Home Surveillance System using Computer Vision and Convolutional Neural Network

Xin Zhang, Won-Jae Yi and Jafar Saniie

Embedded Computing and Signal Processing (ECASP) Research Laboratory (http://ecasp.ece.iit.edu)
Department of Electrical and Computer Engineering
Illinois Institute of Technology, Chicago, Illinois, U.S.A.

***Abstract*—In this paper, we introduce a home surveillance system that utilizes computer vision techniques to recognize intrusions and detailed threat information including classifying types of trespassers and specific weapons used. Process of identifying intrusions is achieved by our Smart Intruder Detection and Surveillance System (SIDSS) which involves 3 stages of computer vision algorithms. 3-stage SIDSS includes an optimized convolutional neural network (CNN) for threat and intrusion detections, cascading classifiers for locating any potential intruders with correcting mechanism to overcome undetected threats from the previous stage, and principal component analysis (PCA) to efficiently train the facial recognizer to accurately differentiate passersby from potential intruders. Our system is scalable for various surveillance events and can be expanded with additional pre-processed datasets added to the SIDSS model to manage greater surveillance areas. Through this enhanced configuration, the system can achieve enhanced accuracy of recognizing broad range of weapons used and intruders.**

I. Introduction

Home surveillance has been an important factor for years in our daily activities. Traditional surveillance techniques have either used wireless sensor networks [1] or closed-circuit televisions (CCTV) [2] where they required individuals to monitor live camera feeds. Recently, advances in surveillance systems were made by the Internet of Things (IoT) devices with intelligence and robustness that can be adopted to the home surveillance systems, which can be scalable by integrating spatially distributed sensors (e.g., temperature sensors, humidity sensors, smoke detectors, cameras and more) to analyze collected data from the physical world for robust decisions.

Computer vision is a widely used method to analyze images/videos for automation applications. Advanced computing techniques such as k-nearest neighbors algorithm [3] and principal component analysis (PCA) [4] can be used in many computer vision applications. Some common applications are facial recognition and scene recognition [4][5]. By distinguishing different objects using these techniques and applications, machines can interact with their environment to serve different needs. For instance, an embedded system with computer vision can detect intruders using recordings from surveillance cameras and warn users.

In this paper, we introduce a smart home surveillance system that utilizes computer vision techniques to identify intruders, their weapons used and seek for any potential trespassers. Common surveillance systems deploy motion sensor [6] or facial detection system [7] based on the IoT scheme. These systems are prone to false alarms, for example, using only motion sensor, strong winds or animals may mistakenly trigger an alarm. Conversely, a surveillance system using only facial recognition application, intruders' faces may not be detected accurately due to different head postures and other factors. Therefore, for robust home surveillance, we introduce Smart Intruder Detection and Surveillance System (SIDSS), a 3-stage computer vision system, to accurately identify intruders and monitor home environment. This 3-stage SIDSS involves multiple layers of intrusion detection which firstly executes an optimized convolutional neural network (CNN) for threat and intruder detections, secondly cascading classifiers for locating potential trespassers that weren't detected in the previous stage, and lastly principal component analysis (PCA) for facial recognition for accurate identification.

II. 3-Stage Smart Intruder Detection and Surveillance System

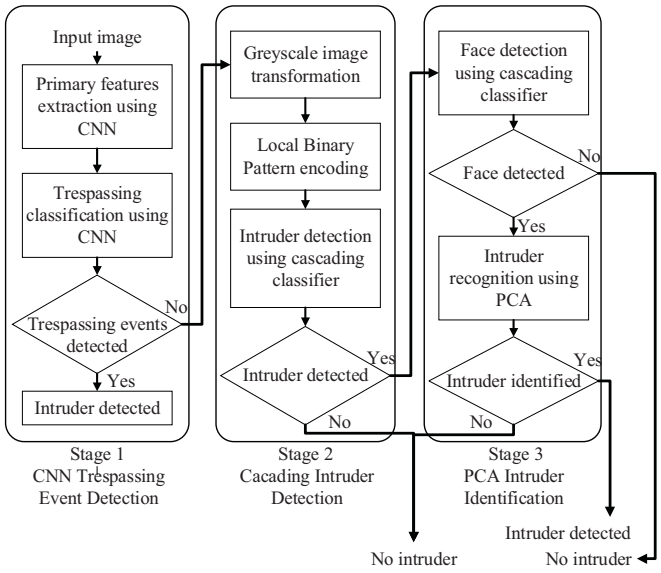

Figure 1. Three-Stage SIDSS Design Flow

Our design flow of the 3-stage SIDSS is shown in Figure 1. In Stage 1, our trained CNN extracts primary features of intrusions and threats from the surveillance camera to identify trespassing events. Compared with other machine learning algorithms such as Support Vector Machine (SVM) [8], the

