, such as a statistical model approach, a classification based approach and a distance based approach. In a statistical model approach, a statistical model such as topic model [11], Bayesian network [3], and Markov random field [14], is estimated from the normal training data. Then, statistical inference is used to determine if the test instance is normal or not. This approach could detect a complex anomaly *e.g.*, indirectly related to observations. However, the statistical model needs to be carefully designed dependently on each target application—a statistical model for some target application would not be applicable to other applications. For example, the statistical model for detecting abnormal objects among pedestrians [3] cannot be applicable to detecting abnormal condition of taxi drivers from their facial expression.

In a classification based approach, a classification boundary is learned from the normal training data. Then, if a test instance does not belong to the inside of the boundary, it is classified as an anomaly [22, 21]. Classification based approach could handle complex normal models using kernel functions and the optimal parameters of the model can be analytically learned. However, the classification based approach would not provide an anomaly score (the degree of anomaly) since it classifies a test instance to normal or anomaly only. In anomaly activity detection, the anomaly score is useful because it indicates how emergent an anomaly event occurs.

In a distance based approach, the distance between a test instance and normal training instances is measured and then an intuitive anomaly score, *e.g.*, k-nearest neighbor distance [4, 5] and the sum of k-nearest neighbor distance [2, 24] is provided. In addition, this approach does not need to design the statistical model for each target application since the distance between instances is simply measured. However, the computational cost and memory requirement for computing the anomaly score is basically dependent on the number of normal training data and thus it was difficult to apply this approach to a large scale problem such as anomaly activity detection.

To alleviate this disadvantage, *locality sensitive hashing* (LSH) such as p-stable hashing [23] and randomized trees [17] have been explored in the field of outlier detection. These approaches compute *approximated* distance by simply counting normal training instances allocated to the same bucket as a test instance. Thus, the computational cost

and memory requirement for computing the anomaly score is *constant*. In addition, it can be implemented without any advanced optimization solvers and cumbersome functions. However, the distance metric used in these approaches considers only individual similarity, *e.g.*, Euclid distance and Hamming distance between two instances. Thus, the approximated distance is not relevant to the property of normal training data and thus its resulting anomaly detection would be unstable, *e.g.*, due to the influence of dense and sparse regions of the data.

Our contribution in this paper is to propose the new hashing scheme that enables accurate anomaly detection by incorporating the normality dependent hash functions. In particular, we select hash functions from random candidates so that instances within the normal region are allocated to the same bucket and instances across the normal region boundary are allocated to the different buckets. In this new hashing scheme, we develop a reliable anomaly detection method while retaining the advantages of LSHs. The theoretical interpretation of the proposed method, called NSH (Normality Sensitive Hashing) is discussed and its effectiveness is demonstrated through experiments on toy data and real world data (i.e., UMN, UCSDped and Subway) for anomaly activity detection.

2. Related Work

In this section, we review the distance based approach for the anomaly detection and a computationally efficient alternative based on locality sensitive hashing (LSH).

2.1. Distance-based Anomaly Detection

First of all, let us assume that *normal* training data $\mathcal{D} = \{\mathbf{p}_n\}_{n=1}^{N}$ is collected at normal situations, where $\mathbf{p}_n$ is a D-dimensional feature vector and N is the number of the training data. Then, we define the term of *anomaly* in distance based approach based on the work [15] as follows:

Definition 1 $\mathbf{DB}(\mathcal{D},\mathbf{q})$ *anomaly: a test instance* $\mathbf{q}$ *is an anomaly if at least fraction* p_t *of the normal training data* $\mathcal{D}$ *lie at a distance greater than* d_t *from* $\mathbf{q}$.

This definition indicates that a test instance $\mathbf{q}$ is anomaly if there are not many enough training instances within the radius d_t from $\mathbf{q}$. The problem detecting such anomaly corresponds to the decision version of nearest neighbor search that the number of training data within the radius d_t from $\mathbf{q}$ is counted.

