International Journal of Computational Intelligence in Control
Vol. 13 No. 1 (June, 2021)

Feature Learning-Based Segmentation Algorithm for Hand Segmentation

Shih-Hung Yang[1*], Yu-Min Tseng[2], and Yon-Ping Chen[2]

[1] *Department of Mechanical and Computer Aided Engineering, Feng Chia University, Taichung, Taiwan, R.O.C.*
[2] *Institute of Electrical and Control Engineering, National Chiao Tung University, Hsinchu, Taiwan, R.O.C.*
[*]E-mail: shyang@fcu.edu.tw

Received : 15th August 2020 ***Revised on 24th September 2020;*** ***Accepted: 13th November 2020***

Abstract:

Feature learning algorithms have been studied extensively for solving many pattern recognition problems, and several effective algorithms have been proposed. This paper proposes a feature learning-based segmentation algorithm (FLSA) for determining appropriate features for hand segmentation. This new approach combines an unsupervised learning phase and a supervised learning phase for hand segmentation. The unsupervised learning phase consists of preprocessing, patch generation, and filter learning through the K-means algorithm. The supervised learning phase consists of feature extraction, classification learning, pixel classification, and morphological operation. The FLSA starts feature learning through the K-means algorithm and then trains a neural network (NN) as the classifier for classifying a pixel into two categories: hand and nonhand. As feature learning progresses, features appropriate for hand segmentation are gradually learned. The emphasis of the FLSA on feature learning can improve the performance of hand segmentation. The Georgia Tech Egocentric Activities data set was used as unlabeled data for feature learning, and the corresponding ground-truth data were used for NN training. The experimental results show that the FLSA can learn features from unlabeled data and verify that feature learning leads to hand segmentation that is more effective than that without feature learning.

Keywords: feature learning, hand segmentation, unsupervised learning.

I. INTRODUCTION

Feature learning, also called representation learning [1], [2], [3], involves a set of techniques in machine learning that learn a transformation of "raw" inputs into a representation that can be effectively exploited in a supervised learning task such as a recognition problem [4], [5], object detection and semantic segmentation [6], [7], scene classification [8], [9], robotic applications [10], and person reidentification [11]. Feature learning algorithms may be either unsupervised or supervised, such as multilayer neural network (NN), autoencoder, K-means, matrix factorization, restricted Boltzmann machine (RBM), and various forms of clustering [12].

Multilayer NNs can be applied to feature learning because they learn a representation of their input at the hidden layers; the representation is subsequently used for classification or regression at the output layer. Feature learning can be performed in an unsupervised manner in which the features are learned from an unlabeled data set. The learned features are then employed using labeled data to improve the performance of classification task in a supervised manner [13].

The K-means algorithm, which has been considered for feature learning in many applications, clusters unlabeled data into k clusters where the centroids of the clusters represent the dictionary of features. These centroids are then used to produce features for a subsequent supervised learning task. The produced features can be derived using several methods, the simplest of which is to add k binary features to each sample. Coates and Ng [14] determined that certain variants of the K-means algorithm behave similarly to sparse coding algorithms.

In comparatively evaluating unsupervised feature learning methods, Coates and Ng [14] found that K-means clustering with an appropriate transformation outperforms the more recently invented autoencoder and RBM algorithms in an image classification task. The K-means algorithm has also been demonstrated to improve performance in the domain of natural language processing, specifically for named-entity recognition, and it competes with Brown clustering and distributed word representations. Therefore, the superiority of the

K-means algorithm is useful for applications involving hand-crafted features, such as image segmentation.

Hand segmentation is a crucial technology for egocentric activity recognition [15]. For example, Google Glass, a wearable computer with an optical head-mounted display, was developed for producing a mass-market ubiquitous computer [16]. It is equipped with an eye-view scope that can be further analyzed for user activity recognition, particularly because the hands of the user generally appear in the eye-view scope. To perform egocentric activity recognition further according to hand activities, hand segmentation is an essential technology for extracting the hands. Other advanced devices, such as GoPro and Kinect, have been applied to capture egocentric images.

Hand segmentation refers to the process of partitioning a digital image into multiple regions and selecting the regions that belong to the hands [17]. The goal of segmentation is to simplify and change the representation of an image into a result that is more meaningful and easier to analyze [18]. Image segmentation mainly groups pixels in homogeneous regions according to common features; more precisely, it is the process of assigning a label to every pixel in an image so that pixels with the same label share certain visual characteristics. Image segmentation is typically applied to locate and analyze objects in images.

