Using Stacked Sparse Auto-Encoder and Superpixel CRF for Long-Term Visual Scene Understanding of UGVs

Zengshuai Qiu, Yan Zhuang, *Member, IEEE*, Huosheng Hu, *Senior Member, IEEE*, and Wei Wang, *Senior Member, IEEE*

***Abstract*—Multiple images have been widely used for scene understanding and navigation of unmanned ground vehicles in long term operations. However, as the amount of visual data in multiple images is huge, the cumulative error in many cases becomes untenable. This paper proposes a novel method that can extract features from a large dataset of multiple images efficiently. Then the membership *K*-means clustering is used for high dimensional features, and the large dataset is divided into *N* subdatasets to train *N* conditional random field (CRF) models based on superpixel. A Softmax subdataset selector is used to decide which one of the *N* CRF models is chosen as the prediction model for labeling images. Furthermore, some experiments are conducted to evaluate the feasibility and performance of the proposed approach.**

***Index Terms*—Conditional random field (CRF), long term navigation, scene understanding, stacked sparse auto-encoder.**

I. Introduction

VISION, as an important sensor for the unmanned ground vehicles (UGVs) to perceive the outdoor environment, has always been attracted many scholars to work on it. It has been widely deployed in multisensor navigation system for simultaneous map construction and positioning. The scene understanding is the key to the development of a vision system, adding attributes to the map and scene constructed by other sensors, such as laser, sonar, and odometry. Meanwhile, scene understanding is one of the most challenging and fundamental problems in computer vision, especially for long-term navigation of UGVs [1]. It aims to assign an object label to each pixel of a given image. The labels correspond to various estimated properties of an image and may be for example an object class label (road, car, sky, building, etc.) in the case of object class image segmentation [2]–[4].

Outdoor scene contains multiple classes of context, such as sky, buildings, plants, and road, which usually have their own specific location. Therefore, it is a big challenge to conduct multiclass object localization and imbalanced data classification for outdoor scenes. Considering the boundary information, context information between regions and context information between objects in the image, Carolina *et al.* [5] proposed a classification model based on three different layers: 1) pixel level; 2) region layer; and 3) object layer. The model is a similarity measure function which is composed of a number of kernel functions so as to explain the interaction of each object. The model combined with conditional random field (CRF) achieved better results in image semantic segmentation. However, the method could not meet the real-time requirements of UGVs due to inference an image depending on each pixel.

Wojek *et al.* [6] proposed a model which was mainly to solve the problem of multiclass object recognition and tracking. In order to improve the accuracy of recognition and tracking, the frame rate of the acquired images was increased and the multiframe images were weighted so that better experimental results are obtained. The focus of this method was to solve the accuracy of multiclass identification but overlooked the fact that UGVs need to work a long time in outdoor scene so as to lead to cumulative error of models.

The inevitable problem of the outdoor scene is the effect of illumination. The recognition performance of different light intensity is different from the same classifier. Upcroft *et al.* [7] put forward an illumination invariant model which transformed the image under different light intensities into the same light intensity and combined with RGB three channel of the image resisted illumination interference. But this paper overlooks the context relationship. Whether the efficiency of the algorithm can meet the UGVs real-time scene understanding is also critical.

In order to meet the real-time requirements of unmanned vehicle outdoor scene understanding, Scharwächter *et al.* [8] proposed a feature extraction method based on sparse representation of continuous multiframe images to estimate the current scene change information and used CRF method to get the scene semantic segmentation. The algorithm greatly reduces the inference time of the CRF model. There is a limitation of the algorithm that recognition accuracy will decreases to some extent.

Manuscript received December 9, 2016; revised May 8, 2017; accepted July 24, 2017. This work was supported in part by the National Natural Science Foundation of China under Grant 61375088 and Grant U1608253, and in part by the Fundamental Research Funds for the Central Universities under Grant DUT16TD15. This paper was recommended by Associate Editor K. Huang. *(Corresponding author: Yan Zhuang.)*

Z. Qiu, Y. Zhuang, and W. Wang are with the School of Control Science and Engineering, Dalian University of Technology, Dalian 116024, China (e-mail: ai_qzs@mail.dlut.edu.cn; zhuang@dlut.edu.cn; wangwei@dlut.edu.cn).

H. Hu is with the School of Computer Science and Electronic Engineering, University of Essex, Colchester CO4 3SQ, U.K. (e-mail: hhu@essex.ac.uk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMC.2017.2735635

In order to improve the recognition accuracy of the outdoor scene, considering the different scenes through the grids, semantic labels and data statistics, Geiger *et al.* [9] proposed likelihood functions and used contrastive divergence (CD) algorithm to get the optimal parameter estimation in [9]. The algorithm has a better effect on small sample data, however, with the increasing number of test samples, the robustness of the algorithm decreases rapidly. Due to the fact that UGVs need a long term operation and keep a high recognition accuracy, designing a single prediction model-based the whole data could result in cumulative error.

In order to reduce the cumulative error of the algorithm, Willem *et al.* [10] applied the online learning method and the semi-supervised learning method to outdoor scene recognition. The method tries to update the original training sample set selectively every time using the recognition results and optimizes the parameter estimation model of the classifier based on the color feature, which could reduce the accumulation of errors caused by long-term recognition. If a new emergent category was not included in the train dataset, this predicting model could not achieve an accurate recognition rate. For a better outdoor scene modeling, Nedovic *et al.* [11] made an in-depth study on the outdoor scene reconstruction by 3-D visual information.

Based on the previous research overviewed above, this paper presents a novel stacked sparse auto-encoder (SSAE) + CRF framework for long term scene understanding of UGVs. Currently, the problem of a large amount of image data classification has been largely solved by the deep learning theory. The features representation of a large amount of image data is difficult. However, the deep learning solved the representation of the high-level features of the image through unsupervised training high-level feature extraction model. The feature extraction model extremely ensures a one-to-one mapping relationship between images and features. The accuracy of the classifier is improved by the effectiveness of the feature extraction. With the high-level features of images, we can cluster the dataset into *N* subdatasets in order to train CRF prediction model. Contextual reasoning techniques, and achieve a better performance than traditional local classifiers [12]. Therefore, in this paper, we consider clustering the dataset into a plurality of subdatasets though the features extracted by SSAE. Then, the prediction model is obtained through off-line training of each subdataset by CRF. Lastly, prediction models will be deployed to complete the selective labeling of outdoor scenes.

This paper is organized as follows. Section II overviews the proposed approach, including experimental platform and algorithm design. Section III describes how to extract features with a high dimension. Section IV presents multiple the training and prediction of CRF models. Experimental results are presented in Section V to verify the feasibility and performance of the proposed method. Finally, a brief conclusion and future studies are given in Section VI.

II. System Overview

Fig. 1 shows our UGV platform used for implementing the algorithms, which is called Smart-Cruiser. It is a home-developed UGV equipped with multiple lasers, industrial personal computers (IPCs) and monocular camera. Our research work in this paper only utilizes monocular vision information from the camera, Fly capture flea3, which is mounted on the top of our Smart-Cruiser UGV platform, as shown in Fig. 1(a). It has 3.2 million pixels (2080 × 1552) and a frame rate of 60 frames/s. Two advantech IPCs with InterCorei7 CPU, 24GRAM, GTX970 graphics card, DDR3 SSD, and one D-Link DKVM-L708H.

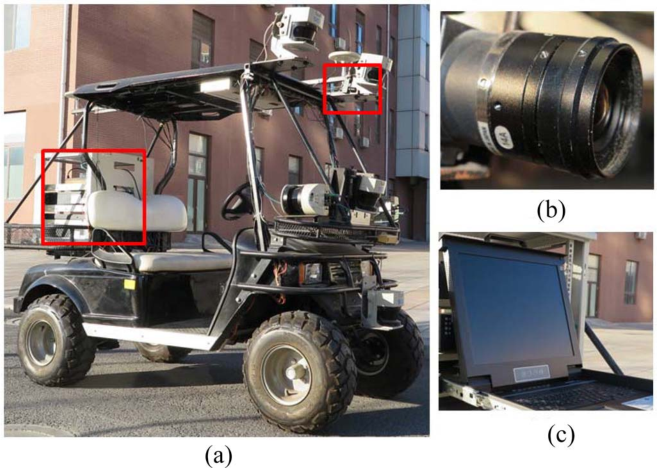


Fig. 1. (a) Smart-Cruiser, a home-developed UGV equipped with multiple lasers and monocular camera. (b) Advantech IPCs. (c) Fly capture flea3 camera.