2.2. Locality Sensitive Hashing

For efficiently counting nearest neighbors from large training data, LSH is a widely known and useful approach. The basic concept of LSH is to allocate similar instances to the same bucket with high probability. Mathematical definition of LSH is as follows [8]:

Definition 2 *A family* $\mathcal{H} = \{h : \mathcal{S} \rightarrow \mathcal{U}\}$ *is called* (r_1, r_2, p_1, p_2) *sensitive if for any two instances* $\mathbf{p}, \mathbf{q} \in \mathcal{S}$*:*

$$\textit{If } d(\mathbf{p},\mathbf{q}) \leq r_1 \textit{ then } pr(h(\mathbf{p}) = h(\mathbf{q})) \geq p_1, \quad (1)$$

$$\textit{If } d(\mathbf{p},\mathbf{q}) \geq r_2 \textit{ then } pr(h(\mathbf{p}) = h(\mathbf{q})) \leq p_2. \quad (2)$$

where $d(\mathbf{p},\mathbf{q})$ is the distance between a training instance $\mathbf{p}$ and a test instance $\mathbf{q}$ in the context of the anomaly detection and $pr(h(\mathbf{p}) = h(\mathbf{q}))$ is called *collision probability*. More specifically, the contrapositive argument of Def. 2 implies that if the probability of a training instance $\mathbf{p}$ stored into the same bucket as a test instance $\mathbf{q}$ is high, then $\mathbf{p}$ is near $\mathbf{q}$. This scheme is known as $(R,1+\epsilon)$*-approximate nearest neighbor search* in particular when $r_1 = R$ and $r_2 = (1+\epsilon)R$ where R is the nearest neighbor distance in $\mathcal{D}$. From this observation, the number $N_{m,\mathbf{q}}$ of training instances stored into the same bucket as $\mathbf{q}$ by a hash function h_m can be used as an anomaly score [23], i.e.,

$$s(\mathbf{q}) = \frac{1}{M}\sum_{m=1}^{M} N_{m,\mathbf{q}} \quad (3)$$

where M is the number of hash functions. Briefly, the lower the value of $s(\mathbf{q})$ is, the less training instances near the test instance $\mathbf{q}$ there are. Thus, when $s(\mathbf{q})$ is lower than a threshold s_t, $\mathbf{q}$ can be detected as anomaly.

The advantages of LSH lie on its constant detection time, low-memory requirement and simple implementation. However, the performance of the anomaly detection based on LSH depends significantly on the metric used to measure the distance. Let us explain this issue briefly using the one of LSHs applied to the anomaly detection, called p-stable hashing [8, 23] defined as

$$h_m^{\mathbf{p-stable}}(\mathbf{q}) \equiv \lfloor \frac{\mathbf{w}_m^{\top}\mathbf{q} + b_m}{r} \rfloor \quad (4)$$

where $\top$ is the transpose operator of a vector, $\mathbf{w}_m$ and b_m are is the normal vector and the bias of hyperplane $\mathbf{w}_m^{\top}\mathbf{q} + b_m = 0$ respectively. $\mathbf{w}_m$ is randomly selected following a p-stable distribution, *e.g.*, $p = 2$ is a normal distribution, and b_m is randomly selected following uniform distribution in the range of $[0, r]$. Then, the collision probability with $p = 2$ can be represented using L2-norm, i.e., Euclidean distance [8].

Euclidean distance is a standard distance metric used in nearest neighbor search. However, the performance of many algorithms is seriously degraded by using Euclidean distance as the distance metric [13, 9, 10]. In particular, the performance of anomaly detection based on Euclid distance could be unstable due to the effect of local densities

of training data (see Sec. 4.1 for the details). Thus, it would be important to design hash functions in consideration of its distance metric. Along this line, we extends the LSH scheme so that the property of entire training data is taken into account.

3. Our Approach

In this section, we propose a new hashing scheme called NSH (Normality Sensitive Hashing).

3.1. Normality Sensitive Hashing

The basic concept of NSH is to allocate instances within the normal region into the same bucket and instances across the region boundary into different buckets. Let us denote a hyperplane by $\mathbf{w}^\top \mathbf{p} - b = 0$ where $\mathbf{w}$ is a D-dimensional normal vector, $\mathbf{p}$ is a D-dimensional feature vector, and b is a scalar bias. From randomly generated candidates of hyperplanes, $\{(\mathbf{w}_l, b_l)\}_l^L$ where L is the number of candidates, we select the one $(\mathbf{w}^*, b^*)$ that minimize the following objective function [1]:

$$\frac{1}{N}\sum_{n=1}^{N} f(\mathbf{w}_l^\top \mathbf{p}_n - b_l) - \lambda b_l \tag{5}$$

where λ is a penalty parameter for a small bias and $f(z)$ is a loss function defined by

$$f(z) \equiv \begin{cases} 0, & \text{if } z \geq 0, \\ z^2, & \text{otherwise.} \end{cases} \tag{6}$$

Given a selected hyperplane $(\mathbf{w}^*, b^*)$, a hash value for a feature vector $\mathbf{p}$ (and $\mathbf{q}$) is computed as follows

