Soft Margin Keyframe Comparison: Enhancing Precision of Fraud Detection in Retail Surveillance

Jiyan Pan, Quanfu Fan, Sharath Pankanti, Hoang Trinh, Prasad Gabbur, Sachiko Miyazawa,
IBM T. J. Watson Research Center, 19 Skyline Drive, Hawthorne, NY 10532

Abstract

We propose a novel approach for enhancing precision in a leading video analytics system that detects cashier fraud in grocery stores for loss prevention. While intelligent video analytics has recently become a promising means of loss prevention for retailers, most of the real-world systems suffer from a large number of false alarms, resulting in a significant waste of human labor during manual verification. Our proposed approach starts with the candidate fraudulent events detected by a state-of-the-art system. Such fraudulent events are a set of visually recognized checkout-related activities of the cashier without barcode associations. Instead of conducting costly video analysis, we extract a few keyframes to represent the essence of each candidate fraudulent event, and compare those keyframes to identify whether or not the event is a valid check-out process that involves consistent appearance changes on the lead-in belt, the scan area and the take-away belt. Our approach also performs a margin-based soft classification so that the user could trade off between saving human labor and preserving high recall. Experiments on days of surveillance videos collected from real grocery stores show that our algorithm can save about 50% of human labor while preserving over 90% of true alarms with small computational overhead.

1. Introduction

A large portion of revenue loss in retail is related to employees and directly caused by fraud or error that occurs in and around the point of sale (POS). For instance, when scanning items during a transaction, a cashier may purposely fail to trigger the barcode scanner in an attempt to give free merchandise to a customer. Such an improper behavior, often called "*sweethearting*" in the retail industry, is a collusion between a cashier and a customer who is usually the casher's friend or family member. In other cases, failure to trigger the barcode scanner is due to the cashier's sloppiness or lack of training, and such failures are usually referred to as "*operational errors*". Both sweethearting and operation errors result in otherwise avoidable loss to retailers, and we call both of them "*fraud*" for simplicity in this paper.

Fraud ranks as one of the most serious problems in the retail industry and causes retail shrinkage with over billions of dollars each year worldwide. Recently, video analytics technologies of cashier fraud detection have increasingly received attention by retailers as a promising means of loss prevention. There are several systems commercially available for detecting cashier fraud [15, 1, 11]. One of the state-of-the-art systems proposed in [6, 7] employs a spatio-temporal method to recognize predominant cashier activities relevant to the checkout process, *i.e*. "visual scan". Each visual scan recognized by the system includes three action primitives from the cashier: item pick-up, item scan and item drop-off (see Fig. 1(a)). A visual scan, if out of alignment with any barcode in the transaction log, is flagged as "suspicious" and subject to manual verification.

To ensure high efficiency, simple features (*i.e*. motion from frame differencing) are used to detect action primitives. Consequently, although the system achieves a high detection recall, its precision is low – a large percentage of system-generated alerts turn out to be false alarms, and much human labor is wasted on manually verifying a large number of alerts of which only a small portion are true positives. A majority of false alarms are due to the errors of action primitive detection, which are caused by belt movement, customer interactions and arbitrary cashier movement. Such false action primitives form invalid visual scans, during which the barcode reader is indeed *not* supposed to be triggered.

To significantly save human labor while maintaining high detection recall, we propose an approach to validate candidate visual scans detected by the system and filter out those that are less likely to be true checkout processes. Instead of directly analyzing the original video, our algorithm focuses on keyframes that capture the essence of action primitives in each candidate visual scan. This is inspired by the fact that humans can quickly eliminate false alarms by simply examining the keyframes without navigating into the video. Based on the keyframes, we develop an effective

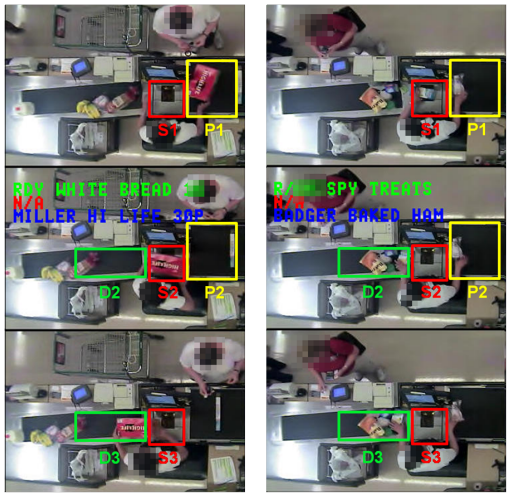
validation criterion by comparing regions of interest (ROIs) of the keyframes after handling complications such as belt movement and local appearance change using augmented motion compensation and max-pooling of sub-block differences. Further inseparability is handled by taking a margin-based approach using learned conservative thresholds, and a soft classification provides users with the flexibility to balance between saving human labor and retaining high recall. Experimental results have shown that our algorithm enhanced detection precision by up to 20%, and halved human labor in manual verification while over 90% of true alarms are preserved, with little computational overhead.