Fig. 2 shows the training process to be conducted in this paper. As can be seen, different scenes in Fig. 2(a) are used for training the feature extraction model by stacked sparse auto-encoder, the extracted features of each image is presented in Fig. 2(c). Then Fig. 2(d) shows the training of the dataset classifier and the CRF predicting model for each subsubset. Fig. 3 describes the prediction process: 1) feature extraction; 2) model selector; and 3) the CRF predicting model for each subdataset.

III. Unsupervised Images Feature Learning With Stacked Sparse Auto-Encoder And Clustering for High Dimensional Feature

In this paper, a stacked sparse auto-encoder is deployed with deep learning theory to extract and cluster features with a high dimension. A number of key issues are addressed for model construction, namely the selection of the number of model layers, the analysis of experimental data, and the selection of the number of nodes on each layer [13]. The final output of the model is deployed to complete the feature extraction in large data sets based on the back-propagation algorithm. The high dimension of features extracted by stacked sparse auto-encoder method on original datasets is divided into *n* subdatasets, and the membership *K*-means algorithm is used to solve the problem [14].

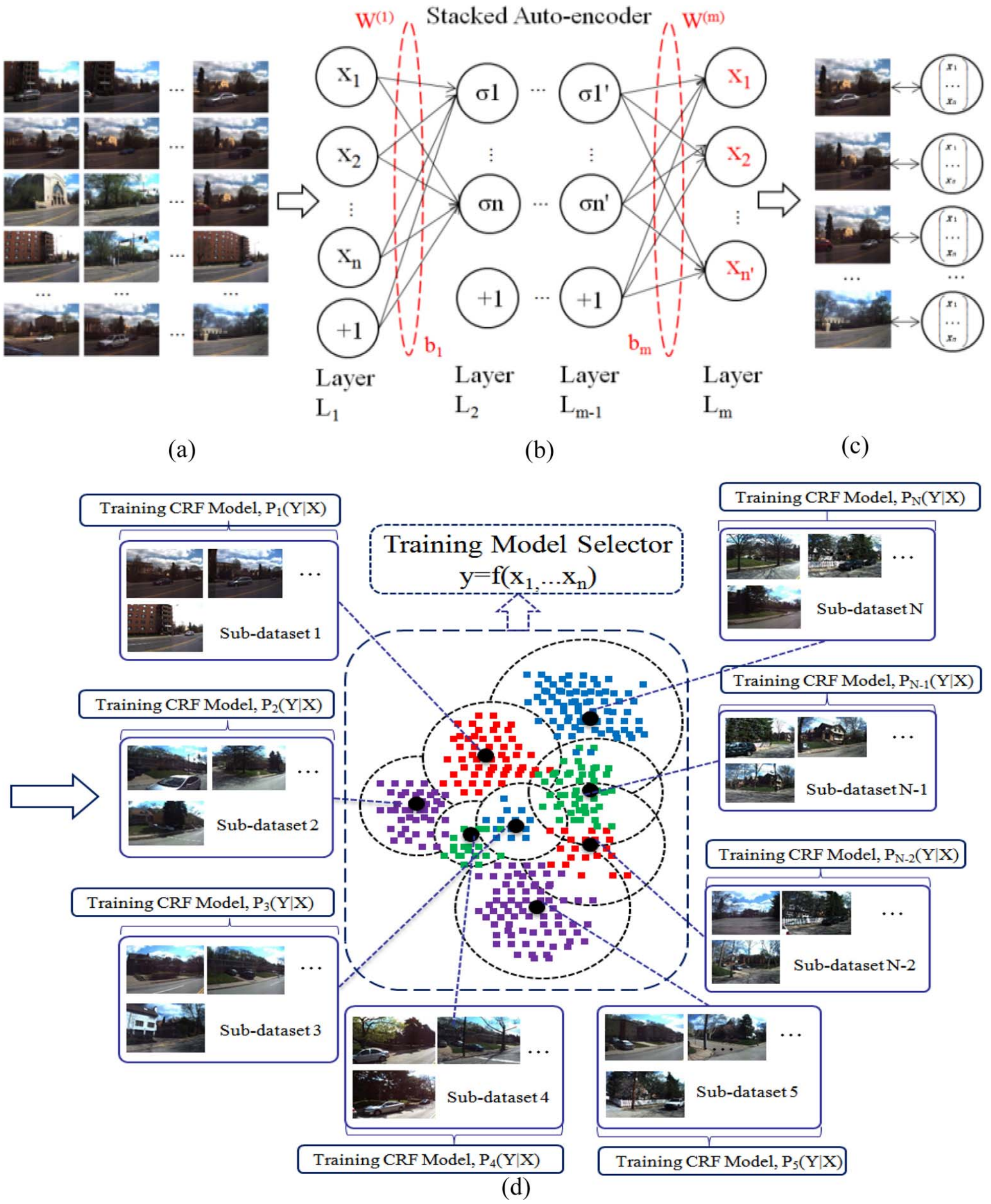

W(1)
Stacked Auto-encoder
W(m)
x1
x2
xn
+1
σ1
σn
+1
σ1'
σn'
+1
x1
x2
xn'
b1
bm
Layer L1
Layer L2
Layer Lm-1
Layer Lm
(a)
(b)
(c)
Training CRF Model, P1(Y|X)
Sub-dataset 1
Training Model Selector y=f(x1,...xn)
Training CRF Model, PN(Y|X)
Sub-dataset N
Training CRF Model, P2(Y|X)
Sub-dataset 2
Training CRF Model, PN-1(Y|X)
Sub-dataset N-1
Training CRF Model, PN-2(Y|X)
Sub-dataset N-2
Training CRF Model, P3(Y|X)
Sub-dataset 3
Sub-dataset 4
Sub-dataset 5
Training CRF Model, P4(Y|X)
Training CRF Model, P5(Y|X)
(d)

Fig. 2. Training process. (a) Different scenes, (b) training feature extraction model by stacked sparse auto-encoder, (c) extracting the features of each image, and (d) training the dataset classifier and the CRF predicting model for each subsubset.

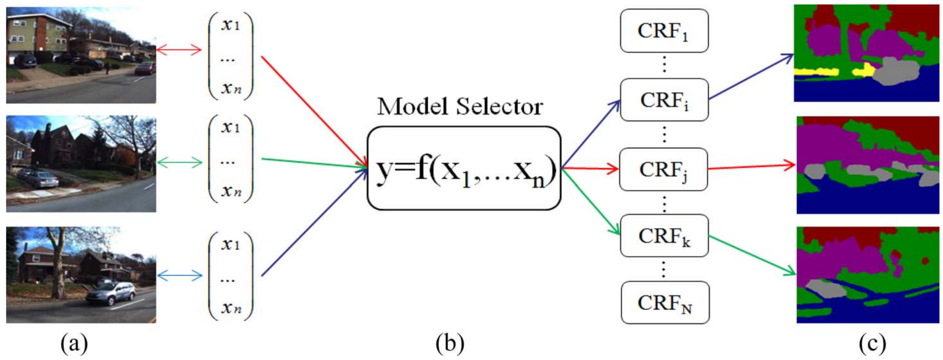


Fig. 3. Predicting, (a) feature extraction, (b) model selector, and (c) CRF predicting model for each subdataset.

A. Feature Learning

1) Auto-Encoders and Sparsity: Fig. 4 shows the schematic of a stacked auto-encoder, i.e., a 3-layer neural network. Its input vector is a set of unlabeled training examples, $\{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\}$, where $x^{(i)} \in \Re^n$. The unsupervised learning is used here for feature learning [26], [27].

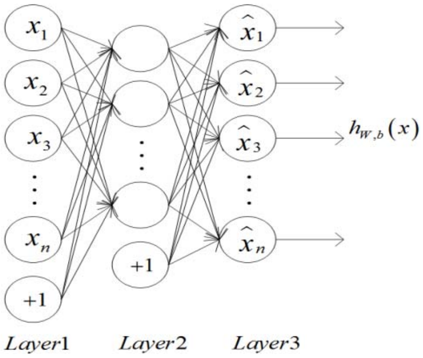


Fig. 4. Schematic of 3-layer auto-encoder.

The auto-encoder tries to learn a function

$$\widehat{x} = h_{w,b}(x) \approx x \tag{1}$$

where x is the input vector and $x \in [0, 1]$, while $w = \{w^{(1)}, w^{(2)}\}$ and $b = \{b^{(1)}, b^{(2)}\}$ represent the weights and the biases of both layers.

