FULL PAPER

An Intelligent Self-Taught Vision System Automated 3D Object Learning and Recognition

Ren C. Luo* and Po Yu Chuang†

*,† *Center for Intelligent Robotics and Automation Research, National Taiwan University, Taipei 10617, Taiwan*

(*v1.0 released January 2015*)

Vision systems for 3D object recognition are widely applied on industrial integrating with robot arm. Conventionally, vision, learning approach, and robot arm are separated into three different systems, so that learning approach can only learn the distributions of input data, but cannot refine qualities of input and output without manual intervention. Therefore, it may cause misfiring performance by erratic input or sustainable growth database. In this paper, we propose an intelligent vision system for 3D object recognition which is able to automatically construct and refine model for 3D object recognition without any manual intervention. Although 2D images and rotation angles of robot arm are different domains, we model 3D objects by multiple 2D images and rotation angles, and information from different domains are integrated by a Hierarchical-Deep (HD) model with parallel branch. The model hierarchically extracts information by domains and levels. The parallel branch distinct labeled and unlabeled data to avoid performance drag by sustainable growth of unlabeled input. The relationships between label and unlabeled data are learned by proposed self-taught approach. The experimental results support the feasibility of proposed structure which can transfer knowledge in different domains with limited prior knowledge, and complete assigned task by only modeling the relation between input and output.

Keywords: Automation; Computer vision; Image recognition; Learning systems; Intelligent robots

1. Introduction

Robot arm with vision has been widely applied in automatic industrial production line for many years [1-4]. New hurdles of conventional vision system arise due to the rise of consumer electronic market. The components in assembling production line are become small volume with large varieties. The types of components are also changed rapidly because of short product life cycle. These conditions are tough for conventional vision system. Model-based recognition methods are commonly used in present industrial applications. The performance is mainly related to the manually labeled data. These manual works not only increase the labor cost, but also point out the dilemma of present vision-based robot is unable to automatically adapt to various assignments, and growing database.

Therefore, we propose an intelligent system which is able to handle sustainable growth of unlabeled input, and refine existed model without any manual intervention. The situation of this paper is that we only provide target face of 3D objects which are intended to be placed on top by robot arm, but the other faces of 3D objects are unknown. The only prior knowledge is target faces, and inputs are arbitrary objects with random faces on top. The input of system is 2D image, and output is rotation angle of robot arm in Cartesian. Since input and output are

*Corresponding author. Email: renluo@ ntu.edu.tw different domains, the traditional single layer models [5,6] which end in a linear or kernel classifier are not enough. We introduce a ***Hierarchical-Deep(HD)*** model[7] to tackle the problems.

The learning of HD model achieves dramatically success recent years. Hinton et al. [7] proposed deep structure learning which hidden layers are formed by lower level feature to higher level hierarchically, and had been successfully applied on different research fields [8-11]. The proposed HD model is constructed by four layers:**Feature**, **Descriptor**, **Object** and **RotationAngle**. Being an automatic system, the ability which could "infer" latent edges between labeled and unlabeled data is needed. Latent edge means two variables in different layers exist an edge in graph model if prior data is sufficient, but, in our case, system only have small amount of prior data. Hence, there are many latent edges which are waiting to be revealed through learning process.

The most challenge part is that the appearance of different faces of a single object might be quite different, so we design three modules to tackle self-taught problem. Firstly, we design a probabilistic based image descriptor. Although many methods [12-16] provide high quality performance by extracting sparse features, the sparse feature is not compact on inferring latent edges. The sparse feature only model strong features of observed face, but most of faces is unknown in our case. We need a descriptor which can provide sufficient information for inference, but still retain scale- and rotation-invariant. Proposed probabilistic based descriptor is established based on the ***Markov Logic Network (MLN)*** [17-20]. MLN is an approach combines first-order logic and probabilistic graphical model. First-order logic enables compactly representing the neighborhood of feature points. Probabilistic graphical model can reveal latent edges by proper inference method, and also handle the uncertainty.