The remainder of this paper is organized as follows. In Section 2 , we briefly review the visual scan detection method that detects and organizes action primitives [6, 7]. Section 3 details our proposed visual scan validation algorithm using soft margin keyframe comparison. Experimental results are given in Section 4, and Section 5 concludes this paper.

2. Visual Scan Detection

The predominant cashier activity during a transaction is characterized by a sequence of repetitive *i.e.* visual scan events, each of which consists of three basic actions (primitives) in sequence: pickup, scan and drop (See Figure 1(a). Based on this observation, a spatiotemporal approach was proposed in [6, 7] to recognize visual scans. The approach identifies checkout-related primitives using the bag of features model (BOF) based on Space-Time-Interest Points (STIP) [12] and histograms of optical flow [14, 4]. The features are fed into an SVM classifier with Multiple-Instance Learning (MIL) [2]. A specialized Hidden Markov Model (HMM) model [3] that considers the strong temporal dependencies between the primitives is then applied to optimally group the primitives into a sequence of visual scans using a specialized Viterbi algorithm. The integrated visual scans are further aligned with transaction data in time to flag suspicious scan activity in a transaction. Due to the limit in space, interested readers please refer to [6, 7] for more details about the algorithm.

While the approach described above has demonstrated good performance in detecting cashier fraud, the computationally expensive STIP features have greatly limited further application of this approach to fraud detection in the real world. As a compromise, the system employs a more efficient method to detect primitives based on thresholding motion energy at pre-specified ROIs. However, a significant increase of false alarms has been observed due to the errors generated by the less accurate primitive detectors. In what follows, we will describe our approach of reducing false alarms by analyzing event keyframes extracted from video.



(a) (b)

Figure 1. (a) An example of a valid visual scan. (b) An example of an invalid visual scan. The three images from the top are the pickup, scan, and drop keyframes, respectively. The rectangular boxes indicate the ROI pairs to compare.

3. Visual Scan Validation

3.1. Keyframe Representation of Visual Scan

In order to achieve a real-time performance, we do not resort to the original video to validate visual scans. Instead, we extract a keyframe for each detected action primitive. More specifically, the frame located in the middle of the duration of an action primitive is selected as the keyframe for that action primitive. As a result, each candidate visual scan consists of three keyframes corresponding to pickup, scan, and drop, respectively, as is shown in Figure 1. We only use the three keyframes to determine the validity of a candidate visual scan, resulting in an efficient real-time algorithm.

3.2. Comparing ROI Pair

Humans can immediately tell whether three keyframes constitute a valid visual scan simply by looking at the pickup, scan, and drop areas. Intuitively, if a visual scan is valid, then both the pickup area and the scan area should undergo major appearance change between the pickup keyframe and the scan keyframe; similarly, both the scan area and drop area should undergo major appearance change between the scan keyframe and the drop keyframe. This is illustrated in Figure 1(a), where the three images from the top are the pickup, scan, and drop keyframes, respectively. Note that *all* the four ROI (region of interest) pairs (*i.e.* P_1 vs. P_2, S_1 vs. S_2, S_2 vs. S_3, D_2 vs. D_3) should have major appearance change before the visual scan can be determined as valid. If any of the four ROI pairs does not meet this requirement, the visual scan is invalid, as is shown in Figure 1(b).

Before directly comparing a ROI pair, we first need to deal with the belt movement in the pickup and drop areas. Even if there is no pickup or drop action, there could still be large appearance change if the belt has moved between two keyframes. In order to remove the influence of belt movement, we perform motion estimation before comparing the appearance of an ROI pair in the pickup or drop region.

Let us use the pickup region as an example. The location of the pickup ROI is selected by user and remain fixed in the scan keyframe, as is illustrated by the solid yellow rectangle $A_2B_2C_2D_2$ in the middle image of Figure 2. Using the pickup ROI in the scan keyframe as a template, we search for the most similar image patch in the pickup region of the pickup keyframe (*i.e.* the top image of Figure 2). In the example shown in Figure 2, the best match is $A_1B'_1C'_1D_1$. Note that the best match might be smaller than the ROI in the scan keyframe. If this occurs, the corresponding portion of the ROI that does not appear in the best match is cropped and not considered for comparison. In this example, region $B'_2B_2C_2C'_2$ is cropped away.