If the number of the hidden layer nodes is S_2, we have

$$a^{(1)} = x, z^{(2)} = W^{(1)}a^{(1)} + b^{(1)}, a^{(2)} = f\left(z^{(2)}\right) \tag{2}$$

where $a^{(1)} \in \Re^n$, $W^{(1)} \in \Re^{s_2 \times n}$, $b^{(1)} \in \Re^n$, $Z^{(2)} \in \Re^{S_2}$, $a^{(2)} \in \Re^{S_2}$. $f(z)$ denotes an element wise application of the logistic sigmoid, $f(z) = 1/1 + \exp(-z)$, Next we have

$$\begin{aligned} z^{(3)} &= W^{(2)}a^{(2)} + b^{(2)} \\ a^{(3)} &= f\left(z^{(3)}\right) = h_{(w,b)}(x) = \widehat{x} \end{aligned} \tag{3}$$

where $b^{(2)} \in \Re^n$, $z^{(3)} \in \Re^n$, $W^{(2)} \in \Re^{n \times s_2}$, and $a^{(3)} \in \Re^n$. The average of feedback error is used as the cost function

$$\begin{aligned} J(W, b) &= \left[\frac{1}{m}\sum_{i=1}^{m} J\left(W, b; x^{(i)}\right)\right] + \frac{\lambda}{2}\sum_{l=1}^{n_l-1}\sum_{i=1}^{s_l}\sum_{j=1}^{s_l}\left(W_{ji}^{(l)}\right)^2 \\ &= \left[\frac{1}{m}\sum_{i=1}^{m}\left(\frac{1}{2}\left\|h_{(w,b)}\left(x^{(i)}\right) - x^{(i)}\right\|^2\right)\right] \\ &\quad + \frac{\lambda}{2}\sum_{l=1}^{n_l-1}\sum_{i=1}^{s_l}\sum_{j=1}^{s_l}\left(W_{ji}^{(l)}\right)^2 \end{aligned} \tag{4}$$

where the first item means square error, the second is a regularization term to make limited range of weights in order to avoid over-fitting phenomenon, λ is the attenuation coefficient of weight balancing the two items in the formula. n_l is the number of the layers of neural network, here $n_l = 3$. S_l is the number of nodes in layer l, m is the total number of examples.

One common method for imposing sparsity is to limit the activation of hidden units a using the Kullback–Leibler (KL) divergence function [15], [16]. Let $a_j^{(i)}(x)$ denote the activation of hidden unit j with respect to the input $x^{(i)}$. Then, the average activation of this hidden unit is

$$\widehat{p_j} = \frac{1}{m}\sum_{i=1}^{m}\left[a_j^{(i)}(x)\right]. \tag{5}$$

To enforce sparsity, we constrain the average activation $\widehat{p_j}$ and p, where p is the sparsity parameter chosen to be a small positive number near 0. To use this constraint in (4), we try to minimize the KL divergence similarity between $\widehat{p_j}$ and p

$$J_{kl}(p\|\widehat{p}) = \frac{1}{m}\sum_{i=1}^{s_2} p \log\frac{p}{\widehat{p_j}} + (1-p)\log\frac{1-p}{1-\widehat{p_j}}. \tag{6}$$

The final cost function for learning an SAE becomes is

$$J_{\text{sparse}}(w, b) = J(w, b) + \beta J_{kl}(p\|\widehat{p}) \tag{7}$$

where β controls the sparsity penalty term.

For the objective function $J_{\text{sparse}}(w, b)$ we use the gradient-descent algorithm to compute optimistic evaluation: (w, b). To incorporate the KL-divergence term into our derivative calculation, a simple-to-implement trick involves a small change to our code. For each output unit i of the layer n_l (output layer), we calculate the residue according to the following formula:

$$\delta_i^{(n_l)} = \frac{\partial}{\partial z_i^{(n_l)}}\frac{1}{2}||x - h_{W,b}(x)^2||^2 = -\left(x_i - a_i^{(n_l)}\right) \bullet f'\left(z_i^{(n_l)}\right) \tag{8}$$

where "$\bullet$" denotes the element-wise product operator.

For each node i in layer $l(1, 2, \ldots, n_l - 1)$, we calculate the residue according to the following formula:

$$\delta_i^{(l)} = \left(\sum_{j=1}^{S_l} W_{ji}^{(l)}\delta_j^{(l+1)} + \beta\left(-\frac{p}{\widehat{p_i}} + \frac{1-p}{1-\widehat{p_i}}\right)\right)f'\left(z_i^{(l)}\right). \tag{9}$$

Then we compute the partial derivatives as follows:

$$\frac{\partial}{\partial W_{ij}^{(l)}}J(W, b; x, y) = a_j^{(l)}\delta_i^{(l+1)} \tag{10}$$

$$\frac{\partial}{\partial b_i^{(l)}}J(W, b; x, y) = \delta_i^{(l+1)}. \tag{11}$$

2) Stacked Sparse Auto-Encoder: As shown in Fig. 4, a stacked auto-encoder is a neural network consisting of three layers of sparse auto-encoders in which the outputs of each layer is wired to the inputs of the successive layer. A good way to obtain accurate parameters for a stacked auto-encoder is to use greedy layer-wise training [17].

To do this, we train the first layer on raw inputs to obtain parameters $W^{(1,1)}$, $W^{(1,2)}$, $b^{(1,1)}$, $b^{(1,2)}$, then use the first layer to transform the raw input into a vector consisting of activation of the hidden units. The second layer is trained on this vector to obtain parameters $W^{(2,1)}$, $W^{(2,2)}$, $b^{(2,1)}$, $b^{(2,2)}$. Repeatedly, we use the output of the previous layer as input for the subsequent layer. In this way, the parameters of each layer are trained individually while freezing parameters for the remainder of the model. The structures are shown in Fig. 5.

B. Dividing Dataset to N-Subdatasets by Membership K-Means

1) Membership K-Means: K-means algorithm consists of four processes [18]. First, select K objects from N data objects randomly as the initial clustering centers. Second, calculate the distances between each data and the clustering centers

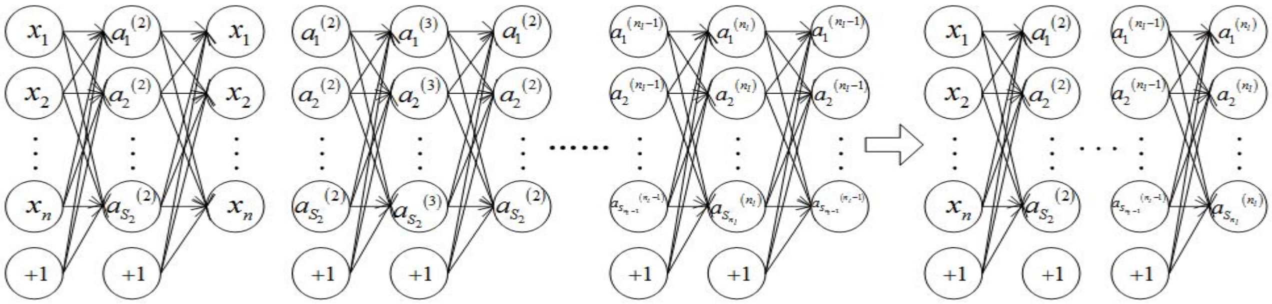


Fig. 5. Structure of stacked sparse auto-encoder.

according to the mean of each cluster object (clustering center) and redivide the corresponding data according to the minimum distance. Third, recalculate the mean of each cluster (clustering center). Finally, repeat the second and third steps until each clustering center no longer changes. But when the dimension of the data is particularly high, the computing time of the algorithm will increase greatly, so it cannot be applied. Since the second step of traditional K-means algorithm needs to calculate the distance to all objects, this will undoubtedly enhance the complexity and redundancy of the algorithm. In this paper, we use the membership K-means algorithm proposed in [14]. There is not necessary to calculate the distances between the center and the data which close to the center when the K-means algorithm has been executed several times. Therefore, the main idea of this algorithm is to define a membership set of clusters for each data. Then we can determine whether the membership set of clusters is valid by setting some numbers. For example, in one K-means algorithm iteration, the cluster is not considered as a membership set if the number of data in a cluster increases. In the next iteration, the distances between the data and the center will not be calculated. On the contrary, the traditional K-means algorithm is used to calculate continuously. This algorithm can greatly reduce the computational complexity and improve the speed and feasibility of high-dimensional data clustering.