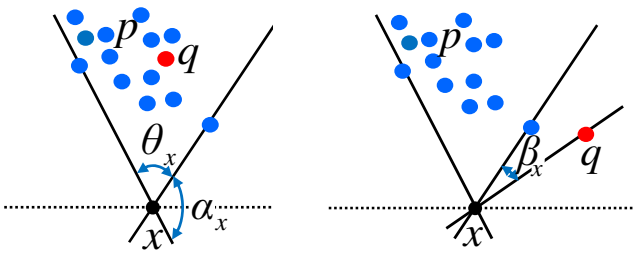
$$h^{\mathbf{NSH}}_{\mathbf{w}^*, b^*}(\mathbf{p}) \equiv \begin{cases} 1, & \text{if } \mathbf{w}^{*\top}\mathbf{p} - b^* \geq 0, \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

The first and second terms of Eq. 5 correspond to the first and second condition of LSH (Def. 2) respectively. The intuition of this hash function is as follows: the fewer number of training data $\mathcal{D}$ a hyperplane $(\mathbf{w}_l, b_l)$ intersects, the smaller the value of the first term of Eq. 5 is. Then, this results in a high collision probability $pr(h(\mathbf{p}) = h(\mathbf{q}))$ since a test instance $\mathbf{q}$ near $\mathbf{p}$ has the same hash value as $\mathbf{p}$ with high probability. In addition, the closer a hyperplane to the training data $\mathcal{D}$ is, the smaller the value of the second term is. Then, this results in a low collision probability since $\mathbf{q}$ far from $\mathbf{p}$ has the same hash value as $\mathbf{p}$ with low probability.

3.2. Theoretical Interpretation

Regarding p_1 and p_2 of Def. 2 for NSH, we have the following lemma:

[1]For simplicity, we introduce to select hash functions that training data are located on the side of its normal vector heading. In practice, we extend Eq. 5 and 7 to select hash functions that training data are located on its either side for computational efficiency.



(a) Maximum angle θ_x in $\mathcal{D}$ (b) Minimum angle β_x in $\mathcal{D}$ from $\mathbf{q}$

Figure 1: Diagram of angles θ_x and β_x used for the region-based distance metric

Lemma 1 *When $D = 2$ and candidate hyperplanes are limited to pass through a point x at the horizontal axis,*

$$p_1 = 1 - \left(\frac{\pi + \theta_x}{2\pi}\right)^L + \left(\frac{\pi - \theta_x}{2\pi}\right)^L, \tag{8}$$

$$p_2 = \left(\frac{2\pi - \beta_x}{2\pi}\right)^L - \left(\frac{\beta_x}{2\pi}\right)^L. \tag{9}$$

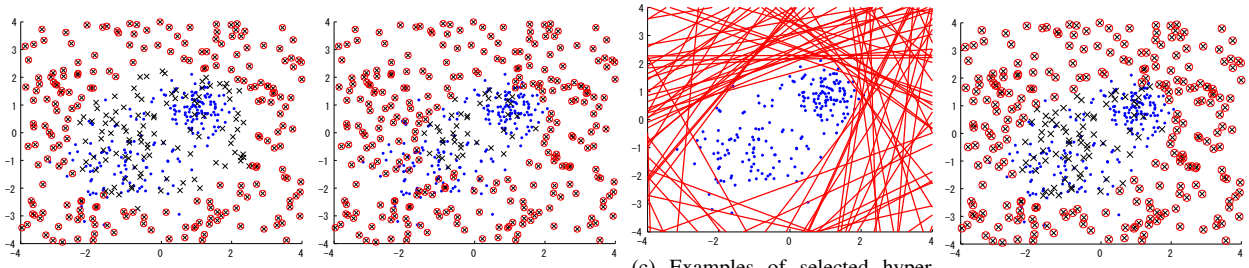
where θ_x is the maximum angle in training data $\mathcal{D}$, and β_x is the minimum angle between $\mathbf{q}$ and $\mathbf{p} \in \mathcal{D}$ with respect to the point $\mathbf{x}$ as shown in Fig. 1. Note that we assume $\theta_x, \beta_x \in \{0, \pi\}$. The proof of Lemma 1 is based on the work [19] and is given in Appendix. This lemma implies that the distance metric of NSH is based on the region of normal training data, represented by angles θ_x and β_x.

3.3. Algorithm

Fig. 2 depicts the pseudo code of learning a normal model by generating hash functions and computing hash values of training data. In addition, Fig. 3 depicts the pseudo code of computing anomaly score for a test instance $\mathbf{q}$. As shown in pseudo codes, the computational cost for learning normality model is linear as $\mathcal{O}(MN)$ where M is the number of selected hash functions. Both the computational cost for detection and memory requirement are constant as $\mathcal{O}(M)$. In addition, these algorithms can be implemented using only multiplication and addition, that any optimization solver and cumbersome functions such as exponential and cosine are not necessary.