However, if we simply compare $A_1B'_1C'_1D_1$ with $A_2B'_2C'_2D_2$, we still cannot capture the appearance difference caused by the cashier's hand in region $A_1D_1E_1F_1$. Therefore, we need to augment the image patch of the pickup keyframe with region $A_1D_1E_1F_1$, and augment the image patch of the scan keyframe with the image of a clear belt (which has been rolled underneath the table in the scan keyframe) as is shown by $A_2D_2E_0F_0$. The final image patches to compare are $B'_1C'_1E_1F_1$ and $B'_2C'_2E_0F_0$.

Similarly, for the drop region, region $G_2J_2I_2H_2$ is selected by user and remain fixed in the scan keyframe, and its best match is searched in the drop area of the drop keyframe. The image patches are augmented the same way as for the pickup region, and the final image patches to compare are $H_2I_2K_0L_0$ and $H_3I_3K_3L_3$.

To make use of the prior knowledge of the direction of belt movement, we only search to the right of the pickup ROI and to the left of the drop ROI. A limited search range is applied vertically.

As the scan region does not move, no motion compensation is performed. The image patch in the scan ROI of the scan keyframe is paired with the image patch at exactly the same location in the pickup keyframe and the drop keyframe, respectively, forming another two pairs of image patches to compare. They are illustrated by the red squares in Figure 2.

To compare two image patches, we divide each patch into a grid of sub-blocks, and difference is computed for each pair of corresponding sub-blocks. The maximum difference over all pairs of sub-blocks is taken as the difference between the two image patches. The reason behind max-pooling of sub-block differences is that in many cases the items are small and they only cause local appearance

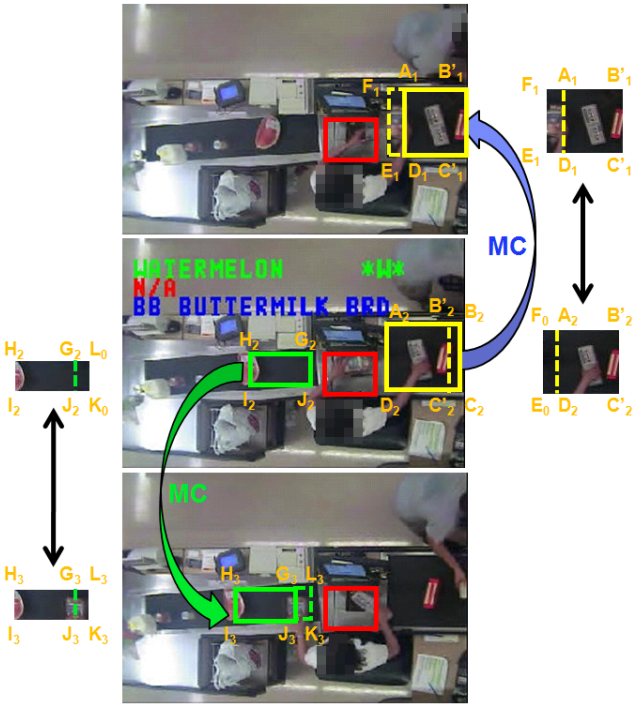


Figure 2. An illustration of generating image patches to compare for the pickup and drop regions. Please see the text for details. (Best viewed in color.)

change in the ROI. If difference is averaged over the entire ROI, such local appearance changes would not be captured.

When computing the difference between a pair of corresponding sub-blocks, we simply compute the pixel-wise mean absolute difference. We do not adopt a histogram-based approach as is proposed in [13, 8, 5, 10, 9] since the image patches to compare have already been registered by motion compensation.

3.3. Generating Threshold-based Features

After comparing ROI pairs, we obtain 4 appearance differences, denoted as $d(P_1, P_2)$, $d(S_1, S_2)$, $d(S_2, S_3)$, and $d(D_2, D_3)$. A naive approach to determine the validity of a visual scan would be directly feeding the appearance differences into a binary classifier such as SVM. However, this approach turns out to yield poor performance. Figure 3 shows a visualization of the distribution of those 4-dimensional data points by projecting them onto several 2-D planes. As we can see, although valid visual scans (red dots) generally have higher appearance differences than invalid visual scans (blue dots), they are highly *inseparable*. How could this happen? This is caused by the fact that some *invalid* visual scans also have large appearance differences for *all* the four ROI pairs. One of the most apparent cases is passing a non-merchandise item (such as a shopping basket). A less apparent example is shown in Figure 4, where the movement of cashier's idle hand results in large