2) Clustering Evaluation: In order to evaluate the clustering results and analysis SSAE feature extraction model, we use two clustering criterions: 1) Calinski–Harabasz and 2) Davies–Bouldin [19], [20].

a) Calinski–Harabasz: Data set $D = \{x_1, x_2, \ldots, x_n\}$, dividing D into NC subsets with hard clustering algorithm, then $D = \{C_1, C_2, \ldots, C_{\mathrm{NC}}\}$, c_i is called subclass of D, c denotes the center of D, c_i denotes the center of C_i, n_i denotes the number of objects in c_i and $d(x_a, x_b)$ denotes the distance between x_a and x_b. In this paper we make the use of Calinski–Harabasz index which is defined as

$$\mathrm{CH}(\mathrm{NC}) = \frac{\frac{1}{\mathrm{NC}-1}\sum_{i=1}^{\mathrm{NC}} n_i d^2(c_i, c)}{\frac{1}{n-\mathrm{NC}}\sum_{i=1}^{\mathrm{NC}}\sum_{x\in c} d^2(c_i, c)}. \tag{12}$$

Calinski–Harabasz index is used to compute the square of the distance between the center of the class and each point in the class to measure the compactness within the class, and compute the square of the distance between the center of the dataset and the center of each class to measure the separativity of the dataset. The larger Calinski–Harabasz index, obtained by the separability to the compactness ratio, means the compact of the class and the separation among classes. This means the better result of a cluster.

b) Davies–Bouldin: Davies-Bouldin criterion is based on a ratio of within-cluster and between-cluster distances. Davies–Bouldin index is defined as

$$\mathrm{DB} = \frac{1}{k}\sum_{i=1}^{k}\max_{j\neq i}\{D_{i,j}\} \tag{13}$$

where $D_{i,j}$ is the within-to-between cluster distance ratio for the ith and jth clusters. In mathematical terms

$$D_{i,j} = \frac{(\overline{d_i} + \overline{d_j})}{d_{i,j}} \tag{14}$$

where $\overline{d_i}$ is the average distance between each point in the ith cluster and the centroid of the ith cluster. $\overline{d_j}$ is the average distance between each point in the jth cluster and the centroid of the jth cluster. $d_{i,j}$ is the Euclidean distance between the centroids of the ith and jth clusters. The maximum value of $D_{i,j}$ represents the worst-case within-to-between cluster ratio for cluster i. The optimal clustering solution has the smallest Davies–Bouldin value. In contrast with Calinski–Harabasz, the lower numbers of DB means the better performance in Davies–Bouldin.

C. Subdatasets Selection

After dividing the datasets into n subdatasets, we need to determine which subdataset the test image belongs to. If the test image is classified into one of subdatasets, we could use the CRF model that has trained on this subdataset to complete the semantic segmentation of the test image. Therefore, we choose Softmax selector [21] to select CRF models that have trained on all subdatasets in this paper. There are two reasons for our choice. The first is that the result of Softmax selector is better than others in our experiment. The second reason is that after this phase of SSAE training is completed, we need Softmax selector to complete subdatasets selection to the test image. Note that the back propagation based fine-tuning of Softmax selector can improve the results and tune the parameters of all layers at the same time.

IV. Scene Understanding by Superpixel CRF

A. Superpixel CRF

CRF can be represented as an undirected graph: $G = (V, E)$, where V represents the all nodes (random variables), E represents the connection between the nodes. Assume that X is the observable random variables (features) and Y are the unobservable random variables (labels). Since the standard CRF model should consider all the pixels in the image, the efficiency of the training and prediction is slow. In order to improve the computational efficiency, we use the superpixel segmentation to preprocess the image, and make each superpixel block be considered as a unit so as to train the CRF model, or predict image [22], [23]. In order to ensure a higher image segmentation result during the driving of unmanned vehicles, so we choose SLIC algorithm.

The labeling of condition random field is over the posterior distribution $P(Y|X)$. According to the Hammersley–Clifford theorem, the posterior probability can be defined as the product of all the potential functions of cliques

$$\begin{aligned} P(Y|X) &= \frac{1}{Z}\prod_{c}\varphi_c(Y_c, X) \\ &= \frac{1}{Z}\exp\left(\sum_{i\in\Omega}\varphi_i(Y_i, X) + \sum_{i\in\Omega}\sum_{i\in N_i}\varphi_{ij}\big(Y_i, Y_j, X\big)\right) \end{aligned} \tag{15}$$

where $\varphi_c(Y_c, X)$ is the potential function, Z is the normalization function: $Z = \sum_c(P|X)$, $\varphi_{ij}(Y_i, Y_j, X)$ is the potential function of c clique; $\varphi_i(Y_i, X)$ is the set of all nodes; N_i is neighborhood of node i; The potential function is composed of two parts. Unary energy term $\varphi_i(Y_i, X)$ is used to measure the probability that a node i is marked as Y_i when X is observation vector. In order to describe the spatial context relations between adjacent nodes i and j in the graph, pairwise energy term $\varphi_{ij}(Y_i, Y_j, X)$ is not only related to the mark of node i, but also to the tag of j of neighboring nodes.

B. Potential Function Selection

In the CRF theory, the selection of potential function is more flexible, and different models can be used in different forms of potential function. In this paper, Unary energy term is defined as follow $\varphi(Y_i, X) = \exp(Y_i w^T g_i(X))$, where $w = [\alpha_0, w_1, w_2, \ldots, w_n]$ is weights of $g_i(X)$ and adjusted in parameter estimation, and α_0 is bias; $g_i(X)$ represents all features of the i superpixel block, $g_i(X) = [1, \psi_i(X)]^T$.

In order to improve recognition accuracy, we use the PCA method to extract the features in each superpixel block, and make these features be CRF model features $\psi_i(X)$. In order to make sure that the CRF model is robust, we normalize the feature vector $\psi_i(X)$ by letting its average be zero and the variance be one.

$$\psi_i'(X) = \frac{\psi_i(X) - \mathrm{mean}(\psi_i(X))}{\mathrm{std}(\psi_i(X))}. \tag{16}$$

Note that the pairwise energy term $\varphi_{ij}(Y_i, Y_j, X)$ not only shows the interaction between adjacent blocks, but also reflects the adjacent block feature vector for its marker. Therefore, it should satisfy the fact that when the difference in the features of adjacent superpixel blocks is close to 0, the value of the potential function should be close to 1. On the contrary, when the difference in the features of adjacent superpixel blocks is close to 1, the value of the potential function is close to 0. Consequently, we define the pairwise energy term as follow:

$$\begin{aligned} \varphi\big(Y_i, Y_j, X\big) &= \exp\big(y_a v^T \mu_{ij}(X)\big) \\ \mu_{ij}(X) &= \mathrm{abs}\big(g_i(X) - g_j(X)\big) \end{aligned} \tag{17}$$

where $v = [v_1, v_2, \ldots v_n]^T$ are the weights of μ_{ij} and adjusted in parameter estimation. Y_i and Y_j are the label of adjacent superpixel blocks, respectively. If $Y_i \neq Y_j$, $y_a = 1$ else $y_a = -1$.

As can be seen from above, adjacent superpixel blocks are similar and have less chance to be marked different.

C. Parameter Estimation

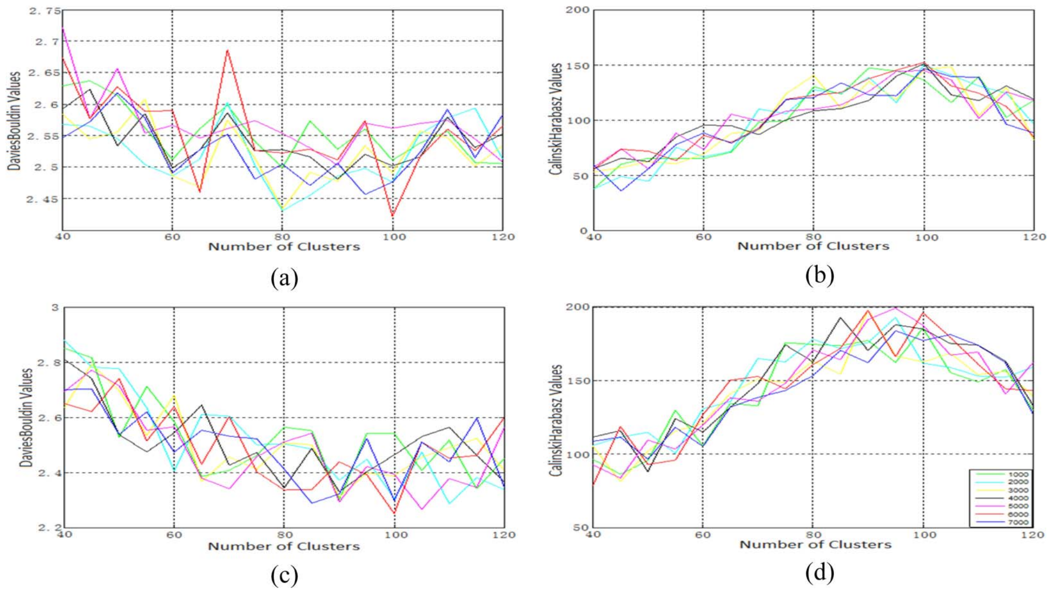
For estimating the parameters $\theta = \{w, v\}$, we assume a set of labeled images $\{X, Y\} = \{X^k, Y^k, k = 1, 2, \ldots K\}$. We train the conditional model discriminatively based on the conditional maximum likelihood criterion [24], which maximizes the log conditional likelihood

$$\begin{aligned} \tilde{\theta} &= \arg_\theta \max\{\log(p(X|Y, \theta))\} \\ &= \arg_\theta \left\{\prod_{k=1}^{K} P\Big(X^k|Y^k, \theta\Big)\right\} \lim_{X\to\infty}. \end{aligned} \tag{18}$$