978-1-7281-0927-5/19/$31.00 ©2019 IEEE

CNN has higher detection accuracy once trained on large datasets. In case if the Stage 1 was unable to detect any intruder, our system utilizes cascading classifier to further detect any potential intruders in Stage 2. At this stage, the algorithm transforms images to greyscale and uses Local Binary Pattern (LBP) algorithm for fast identification. The cascading classifier is used to detect any undetected intruder from the previous stage. If Stage 2 detected any intruder, trained cascading face detector and recognizer can identify intruders using principal component analysis (PCA) in Stage 3.

III. Convolutional Neural Network for Trespassing Detection

In Stage 1 of SIDSS, we trained a multilayer convolutional neural network (CNN) with real surveillance images of large datasets to predict multiple suspicious objects. For training, we created new datasets for SIDSS by collecting approximately 3,000 surveillance images for each category with size of 96x96 pixels. Per category of datasets include Handgun Dataset [9], Knife Images Dataset [10] and INRIA Person Dataset [11]. Datasets have been classified into 5 categories corresponding to 5 types of trespassing events. In order to fully utilize our new datasets, we applied 10-fold cross-validation [12] for training the dataset. Meanwhile, data augmentation method, such as image rotation, image height or width shift, is used to enrich the datasets to avoid overfitting. Our trained CNN achieves 97.12% detection accuracy for the testing dataset. The training is implemented on an NVIDIA Quadro 4000 GPU processing in parallel with 8 logical CPU cores for computation to increase the training speed. Specific structure of our proposed neural network is shown in Figure 2.

Stage 1, using CNN, can predict 5 types of trespassing events: an intruder with the gun, an intruder with the knife, an intruder in the burglar mask, an intruder with no weapon, and no suspicious trespassing activity. Each trespassing event is predicted with scores of possibilities which indicate how confident the neural network is about its predictions. If the prediction score of a suspicious activity is above 50%, it indicates that our CNN determines the existence of the suspicious activity. Any suspicious subject or object detected with the top two highest scores are combined to predict a trespassing event, as shown in the output image of Figure 2.

Our CNN structure is composed of 4 blocks. Blocks from 1 to 3 are convolutional layers and pooling layers to extract the primary features of each input image. Block 4 is the hidden and output layer to analyze the extracted features from Blocks 1 to 3 for final classifications. In Block 1 of our CNN, there are 32 convolutional filters in size of 3x3 pixels. Each filter is the feature detector to extract certain features, such as edges, from the input image. Rectified Linear Units (ReLU) function is applied to the convolutional layer to reduce the linearity, efficiently utilizing the non-linear structure of the neural network to extract image features [13]. Then, we further optimize our neural network by using batch normalization, which gives improvement by almost factor of 10 in our training speed and reduces the covariate shift for the convolutional layer. With the normalized convolutional layer, we apply the max pooling to greatly reduce the parameters that need to be trained to avoid overfitting while maintaining important features of input images, enhancing the spatial invariance of our CNN.