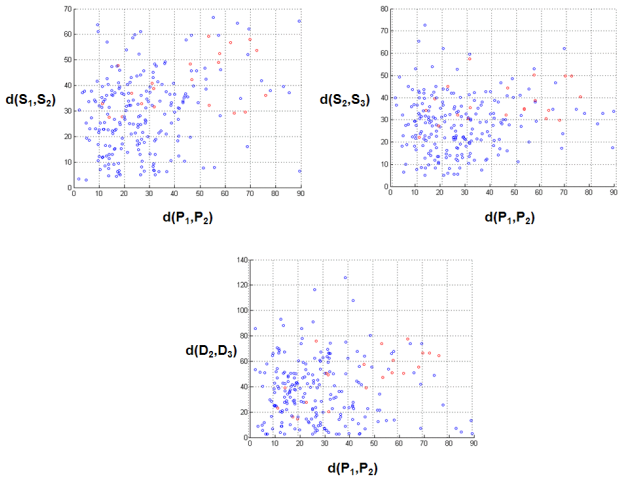


Figure 3. Visualization of data point distribution for valid visual scans and invalid visual scans. The red and blue dots indicate valid and invalid visual scans, respectively. (Best viewed in color.)

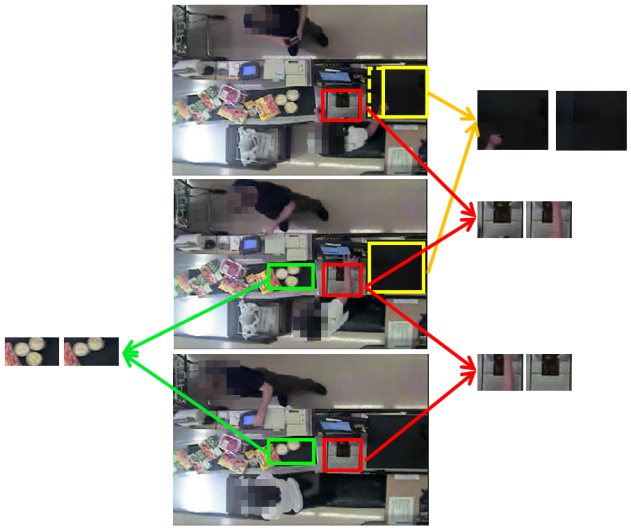


Figure 4. An example of an invalid visual scan which has large appearance differences for all the four ROI pairs.

$d(P_1, P_2)$, $d(S_1, S_2)$, and $d(S_2, S_3)$, while an item taken away in the drop area leads to a large $d(D_2, D_3)$. More sophisticated object appearance models and recognition algorithms could be applied to further disambiguate these cases, yet they are not feasible for real-time applications.

In order to overcome this inseparability problem while maintaining real-time performance, we do not directly classify those data points. Rather, we take a conservative approach which preserves all valid visual scans in the training set. More specifically, we compute four thresholds

$$t_1 = \min_{i \in T} d_i(P_1, P_2), \quad t_2 = \min_{i \in T} d_i(S_1, S_2),$$
$$t_3 = \min_{i \in T} d_i(S_2, S_3), \quad t_4 = \min_{i \in T} d_i(D_2, D_3), \tag{1}$$

where T is the training set. A test visual scan is determined as valid only when the appearance differences of *all* the four ROI pairs are larger than their corresponding thresholds. As many invalid visual scans do not satisfy this criterion, they are discarded without being checked by the user.

3.4. Soft Classification Using Margin

Although the thresholds obtained in Equation 1 guarantee an 100% recall on the training set, some test data points which are valid visual scans might still fall below one of the four thresholds and are mistakenly discarded. It would be more desirable if a soft classification is enabled where each detected visual scan is given a score and the user controls the number of preserved visual scans (i.e. those classified as being valid), and thus the amount of human labor, according to those scores.

To soften the classification, we first need to compute the margin of each detected visual scan with respect to the thresholds in Equation 1. The margins associated with the thresholds t_1 through t_4 are $m_1 = d(P_1, P_2) - t_1$, $m_2 = d(S_1, S_2) - t_2$, $m_3 = d(S_2, S_3) - t_3$, and $m_4 = d(D_2, D_3) - t_4$. Naively, to obtain a soft classification, we could simply train a logistic regression model [3] directly using m_1 through m_4 as its covariates. However, what really matters is the "worst-case" margin. For example, a candidate visual scan is invalid as long as one margin is negative; the other three margins are irrelevant. Incorporating all the four margins would introduce noise and therefore adversely affect the classification performance. To compute the worst-case margin, we should treat two different cases separately.