4. Evaluation

In this section, we evaluate our proposed NSH through experiments on a toy example and the anomaly activity detection using UMN [20], UCSDped [18] and Subway [1] datasets. In the sequel, we assume 'normal training data' is collected at 'normal' situation.



(a) p-stable hashing with $s_t = 120$ (b) p-stable hashing with $s_t = 140$ (c) Examples of selected hyperplanes in NSH (d) NSH with $s_t = 199$

Figure 4: Toy examples of the anomaly detection using p-stable hashing and NSH. A blue dot is a normal training instance, a black cross is a test instance, and a red circle is a the test instance detected as anomaly.

4.1. Toy Example

To show the qualitative behavior of p-stable hashing and NSH, we use toy data with the dimension $D = 2$, and $N = 200$ normal training data, each dimension of which is following the mixture of two Gaussian distributions $0.5\mathcal{N}(1, 0.25) + 0.5\mathcal{N}(-1, 1)$. That is, there are a dense region (around $(1, 1)$) and sparse region (around $(-1, -1)$) in training data. Each dimension of a test instance is following a uniform distribution $\mathbf{q}_n \sim \mathcal{U}(-5, 5)$ and parameters of p-stable hashing and NSH are set as $M = 100$, $L = 200$, $r = 5$ and $\lambda = 0.0001$. Fig. 4a and Fig 4b depict examples of detected anomalies by p-stable hashing. Fig. 4a shows that when the threshold is relatively low, $s_t = 120$, test instances outside the dense region (around $x = 2$ or $y = 2$) cannot be detected as anomalies. Meanwhile, Fig. 4b shows that when the threshold is relative high, $s_t = 140$, test instances inside the sparse region (around $(-1, -1)$) are wrongly detected as anomalies. The former case causes a high false-negative rate, and the latter case causes a high false-positive rate and thus there is a trade-off on adjusting threshold parameter s_t. This implies that p-stable hashing based on L2-norm distance is seriously influenced by local densities of the training data.

On the other hand, Fig. 4d obtained by NSH shows that when the threshold is high enough *e.g.*, $s_t = 199$ (the maximum threshold is the number of training data $N = 200$), test instances outside the dense region and inside the sparse region are classified correctly. This results in a high accuracy on the anomaly detection, implying that NSH (considering the region of the training data) can handle well the influence of these local densities. Fig. 4c depicts $M = 100$ selected hash functions. This shows that hash functions are located along the region of the normal training data.

4.2. UMN dataset

The UMN dataset [20] consists of three different scenes each of which repeats several intervals of normal (walking) and abnormal (escaping) crowd activities. The frame-rate is 30 per-sec and the resolution is 320×240 pixel. We use 400 frames [2] of each interval as a normal training part and the rest of the interval as a test part. In order to extract features, we estimate the magnitude and orientation of optical flow [16] at each pixel and then compute a multi-scale histogram of optical flow (MHOF) [7] from each of 4×5 regions at each frame as shown in Fig. 5. That is, the number of normal training data is 8000 (20 regions $\times$ 400 frames) and the dimension of data is 16. The threshold for MHOF is set to the average of 1-percentile of the magnitude of all the normal training data.

Method	Average AUC (SD)
k-nearest neighbor	86% (15%)
p-stable hashing	82% (12%)
NSH	88% (12%)

Table 1: Average AUC and its standard deviation on UMN dataset.

The average anomaly score over 20 regions is calculated at each test frame and used for determining if an anomaly behavior occurs. Table 1 depicts the average AUC over all intervals for three methods: the sum of k-nearest neighbor with $k = 15$ as a baseline method, p-stable hashing with $M = 50$, $B = 5$ and $r = 2$ and our proposed NSH with $M = 50$, $L = 1000$, $B = 5$, $\lambda = 0.001$ and $r = 2$, where B is the number of concatenated hash functions [8]. Note that we label frames containing an escape behavior as anomaly. This table shows that our proposed method NSH outperforms a baseline method k-nearest neighbor and p-stable hashing.

Fig. 6 depicts examples of average anomaly scores (normalized to 0 to 1) obtained by NSH, and its corresponding *ground truth* and image frames. These figures show that the anomaly score keeps close to 1 during people waking and then starts decreasing as people start running (*e.g.*,

[2] If there are not 400 frames before the escaping activity, we use 200 frames as the normal training part.