The result of image segmentation is a set of segments that collectively cover the entire image, or a set of contours extracted from the image. All of the pixels in a region are similar with respect to a particular characteristic or computed property such as color, intensity, or texture. Adjacent regions are distinct with respect to the same characteristics. Segmentation is mainly used in medical imaging, face recognition, fingerprint recognition, traffic control systems, and machine vision. However, selecting appropriate features for hand segmentation is difficult because the features are generally hand-crafted features that are not easily designed. Thus, developing a feature learning algorithm is necessary to extract appropriate features for specific tasks.

This paper describes a feature learning-based segmentation algorithm (FLSA) for determining appropriate features for hand segmentation. The new approach combines an unsupervised learning phase and a supervised learning phase for pixel-based hand segmentation. It starts feature learning through the K-means algorithm and then trains a

classifier to classify a pixel into two categories: hand and nonhand. As feature learning progresses, features appropriate for hand segmentation are gradually learned. The effects of the FLSA on feature learning can improve the performance of hand segmentation.

The remainder of this paper is organized as follows. Section II describes the proposed algorithm in detail, and Section III presents the experimental results. Finally, Section IV concludes this paper with a brief summary and a few remarks.

Ⅱ. FEATURE LEARNING-BASED SEGMENTATION ALGORITHM

This section presents the proposed FLSA in detail. The main objective of the FLSA is to perform both feature learning and classifier learning through unsupervised and supervised learning, respectively. Fig. 1 shows a flowchart of the FLSA, which includes two main parts: unsupervised feature learning and supervised classifier learning. The unsupervised feature learning consists of preprocessing, patch generation, and filter learning through the K-means algorithm. The supervised classifier learning consists of feature extraction, classification learning through an NN, pixel classification, and morphological operation. These two main parts are further explained as follows.

A. Unsupervised feature learning

This section describes the framework of an unsupervised feature learning algorithm for hand segmentation. The framework involves several stages and is similar to other feature learning works, e.g., ICA [19] and sparse coding [20]. The flowchart of the unsupervised feature learning algorithm is shown in Fig. 2. First, the acquired raw data must include scenarios that are related as closely as possible to the objective application. First, the patches are generated from the raw data for the further dictionary construction. Then, a pre-processing is applied to remove the correlations between nearby pixels. Finally, the dictionary is learned by K-means algorithm and could be employed to extract features for hand segmentation. The following section will introduce the patch generation and pre-processing in detail.

This learning framework starts with patch generation while the raw data are acquired from the scenarios related to the objective application. Furthermore, in order to ensure a practical number of inputs for each cluster in K-means algorithm, a sufficient number of patches are necessary to train a K-means dictionary. In practice, 100,000 patches are enough for 16-by-16

gray patches [4]. The patch generated from unlabeled raw data has dimension *r*-by-*r* and *c* channels, where *r* refers to as the receptive field size [14] and *c* = 3 in RGB color image and *c* = 1 in gray level image. Each patch can be represented as a vector in $\Re^n$ of pixel intensity values, with $n = r \times r \times c$. Then a set of patches could be sampled from the raw data as $x^{(1)}, x^{(2)}, \ldots, x^{(m)}$, where $x^{(i)} \in \Re^n$. Thus a data matrix $\boldsymbol{X}$ could be constructed with *n* rows and *m* columns and is used to build dictionary in feature learning process. Then the dictionary could be adopted to construct a new representation of data with minimal error in reconstruction.

The pre-processing starts from normalizing the contrast and brightness of the patches. It could be achieved by

$$\tilde{\boldsymbol{X}} = \frac{\boldsymbol{X} - \mu_X}{\sqrt{\sigma_X^2 + 10}} \quad (1)$$

where μ_X and σ_X^2 are mean and variance of $\boldsymbol{X}$, respectively. Since K-means is likely to achieve highly correlated centroids, it may result in poor recognition or prediction performance [4]. Therefore, whitening is applied to remove the correlations between nearby pixels and is achieved by Zero Component Analysis (ZCA) whitening transformation. First, the eigenvalues, $\boldsymbol{D}$, and eigenvactors, $\boldsymbol{V}$, are obtained via the eigenvalue decomposition of the covariance of the data as

$$\boldsymbol{V}\boldsymbol{D}\boldsymbol{V}^{\mathrm{T}} = cov\left(\tilde{\boldsymbol{X}}\right) \quad (2)$$