Firstly, if $\forall p \in 1, 2, 3, 4, \ m_p \geq 0$ for an example, then this example will be classified as a valid visual scan according to Equation 1. Therefore, the margin M for this example (*i.e.* the worst-case margin) is the smallest margin among m_1 through m_4:

$$M = \min_{p \in \{1,2,3,4\}} m_p \tag{2}$$

On the other hand, if $\exists \ p \in 1, 2, 3, 4, m_p < 0$ for an example, then this example will be classified as an invalid visual scan according to Equation 1. Therefore, the margin M for this example (*i.e.* the worst-case margin) is the largest *negative* margin:

$$M = \max_{p \in \{1,2,3,4\}, m_p < 0} m_p \tag{3}$$

These two cases are illustrated in Figure 5, where the left figure shows the margin for a valid visual scan and the right figure for an invalid visual scan. Note that in the right figure, although $d(P_1, P_2)$ has the smallest absolute margin, it is not meaningful for an invalid visual scan, as the margin of such examples should be defined from the negative side.

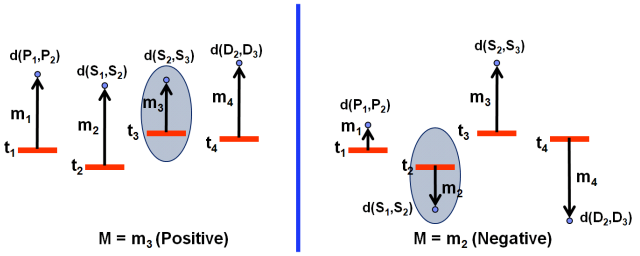


Figure 5. An illustration of computing the margin of a detected visual scan in two different cases. The ellipses indicate the selected margin.

Having obtained the margin of each training example, now we could soften the classification by learning a logistic regression model [3] over the margin to compute the validity scores of the examples. During training, the label value is 0 for invalid visual scans and 1 for valid ones. This learning task can be formulated as

$$(w^*, b^*) = \arg\min_{w,b} \prod_{i \in T} \left(\frac{e^{wM_i+b}}{e^{wM_i+b}+1}\right)^{y_i} \left(\frac{1}{e^{wM_i+b}+1}\right)^{1-y_i} + \lambda(w^2+b^2), \quad (4)$$

where w and b are the parameters in the logistic regression model, y_i is the label for training example i, and λ is the ridge regularization parameter.

For a testing example, its margin is computed exactly the same way as a training example — not knowing its ground-truth label does not pose any problem. Its score is then computed by the learned logistic regression model:

$$S = \frac{e^{w^*M+b^*}}{e^{w^*M+b^*}+1}, \quad (5)$$

where M and S are the margin and score of the testing example, respectively.

An example of the mapping from margin to score is illustrated in Figure 6, where both the training and testing examples are shown. The black curve is the mapping learned from the training examples. During testing, the user selects a score threshold, as is indicated by the dashed purple line in Figure 6. All the testing examples whose scores are higher than the score threshold (*i.e.* the testing points to the right of the solid purple line in Figure 6) are classified as valid visual scans. We can see that increasing the score threshold leads to a higher precision at the risk of a lower recall.

In order to give the user a reference, the algorithm provides a reference S_0 score which corresponds to zero margin:

$$S_0 = \frac{e^{b^*}}{e^{b^*}+1}. \quad (6)$$

In Figure 6, the reference score and its associated decision boundary are illustrated by the dashed and solid green lines, respectively.

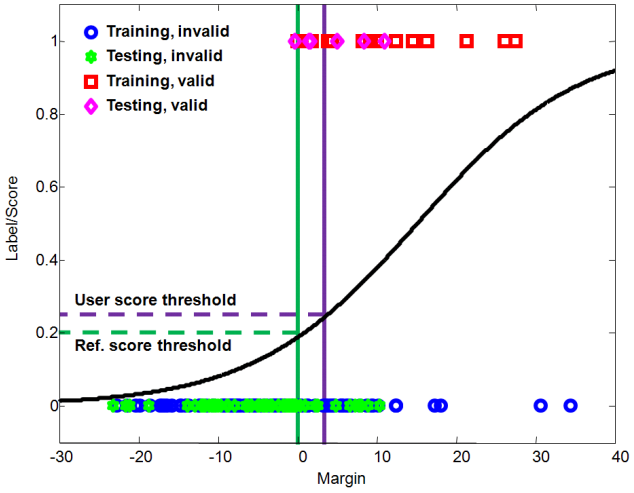


Figure 6. An example of mapping margins to scores, as well as an illustration of the reference and user score thresholds. (Best viewed in color.)

Experiment 1	Counter	Total #	Valid #	Prec.
Train	A	174	35	0.20
Test	A	175	37	0.21
Experiment 2	Counter	Total #	Valid #	Prec.
Train	B	303	34	0.11
Test	A	349	72	0.21

Figure 7. Description of the training and testing data sets in two experiments. Please see text for details.

4. Experimental Results

The effectiveness of our algorithm is examined using retail surveillance videos collected from real grocery stores.

