Recognition of Untrustworthy Face Images in ATM Sessions using a Bio-inspired Intelligent Network

R. Škoviera[1,2], K. Valentín[1,2], S. Štolc[3,1] and I. Bajla[1]
[1]*Institute of Measurement Science, Slovak Academy of Sciences, Bratislava, Slovakia*
[2]*Faculty of Mathematics, Physics, and Informatics, Comenius University, Bratislava, Slovakia*
[3]*AIT Austrian Institute of Technology GmbH, Seibersdorf, Austria*

Keywords: Anomaly Face Classification, Hierarchical Temporal Memory, Automatic Teller Machine, Kinect, Surveillance System.

Abstract: The aim of this paper is to report on a pilot application of a bio-inspired intelligent network model, called Hierarchical Temporal Memory (HTM), for recognition (detection) of untrustworthy manipulation with an Automatic Teller Machine (ATM). HTM was used as a crucial part of an anomaly detection system to recognize hard-to-identifiable faces, i.e., faces with a mask, covered with a scarf, or wearing sunglasses. Those types of face occlusion can be a good indicator of potentialy malicious intentions of an ATM user. In the presented system, the Kinect camera was used for acquisition of video image sequences. The Kinect's depth output along with skeleton tracking was used as a basis of the color image segmentation. To test the proposed system, experiments have been carried out in which several participants performed normal and untrustworthy actions using an ATM simulator. The output of the face classification system can assist a security personnel in surveillance tasks.

1 INTRODUCTION

Application of surveillance systems, capable to automatically evaluate various untrustworthy human activities and generate an on-line alarm indication, can further improve the security of Automatic Teller Machines (ATM) in practice. According to EAST 2012 report, the vast majority of losses due to ATM related fraud attacks are caused by card skimming attacks. Additionally, in 2011, there was a surge in cash trapping activity. Those types of activities usually need some period of time during which offenders try to disguise themselves by masks, sunglasses, scarfs, etc., while manipulating with the ATM, for instance, installing various illegal devices. The surveillance systems can often record images of the customer's face, however, having the face covered, the identification of a possible offender for a follow-up criminal investigation is substantially degraded. Recently, several approaches to detection of partially covered faces have been published (Wen et al., 2005; Dong and Soh, 2006; Lin and Liu, 2006; Suhr et al., 2012). They are based on computationally demanding calculation of various facial features and/or using different forms of template matching.

In our paper we present a different "global" approach that could be promising especially for real-time security surveillance applications. This approach is based on an intelligent (bio-inspired) learning network for classification of image sequences of users' faces, extracted from video, into normal and untrustworthy (anomalous) categories. In particular, for classification of untrustworthy faces, we propose to use a bio-inspired Hierarchical Temporal Memory (HTM) learning network (George and Hawkins, 2005; George, 2008) that solves the two-class classification problem without the need of computing any sophisticated image features. The HTM network enables for calculating real values of beliefs in defined classes in the range $[0,1]$, which can be afterwards used for calculation of a real-valued score as a function of time.

For generation of input visual data, i.e., image segments containing users' faces, we have utilized the commercial 3D camera system (Kinect device) originally developed for game industry for acquisition and segmentation of visual scenes. From a video sequence it produces a sequence of human body skeletons which we have used for extraction of the image segments containing the customer's face. We have evaluated the classification accuracy into normal and anomalous face categories (achieved by the HTM) in real experiments and compared it with conventional classifiers.

Škoviera R., Valentín K., Štolc S. and Bajla I. (2013).
Recognition of Untrustworthy Face Images in ATM Sessions using a Bio-inspired Intelligent Network.
In *Proceedings of the 2nd International Conference on Pattern Recognition Applications and Methods*, pages 511-517
DOI: 10.5220/0004195605110517

2 BASIC METHODOLOGY

The project was focused on the research into users' face classification method based on a bio-inspired HTM network. The proposed classification system consists of:

- An Input Image Acquisition Unit.
  - In our project, we decided to use the Kinect device for image acquisition since it can provide a stream of RGB images as well as other types of data (see Section 4) that could be used to simplify the face recognition task.
- An ATM Simulator.
  - To simulate a realistic situation at the ATM we decided to use a software based ATM simulator that would run on a PC and would compel the test subjects to behave like they would while interacting with a real ATM.
- An Application of the Optimized HTM Network.
  - For the task of classification of the face images we will use the HTM network. However, since the images of the whole person at the ATM with background clutter would not be suitable as an input for the HTM, we will also have to devise a program that will produce images with only the faces from the recorded data (see Section 5.3).