Then the whitened data could be computed by

$$\overline{\boldsymbol{X}} = \boldsymbol{V}\left(\boldsymbol{D} + \varepsilon \boldsymbol{I}\right)^{-1/2} \boldsymbol{V}^{\mathrm{T}} \tilde{\boldsymbol{X}} \quad (3)$$

where ε is a small constant.

The unsupervised feature learning algorithm adopts the K-means algorithm to learn the dictionary of the features. The K-means algorithm is one of the unsupervised learning methods used for partitioning *n* observations into *k* clusters by minimizing the variance of each cluster. The technique consists of three main steps: initializing cluster centroids, assigning clusters, and updating cluster centroids. The first step involves randomly selecting *k* patches from the whole data set $\boldsymbol{X}$ and adopting them as the cluster centroids (filters). In the second step, the cluster assignment is achieved by assigning each observation to the cluster with the nearest mean where the distance could be either the Euclidean distance or the Manhattan distance. The

final step entails calculating the gravity point for each cluster to be the new cluster centroids. The process iterates for a fixed number of iterations or stops when the convergence of clusters is achieved.

Two types of distance function are commonly used for the K-means algorithm: the Euclidean distance and the Manhattan distance. The formula of the Euclidean distance is shown as follows:

$$D_E(\boldsymbol{p},\boldsymbol{c}) = \sqrt{\sum_{i=1}^{n}(p_i - c_i)^2} \tag{4}$$

where $\boldsymbol{p}$ is the position of a data point, $\boldsymbol{c}$ is the position of a cluster center, and n is the dimension. However, the Euclidean distance requires considerable computational resources because of the square root operation. To manage the problem, an alternative function called the Manhattan distance is used; its algorithm is expressed as

$$D_M(\boldsymbol{p},\boldsymbol{c}) = \sum_{i=1}^{n}|p_i - c_i| \tag{5}$$

where the computation time is less than that of the Euclidean distance. The computation time of the K-means algorithm depends on the size of the data set and the number of clusters; that is, the higher the number of clusters is, the longer the computation time. Therefore, using a simpler distance function can accelerate the K-means algorithm. Although the accuracy of the Manhattan distance is slightly lower than that of the Euclidean distance, the result is still acceptable. The learned centroids are the filters used to extract the feature for further supervised classifier learning.

B. Supervised classifier learning

This section describes the supervised classifier learning framework, which consists of feature extraction, classifier learning, pixel classification, and morphological operation. The supervised classifier learning starts with feature extraction, where the features of each pixel are extracted from the filters learned by the unsupervised feature learning framework. The features and their corresponding ground-truth data are then used to train the classifier. The pixels are classified into two categories—hand and nonhand—through the learned classifier according to the features. Finally, a morphological operation is used to eliminate noise.

The supervised classifier learning adopts an NN as the classifier because of its classification ability [21]. The structure of an NN consists of input neurons, hidden neurons,

output neurons, and connections between each layer. The number of neurons in the input and output layer is problem dependent. The input vector is linearly combined at the hidden neuron and then processed by the activation function, which can be one of the continuous neuron models (e.g., logistic, hyperbolic tangent, linear threshold, exponential, and Gaussian signal function) [22]. Regarding the training algorithm, the Levenberg–Marquardt algorithm [23] is used to update the weights as follows:

$$\boldsymbol{v}' = \boldsymbol{v} - \left[G^T G + \mu I\right]^{-1} G^T \varepsilon \tag{6}$$

where $\boldsymbol{v}$ is the weight vector, $\boldsymbol{G}$ is the Jacobian matrix, $\boldsymbol{\varepsilon}$ is the network error vector, and μ is a positive scalar parameter.