The stochastic gradient descent algorithm is usually used to solve the optimal solution of the maximum likelihood function. However, the training process is not feasible due to the large number of target model parameters and the complex model structure. There are usually two methods to solve this problem. One is to find a function, which is called surrogate function, to replace the original objective function. Another method is to use the edge probability distribution to solve the problem of original maximum likelihood indirectly. Markov chain Monte Carlo (MCMC) sampling is a typical method for the problem of maximum posterior marginal (MPM). MCMC sampling need to consider a lot of different parameters in order to ensure convergence, which leads to more iterations. We use CD algorithm to solve this problem [23]. CD initializes the Markov chain by preselecting the better parameter set, which greatly reduces the number of iterations of the solution gradient and obtains a relatively good parameter estimation result.

D. Inference for Labeling Image

We need to infer the label of each pixel by CRF model to label a new image. The problem is actually looking for a group Y_i^* that lets $Y_i^* = \arg\max P(Y_i|X, \theta)$ true. Maximum *a posteriori* (MAP) and MPM are usually used to infer labels from the posterior distribution [25]. It is hard to the find the optimal solution of MAP because of the different cases of Y_i^*. The usual solution is to solve the MPM rather than solving the MAP. In this way, the amount of computation is greatly reduced. The computational complexity will be is greatly reduced in this way. MCMC sampling is usually used



(a) (b)

(c) (d)

Fig. 6. (a) and (c) show the performance of Davies–Bouldin values under RSC-*K*-means and membership *K*-means, (b) and (d) show the performance of Calinski–Harabasz values under RSC-*K*-means and membership *K*-means.

to approximate the edge distribution, generating in a series of sample values and completing the optimal solution. Gibbs sampling is an example of MCMC sampling. In this paper, we use Gibbs sampling to complete the inference of the image.

V. Experiments

In order to test outdoor scene understanding of UGV in long time operations, we have chosen the data in this paper that covers the scene in different seasons. With the long term operations of UGV in outdoor scene, the number of test images will also continue increasing. Different sizes of dataset are used to evaluate the performances under different working time. The different performances between single prediction model and multistage prediction model are compared with different data.

A. Datasets

Our framework, presented in this paper, is evaluated on two outdoor scene datasets, one of which is publicly available, namely CMU dataset, and the other one is created by us for the purpose of examining the illumination invariance property. More specifically, CMU dataset is widely used in computer vision in particular (http://3dvis.ri.cmu.edu/data-sets/localization/). Its dataset, covering 12 month outdoor scene, contains 30 000 images of resolution 768 × 1024 pixels with eight classes of objects.

DLUT Campus dataset (http://pan.baidu.com/s/1c10iNlu/) is collected by ourselves in our university campus by using our Smart-Cruiser robot. It was created by our laboratory, equipped with a forward facing camera in order to evaluate the performance and robustness of our framework under the changing lighting conditions. DLUT Campus dataset contains 20 000 images of resolution 480*640 pixels with eight classes of objects. In order to create illumination variation, the robot was driven through the same route at cloudy day, rainy day, and three times of the fine day (spcifically starting at morning, noon, and afternoon, respectively).

B. Feature Extraction and Clustering Analysis

In order to design the data set feature extraction model and analyze the data clustering effectively, as well as find the optimal relationship between them, we need to confirm more accurate feature extraction model parameters, such as the number of layers and nodes, clustering algorithm, and cluster number according to experimental results in a large number of different cases.

1) SSAE Model Selection: In our experiment, for 16 000 images from CMU dataset, we use a 4-layer SSAE network, and the number of nodes of the first three layers is 57 600, 25 088, and 12 544, respectively. The number of the nodes in the first layer is equal to the size of image which is resized 120 × 160 × 3. For the number of hidden nodes of the last layer, we change the number of hidden nodes from 1000 to 7000, and consider that relaying on two clustering algorithms for high dimensional data (membership *K*-means and RSC-*K*-means [28]), and two cluster evaluation criterions (Davies–Bouldin and Calinski–Harabasz). The different combinations between them are shown in Fig. 6.

As can be seen from the curves in Fig. 6, the Davies–Bouldin value is the better while it is lower, and the Calinski–Harabasz value is the better while it is higher. We can see that optimal number of clusters should be between 90 and 110, and therefore we choose 100. When the number of clusters is 100, we can see from the figures that the Davies–Bouldin value on membership *K*-means is lower than on RSC-*K*-means and the Calinski–Harabasz value on membership *K*-means is larger than on RSC-*K*-means, so membership *K*-means is better than RSC-*K*-means according to the performance of two cluster evaluation criterions. Thus, we choose membership *K*-means as our cluster algorithm in

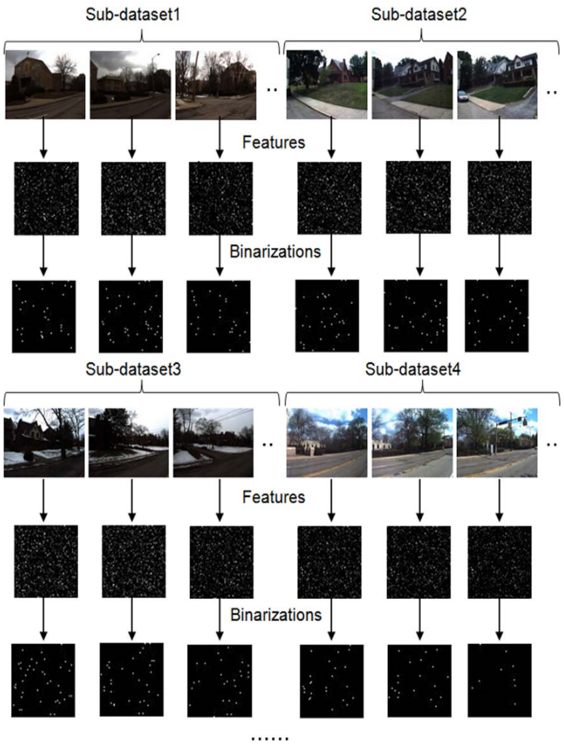


Fig. 7. Features visualization process.

this paper. In order to choose the best SSAE model, we consider optimal number of clusters and clusters algorithm should be 100, membership K-means, respectively. The red curve is better than other curves.

Consequently, we can draw a conclusion that the number of hidden nodes of the last layer should be 6000, to be taken as the features of the input image. The visualization of the features is given in Fig. 7 to facilitate the observation of the relationship of the features. First, the first 5929-dimensional elements are extracted from the 6000-dimensional features and the feature values are normalized to 0–255. Then, the 5929 dimensional feature vector is reshaped into a 77*77 dimensional matrix and displaying the feature map. Finally, the feature map is binarized and threshold value is set to 150.

2) Comparison of Feature Learning Methods: In order to analyze the unsupervised feature learning method used in this paper, we compare it with several commonly used nonlinear unsupervised feature learning methods: locally linear embedding (LLE) and kernel principal component analysis (KPCA). The main idea of LLE is to use the local linearity of the data to approximate the global linearity. It is assumed that any sample point can be expressed as a linear combination of its neighboring sample points. Keeping the neighborhood linearly invariant while finding the low-dimensional embedding of the data, which can make the data had dimensionality reduced maintain the original topology invariant.