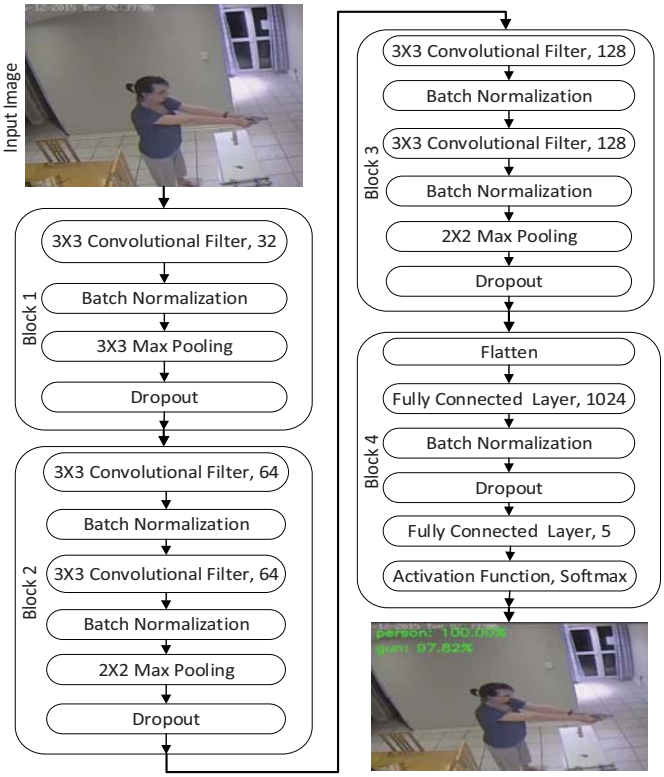

Figure 2. Flow Chart of Trespasser Detection using CNN

Lastly, dropout is used for our CNN with dropping rate of 0.25 to further avoid overfitting. Similarly, in Block 2 and Block 3 of our CNN, features from input images are extracted more specifically using more convolutional filters and then optimized with batch normalization, max pooling and dropout. In Block 4, feature maps from Block 3 are flattened into a high-dimensional vector as the input layer. Then, they are densely connected with the hidden layer of 1024 neurons. For the output layer, we use 5 neurons as there are 5 classes in our datasets and use Softmax as the activation function [14]. Our CNN is trained with epoch of 75 and 30 batches for each epoch for the datasets. We use the cross-entropy loss [15] and Adam optimizer to update weights, which outperforms other stochastic optimization methods [16].

By training our datasets with 96×96×3 images, CNN can extract features, such as eyes and faces, from each layer and detect trespassing objects in various conditions. Figure 3 shows the training loss, validation loss, training accuracy and validation accuracy of the trained CNN, which are 5.92%, 8.68%, 97.73%, 97.12% respectively for the final epoch. The CNN is trained by datasets with 75 epochs and 30 batches for each epoch. The accuracy is measured by the number of correct predictions divided by the total number of predictions. The loss is measured by the cross-entropy loss function, which outperforms other error loss functions (e.g., mean square error) for better performance [17]. Our trained CNN achieves higher accuracy of 97.12% to detect multiple trespassing objects, compared to 75% for Multi SVM algorithm and 90.64% for LMKNCN algorithm [8].

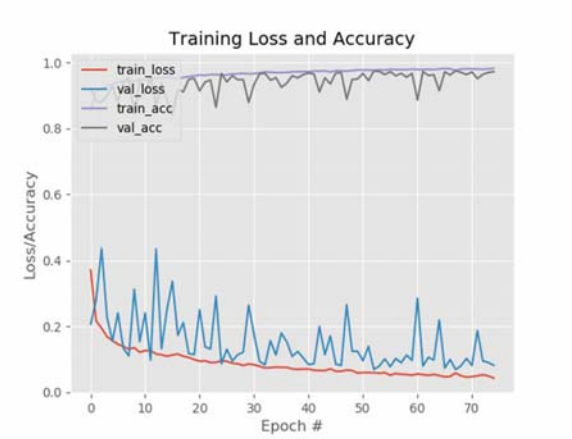

Figure 3. Training Loss and Accuracy of CNN

Figure 4 shows examples outputs from our trained CNN which can detect potential trespassing activities in complex backgrounds. Scores of possibilities for the 5 suspicious trespassing objects and subjects are predicted and two trespassing objects with the highest scores are shown in each output image. For instance, in Figure 4(b), scores of possibilities for intruders, guns, knives, burglar masks, no trespassing object are 100.00%, 99.95%, 45.23%, 71.64%, 4.35%, respectively, indicating that a trespassing event of an intruder with a gun. Figure 4(c) shows a trespassing activity of an intruder with a knife detected and Figure 4(e) shows the result of surveillance detection if there are no trespassers. The "none" indicates no detection results when no suspicious trespassing activities nor objects have been detected.

Figure 4. Intrusion Detection in Complex Backgrounds using CNN

IV. Cascading Classifier for Intruder Detection

In Stage 2 of SIDSS system, after using CNN for the intruder detection and to identify any undistinguished intruders, we detect the number of potential intruders and their locations by training a cascading detector. We achieve this with the HDA Person Dataset [18], which includes approximately 75,000 passersby images with 85 different views of passersby with dissimilar poses and clothes. Figure 5 is the flowchart of the passersby detection using cascading method.