Secondly, transfer information module is tailored to handle unlabeled data. Transfer information module is realized by Self-taught Clustering algorithm [21]. Self-taught Clustering algorithm is a transfer learning method [22-26] which is built for enhancing model through large amount of auxiliary unlabeled data. The input face can be considered auxiliary unlabeled data, and find co-cluster between prior faces in the dataset. The distribution of co-cluster is further utilized to infer the possible rotation angle for robot arm, and robot arm will rotate target object from input face to output face. Finally, the validation module is an eye-to-hand camera which used to validate the error between the output face and desired target. Then, the validation module returns the error to the model in order to refine the existed model. Through these three modules, proposed system can automatically learn the relations between input images and corresponding rotation angles with only labeled the target face of each object.

In this paper, we start with briefly overview of system design and structure in section 2. The MLN-based descriptor for recognizing object is described in section 3. Section 4 introduces how to model the proposed hierarchical networks, and learn by self-taught learning. Then, we compare the performance of proposed system with several states of development in section 5. Finally, reviewing performance and conclusions are presented in the final section.

2. System architecture

The main purpose of this system is to automatically derive the relationship between input face and corresponding rotation angle to make robot arm can rotate objects to the assigned faces. The only prior knowledge are the target face. Input is arbitrary assigned object with random face on top, so input is very likely an unknown face of assigned object showed-up rather than prior target face. Therefore, system has to infer the correlation between input and existed priors. Proposed system is shown in Fig. 1. Camera 1 captures images of all input objects with random faces on top, and constructs MLN-based descriptor for each input. Then, system matches the input with data in database and output rotation angle for robot arm. After robot arm placing an object, camera 2 will validate result, and feedback error for refining existed model. The system architecture in Fig. 1 is realized by a hierarchical-deep model in Fig. 2.