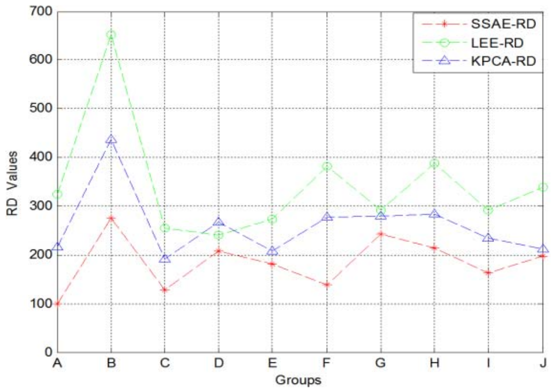


Fig. 8. Result of RD.

The main idea of KPCA is to project the input space into the new feature space through the nonlinear mapping, and then do PCA processing in the new feature space, which has a strong nonlinear processing ability. In this paper, the Gaussian radial basis function kernel function is chosen as the KPCA kernel function. In order to compare the SSAE used in this paper with LLE algorithm and KPCA algorithm, we prepare ten groups of image data that are recorded as A, B, C, D, E, F, G, H, I, and J which are similar in the same scene and quite different between each other. Each group of image data is composed of two scenes and each scene contains 320 images. This makes it easier for us to compare the two feature extraction algorithms.

Then we calculate the ratio of the distance between classes and the distance within the class. The distance between classes can express the difference between the two classes. The sum of the distance within the class can express the difference in the size of each category. Because of the diversity of the data group, it cannot rely solely on these two statistics. Therefore, in this paper the size of RD is used to measure the result of feature extraction

$$\mathrm{Dis1} = \sum_{i=1}^{2} \mathrm{sqrt}\left(\sum_{j=1}^{320} \left(x_j - u_i\right)^T \left(x_j - u_i\right)\right) \tag{19}$$

$$u_i = \frac{1}{320} \sum_{j=1}^{320} x_{ij},\ i = 1, 2 \tag{20}$$

$$\mathrm{Dis2} = \mathrm{sqrt}\left((u_1 - u_2)^T (u_1 - u_2)\right) \tag{21}$$

$$\mathrm{RD} = \mathrm{Dis1}/\mathrm{Dis2} \tag{22}$$

where the u_i is the center of ith cluster, Dis1 is the sum of intraclass distance, Dis2 is interclass distance, RD is ratio of Dis1 and Dis2. From the definition of RD, we can know that the larger the RD, the worse the feature extraction results. The feature dimension of SSAE is 6000, and the feature dimension of LLE and KPCA is set to 6000 and the eigenvector element is normalized [0, 1] for convenience comparison. Dis1, Dis2, and RD were counted for ten sets of data, shown in Fig. 8.

From the figures, we can know the algorithm of SSAE is better than the latter two algorithms by comparing SSAE_RD, LLE_RD, and KPCA_RD. For the latter two algorithms, KPCA algorithm is stronger than the LLE algorithm in terms of stability.

3) Subdataset Selection: After features extraction of original dataset, we use membership K-means to divide the original dataset into 100 subdatasets (called target datasets) which were labeled as $1, 2, \ldots, 100$. Our purpose is that giving an input image I, we discriminate which target dataset it belongs to.

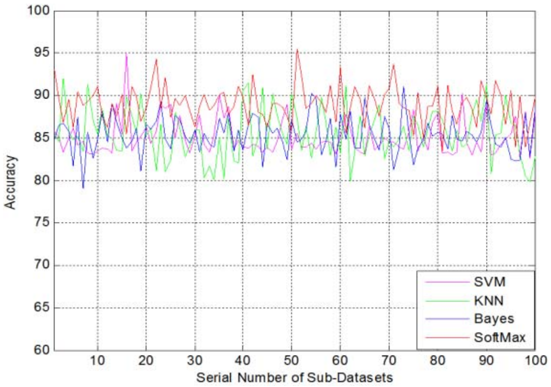


Fig. 9. Serial number of subdatasets classification accuracy.

In this part, we utilized four common classifiers: 1) K-nearest neighbor; 2) Bayes; 3) support vector machine (SVM); and 4) Softmax classifiers. Fig. 9 shows the classification results of different methods on the subdatasets which are evaluated in terms of accuracy. From the graph, we can see that Softmax classifier is better than the other classifiers in our experiment. Therefore, Softmax classifier is chosen for subdatasets selection in this paper finally.

C. Image Semantic Segmentation

The superpixel segmentation is generally used for image preprocessing and each superpixel block is treated as a target to be identified. Since the feature extraction of superpixel blocks determines the result of semantic segmentation to a certain extent, in order to extract the features effectively, the PCA algorithm is used to extract the feature of superpixel block. In the experiment, the images are resized to 240*320. First, the superpixel segmentation and the grayscale processing are used for the original images. Second, a square region with a length of 14 pixels is determined from the superpixel block and center of the square is equal to the center of the superpixel block. If the square region is completely included in the superpixel block, the 14*14 square is reshaped into 196*1 vector, and then PCA algorithm is used to reduce the vector to 50. If the square region is not completely included in the superpixel block, the upsampling process is performed before the PCA processing. The features processed by PCA are used as the input features of the classifier and classification is completed. The details are shown in Fig. 10. In our experiment, we have utilized four methods (SVM, Adaboost, Bayes, and CRF) to process image semantic segmentation, and the results are shown in Fig. 11.

We understand the fact that images within one superpixel block may have sky on the top, buildings in the middle, and roads on the bottom. It is obvious from the above graph that SVM, Adaboost, and Bayes method only take a look at images within each superpixel block. Therefore, the object classes shall consider spatial relationship and global appearance between superpixel blocks in order to achieve better results. This is an important reason why we utilize the CRF method to conduct image semantic segmentation.

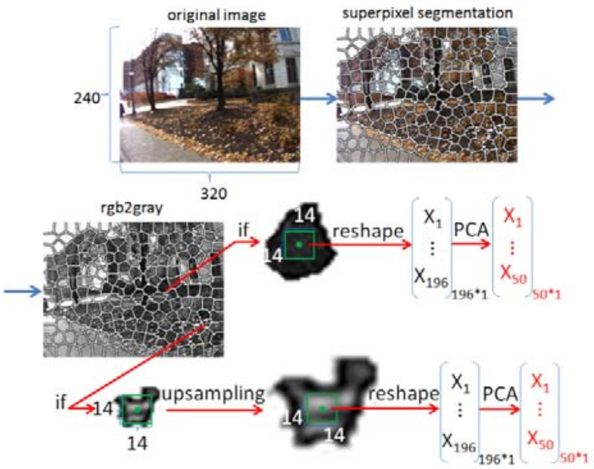


Fig. 10. Feature extraction process of superpixel blocks.

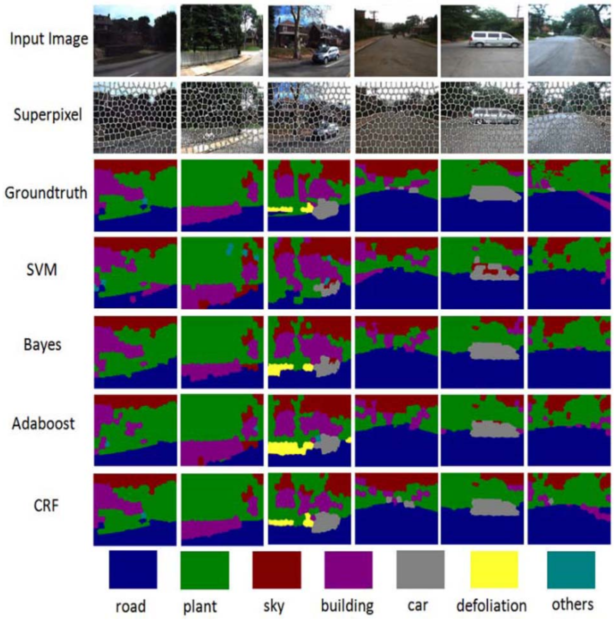


Fig. 11. Representative classification results on testing images from the dataset. 1st-row: input images. 2nd-row: superpixel segmentation results. 3rd-row: labeled images. 4th-row to 7th-row: classification results using the SVM classifier, the Bayes classifier, Adaboost classifier, and the CRF model, respectively.