The project should provide an information, whether the proposed philosophy of improving the ATM security is, in principle, capable for deployment in a semi-autonomous on-line monitoring of the user behavior and in early detection of unauthorized and/or untrustworthy activities.

3 HIERARCHICAL TEMPORAL MEMORY

The Hierarchical Temporal Memory (HTM) is a memory-prediction network proposed initially by (George and Hawkins, 2009; Hawkins and Blakeslee, 2004), and distributed as a free software package, called NuPIC[1], by Numenta, Inc. The HTM has been explored and further extended by several other authors, e.g., (Thornton et al., 2008; Kostavelis and Gasteratos, 2012; Rozado et al., 2012; Štolc and Bajla, 2010a; Štolc and Bajla, 2010b; Štolc and Bajla, 2009).

Structurally, HTM network can consist of several layers (levels) of elementary units, called *nodes,* which use the same algorithms. The effective area from which a node receives its input is called *field of view* or *receptive field* of the node. The individual levels are ordered in a hierarchical tree-like structure (see Fig. 1 for a prototypical example of the HTM network). There is a zero sensory level of the HTM which serves as an input to the first level of nodes. In our case, zero level represents a visual field of image pixels. At the top level, there is only one node that serves for classification. In this role various classifiers can be applied.

Since the use of smooth temporal dependencies of input spatial patterns is essential characteristic of the HTM, its learning process utilizes either native sequence of images (e.g., video captured by a camera), or (in case of static images) an artificially generated sequence of images using various exploring schemes.

The HTM node operates in two distinct stages – *learning* and *inference*, the algorithms of which will be briefly described in the following section.

3.1 Learning in a Node

The learning is done by training the network level by level starting at the bottom level. Nodes in the level that is learning can be trained separately or using shared representations. After the nodes in a level are trained, they switch into inference mode and produce input for the next level. When the learning process is finished in all the levels, the HTM network can classify any unknown pattern into trained categories.

3.1.1 Memorization of Input Representative Patterns

In the first step of the learning process, the node memorizes the representative spatial patterns from its receptive field. This memorization process can be considered as an on-line quantization of the input image space (Numenta, 2009). The quantization works in the following way: when the Euclidean distance between the input pattern and the nearest existing quantization point exceeds the value of the parameter *maxDistance,* the input pattern becomes a new quantization point. The memorized quantization points (also called *coincidences*) represent centroids of the pools covering one or more input patterns. After having processed all available input patterns (or reaching the requested number of quantization points), the memorization process is finished.

3.1.2 Learning Transition Probabilities

The ultimate goal of the HTM learning is to detect correct invariant representations of the input world

[1]NuPIC stands for Numenta Platform for Intelligent Computing.

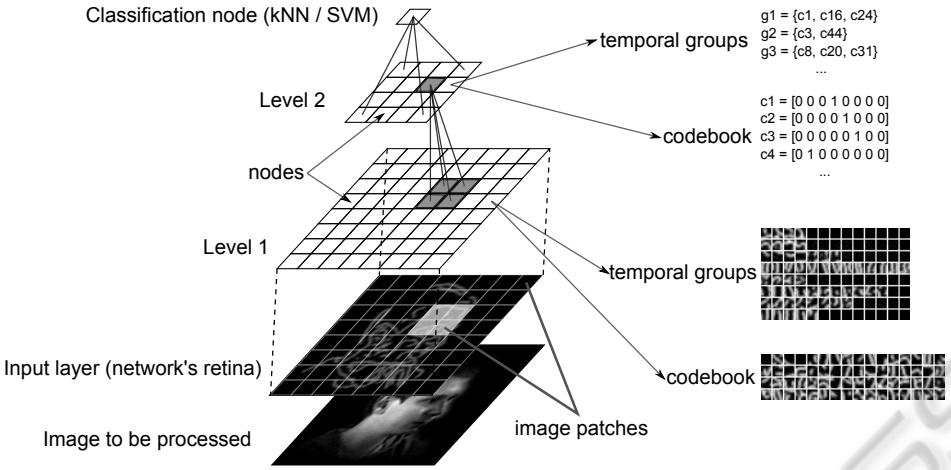

Figure 1: An example of the 2-level HTM network for learning invariant representations of faces. The nodes at the level-1 are arranged in an 8×8 grid and they cover the whole retina without overlap. The input image has 128×128 pixels. Each level-1 node receives its input from a 16×16 pixel patch of the retina. Each level-2 node receives its input from four child nodes at the level-1. The top-most classifier node is connected to all level-2 nodes.