With sufficient training epochs, an NN can be learned to classify pixels. The preliminary hand segmentation results with noise can then be obtained. To eliminate the noise from the segmentation results, morphological operations including erosion and dilation are adopted. The morphological erosion operation can be expressed as

$$A \Theta B = \left\{ x : (B)_x \subseteq A \right\} \tag{7}$$

where A is the original image and B is a disk-shaped structuring element with a fixed radius. It is expected that the noise in an image can be erased after the operation. However, some gaps may also be generated in isolated regions after erosion. To repair these gaps, the morphological dilation operation is further employed and expressed as

$$A \oplus C = \left\{ x : (\hat{C})_x \cap A \neq \phi \right\} \tag{8}$$

where C is a disk-shaped structuring element with a fixed radius. It is expected that the gaps in image A can be repaired after operation.

Ⅲ. EXPERIMENTAL RESULTS

The goal of the experiment was to segment hands from an image by using the proposed FLSA. A Georgia Tech Egocentric Activities (GTEA) data set [1] was adopted as the data set in the experiment, which entailed seven daily activities from an egocentric perspective performed by four subjects. A GoPro camera was mounted on a baseball cap worn by each subject to capture the area in front of the subject's eyes. The image sequence was captured at a resolution of 1280×720. The data set contains the following scenarios: Hot Dog, Instant

Coffee, Peanut Butter Sandwich, Peanut Butter and Jelly Sandwich, Sweet Tea, Coffee and Honey, and Cheese Sandwich. The image sequence showed that the subjects' hands frequently appeared and interacted with the objects, such as the coffee, water, cup, and tea, in the corresponding scenarios. Therefore, hand segmentation is the key technique for egocentric activity recognition and is considered the benchmark problem. The images were downsampled because of the hardware computational limitation. Fig. 3 shows examples of the GTEA data set, where Fig. 3(a), (b), (c), and (d) present the cheese, coffee, tea, and hot dog activities.

The proposed algorithm was implemented using Matlab on a desktop PC that had an Intel Core I5-2550K 4.5GHz and 32 GB of DDR3 1600MHz RAM. In this experiment, the system comprises three parameters: (i) the number of features k, (ii) stride s, and (iii) receptive field size w. The parameters used for feature learning are set at $k = 16$, $s = 5$, and $w = 5$. For ZCA whitening transformation, ε is set to 0.01 for 16-by-16 pixel patches according to the suggestion in [4].The features are extracted via a 16-by-16 receptive field. Regarding the classifier learning, the structure of the NN is designed as follows. The input layer consists of 48 input nodes that receive the features, whereas the output layer consists of one output node that represents whether the pixel belongs to the hand. Furthermore, one hidden layer that uses a logistic sigmoid function as the activation function is adopted. The number of training epochs is 10,000.

To determine the parameter k in the K-means algorithm, a set of experiments with various k were performed. The results are shown in Fig. 4, where the numbers on the horizontal axis represent various k, whereas the numbers on the vertical axis represent the hand segmentation accuracy. As shown, $k = 16$ has the highest accuracy. Therefore, k is set at 16 in the experiments. According to the same concept, the number of hidden neurons was designed through a set of experiments with various numbers of hidden neurons. The results are shown in Fig. 5, where the numbers on the horizontal axis represent the numbers of hidden neurons, whereas the numbers on the vertical axis represent the hand segmentation accuracy. As shown, the NN with 40 hidden neurons demonstrated the highest performance and was thus selected as the structure of the NN.

The centroids (or dictionary) learned by the K-means algorithm are shown in Fig. 6. The receptive field incorporates filters with different orientations but various phases and

frequencies. Some receptive fields perform the functions similar with Sobel filters, whereas some receptive fields carry out the functions similarly with an average filter, such as the final four filters in the final row of Fig. 6.

For illustrating the performance of the proposed FLSA further, Fig. 7 shows an example of the ground-truth data and the segmented result of the FLSA. Although the result still contains the noise, the outline of the hands can be extracted. A morphological operation can then be applied to further eliminate the noise. For comparing the FLSA with those of other studies, Table I presents a comparison of segmentation results at different image resolutions. The first method involves using the NN [23] to segment the hands without feature learning. The features used in the NN [23] are the raw RGB values of the pixel and the surrounding eight pixels. Notably, the centroids employed in the FLSA were learned in RGB channels. The results show that the accuracy with high resolution images of 405×720 is higher than that with low resolution images of 203×360. Hence, the higher the resolution is, the higher the accuracy. However, the higher resolution did not substantially improve the accuracy but required longer computational time. According to the experimental results, the resolution of the images does not critically affect the accuracy. Moreover, the FLSA achieved higher accuracy than the NN did [23] because of the integration of unsupervised feature learning. The experimental results therefore indicate the superiority of integrating the phases of unsupervised and supervised learning.