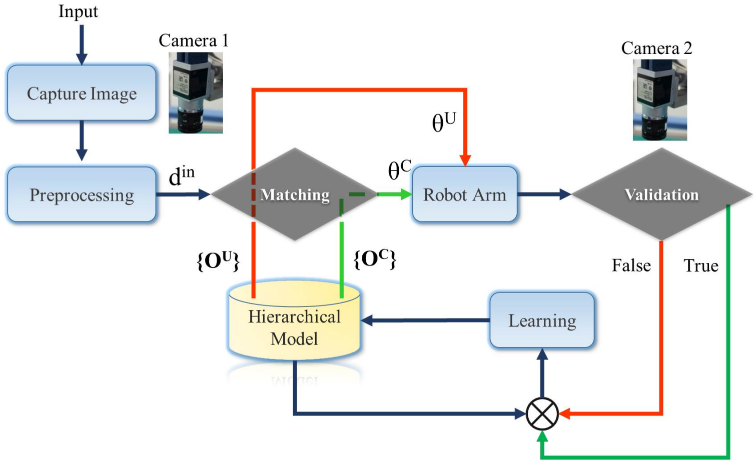


Figure 1. System architecture

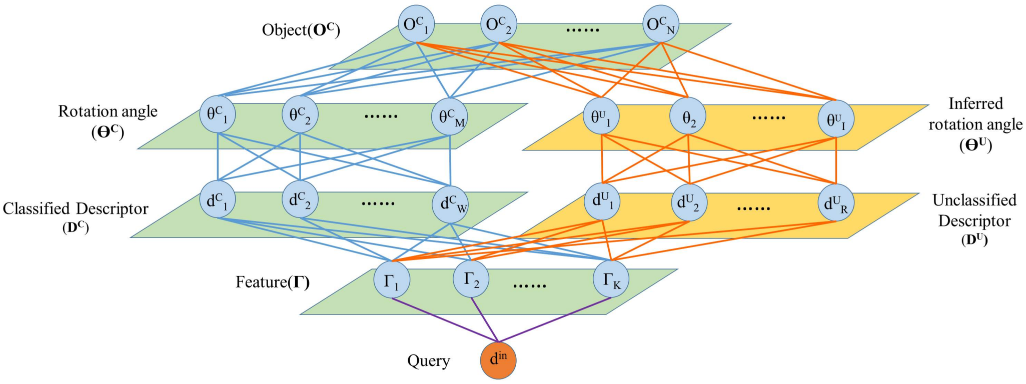


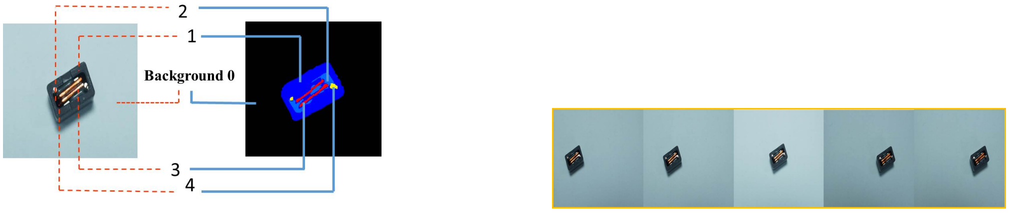
Figure 2. Hierarchical-deep model for self-taught system

The variables in the same layer are independent, and vertical adjacent two layers are full connected. Variables in **Feature($\mathbf{\Gamma}$)** layer are extracted image features, and variables in both **Classified Descriptor($\mathbf{D^C}$)** and **Unclassified Descriptor($\mathbf{D^U}$)** are MLN-based descriptor. Variables in **Rotation angle($\mathbf{\Theta^C}$)** and **Inferred Rotation angle($\mathbf{\Theta^U}$)** are set of rotation angles { Row (α), Pitch(β) , Yaw(γ) } respect to target faces. Finally, variables in **Object($\mathbf{O^C}$)** are combinations of rotation angles.

The difference between classic **Deep Belief Networks**(**DBN**) is that proposed model exist two parallel parts in Fig. 2. $\mathbf{D^C}$-$\mathbf{\Theta^C}$ and $\mathbf{D^U}$-$\mathbf{\Theta^U}$ have no connection between each other, but both have full connection with deepest layer $\mathbf{O^C}$ and first layer $\mathbf{\Gamma}$. To handle tons of unknown data, the structure of connection will dynamically change with observed evidences. Sparse coding method is used to constructs edge in the model, most of connection is zero which is called latent edge in this paper. Latent edge might become non-zero while some new evidences have been discovered. For variable d^C_w in layer $\mathbf{D^C}$, the sparse coding result should be formulated as:

$$d^C_w = \sum_{i \in d^C_w} a_i \Gamma_i + \sum_{j \notin d^C_w} b_j \Gamma_j \tag{1}$$

Although Eq.(1) can handle the problem of latent edge, it's impractical to sample all possible conditions whenever new evidence showing up. Therefore, proposed model separate descriptor layer into two parallel parts as:



(a) Result of background subtracton and clustering

(b) Serial captured frames

Figure 3. Preprocessing of input objects

$$d_w^C = \sum_{\Gamma_i \in d_w^C} a_i \Gamma_i \tag{2}$$

$$d_r^U = \sum_{\Gamma_j \in d_r^U} b_j \Gamma_j \tag{3}$$

d_w^C is a prior descriptor which the edge between $\mathbf{\Theta^C}$ and $\mathbf{O^C}$ had been established. d_r^U is a descriptor in $\mathbf{D^U}$ layer which match none of descriptors in $\mathbf{D^C}$. Therefore, we propose an inference method to infer the possible rotation angle, and camera 2 will check inferred results. If inference is success, latent edge between d_r^U and θ^U will be established, and become stronger as more successful inferences.

$$d_w^C = \sum_{i \in d_w^C} a_i \Gamma_i + \sum_{j \notin d_w^C} b_j \Gamma_j \tag{4}$$