based on the temporal relations contained in the learning sequence. To achieve this, one needs to evaluate a frequency of transition events, i.e., co-occurrences of the memorized coincidences in adjacent time instances. A sequence of the input patterns generates a sequence of the coincidences within the node. In the HTM theory (George, 2008), the temporal relations are described in a form of the first-order Markov graph in which the vertices represent coincidences and weighted edges express how strong is the temporal relation. In each HTM node, an individual Markov graph is constructed and maintained.

3.1.3 Temporal Grouping

The last step of the learning process within each HTM node is to analyze the constructed Markov graph and partition it into a set of temporal groups. The partitioning is done in such a way that the vertices of the same temporal group have strong temporal relation. To form the temporal groups, the nodes use the Agglomerative Hierarchical Clustering (AHC) method (Johnson, 1967).

3.2 Inference in a Node

A node that has completed its learning phase can be switched into the inference mode. In this mode, the node produces an output vector for every input pattern provided. This vector indicates the degree of membership of the input pattern into each of the temporal groups. There are two phases of the inference process – inference in the "spatial pooler" followed by inference in the "temporal pooler".

Typically, most of the input patterns do not perfectly match any of the patterns stored in the node's memory. Therefore, a closeness to every memorized coincidence must be calculated. Let d_i be the Euclidean distance of the i-th stored pattern from the input pattern. The larger is the distance, the smaller should be the match between the input pattern and the stored coincidence (pattern). It can be assumed that the match of the patterns can be expressed as a Gaussian function of their Euclidean distance, with the zero mean: $y_i = e^{-d_i^2/\sigma^2}$, where σ is a parameter of the node. By calculating this quantity for all n memorized coincidences, one can produce an overall belief vector $\mathbf{y} = [y_1, y_2, \ldots, y_n]$ that represents closeness of the input pattern to all memorized coincidences (George and Hawkins, 2009). Such a vector is inferred in the spatial pooler for every input pattern.

In the second phase of the inference, the temporal pooler makes use of the learned temporal groups and calculates the output vector for the nodes that are above in the HTM hierarchy. Each individual component of the output vector represents belief that $\mathbf{y}$ comes from a particular temporal group.

4 IMAGE ACQUISITION IN THE SYSTEM

Since the users' face images serve as an input to the HTM network, and our intention was to simulate a real-time image acquisition process, we needed a generator of video image sequences. For this purpose, various cameras and image segmentation algorithms

could be applied to the task of input data generation, however, to be able to concentrate on the main problem, we have utilized an existing convenient tool that could simplify this auxiliary task – Kinect.

Kinect is an input device to the Microsoft Xbox 360 game console that enables a novel interaction of players in terms of video, movement, and sound. This device contains a color VGA camera and a VGA depth image sensor with 2048 possible distance values. The basic capability of Kinect is to track skeletons of up to two players. The skeleton is represented by 20 joints that cover the whole body. This was an important attribute of Kinect, since localization of representative parts of the body enables to solve the problem of the image segmentation required by the HTM network. Kinect can be connected to a standard PC as an input device and operated using official SDK[2] by Microsoft[3]. The SDK enables acquisition of four types of data: RGB video, depth video, skeletons, and audio. For the purposes of our project, it was sufficient to use the first three data types. Because of the data resolution limitation for the depth image (QVGA resolution), it was necessary to warp it into the VGA video resolution (see Section 5.3).

5 DATA GENERATION AND STORAGE

5.1 ATM Simulator and Data Synchronization (Kinect – ATM-S)

In our project, we decided to substitute a real ATM by a software model that could operate on a notebook and imitate real interactions with the user being scanned by the Kinect. After the analysis of solutions currently available on the Internet, we decided to implement our own software ATM simulator (ATM-S). The ATM-S is designed as a state automaton that records state changes and user inputs by means of a system of events.