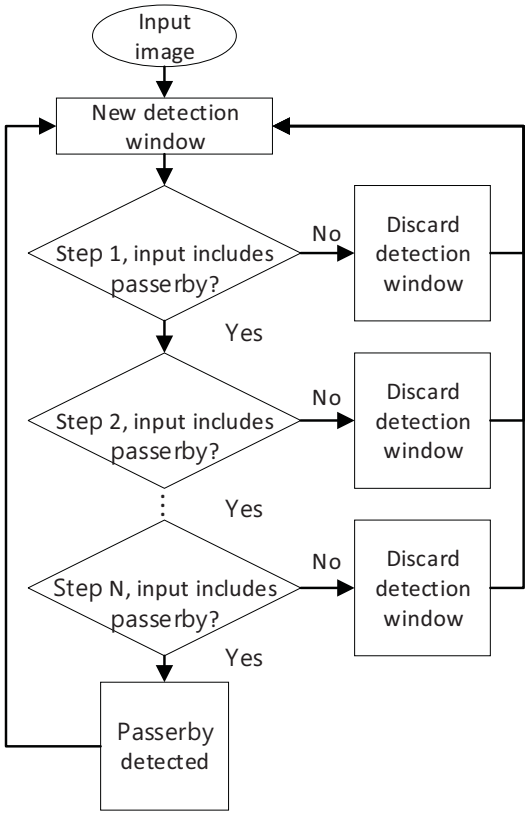

Figure 5. Passersby Detection using Cascading Method

This potential intruder or passersby detector is built on grey-scale images and can classify a person from a complex background and predict one's position. Most traditional methods for passersby detection are based on color images, which increases data complexity, reduces the processing speed and detection accuracy [19]. At first, the classifiers utilize $LBP^{u2}_{(8,2)}$ (Local Binary Pattern) descriptor [20] for extracting passerby features. Then, they are trained with Gaussian distribution for finding the threshold. In the training, $LBP^{u2}_{(8,2)}$ descriptor divides input image into different sub-regions and encodes each pixel of the sub-region. The encoded pixel value represents the local pattern including edges or corners. A high-dimensional vector extracted from LBP encoded sub-region of sample image is compared with vector of the template image, which is trained by Gaussian distribution to obtain the threshold. Therefore, we train different classifiers based on various sub-regions of the sample images. However, these classifiers perform passersby detection with comparatively low accuracy. Thus, we apply the AdaBoost algorithm [21], which iteratively ranks the weak classifiers based on performance, and combine

the best weak classifiers in a cascading process into a strong classifier, resulting detection accuracy enhancement. In passersby detection, within each detection window, there is countless background information that degrades the detection process. To overcome this issue, the cascading process trains the strong classifier in many stages which each stage contains many weak classifiers. The process by using cascading method for detecting passersby is shown in Figure 5.

Figure 6 shows that our cascading classifier can detect passersby and their locations in different conditions, such as various views of the passersby and multiple passersby in complex backgrounds. We executed our trained detector on a computer with the Intel Core i7-3537U CPU at 640x480 resolution. The processing time for detecting every passerby and one's position was 8.3ms, which is approximately 18 times faster than the Intersection Kernel SVM passerby detector [22] and approximately 96 times faster than the linear SVM passerby detector [8]. Meanwhile, our passersby detector achieves 93.47% detection accuracy which outperforms the linear SVM passersby detector with accuracy of 87.46% [8].

Figure 6. Passersby Detection in Different Complex Backgrounds

V. Principal Component Analysis for Intruder Identification

In Stage 3 of SIDSS, we trained a facial recognizer to identify intruders for more detailed recognition. This facial recognizer is at first trained with cascading algorithms as our cascading intruder detector to locate faces, then trained with Principal Component Analysis (PCA) to recognize intruders' faces. MIT CBCL Face Database [23] is used to train the cascading face detector and Facial Recognition Technology Database [24] is used to train facial recognizer with PCA. PCA can represent the high-dimensional samples with fewer dimensions, which are called principal components. The principal components are obtained by maximizing the variance of training data on each component and minimizing the mean squared errors between real and estimated values. The largest variance of data is contained in the first principal component and each succeeding component in turn has the largest variance. Every principal component is orthogonal to each other, so that the variable on each component is uncorrelated. The processed face image has 70×70 = 4900 pixel-values and each pixel-value represents a feature of the image. Therefore, the processed face image is represented by a column vector with 4,900 dimensions.