Although Eq.(1) can handle the problem of latent edge, it's impractical to sample all possible conditions whenever new evidence showing up. Therefore, proposed model separate descriptor layer into two parallel parts as:

$$d_w^C = \sum_{\Gamma_i \in d_w^C} a_i \Gamma_i \tag{5}$$

$$d_r^U = \sum_{\Gamma_j \in d_r^U} b_j \Gamma_j \tag{6}$$

d_w^C is a prior descriptor which the edge between $\mathbf{\Theta^C}$ and $\mathbf{O^C}$ had been established. d_r^U is a descriptor in $\mathbf{D^U}$ layer which match none of descriptors in $\mathbf{D^C}$. Therefore, we propose an inference method to infer the possible rotation angle, and camera 2 will check inferred results. If inference is success, latent edge between d_r^U and θ^U will be established, and become stronger as more successful inferences.

Table 1. Example of predicates

Key Atom	1	2	3	4
Predicates	ne(1,2)	ne(2,1)	ne(3,1)	ne(4,1)
	ne(1,3)	ne(2,3)	ne(3,2)	ne(4,2)
	ne(1,4)	ne(2,4)		
	ne(1,0)			

3. MLN-based Descriptor

3.1 *The concept of constructing MLN-based descriptor*

Being a self-taught system, deriving more valuable information from raw data helps system deriving more reliable results with scarce prior knowledge. Most of present image descriptors [12-16] are constructed based on strong sparse feature point, because these points are consistent even in different environment. These kinds of descriptor can efficiently and precisely match given image. Nevertheless, most of observed face is not in prior data, so we need a descriptor which can infer the relation between observations and priors. To avoid losing information, we choose normal distributed feature instead of sparse feature. Since different faces of an object may derive different strong features, normal distributed feature is more suitable for our case.

For prepossessing of input images, each channel of RGB domain is classified into 5 parts, and get 125 classes in RGB domain. Every input image is segmented by these classes. In Fig. 3(a), an observed face of input object is segmented into 4 classes, and class 0 is background. Hereafter, predicates for MLN networks are constructed by segmented results. We have only two kinds of predicate *ne(a,v)* and *des(x)* for MLN model. Variable a is an atom cluster, and variable v is a neighbor of atom cluster, so predicate *ne(a,v)* represent adjacency of atom cluster. Variable x in *des(x)* is a MLN-based descriptor. The variables of $\mathbf{\Gamma}$ layer in Fig. 2 are predicates *ne(a,v)*. Since every class can be the atom cluster, we have $\binom{125}{2}$ binary variables in $\mathbf{\Gamma}$ layer.

Taking Fig. 3(a) as an example, the predicates of preprocessed image are shown in Table I, and first order logic is formulated as:

$$\forall a \forall v \quad ne(a{,}v) \Rightarrow des(x) \tag{7}$$

Each image will further be down sampled, and derived several images with different scales. For each image, we derive **F*S** formulas where F is number of serial captured images and S is number of images with different scales. Through these formulas, a MLN model can be constructed. The probability distribution over possible world d^{in} specified by MLN is given by:

$$P(\mathbf{D^{in}} = d^{in}) = \frac{1}{Z} exp(\sum_{j=1}^{F*S} w_j n_j(d^{in}))$$

$$Z = \sum_{d^{in} \in \mathbf{D^{in}}} exp(\sum_{j=1}^{F*S} w_j n_j(d^{in})) \tag{8}$$

Where d^{in} is the descriptor of input image. $n_j(d^{in})$ is the number of true grounding of formula j in d^{in}, and w_j is weight of formula j .

Consequently, probability distribution Eq.(5) is MLN-based descriptor for each 2D faces within input 3D object.