However, ATM-S events (that can be used to trigger face recognition in real world applications) as well as the three data types, generated by Kinect for each frame, occur asynchronously. To synchronize them, we have created a program called Logger. This program subscribes to events generated by both Kinect and ATM-S, synchronizes them according to their time stamps, and stores them in a synchronous format. The Logger is also used for reduction of the amount of data coming from Kinect.

[2]Software development kit.

[3]SDK is available at http://www.microsoft.com/en-us/kinectforwindows/download/http://www.microsoft.com/en-us/kinectforwindows/download/

5.2 Acquisition of Input Data

The acquisition of input data was carried out in an office environment where ATM-S was set up. The system included a notebook with the running ATM-S and Logger. Among various anomalous behaviors (listed in Section 2), we focused on the acquisition of images of faces. Kinect was placed so that the skeletons could be tracked for a person using ATM-S while the RGB camera was able to capture his/her face as well as hands. Participants of the experiments accomplished several trials, in which they simulated normal and anomalous behavior while using ATM-S.

5.3 Data Processing

Due to different resolutions and camera positions, the pixels with the same positions in the depth and RGB images do not correspond to the same point in the real scene. The deviation is greater the closer is the object to the camera plane. For mapping of the RGB image to the geometry of the depth image, the SDK provides a special transformation map that however downgrades the RGB image to QVGA resolution. However, to preserve the VGA resolution of the RGB image, we implemented a backmapping warp transformation that maps the input depth image into the geometry of the RGB image.

The next step was to segment out the faces from the RGB images (using the skeleton data and depth image). To remove most of the background information, we cut out the face area based on the positions obtained from the skeleton. Due to the properties of the RGB camera and the distance of the user, the face images had resolution of 128×128 pixels. To suppress the rest of the background, a gray-scale mask was generated from the depth image of the tracked user. The potential artifacts in the generated mask were removed by application of the binary morphological closing. To prevent from forming of artificial edges around masked object, the resulted mask was then smoothed by a Gaussian filter. To remove the neck from the face image, we also applied a Blackman window mask. The final segmented image was obtained by pixel-wise multiplication of the face area RGB image by the prepared mask. An illustrative example of this process is depicted in Fig. 2.

Since currently HTM accepts only gray-scale images, the obtained RGB images were finally transformed to the gray-scale format.

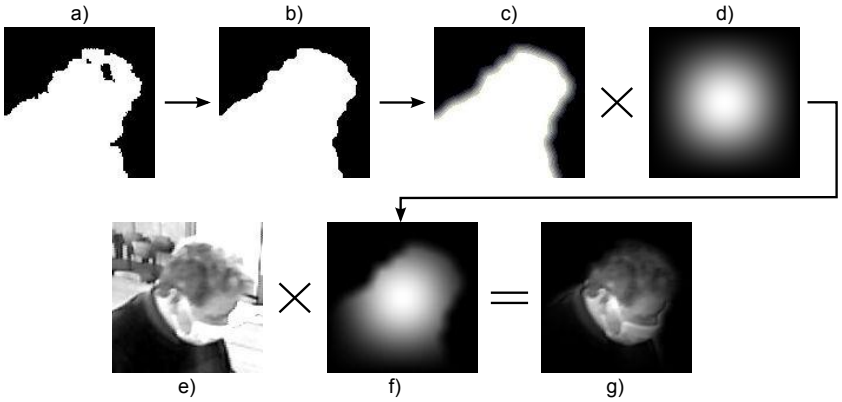

Figure 2: Face image pre-processing and application of the mask: a) thresholded mask, b) mask after binary morphological closing, c) smoothed mask, d) Blackman window, e) face cut-out, f) final mask image, g) segmented face image. The "×" symbol represents pixel-wise multiplication.

Table 1: Changes in HTM learning and inference parameters made to the original setup from NuPIC.

level	type	parameter name	value
0	sensor	image width and height	128
1	spatial pooler	gaborNumOrients	4
2	spatial pooler	maxCoincidenceCount	512
3	spatial pooler	maxCoincidenceCount	512
4	classifier	outputElementCount	2
		autoTuningData	False

Figure 3: Examples of selected face images. In the first row, there are examples of the anomaly category. Second row contains examples of the normal category.