In our system, we use PCA to extract the first 100 principal components by building a new coordinate system to represent the processed face image. The dimensionality reduction lowers calculation complexity and removes noises included in irrelevant features. During the intruder identification, homeowners' faces have been collected as datasets in our surveillance system for training the facial recognizer. If the homeowner's face is recognized, then the homeowner's name is shown on the image. Otherwise, the detected face is considered as an intruder. Figure 7 shows the process of intruder recognition with PCA.

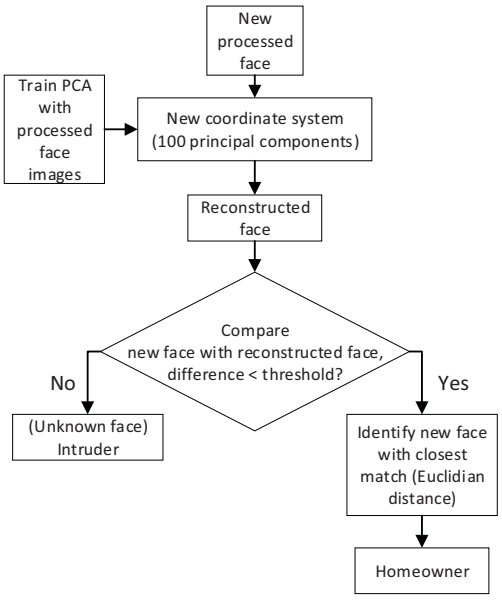

Figure 7. Flowchart of Intruder Recognition with PCA

After the new coordinate system has been trained with PCA algorithm utilizing face images, every new input face can be reconstructed with the new coordinate system. If the difference between the reconstructed face and new input face is above the threshold, which is 0.4 in our system, the new face is classified as an intruder's face. Otherwise, we use k-nearest neighbors algorithm [3] to find the close match for identification as the homeowner. Figure 8 shows the identification results of our facial recognizer. Faces of homeowner and intruder are differentiated and intruder is identified using our SIDSS. Our facial recognizer achieves 98.8% true-positive recognition of accuracy which is higher than other face detection algorithms, for example, HeadHunter achieves 97.14% of accuracy [25].

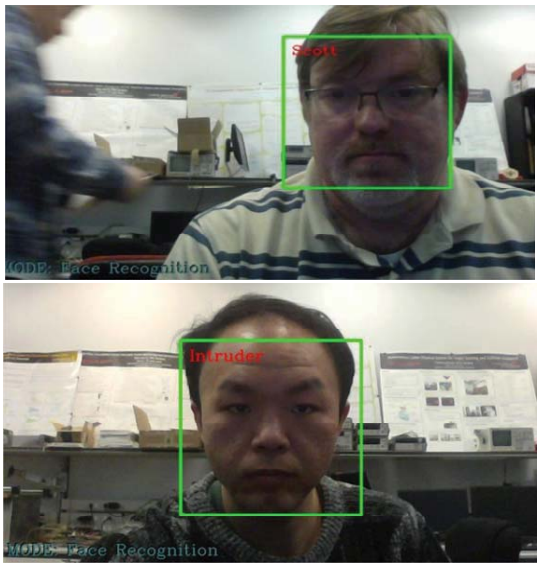

Figure 8. Output of Intruder Recognition with PCA

VI. Conclusion

In this paper, we introduced a design flow for smart home surveillance which utilizes an innovative 3-stage SIDSS to identify intruders accurately in complex scenes. Large datasets with real surveillance images are created to train the SIDSS, various optimization algorithms including batch normalization, adaptive boosting and cascading methods are used to enhance robustness of the SIDSS. Although the proposed surveillance system is robust and scalable for various surveillance events, our system can be expanded with more pre-processed datasets added to the SIDSS model to manage greater surveillance areas.

References

[1] W. Chen, P. Chen, W. Lee, C. Huang, "Design and Implementation of a Real Time Video Surveillance System with Wireless Sensor Networks," *IEEE Vehicular Technology Conference*, May 2008.

[2] H. Kruegle, CCTV Surveillance. Burlington, MA: *Elsevier Butterworth-Heinemann*, 2006.

[3] H. Feng, D. Eyers, S. Mills, Y. Wu, Z. Huang, "Principal Component Analysis Based Filtering for Scalable, High Precision k-NN Search", *IEEE Transactions on Computers*, vol. 67, no. 2, Feb. 2018.