4.1. Generalization Over Cashiers

Dataset description. We first evaluate our algorithm on the surveillance video taken over the same checkout counter yet containing multiple cashiers. The video sequence captures the activities of four different cashiers working at different shifts over an entire day. The details of the data in this experiment are listed in the upper half of Figure 7, where "Total # "means the number of candidate visual scans returned by the visual scan detection algorithm described in Section 2. No barcode reading is present in any of those candidate visual scans."Valid # "means the ground-truth number of valid visual scans. "Prec."means the precision of the detection algorithm. Note that the two cashiers in the testing set never appear in the training set. Also note that the ground-truth labels for training are given beforehand.

Precision-recall curve. In our proposed visual scan validation algorithm, increasing the score threshold results in

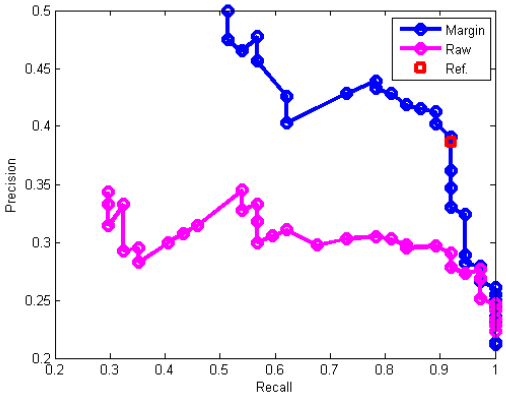


Figure 8. Precision-recall curve for experiment 1. The blue curve shows the result using the worst-case margin. The red square indicates the working point when the reference score threshold is used. The magenta curve displays the result using the raw margins. The same representation convention applies to all the other plots that follow.

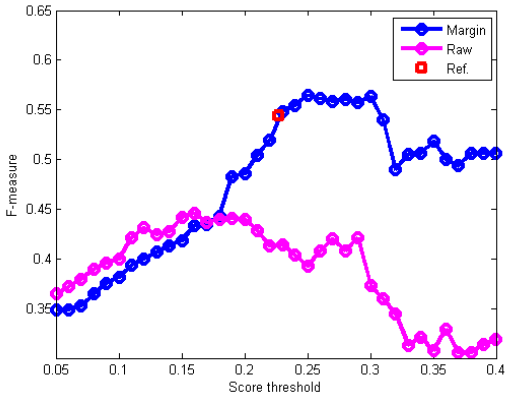


Figure 9. Relation between F-measure and score threshold for experiment 1.

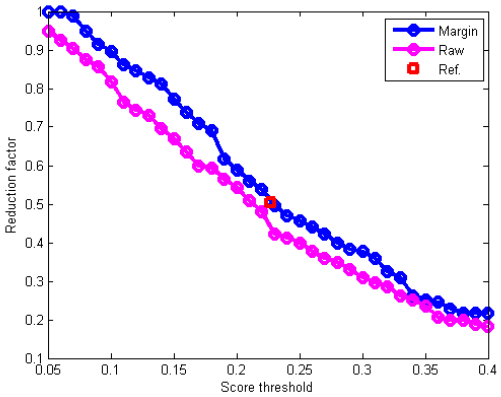


Figure 10. Relation between reduction factor and score threshold for experiment 1.

fewer visual scans that pass the validation, and therefore less human labor to manually check if fraud indeed occurs during those visual scans. In the meantime, a higher score threshold leads to a greater risk of missing visual scans that are actually valid. In other words, the score threshold trades off between precision and recall. The precision-recall curve of the testing set is shown in Figure 8, where the blue and magenta curves show the results when using the worst-case margin M and the raw margins m_1 through m_4, respectively. The red square indicates the working point when the automatically-generated reference score threshold (which is 0.2271) is used under the worst-case margin. This representation convention applies to all the remaining plots in this paper. We could see that using the worst-case margin well outperforms the raw margins.

F-measure. Figure 9 shows the F-measure (defined as the harmonic mean of precision and recall) as a function of the score threshold. As we can see, the reference score threshold gives the near-optimal F-measure. Worst-case margin achieves much higher F-measures than raw margins under most score thresholds.

Reduction in human labor. The reduction in human labor can be measured by the reduction factor, defined as the ratio of the number of the visual scans output by the visual scan validation algorithm, to the number of the visual scans generated by the visual scan detection algorithm. The relation between the reduction factor and the score threshold is displayed in Figure 10. We can see that using the reference score threshold reduces human labor by half. The fact that using the raw margins retrieve fewer visual scans under the same score threshold is meaningless, since it does not reflect anything related to accuracy.