6 HTM IMPLEMENTATION

The structure of our HTM network was inspired by the NuPIC template set of parameters[4] dedicated for the visual object recognition domain. The HTM network consisted of three layers of nodes followed by a classifier node on the top. The spatial pooler in the first layer of the HTM network used Gabor filters. The Gabor filters play the role of edge detectors in certain directions and frequencies, and moreover, they are biologically plausible (Daugman, 1985). See Table 1 for information about additional parameters.

To better evaluate the influence of the HTM network on the change of the classification complexity, we used the *k*-nearest neighbor (*k*-NN) classifier as the baseline. Along with *k*-NN we also investigated the Support Vector Machine (SVM) classifier with a Gaussian kernel function.

6.1 Learning and Testing

After the pre-processing of all acquired images, we selected 190 face images (see Fig. 3 for examples). To increase the variability of the data set (and thereby the generalization of the image classification system), we extended this initial set by application of rotational (up to 45 degrees), translational, and mirroring transformations. As a result, we obtained 21 new images for each original image, that made altogether 4180 face images.

Before initiating the learning process, we randomly divided the data set into two disjunctive groups – testing and training set. To analyze the influence of the training data size on the classification accuracy, we created several data set divisions, ranging from 5% to 50%. To validate our results, for every ratio we repeated the training and testing process 20 times. Finally, we calculated the arithmetic mean and standard deviation of the classification accuracy.

We have realized two scenarios:

Scenario 1: The whole data set was used and *k*-NN classifier in the input space was compared with the combination of the HTM and the *k*-NN classifier.

Scenario 2: Translated images were excluded from the training; *k*-NN classifier in the input space was compared with the combination of the HTM and

[4]The file with the default HTM parameters is placed in "Numenta/nupic-1.7.1/share/vision/experiments/toolkitNetworks/SVMnetwork/params.py" after installation of NuPIC software on Windows operating system.

Table 2: Results of the classification of faces to anomaly and normal classes. In both cases, k-NN was used with $k = 1$.

Training data ratio in %	Classification accuracy in % (stdev)					
	Scenario 1			Scenario 2		
	k-NN	HTM+k-NN	HTM+SVM	k-NN	HTM+k-NN	HTM+SVM
5	74.92 (1.21)	78.73 (1.81)	80.59 (1.85)	73.28 (1.71)	75.44 (1.63)	75.99 (2.94)
10	80.12 (0.82)	83.35 (1.24)	86.46 (1.48)	78.05 (1.29)	82.15 (1.33)	82.46 (1.69)
20	85.84 (0.60)	88.83 (0.81)	92.57 (0.75)	84.14 (1.17)	88.02 (1.76)	88.56 (1.56)
30	89.01 (0.55)	91.29 (0.66)	94.56 (0.80)	86.50 (0.97)	91.53 (0.97)	91.36 (1.37)
40	91.06 (0.65)	93.38 (0.71)	97.39 (0.25)	88.24 (0.69)	94.10 (0.87)	93.44 (1.06)
50	92.56 (0.63)	94.57 (0.52)	98.22 (0.38)	89.01 (0.75)	94.86 (0.78)	94.19 (0.61)

the SVM classifier.

7 HTM APPLICATION RESULTS

The results of our computer experiments with the anomalous and normal faces, carried out according to the two described scenarios, are summarized as values of classification accuracy (CA) in Table 2. Based on their analysis, the following conclusions can be derived:

- There are differences observed between the results obtained by Scenario 1 and Scenario 2: (i) the CA values for k-NN classifier reached within Scenario 1 are greater in comparison to those obtained by Scenario 2; the difference is increasing with the increasing size of the training set, (ii) for the vector space generated by HTM and k-NN classifier applied to this space, the CA values obtained by the individual Scenarios differ less; for larger training sets the CA values are aï¿œbit greater in the case of Scenario 2,
- The average loss in CA of k-NN classifier in Scenario 2 comparing to Scenario 1 is 2.382 percentage points, whereas when combined with HTM network, the average loss is only 0.675 percentage points, this suggests that HTM has an ability to learn representations more invariant to translation than k-NN alone (especially in the case of lesser training data size),
- The application of the optimized HTM network to our classification problem outperforms the application of the simple k-NN classifier in the input space,
- Finally, the best CA is achieved for HTM combined with the SVM classifier in Scenario 1 (98.22%).