IV. CONCLUSION

This paper proposes an FLSA that combines unsupervised feature learning and supervised classifier learning algorithms for hand segmentation. The framework involves two learning phases. The first is unsupervised feature learning, which consists of preprocessing, patch generation, and filter learning by using unlabeled data through the K-means algorithm. The second is supervised classifier learning, which consists of feature extraction, classification learning through the NN, pixel classification, and morphological operation. The main advantage of feature learning through the K-means algorithm is the simplicity and low computational cost on a large scale. Furthermore, the design of features is not necessarily hand-crafted. In experiments conducted to evaluate the performance of the proposed FLSA, a

GTEA data set was used as the unlabeled data for feature learning, and the corresponding ground-truth data were used to train the NN. The experimental results show that features can be learned from unlabeled data and verify that feature learning leads to hand segmentation that is more effective than that without feature learning. Furthermore, the resolution of the images does not critically affect the accuracy. Future work will focus on the development of region-based hand segmentation and its corresponding feature learning algorithm.

V. ACKNOWLEDGMENTS

This work was supported by the National Science Council and the Ministry of Science and Technology of the Republic of China (Contract No.: NSC 100 - 2410 - H - 035 – 041- and MOST 103-2218-E-035-014-).

REFERENCES

[1] Y. Bengio, A. Courville, P. Vincent. "Representation Learning: A Review and New Perspectives". *IEEE Trans. PAMI, special issue Learning Deep Architectures*, 2013.

[2] Le, Q.V., "Building high-level features using large scale unsupervised learning," *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pp.8595-8598, 2013.

[3] Rigamonti, R.; Sironi, A.; Lepetit, V.; Fua, P., "Learning Separable Filters," *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pp.2754-2761, 2013.

[4] Adam Coates, Andrew Y. Ng. "Learning Feature Representations with K-means". *In Neural Networks: Tricks of the Trade, Reloaded*, Springer LNCS, 2012.

[5] L. Shao, L. Liu, and X. Li, "Feature learning for image classification via multiobjective genetic programming," *Neural Netw. Learn. Syst. IEEE Trans. On*, vol. 25, no. 7, pp. 1359 – 1371, 2014.

[6] Girshick, R.; Donahue, J.; Darrell, T.; Malik, J., "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," *Computer Vision and Pattern Recognition (CVPR)*, pp.580-587, 2014.

[7] J. Arevalo, A. Cruz-Roa, V. Arias, E. Romero, and F. A. González, "An unsupervised feature learning framework for basal cell carcinoma image analysis," *Artif. Intell. Med.*, 2015.

[8] Cheriyadat, A.M., "Unsupervised Feature Learning for Aerial Scene Classification," *Geoscience and Remote Sensing, IEEE Transactions on*, vol.52, no.1, pp.439-451, 2014.

[9] G. Mesnil, S. Rifai, A. Bordes, X. Glorot, Y. Bengio, and P. Vincent, "Unsupervised Learning of Semantics of Object Detections for Scene Categorization," in *Pattern Recognition Applications and Methods*, Springer, 2015, pp. 209 – 224.

[10] Madry, M.; Liefeng Bo; Kragic, D.; Fox, D., "ST-HMP: Unsupervised Spatio-Temporal feature learning for tactile data," *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pp.2262-2269, 2014.

[11] Figueira, D.; Bazzani, L.; Ha Quang Minh; Cristani, M.; Bernardino, A.; Murino, V., "Semi-supervised multi-feature learning for person re-identification," *Advanced Video and*

Signal Based Surveillance (AVSS), 2013 10th IEEE International Conference on, pp.111-116, 2013.

[12] Bengio, Y.; Courville, A.; Vincent, P., "Representation Learning: A Review and New Perspectives," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol.35, no.8, pp.1798-1828, 2013.

[13] Joseph Turian; Lev Ratinov; Yoshua Bengio. "Word representations: a simple and general method for semi-supervised learning," *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, 2010.

[14] Adam Coates , Andrew Y. Ng. "An analysis of single-layer networks in unsupervised feature learning," *Int'l Conf. on AI and Statistics (AISTATS)*, 2011.