3.2 *Inference and Weight learning of MLN-based descriptor*

The weights of MLN-based descriptor are learned by maximizing the pseudo-log likelihood. Since each descriptor can be consider as a closed world, we only need to consider the atoms which derive from captured serial frames. Comparing with uniform sampling approach, maximizing pseudo-log-likelihood is more efficient, because pseudo-log likelihood only need to consider relational data. The pseudo-log likelihood of Eq.(5) can be written as:

$$\log P_w^*(\mathbf{D^{in}} = d^{in}) = \sum_{j=1}^{F*S} \log P_w(\mathbf{D^{in}} = d^{in} | \mathbf{MB}(d^{in})) \tag{9}$$

Where $\mathbf{MB(d^{in})}$ is Markov blanket while d^{in} is observed. The MLN weights are learned generatively by maximizing the pseudo-log likelihood of Markov blanket. The gradient of the pseudo-log likelihood with respect to the weight is:

$$\begin{aligned} &\frac{\partial}{\partial w_i} \log P_w^*(\mathbf{D^{in}} = d^{in}) = \\ &\textstyle\sum_{j=1}^{F*S} \{ n_i(d^{in}) - P_w(\mathbf{D^{in}} = 0 | \mathbf{MB}(d^{in})) n_i(d^{in} = 0) \\ &\qquad\qquad -P_w(\mathbf{D^{in}} = 1 | \mathbf{MB}(d^{in}) n_i(d^{in} = 1) \} \end{aligned} \tag{10}$$

Where $n_i(d^{in} = 0)$ is the number of true grounding of j^{th} formula while force $\mathbf{d^{in}} = 0$, and similar for $n_i(d^{in} = 1)$. The learning of pseudo-log-likelihood in our approach are further boosted by ***Limited-memory Broyden-Fletcher-Goldfarb-Shanno(L-BFGS)*** optimizer [20] to make entire process become more efficiency.

3.3 *Matching of MLN-based descriptors*

For each constructed input descriptor d_k^{in}, system would search for the matched descriptor in the database, and further arrange it to the proper layer of $\mathbf{D^C}$ or $\mathbf{D^U}$ as shown in Fig.2. Since input is possible to be assigned to one of parallel layers, matching step is separated into two parts. One is using pseudo-log likelihood for deciding observation should be assigned to which layer. The pseudo-log likelihood of descriptors matching could be formulated as:

$$\begin{aligned} &\mathbf{argMax} P(\mathbf{D^C} = d_w^C | \mathbf{D^{in}} = d^{in}, \mathbf{\Gamma}_{k \in d^{in}}) \\ &= \mathbf{argMax} P(d^{in} | \mathbf{MB}(d^{in}) P(d_w^C | \mathbf{MB}(d^{in})) \end{aligned} \tag{11}$$

If input descriptors match none of descriptor in $\mathbf{D^C}$ layer, the descriptor become a variable of $\mathbf{D^U}$ layer. For a variable in $\mathbf{D^U}$, we infer rotation angle to make input object which can be placed on corresponding target face. Since the rotation angles for descriptors in $\mathbf{D^U}$ are unidentified, the second part for matching is trying to find a descriptor in $\mathbf{D^C}$ which have max co-cluster with input descriptor. Finding max co-cluster can be alternately considered as minimizing information loss as:

$$\mathbf{argMin}(I(d^i n, \mathbf{\Gamma}_{k \in d^{in} \cap d_w^C}) - I(d_w^C, \mathbf{\Gamma}_{k \in d^{in} \cap d_w^C})) \tag{12}$$

The common feature $\mathbf{\Gamma}_{k \in d^{in} \cap d_w^C}$ is further represented by Markov Blanket of d^{in} and d_w^C, and the loss of mutual information can be further formulated by ***KullbackLeibler divergence(KL divergence)***[28] as:

$$\begin{aligned} &\mathbf{argMin} D(P(d^{in}, \mathbf{MB}(d^{in}, d_w^c))||P(d_w^c, \mathbf{MB}(d^{in}, d_w^c))) \\ &= \mathbf{argMin} \textstyle\sum_{\Gamma_k \in \mathbf{MB}(d^{in}, d_w^c)} P(\Gamma_k) D(P(d^{in}|\Gamma_k)||P(d_w^C|\Gamma_k)) \end{aligned} \tag{13}$$