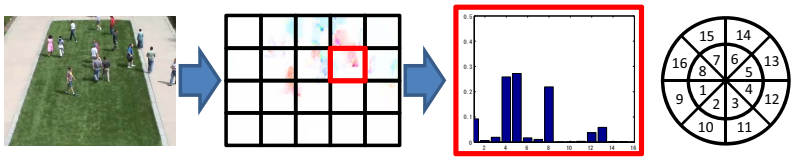
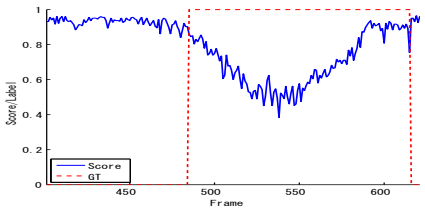


Figure 5: The diagram of flow of feature extraction. From left figure, an input frame, optical flow with regions, a multi-scale histogram of optical flow (MHOF) [7] for the region in red box, and the index of 8 orientations and two levels of magnitude of optical flow.

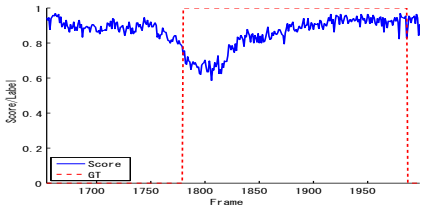
(a) 401th frame (b) 485th (c) 540th (d) 600th

(f) 1655th frame (g) 1780th (h) 1805th (i) 1905th

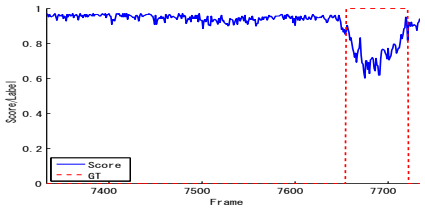
(k) 7333th frame (l) 7655th (m) 7683th (n) 7721th



(e) Anomaly score for scene 1-1



(j) Anomaly score for scene 2-1



(o) Anomaly score for scene 3-3

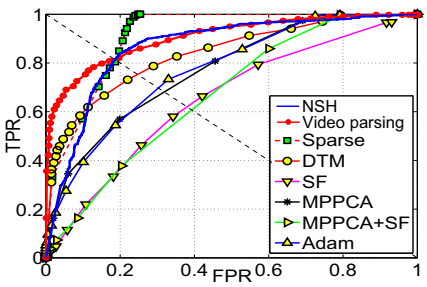
Figure 6: Examples of average anomaly scores obtained by NSH and these corresponding image frames for UMN dataset.

485 frames). The value of anomaly score takes the minimum value when the escaping behavior is at peak levels (*e.g.*, 540th frame), and increases as only a few people remain. These figures imply that NSH can capture correctly the crowd anomaly behavior.

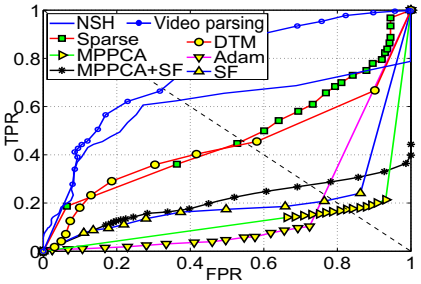
4.3. UCSDped1 dataset

The UCSDped1 dataset [18] consists of 34 normal training and 36 test video clips each of which contain 200 frames and 238×158 resolution. Normal training videos contain only pedestrians but test videos contain also non-pedestrians such as bikers and skaters. Similarly with UMN dataset, we computed MHOF from each of 13×10 regions of each frame with 15-pixel overlap—its threshold is set to the average of 5-percentile of the magnitude of all normal training data. Unlike UMN dataset, we constructed a normal model at each region since appropriate normal models would be diverse over regions. At each frame, the region that the average magnitude of optical flow are small is filtered out since there must not be any pedestrian in such region. Then, the average anomaly score of the current frame, and ten frames before and after is calculated for each region and used for determining if an anomaly behavior occurs at each region. Note that we use the ground truth label provided with the UCSDped1 dataset [18].

Fig. 7 depict results of UCSDped1 dataset for our proposed NSH with $M = 300$, $L = 1000$, $B = 5$, $r = 2$ and $\lambda = 0.0001$ in comparison with the state-of-art methods [3, 7, 1, 18]. Both frame-level and pixel-level results are calculated following the definition described in [18]. This results show that our proposed method, NSH, are well comparable with the state-of-arts (*e.g.*, [7, 3]).