[15] Alireza Fathi, Xiaofeng Ren, James M. Rehg, "Learning to Recognize Objects in Egocentric Activities," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[16] Miller, Claire Cain. "Google Searches for Style". *The New York Times*, Retrieved 2013.

[17] Kjeldsen, R.; Kender, J., "Finding skin in color images," *Automatic Face and Gesture Recognition, 1996., Proceedings of the Second International Conference on*, pp.312-317, 1996.

[18] W. Gonzalez, and S. Eddins, *Digital Image Processing Using Matlab*, Image Processing.

[19] Hyvarinen, A., Oja, E.: "Independent component analysis: algorithms and applications," *Neural Networks*, pp. 411-430, 2000.

[20] Jarrett, K., Kavukcuoglu, K., Ranzato, M., LeCun, Y. "What is the best multistage architecture for object recognition," *In 12th International Conference on Computer Vision*, pp. 2146-2153, 2009.

[21] Tatt Hee Oong, Isa, N.A.M., "Adaptive Evolutionary Artificial Neural Networks for Pattern Classification," *IEEE Transactions on Neural Network*, vol. 22, no. 11, 2011.

[22] B. Kosko, *Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence*, Prentice-Hall, Upper Saddle River, NJ, USA, 1992, ch2.

[23] M. T. Hagan and M. B. Menhaj, "Training feedforward networks with the Marquardt algorithm," *IEEE Trans. on Neural Networks*, vol. 5, no. 6, pp.989–993, 1994.

Table 1. Comparison of segmentation results at different resolutions.

Method	Resolution	# of pixels	# of features	Accuracy (%)
NN [23]	203×360	7308000	27	80.70
NN [23]	405×720	29160000	27	81.08
FLSA	203×360	7308000	16	82.43
FLSA	405×720	29160000	16	82.62

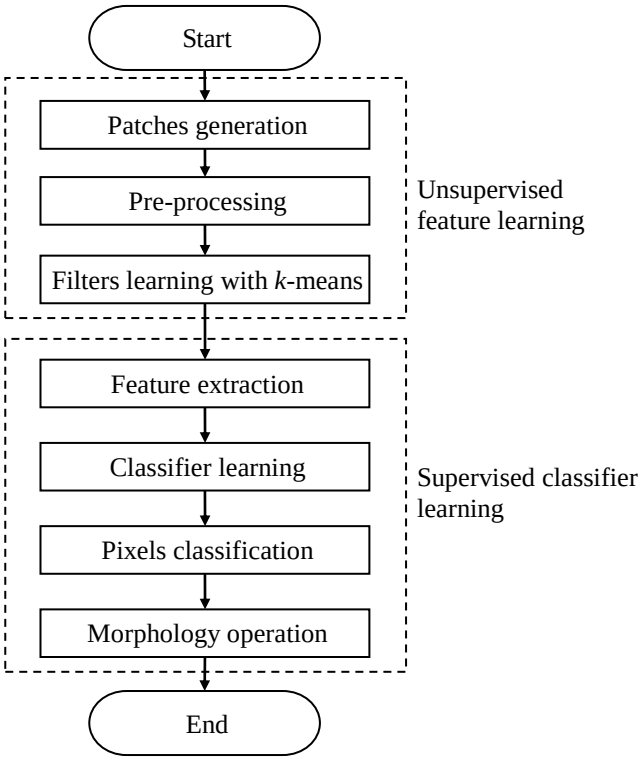


Fig. 1. Feature learning-based segmentation algorithm flowchart.

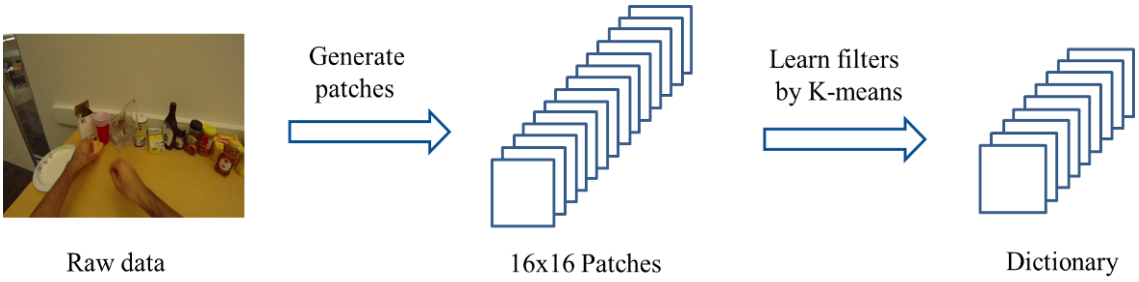


Fig. 2. Feature learning flowchart.