Recall-reduction curve. The effectiveness of our proposed algorithm is best demonstrated by the relation between recall and the reduction factor, which is shown by the blue curve in Figure 11. The green dashed line along the diagonal depicts the performance of randomly picking candidate visual scans as valid ones — in this case, recall is always equal to the reduction factor. The higher the actual curve over the green dashed line, the more effective the algorithm is. From the figure, we can see that our proposed algorithm performs way better than chance. We can also see that using the worst-case margin yields consistently better performance than using the raw margins.

Numerical results. The numerical results of maintaining a 90% recall are shown in the left part of Figure 12. Here, "Prec.", "Reca". and "Redu". are the abbreviations of precision, recall, and reduction factor, respectively. The columns titled "Original ", "SMKC ", and "Chance "show the statistics of the original data set, the retrieved data set by our proposed algorithm, and by random selection, respectively. We can see that our algorithm almost doubles the precision and reduces human labor to 46% while keeping a 90% recall.

The results mentioned above indicate that our algorithm generalizes well to different cashiers.

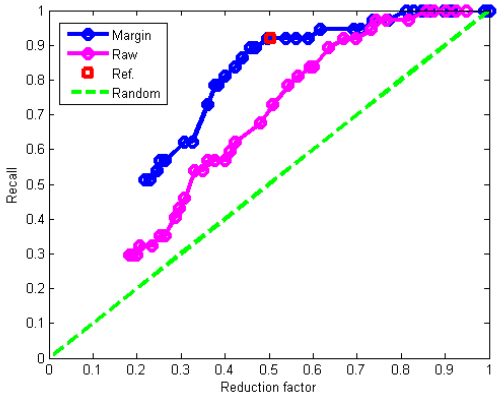


Figure 11. Relation between recall and reduction factor for experiment 1. The green dashed line depicts the performance when valid visual scans are picked randomly.

	Experiment 1			Experiment 2		
	Original	SMKC	Chance	Original	SMKC	Chance
Prec.	0.21	0.40	0.21	0.21	0.34	0.21
Reca.	1.00	0.90	0.90	1.00	0.90	0.90
Redu.	1.00	0.46	0.90	1.00	0.55	0.90

Figure 12. Numerical results of our proposed algorithm under a 90% recall. Please see text for details.

4.2. Generalization Over Cashiers and Counters

Dataset description. To see how well our algorithm generalizes to even greater scenario differences, we trained it using a video taken at one checkout counter, and tested it on a video taken at another checkout counter. Now both the counter layout and the cashiers are different between training and testing sets. An example is shown in Figure 13. The training and testing data sets are described in the lower part of Figure 7. Note that the testing data set is very different from the training one also in terms of precision.

P-R curve and F-measure. The precision-recall curve and the F-measure - threshold curve are displayed in Figures 14 and 15, respectively. We can see that the maximum F-measure is only about 5% lower compared to the same-counter case, although the layout of checkout counters are disparate. When the reference score threshold (0.1420) is used, precision is enhanced by over 10% while recall is still close to 100%. The performance using the raw margins is much worse.

Reduction in human labor and reduction-recall curve. Figures 16 and 17 show the reduction-factor - threshold curve and the recall - reduction-factor curve. Again, the performance of our algorithm is significantly better than chance — about 45% of reduction in human labor can be achieved while 90% of valid visual scans are preserved. By contrast, the performance degenerates to nearly chance if the raw margins are used. From Figure 17, we could see that if the reference score threshold is used, over 30% of human labor is saved while almost all valid visual scans are captured.

(a) (b)

Figure 13. Training and testing sets have different cashiers and counter layouts. (a) A training example. (b) A testing example.

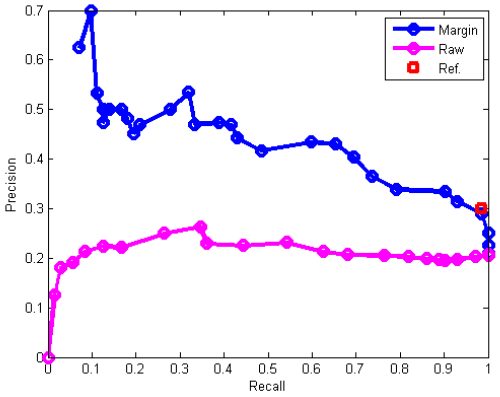


Figure 14. Precision-recall curve for experiment 2.