D. Outdoor Scene Understanding

This paper mainly investigates how UGVs could maintain high recognition accuracy for outdoor scene understanding under long term conditions. Hence, we take a comparison between conventional CRF method and our method (SSAE+CRF) on three different sizes of CMU dataset and DLUT Campus dataset respectively. There are two main differences between conventional CRF and our method (SSAE + CRF). First, feature extraction of conventional CRF is more effective than our method (SSAE + CRF). Second conventional CRF is a single model and our method (SSAE + CRF) contains multiple CRF models and a Softmax model selector. The conventional CRF model is designed to be the

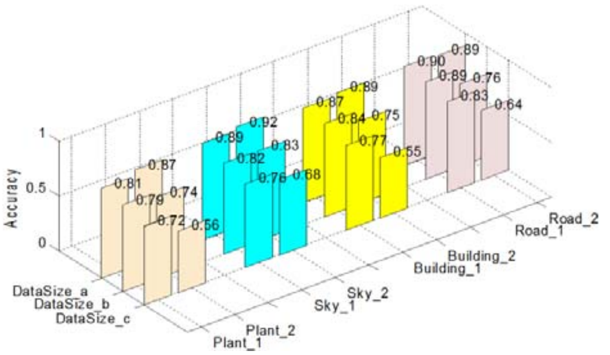


Fig. 12. Accuracy of four categories (plant, sky, building, road) from CUM dataset.

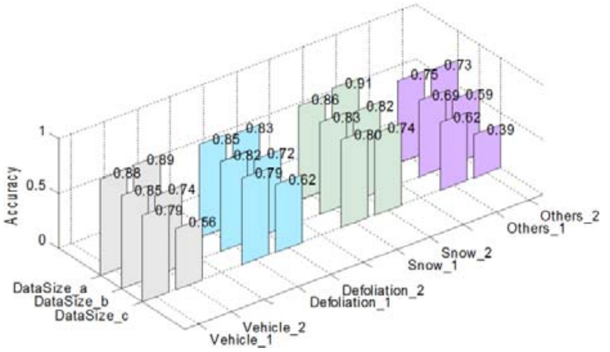


Fig. 13. Accuracy of four categories (vehicle, defoliation, snow, others) from CUM dataset.

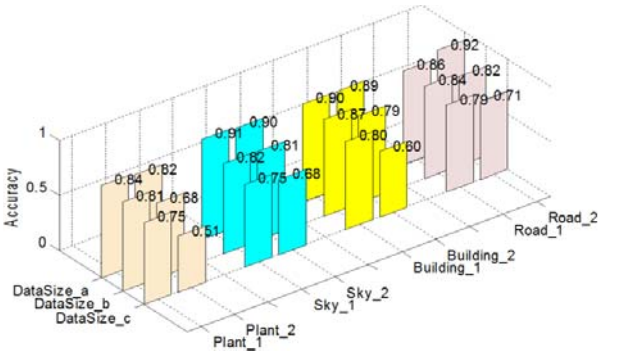


Fig. 14. Accuracy of four categories (plant, sky, building, road) from the DLUT Campus dataset.

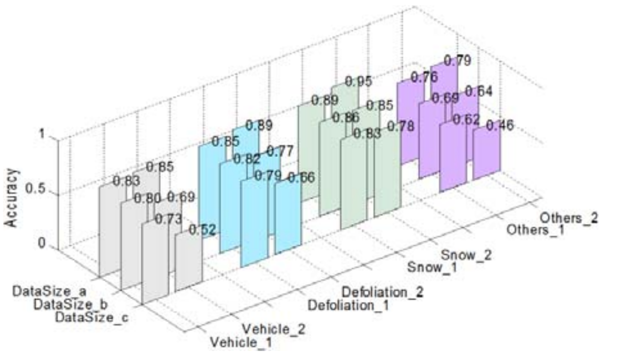


Fig. 15. Accuracy of four categories (vehicle, defoliation, snow, others) from the DLUT Campus dataset.

two-order clique, which can be pixel-based CRF or superpixel-based CRF. Due to the real-time requirement of UGVs, it is designed to be superpixel-based CRF in this paper. Feature extraction is performed for each superpixel, and color histogram is used as color feature, and amplitude frequency of fast Fourier transform is used as texture feature. Considering the fact that our method (SSAE + CRF) contains a Softmax model selector and balances the computing efficiency and recognition accuracy, we reduce the complexity of feature extraction. Fig. 10 has shown in this paper. Both our method (SSAE+CRF) and conventional CRF method are, respectively, used to train three models on the CMU dataset and DLUT Campus dataset according to three different data sizes. In order to facilitate the interpretation of statistical results, we use _1 to mark the results from our method (SSAE+CRF) and _2 to mark the results from the conventional CRF method, respectively.

Figs. 12 and 13 show the performances on CMU dataset. Please note that DataSize_a, DataSize_b, and DataSize_c are three subdatasets extracted from CMU dataset. In DataSize_a, 5000 images are used as training set and other 1300 images are used as testing set. Similarly, in DataSize_b, 8000 images are used as training set and other 2000 images are used as testing set; in DataSize_c, 10 000 images are used as training set and other 3000 images are used as testing set.

For DataSize_a, the conventional CRF method performs slightly better than our method for most categories (plant, sky, building, vehicle, and snow). The main reason is that when dataset DataSize_a is relatively small, differences of the features extracted by SSAE are not obvious so as to decrease distances between clusters, which leads to decreasing the accuracy of SoftMax model selector and CRF models. The conventional CRF method is more suitable for the small dataset application.

For DataSize_b, our method performs better than the conventional CRF method for most categories, but not all. The difference in the identification accuracy of the sky and snow is not obvious, because CRF model considers the contextual relationship of the outdoor scene, and the features of the sky and snow are more obvious.

For DataSize_c, it is clear that our method performs better than the conventional CRF method for all categories on DataSize_c. The main reason is that SSAE method could fully perform unsupervised features learning with the increasing of the data capacity. With better features extracted by SSAE, the distances between the classes increase and the inner distances of classes reduce, which improve the accuracy of the SoftMax model selector and CRF models greatly.

Figs. 14 and 15 show the performances on the DLUT Campus dataset. Please note that DataSize_a, DataSize_b, and DataSize_c are three subdatasets extracted from DLUT dataset. In DataSize_a, 4500 images are used as training set and other 1000 images are used as testing set. Similarly, in DataSize_b, 7800 images are used as training set and other 1800 images are used as testing set; in DataSize_c, 96 000 images are used as training set and other 2700 images are used as testing set. The change trend of statistical data is roughly the same as that of Figs. 12 and 13.

From what we have been discussed above, the performance of our method (SSAE + CRF) is more robust than the conventional CRF method when UGVs is in a long time of operation. Hence, we can draw a conclusion that simple conventional CRF model cannot guarantee the recognition accuracy of outdoor scene as UGVs need understand outdoor scenes and grab an increasing number of images that leads to model cumulative errors. We use SSAE to extract features of dataset and divide dataset to N subdatasets in order to train N CRF models, and then use Softmax classifier to select subdatasets and train CRF models for each subdataset.

VI. Conclusion

In this paper, we have introduced a novel approach that enables the analysis of outdoor scene understanding over large image sets for UGVs in a long term of operation. The important contribution in this paper lies on the realization of designing relatively efficient SSAE model to extract features on the large dataset. According to clustering algorithm for high dimensional data and cluster evaluation criterions and data size, the dataset is divided into N subdatasets to train N prediction models. This effectively reduces the complexity of the direct prediction images on the large dataset, and enhances the practicability of long term outdoor scene understanding by UGVs.

In the future, we intend to study the generalization of the developed framework to both feature extraction models and semantic segmentation models to deal with large outdoor scene dataset for UGVs in a long term operation.

References

[1] W. Churchill and P. Newman, "Experience-based navigation for long-term localisation," *Int. J. Robot. Res.*, vol. 32, no. 14, pp. 1645–1661, 2013.

[2] S. Minaeian, J. Liu, and Y.-J. Son, "Vision-based target detection and localization via a team of cooperative UAV and UGVs," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 46, no. 7, pp. 1005–1016, Jul. 2016.

[3] D. I. Katzourakis, J. C. F. De Winter, M. Alirezaei, M. Corno, and R. Happee, "Road-departure prevention in an emergency obstacle avoidance situation," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 44, no. 5, pp. 621–629, May 2014.

[4] M. Rokunuzzaman, T. Umeda, K. Sekiyama, and T. Fukuda, "A region of interest (ROI) sharing protocol for multirobot cooperation with distributed sensing based on semantic stability," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 44, no. 4, pp. 457–467, Apr. 2014.

[5] G. Carolina, M. Brian, B. Serge, and G. Lanckriet, "Multi-class object localization by combining local contextual interactions," in *Proc. IEEE Comput. Soc. Conf. Comput. Pattern Recognit. (CVPR)*, San Francisco, CA, USA, 2010, pp. 113–120.