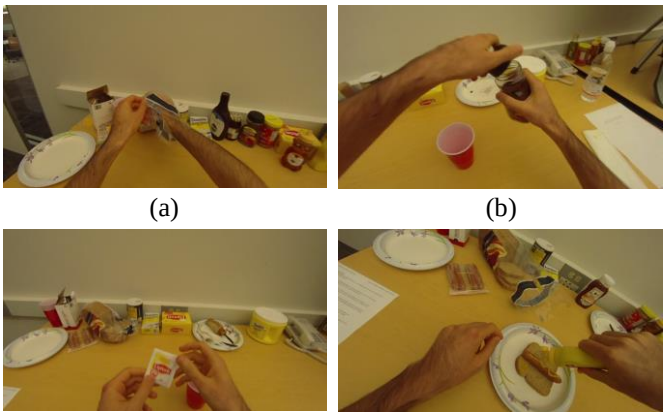


Fig. 3. Examples of a GTEA data set: (a) cheese, (b) coffee, (c) tea, and (d) hot dog.

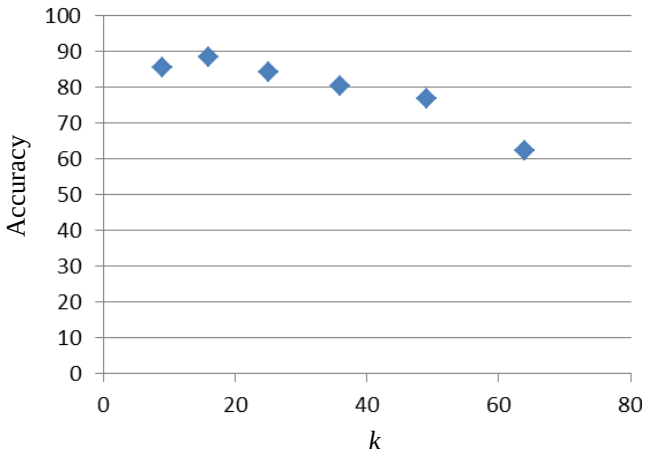


Fig. 4. Decision of the parameter *k* in K-means clustering.

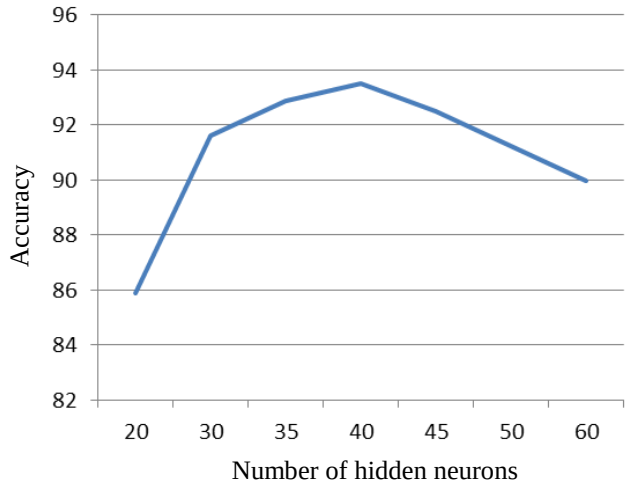
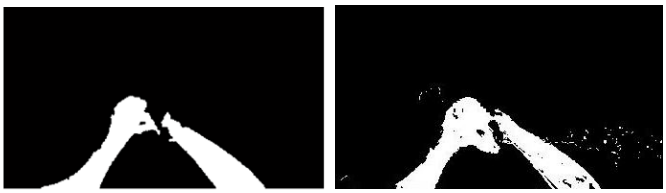


Fig. 5. Decision of the number of hidden neurons in the NN.

Fig. 6. Centroids learned by the K-means algorithm.



(a) (b)

Fig. 7. Examples of hand segmentation: (a) ground-truth data and (b) segmented result from using the FLSA.