(a)



(b)

Method	EER	RD	AUC
SF [18]	31%	21%	17.9%
MPPCA [18]	40%	18%	20.5%
SF-MPPCA [18]	32%	18%	21.3%
MDT [18]	25%	45%	44.1%
Adam [1]	38%	24%	13.3%
Sparse [7]	19%	46%	46.1%
Video parsing [3]	18%	68%	76%
NSH	18.7%	63.4%	60.3%

(c)

Figure 7: The results of UCSDped1 dataset. (a) Frame-level ROCs, (b) Pixel-level ROCs, and (c) Quantitative comparison of out proposed NSH with state-of-art methods [3, 7, 1, 18]: EER is frame-level equal error rate, RD is pixel-level rate of detection, and AUC is pixel-level area under ROC.

Figure 8: Examples of detected anomalies by NSH in UCSDped1 dataset. Red bounding boxes are the regions taking the lower anomaly score than a fixed threshold.

Especially, our framework of NSH with region dependant normality model (i.e., 13×10 models) outperforms *Sparse* [7] and *MPPCA* [18] which use the same type of low-level features, *e.g.*, optical flow. Furthermore, NSH has advantages on constant detection time, low memory requirement and simple implementation. Meanwhile, *Sparse* [7] needs to solve a linear programming problem for computing anomaly score, and thus its computational cost depends on the number of iterations—the cost in the worst case is known to be $\mathcal{O}(C^{3.5}T)$ [12] where C is the number of dictionaries and T is the number of bits in a test instance.

Fig. 8 depict examples of detected anomaly objects by NSH. These figures show that our proposed method NSH can capture correctly anomalies such as bikers, skaters, carts, persons running, and persons walking in the grass.

4.4. Subway exit dataset

The Subway exit dataset [1] consists of a surveillance video clip (43 minutes) captured at the exit gate of a subway station. We used the first 10 minutes of the clip as normal training data and the rest of the clip as test data. We resized the frames from 512×384 to 320×240 and divided the resized frames into 10×10 regions with 10-pixel overlap. Similarly with UCSD dataset, we computed MHOF from each of regions with the threshold set to the average of 1-percentile of the magnitude of all normal training data,

Method	Detection rate	False alarms
Adam [1]	100%(9/9)	2
Sparse [7]	100%(9/9)	0
NSH	100%(9/9)	1

Table 2: Detection rate and false alarms on Subway exit dataset.

and constructed a normal model at each region. Table. 2 depicts results of Subway dataset for our proposed NSH with $M = 300$, $L = 1000$, $B = 5$, $r = 2$ and $\lambda = 0.0001$ in comparison with the state-of-art methods [1, 7]. We note that the area of detection is limited to the rectangle from the point $(40, 60)$ with the width of 160 and the height of 50, corresponding to an upper part of the exit gate. Fig. 9 depict examples of detected anomaly activities by NSH. These results show that our framework of NSH with region-dependent normality model (i.e., 10×10 models) perform reasonably well even in complex anomaly detection problems.

5. Conclusions

We proposed a new hashing scheme, NSH, that hashing function is specially designed for anomaly detection.

Algorithm 3.1: LEARNNORMALMODEL($M, L, \mathcal{D}, \lambda, r$)

// Initialize counter
$C \leftarrow \mathbf{0}$
for $m \leftarrow 1$ **to** M
{
  // Collect hash candidates
  $\{(\mathbf{w}_l, b_l)\}_l^L \leftarrow$ COLLECTHASH(r, L)

  // Select a hash function
  $(\mathbf{w}_m^*, b_m^*) \leftarrow$ SELECTHASH($\{(\mathbf{w}_l, b_l)\}_l^L, \lambda$)

  // Update counter
  for $n \leftarrow 1$ **to** N
  {
    if $h^{\mathbf{NSH}}_{\mathbf{w}_m^*, b_m^*}(\mathbf{q}_n) = 0$
      then $C_{m,0} = C_{m,0} + 1$
      else $C_{m,1} = C_{m,1} + 1$
  }
return $(\{\mathbf{w}_m^*, b_m^*\}_m^M, C)$

Figure 2: Pseudo code for learning a normality model. By the *CollectHash* function, the candidates $\{(\mathbf{w}_l, b_l)\}_l^L$ of hyperplanes are collected following $\mathbf{w}_{l,d} \sim \mathcal{N}(0, 1)$ where $\mathbf{w}_{l,d}$ is the dth element of $\mathbf{w}_l$, and $b_l \sim \mathcal{U}(-r, r)$. By the *SelectHash* function, a hash function $(\mathbf{w}_m^*, b_m^*)$ is selected using Eq. 5. $C_{m,z}$ is the matrix containing the number of training data having the hash value z by the hash function $(\mathbf{w}_m^*, b_m^*)$.