[6] C. Wojek, S. Walk, S. Roth, K. Schindler, and B. Schiele, "Monocular visual scene understanding: Understanding multi-object traffic scenes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 4, pp. 882–897, Apr. 2013.

[7] B. Upcroft, C. Mcmanus, W. Churchill, W. Maddern, and P. Newman, "Lighting invariant urban street classification," in *Proc. IEEE Conf. Robot. Autom.*, Hong Kong, 2014, pp. 1712–1718.

[8] T. Scharwächter, M. Enzweiler, U. Franke, and S. Roth, "Stixmantics: A medium-level model for real-time semantic scene understanding," in *Computer Vision—ECCV*. Zürich, Switzerland: Springer, 2014, pp. 533–548.

[9] A. Geiger, M. Lauer, C. Wojek, C. Stiller, and R. Urtasun, "3D traffic scene understanding from movable platforms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 5, pp. 1012–1025, May 2014.

[10] W. P. Willem, G. Dubbelman, and P. H. N. de With, "Extending the stixel world with online self-supervised color modeling for road-versus-obstacle segmentation," in *Proc. IEEE Int. Conf. Intell. Trans. Syst.*, Qingdao, China, 2014, pp. 1400–1407.

[11] V. Nedovic, A. W. M. Smeulders, A. Redert, and J.-M. Geusebroek, "Stages as models of scene geometry," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1673–1687, Sep. 2010.

[12] Q. Huang, M. Han, B. Wu, and S. Ioffe, "A hierarchical conditional random field model for labeling and segmenting images of street scenes," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Colorado Springs, CO, USA, 2011, pp. 1953–1960.

[13] Y. Qi, Y. Wang, X. Zheng, and Z. Wu, "Robust feature learning by stacked autoencoder with maximum correntropy criterion," in *Proc. IEEE Conf. Acoust. Speech Signal Process.*, Florence, Italy, 2014, pp. 6716–6720.

[14] S. Munoz-Romero, V. Gomez-Verdejo, and J. Arenas-Garcia, "Regularized multivariate analysis framework for interpretable high-dimensional variable selection," *IEEE Comput. Intell. Mag.*, vol. 11, no. 4, pp. 24–35, Nov. 2016.

[15] Y. Wang *et al.*, "Robust active stereo vision using Kullback–Leibler divergence," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 3, pp. 548–563, Mar. 2012.

[16] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning hierarchical features for scene labeling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1915–1929, Aug. 2013.

[17] D. Tuia, R. Flamary, and N. Courty, "Multiclass feature learning for hyperspectral image classification: Sparse and hierarchical solutions," *ISPRS J. Photogrammetry Remote Sens.*, vol. 105, pp. 272–285, Jul. 2016.

[18] V. K. Sharma and A. Bala, "Clustering for high dimensional data," in *Proc. IEEE Int. Conf. Netw. Soft Comput.*, 2014, pp. 365–369.

[19] O. Arbelaitz, I. Gurrutxaga, J. Muguerza, J. M. Pérez, and I. Perona, "An extensive comparative study of cluster validity indices," *Pattern Recognit.*, vol. 46, no. 1, pp. 243–256, 2013.

[20] C.-T. Lin, M. Prasad, and A. Saxena, "An improved polynomial neural network classifier using real-coded genetic algorithm," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 45, no. 11, pp. 1389–1401, Nov. 2015.

[21] H.-C. Shin, M. R. Orton, D. J. Collins, S. J. Doran, and M. O. Leach, "Stacked autoencoders for unsupervised feature learning and multiple organ detection in a pilot study using 4D patient data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1930–1943, Aug. 2013.

[22] C. Zhao *et al.*, "A Markov chain-based testability growth model with a cost-benefit function," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 46, no. 4, pp. 524–534, Apr. 2016.

[23] L. Ladický, C. Russell, P. Kohli, and P. H. Torr, "Inference methods for CRFs with co-occurrence statistics," *Int. J. Comput. Vis.*, vol. 103, no. 2, pp. 213–225, 2013.

[24] E. Fosler-Lussier, Y. He, P. Jyothi, and R. Prabhavalkar, "Conditional random fields in speech, audio, and language processing," *Proc. IEEE*, vol. 101, no. 5, pp. 1054–1075, May 2013.

[25] M. Abboud, B. Németh, and J. C. Guillemin, "Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription," *Chem. Eur. J.*, vol. 18, no. 13, pp. 3981–3991, 2012.

[26] S. Lange and M. Riedmiller, "Deep auto-encoder neural networks in reinforcement learning," in *Proc. IEEE Int. Conf. Neural Netw.*, Barcelona, Spain, 2010, pp. 1–8.

[27] X. Cao and W. A. Chaovalitwongse, "Optimization models for feature selection of decomposed nearest neighbor," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 46, no. 2, pp. 177–184, Feb. 2016.

[28] P. A. Traganitis, K. Slavakis, and G. B. Giannakis, "Clustering high-dimensional data via random sampling and consensus," in *Proc. IEEE Signal Inf. Process.*, Atlanta, GA, USA, 2014, pp. 307–311.

Zengshuai Qiu received the bachelor's degree in automation from the City Institute, Dalian University of Technology, Dalian, China, in 2010 and the master's degree in control theory and engineering from the Shenyang University of Technology, Shenyang, China, in 2013. He is currently pursuing the Ph.D. degree with the School of Control Science and Engineering, Dalian University of Technology.

His current research interests include robotics, deep learning, image processing, and semantic scene understanding.

Yan Zhuang (M'11) received the bachelor's and master's degrees in control theory and engineering from Northeastern University, Shenyang, China, in 1997 and 2000, respectively, and the Ph.D. degree in control theory and engineering from the Dalian University of Technology, Dalian, China, in 2004.

He joined the Dalian University of Technology as a Lecturer in 2005, and became an Associate Professor, in 2007, where he is currently a Professor with the School of Control Science and Engineering. His current research interests include mobile robot 3-D mapping, outdoor scene understanding, 3-D-laser-based object recognition, and 3-D scene recognition and reconstruction.

Huosheng Hu (M'94–SM'01) received the M.Sc. degree in industrial automation from Central South University, Changsha, China, in 1982 and the Ph.D. degree in robotics from the University of Oxford, Oxford, U.K., in 1993.

He is a Professor with the School of Computer Science and Electronic Engineering, University of Essex, Colchester, U.K., leading the Robotics Research Group. His current research interests include behavior-based robotics, human–robot interaction, service robots, embedded systems, data fusion, learning algorithms, mechatronics, and pervasive computing. He has published around 450 papers in journals, books, and conferences in the above areas.

Dr. Hu was a recipient of the number of best paper awards. He currently serves as the Editor-in-Chief for the *International Journal of Automation and Computing,* online *Robotics* Journal, and an Executive Editor of the *International Journal of Mechatronics and Automation.* He has been the Program Chair or a member of Advisory/Organizing Committee for many international conferences, such as IEEE International Conference on Robotics and Automation, Intelligent Robots and Systems, International Conference on Mechatronics and Automation, International Conference on Robotics and Biomimetics, International Conference on Automation and Information, International Conference on Automation and Logistics, and the International Association of Science and Technology for Development Robotics and Applications, Control and Applications, and Computational Intelligence Conferences. He is a Founding Member of IEEE Robotics and Automation Society Technical Committee on Networked Robots. He is a fellow of IET and InstMC and a Senior Member of ACM.

Wei Wang (SM'01) received the bachelor's, master's, and Ph.D. degrees in industrial automation from Northeastern University, Shenyang, China, in 1982, 1986, and 1988, respectively.

He is a Post-Doctoral Fellow with the Division of Engineering Cybernetics, Norwegian University of Science and Technology, Trondheim, Norway, from 1990 to 1992, and a Research Fellow with the Department of Engineering Science, University of Oxford, Oxford, U.K., from 1998 to 1999. He is currently a Professor with the School of Control Science and Engineering, Dalian University of Technology, Dalian, China. He has published over 200 papers in international and domestic journals. His current research interests include adaptive control, predictive control, robotics, computer integrated manufacturing systems, and computer control of industrial process.

Dr. Wang was a recipient of the National Distinguished Young Fund of the National Natural Science Foundations of China in 1998. He was a member of IFAC Technical Committee of Mining, Mineral and Metal Processing, in 1999, and the Chair of IFAC Technical Committee on Cost Oriented Automation, from 2005 to 2008, a Steering Commission Member of Asian Control Association in 2011.