8 CONCLUSIONS

In this pilot project (a feasibility study), we have focused on the face classification problem. The system can work well under moderate lighting changes and even on rather small images containing some amount of noise. It should be noted that our system can serve as a convenient basis for exploring a more complex detection system.

In conclusion we can summarize individual contributions made within the reported pilot project:

- development of an ATM program simulator (ATM-S),
- development of a program Logger for synchronization between Kinect and ATM-S,
- development of a program module necessary for warp transformation that maps the input depth image to the geometry of the RGB original image and pre-processing the resulting images to yield final face images suitable for input in the HTM network,
- finding optimum parameters of the HTM network by conducting a large set of computer experiments,
- comparison of the performance of several well-known classifiers working in tandem with HTM.

ACKNOWLEDGEMENTS

This work has been supported by the Slovak Grant Agency for Science (VEGA research project No. 2/0019/10) and partially by the Tatra banka Foundation, through the program E-talent for young scientists in the field of applied informatics (project No. 2010et019).

REFERENCES

Daugman, J. G. (1985). Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *Journal of the Optical Society of America A Optics and image science*, 2(7):1160–1169.

Dong, W. T. and Soh, Y. S. (2006). Image-based fraud detection in automatic teller machine. *IJCSNS*, 6(11):13.

George, D. (2008). *How the brain might work: A hierarchical and temporal model for learning and recognition*. PhD thesis, Stanford.

George, D. and Hawkins, J. (2005). A hierarchical Bayesian model of invariant pattern recognition in the visual cortex. In Prokhorov, D., editor, *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, volume 3, pages 1812–1817.

George, D. and Hawkins, J. (2009). Towards a mathematical theory of cortical micro-circuits. *PLoS Computational Biology*, 5(10):e1000532.

Hawkins, J. and Blakeslee, S. (2004). *On intelligence*. Henry Holt and Company, New York.

Johnson, S. (1967). Hierarchical clustering schemes. *Psychometrika*, 32:241–254.

Kostavelis, I. and Gasteratos, A. (2012). On the optimization of hierarchical temporal memory. *Pattern Recognition Letters*, 33(5):670–676.

Lin, D. T. and Liu, M. J. (2006). Face Occlusion Detection for Automated Teller Machine Surveillance. In Chang, L.-W. and Lie, W.-N., editors, *Advances in Image and Video Technology*, volume 4319 of *Lecture Notes in Computer Science*, pages 641–651. Springer Berlin / Heidelberg.

Numenta (2009). *Numenta node algorithms guide, NuPIC 1.7*.

Rozado, D., Agustin, J. S., Rodriguez, F. B., and Varona, P. (2012). Gliding and saccadic gaze gesture recognition in real time. *ACM Transactions on Intelligent Interactive Systems*, 1(2):1–27.

Suhr, J. K., Eum, S., Jung, H. G., Li, G., Kim, G., and Kim, J. (2012). Recognizability assessment of facial images for automated teller machine applications. *Pattern Recognition*, 45(5):1899–1914.

Thornton, J., Faichney, J., Blumenstein, M., and Hine, T. (2008). Character recognition using hierarchical vector quantization and temporal pooling. In *Proceedings of the 21st Australasian Joint Conference on Artificial Intelligence: Advances in Artificial Intelligence*, AI '08, pages 562–572, Berlin, Heidelberg. Springer-Verlag.

Štolc, S. and Bajla, I. (2009). Image object recognition based on biologically inspired Hierarchical Temporal Memory model and its application to the USPS database. In Tyšler, M., editor, *7th International Conference MEASUREMENT 2009*, pages 23–27, Smolenice, Slovak Republic.

Štolc, S. and Bajla, I. (2010a). Application of the computational intelligence network based on hierarchical temporal memory to face recognition. In Hamza, M. H., editor, *10th IASTED International Conference on Artificial Intelligence and Applications AIA 2010*, pages 185–192, Innsbruck, Austria. ACTA Press.

Štolc, S. and Bajla, I. (2010b). On the Optimum Architecture of the Biologically Inspired Hierarchical Temporal Memory Model Applied to the Hand-Written Digit Recognition. *Measurement Science Review*, 10(2):28–49.

Wen, C., Chiu, S., Tseng, Y., and Lu, C. (2005). The mask detection technology for occluded face analysis in the surveillance system. *Journal of Forensic Sciences*, 50(3):1–9.