Algorithm 3.2: COMPUTESCORE($\{\mathbf{w}_m^*, b_m^*\}_m^M, C, \mathbf{q}$)

// Initialize anomaly score
$s \leftarrow 1$
for $m \leftarrow 1$ **to** M
{
  // Update anomaly score
  if $h^{\mathbf{NSH}}_{\mathbf{w}_m^*, b_m^*}(\mathbf{q}_n) = 0$
    then $s = s + C_{m,0}$
    else $s = s + C_{m,1}$
}
$s = \frac{s}{M}$
return (s)

Figure 3: Pseudo code for computing anomaly score.

Through experiments on a toy example and anomaly activity detection using UMN, UCSDped1 and Subway datasets, we experimentally confirmed that NSH can capture anomalies with a high accuracy, comparable to state-of-the-arts. We showed that the algorithm of NSH needs only a constant computational complexity, memory requirement with simple implementation. Overall, NSH could be useful in a resource-constrained device such as surveillance camera. Further comparison with other state-of-the-art hashing techniques would be future work.

The performance of NSH could be further improved when training data with anomaly labels are available since more reliable region could be defined by adding the loss-function regarding anomaly labels to Eq. 5, *e.g.*,

$$\frac{N_n}{N}\sum_{n=1}^{N_p} f(\mathbf{w}_l^\top \mathbf{x}_n - b_l) + \frac{N_p}{N}\sum_{n=1}^{N_n} f(-(\mathbf{w}_l^\top \mathbf{x}_n - b_l)) - \lambda b_l \tag{10}$$

where N_p is the number of normal training data, N_n is the number of abnormal training data, and $N = N_p + N_n$.

Furthermore, in NSH, we can select new hashing functions in on-line manner. That is, we first evaluate hash functions $(\mathbf{w}_m^*, b_m^*)$ selected for previous training data using Eq. 5 with new training data. Then, we reuse highly evaluated hashing functions, *e.g.*, low value of Eq. 5 as new hashing functions for new data. This on-line update would be effective in the case that the normality model changes gradually in space and time, *e.g.*, UCSDped1 and Subway dataset are the cases of the spatial change.

Figure 9: Examples of detected anomalies by NSH in Subway exit dataset.

Appendix

Proof of p_1

A hyperplane passing through the angle α_x (see Fig. 1a) holds the equality $h(\mathbf{p}) = h(\mathbf{q})$ for any $(\mathbf{p}, \mathbf{q}) \in \angle\theta_x$. Thus, we consider the probability of a hyperplane in the angle α_x being selected by our proposed method, NSH. In the angle α_x, the hyperplane whose normal vector is heading/not heading to the training data has the lower/higher value than the ones in other regions in Eq. 5 with low enough value of λ. From this fact, there are two cases that a hyperplane in the angle α_x is selected—when at least one heading candidate is in the angle α_x, or when all L non-heading candidates are in the angle α_x. The probability of the former one is $1 - \left(\frac{\pi+\theta_x}{2\pi}\right)^L$ and the probability of the latter one is $\left(\frac{\pi-\theta_x}{2\pi}\right)^L$. Then, the summation of these probabilities results in Eq. 8.

Proof of p_2

A hyperplane passing through the angle β_x always holds $h(\mathbf{p}) \neq h(\mathbf{q})$ for any $\mathbf{q} \notin \angle\theta_x$. Here, we consider the

probability of a hyperplane passing through the angle β_x being selected by our proposed method, NSH. The hyperplane whose normal vector is heading and close to the training data has the lower value in Eq. 5 with low enough (non-zero) value of λ. From this fact, there are two cases that a hyperplane in the angle β_x is selected—when at least one heading candidate is in the angle β_x, or when all L non-heading candidates are in the angle β_x. The probability of the former one is $1 - \left(\frac{2\pi-\beta_x}{2\pi}\right)^L$ and the probability of the latter one is $\left(\frac{\beta_x}{2\pi}\right)^L$. The summation of these probabilities results in the lower-bound of the probability $pr(h(\mathbf{p}) \neq h(\mathbf{q}))$, i.e.,

$$pr(h(\mathbf{p}) \neq h(\mathbf{q})) \geq 1 - \left(\frac{2\pi - \beta_x}{2\pi}\right)^L + \left(\frac{\beta_x}{2\pi}\right)^L. \quad (11)$$

Then, $1 - pr(h(\mathbf{p}) \neq h(\mathbf{q}))$ results in the upper-bound p_2 of the probability $pr(h(\mathbf{p}) = h(\mathbf{q}))$ in Eq. 9.