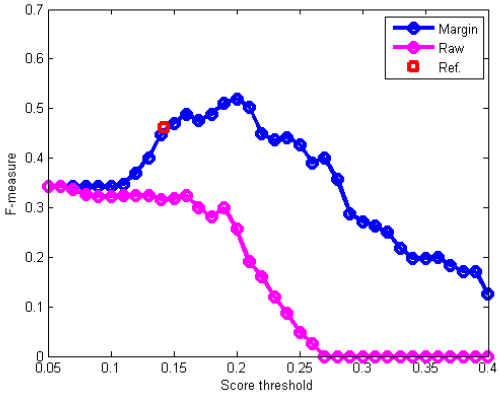


Figure 15. Relation between F-measure and score threshold for experiment 2.

Numerical results. The numerical results of maintaining a 90% recall are shown in the right part of Figure 12.

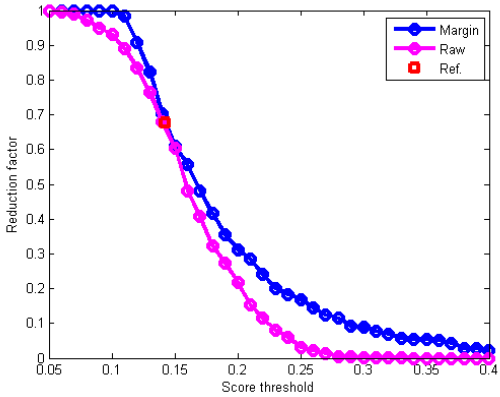


Figure 16. Relation between reduction factor and score threshold for experiment 2.

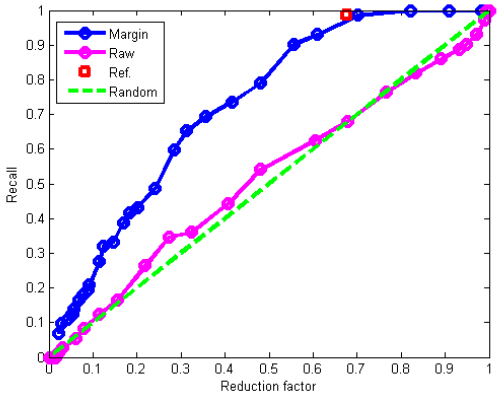


Figure 17. Relation between recall and reduction factor for experiment 2. The green dashed line depicts the performance when valid visual scans are picked randomly.

Again, precision is enhanced by 14%, and human labor is almost halved.

The results show that our algorithm generalizes well to both different cashiers and different checkout counters.

5. Conclusion

In this paper, we propose an effective visual scan validation algorithm to enhance the precision of visual scan detection in retail surveillance applications. The algorithm validates candidate visual scans by comparing ROIs of the keyframes associated with the candidate visual scans. Belt movement and local appearance change are handled by augmented motion compensation and max-pooling of sub-block differences. Furthermore, the problem of inseparability caused by spurious motions are solved by computing worst-case margins with respect to learned conservative thresholds and training a logistic regression model for soft classification based on the worst-case margins. Our proposed algorithm significantly increases the precision of visual scan detection, and therefore considerably reduces human labor, in real-world surveillance videos where both cashiers and checkout counters have large variations.

As future work, we plan to further analyze the distribution of feature vectors and develop a semi-supervised learning scheme to relieve human labeling efforts.

References

[1] Agilence. http://www.agilenceinc.com/.

[2] S. Andrews, T. Hofmann, and I. Tsochantaridis. Multiple instance learning with generalized support vector machines. *Artificial Intelligence*, 2002.

[3] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[4] R. Chaudhry, A. Ravichandran, G. Hager, and R. Vidal. Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human actions. *CVPR*, 2009.

[5] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. *CVPR*, 1:886–893, 2005.

[6] Q. Fan, R. Bobbitt, Y. Zhai, A. Yanagawa, S. Pankanti, and A. Hampapur. Recognition of repetitive sequential human activity. *CVPR*, 2009.

[7] Q. Fan, A. Yanagawa, R. Bobbitt, Y. Zhai, R. Kjeldsen, S. Pankanti, and A. Hampapur. Detecting sweethearting in retail surveillance videos. *ICASSP*, 2009.

[8] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. *CVPR*, 2008.

[9] C. Galleguillos, B. McFee, S. Belongie, and G. Lanckriet. Multi-class object localization by combining local contextual interactions. *CVPR*, 2010.

[10] N. He, J. Cao, and L. Song. Scale space histogram of oriented gradients for human detection. *International Symposium on Information Science and Engieering*, 2:167–170, 2008.

[11] Intellivid. http://www.americandynamics.net/.

[12] I. Laptev and T. Lindeberg. Space-time interest points. *ICCV*, 2003.

[13] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[14] J. Pers, V. Sulic, M. Kristan, M. Perse, K. Polanec, and S. Kovacic. Histograms of optical flow for efficient representation of body motion. *Pattern Recognition Letters*, 2010.

[15] StopLift. http://www.stoplift.com/.