[4] X. Zhang, T. Gonnot and J. Saniie, "Real-Time Face Detection and Recognition in Complex Background," *Journal of Signal and Information Processing*, vol. 8, pp. 99-112, May 2017.

[5] C. Weng, H. Wang, J. Yuan, X. Jiang, "Discovering Class-Specific Spatial Layouts for Scene Recognition," *IEEE Signal Processing Letters*, vol. 24, no. 8, Aug. 2017.

[6] S. Sruthy, S. George, "WiFi Enabled Home Security Surveillance System Using Raspberry Pi and IoT Module," *2017 IEEE International Conference on Signal Processing, Informatics, Communication and Energy Systems (SPICES)*, Nov. 2017.

[7] I. Aydin, N. Othman, "A new IoT combined face detection of people by using computer vision for security application," *2017 International Artificial Intelligence and Data Processing Symposium (IDAP)*, Sep. 2017.

[8] N. Dalal, B. Triggs, "Histograms of oriented gradients for human detection," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Jun. 2005.

[9] Sci2s.ugr.es. (2019). *Weapons Detection | Soft Computing and Intelligent Information Systems*. [online] Available at: https://sci2s.ugr.es/weapons-detection#RP [Accessed 28 Feb. 2019].

[10] M. Grega (2019). *Automated Detection of Firearms and Knives in a CCTV Image*. [online] Kt.agh.edu.pl. Available at: http://kt.agh.edu.pl/~matiolanski/KnivesImagesDatabase/ [Accessed 28 Feb. 2019].

[11] Pascal.inrialpes.fr. (2019). *INRIA Person dataset*. [online] Available at: http://pascal.inrialpes.fr/data/human/ [Accessed 28 Feb. 2019].

[12] S. Arlot, A. Celisse, "A survey of cross-validation procedures for model selection," *Statistics Surveys*, vol. 4, pp. 40–79, 2010.

[13] C. Kuo, "Understanding Convolutional Neural Networks with A Mathematical Model," *arXiv preprint* arXiv:1312.6034, Nov. 2016.

[14] W. Liu, Y. Wen, Z. Yu, M. Yang, "Large-Margin Softmax Loss for Convolutional Neural Networks," *Proceedings of the 33rd International Conference on Machine Learning*, vol. 48, 2016.

[15] P. Boer, D. Kroese, S. Mannor, R. Rubinstein, "A Tutorial on the Cross-Entropy Method," *Annals of Operations Research*, vol. 134, no. 1, pp. 19–67, Feb. 2005.

[16] D. Kingma, J. Ba, "Adam: A Method for Stochastic Optimization," *3rd International Conference for Learning Representations*, *arXiv preprint,* arXiv:1412.6980, Jan. 2017.

[17] K. Janocha, W. Czarnecki, "On Loss Functions for Deep Neural Networks in Classification," *Theoretical Foundations of Machine Learning*, *arXiv preprint* arXiv: 1702.05659, Feb. 2017.

[18] D. Figueira, M. Taiana, A. Nambiar, J. Nascimento and A. Bernardino, "The HDA+ data set for research on fully automated reidentification systems," *ECCV workshop*, 2014.

[19] X. Zhu, D. Ramanan, "Face Detection, Pose Estimation, and Landmark Localization in the Wild," *IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2012.

[20] T. Ahonen, A. Hadid, M. Pietikainen, "Face Description with Local Binary Patterns: Application to Face Recognition," *Pattern Analysis and Machine Intelligence*, vol. 28, no. 12, Dec. 2006.

[21] K. Mehmood, B. Ahmad, "Implementation of Face Detection System using Adaptive Boosting Algorithm," *International Journal of Computer Applications*, vol. 76, no. 2, Aug. 2013.

[22] B. Jain Stoble, M. Sreeraj, "Multi-posture human detection based on hybrid HOG-BO feature," *Fifth International Conference on Advances in Computing and Communications*, 2015.

[23] Ai.mit.edu. (2019). *CBCL SOFTWARE*. [online] Available at: http://www.ai.mit.edu/projects/cbcl.old/software-datasets/FaceData2 html [Accessed 28 Feb. 2019].

[24] NIST. (2019). *Face Recognition Technology (FERET).* [online] Available at: https://www.nist.gov/programs-projects/face-recognition-technology-feret [Accessed 28 Feb. 2019].

[25] M. Mathias, R. Benenson, M. Pedersoli, L. Van Gool, "Face Detection Without Bells and Whistles," *In Computer Vision – ECCV* 2014.