References

[1] A. Adam, E. Rivlin, I. Shimshoni, and D. Reinitz. Robust real-time unusual event detection using multiple fixed-location monitors. *IEEE Transactions of Pattern Analysis and Machine Intelligence*, 30(3):555–560, 2008.

[2] F. Angiulli and C. Pizzuti. Fast outlier detection in high dimensional spaces. In *Proceedings of the 6th European Conference on Principles of Data Mining and Knowledge Discovery (PKDD2002)*, pages 15–26, 2002.

[3] B. Antic and B. Ommer. Video parsing for abnormality detection. In *Proceedings of 2011 IEEE International Conference on Computer Vision (ICCV2011)*, pages 2415–2422, 2011.

[4] S. D. Bay and M. Schwabacher. Mining distance-based outliers in near linear time with randomization and a simple pruning rule. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 29–38, 2003.

[5] K. Bhaduri, B. L. Matthews, and C. R. Giannella. Algorithms for speeding up distance-based outlier detection. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 859–867, 2011.

[6] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Computing Surveys*, 41(3):1–58, 2009.

[7] Y. Cong, J. Yuan, and J. Liu. Sparse reconstruction cost for abnormal event detection. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR2011)*, pages 3449–3456, 2011.

[8] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In *Proceedings of the twentieth annual symposium on Computational geometry*, pages 253–262, 2004.

[9] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon. Information-theoretic metric learning. In *Proceedings of the 24th International Conference on Machine Learning (ICML2007)*, pages 209–216, 2007.

[10] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov. Neighbourhood component analysis. In *Advances in Neural Information Processing Systems 17 (NIPS2005)*, pages 513–520, 2005.

[11] T. S. F. Haines and T. Xiang. Delta-dual hierarchical dirichlet processes: A pragmatic abnormal behaviour detector. In *Proceedings of 2011 IEEE International Conference on Computer Vision (ICCV2011)*, pages 2198–2205, 2011.

[12] N. Karmarkar. A new polynomial-time algorithm for linear programming. In *Proceedings of the sixteenth annual ACM symposium on Theory of computing (STOC1984)*, pages 302–211, 1984.

[13] Q. W. Killian and K. S. Lawrence. Distance metric learning for large margin nearest neighbor classification. *The Journal of Machine Learning Research*, 10:207–244, 2009.

[14] J. Kim and K. Grauman. Observe locally, infer globally: a spase-time mrf for detecting abnormal activities with incremental updates. In *Proceedings of 2009 IEEE International Conference on Computer Vision and Pattern Recognition (CVPR2009)*, pages 2921–2928, 2009.

[15] E. M. Knorr and T. N. Raymond. Algorithms for mining distance-based outliers in large datasets. In *Proceedings of the 24th International Conference on Very Large Data Bases (VLDB1998)*, pages 392–403, 1998.

[16] C. Liu, W. Freeman, E. Adelson, and Y. Weiss. Human-assisted motion annotation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR2008)*, pages 1–8, 2008.

[17] F. T. Liu, K. M. Ting, and Z. H. Zhou. Isolation-based anomaly detection. *ACM Transactions on Knowledge DIscovery from Data*, 6(3):1–3, 2012.

[18] V. W. L. Mahadevan, V. Bhalodia, and N. Vasconcelos. Anomaly detection in crowded scenes. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR2010)*, pages 1975–1981, 2010.

[19] S. C. Moses. Similarity estimation techniques from rounding algorithms. In *Proceedings of the thirty-fourth annual ACM symposium on Theory of computing (STOC2002)*, pages 380–388, 2002.

[20] U. of Minesota. Unusual crowd activity dataset of university of minesota. http://mha.cs.umn.edu/movies/crowdactivity-all.avi.

[21] V. Roth. Kernel fisher discriminants for outlier detection. *Neural Computation*, 18(4):942–960, 2006.

[22] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7):1443–1471, 2001.

[23] Y. Wang, S. Parthasarathy, and S. Tatikonda. Locality sensitive outlier detection: A ranking driven approach. In *Proceedings of the IEEE 27th International Conference on Data Engineering (ICDE2011)*, pages 410–421, 2011.

[24] J. Zhang and H. Wang. Detecting outlying subspaces for high-dimensional data: the new task, algorithms, and performance. *Knowledge and Information Systems*, 10(3):333–355, 2006.