In Eq.(10), classified descriptor d_w^C with min KL divergence is considered as acquired max co-cluster with d^{in}. The relation between the co-cluster become the evidence for inferring rotation angle of d^{in}. Through Eq.(8) and Eq.(10), the input descriptors are classified to corresponding layer, and become inputs $\mathbf{\Theta^U}$ or $\mathbf{\Theta^C}$ layer.

4. Hierarchical model

4.1 *Inference of rotation angle in $\mathbf{\Theta^U}$ layer*

Inference rotation angle $\theta_\mathbf{i}^\mathbf{U}$ is based on max co-cluster between d^{in} and d_w^C. A set of co-cluster $\{C_{w1}, C_{w2}, ..., C_{wL}\}$ can be derived by minimizing KL divergence. The center of co-cluster with respect to center of camera in Cartesian space can be derived into two sets $\mathbf{V^{in}}$=$\{v_1^{in}$, v_2^{in}, ..., v_L^{in} $\}$ and $\mathbf{V_w^C}$=$\{v_{w1}^C$, v_{w2}^C, ..., v_{wL}^C $\}$. $\mathbf{V^{in}}$ is a set of co-cluster position in input descriptor, and $\mathbf{V_w^C}$ is a set of co-cluster position in d_w^C. The roll angle α of robot arm is calculated by:

$$\alpha = \cos^{-1} \frac{1}{L} \sum_{l=1}^{K} \frac{v_{wl}^C - v_l^{in}}{|v_{wl}^C - v_l^{in}|} \tag{14}$$

Where roll angle α is the mean angle of co-cluster in two descriptors. As for pitch angle β and yaw angle γ, the pitch and yaw angle are hard to be estimated by 2D descriptor directly. We make random sample these two angles in value $\pi/2$, and $-\pi/2$ initially, and approximate to actual angles by algorithm 1.

4.2 *Inference and learning of hierarchical-deep model*

Proposed hierarchical model is a generative model of ***Deep Belief Network (DBN)***. Structure between each layer is shown in Fig. 4. each layer is considered as a ***Restricted Boltzmann Machine (RBM)***[8] except $\mathbf{\Gamma}$, $\mathbf{D^C}$, and $\mathbf{D^U}$. The MLN is trained by pseudo-log likelihood as mentioned previously, and RBM is trained by greedy layer-wise training [30].

Initially, left part of model ($\mathbf{\Gamma}$-$\mathbf{D^C}$-$\mathbf{\Theta^C}$-$\mathbf{O^C}$) are trained with prior target face of objects, and number of variables $\mathbf{N}$ in $\mathbf{O^C}$ equal to the number of prior target faces. The right part of model is activated only when a new observation is classified into $\mathbf{D^U}$. The activation probability of θ_i^U is a sigmoid activation function:

$$\begin{aligned} &P(\theta_i^U|\mathbf{MB}(\mathbf{d^{in}})) = \tfrac{\mathbf{1}}{\mathbf{1+exp}(\mu * \mathbf{b} - \sum_\mathbf{r} \mathbf{d_r^U w_r})} \\ &\mu = \begin{cases} 0 & \text{, if inference succeed} \\ 1 + logP(\theta_i^U|\mathbf{\Theta^U}) & \text{, if inference fail} \end{cases} \end{aligned} \tag{15}$$

Algorithm 1 Inferring rotation angle from co-cluster

```
Function inferringTheta($d^{in}, \mathbf{D^C}, \mathbf{D^U}$)
Input :
$d^{in}$, input descriptor
$\mathbf{D^C}$, descriptors in $\mathbf{D^C}$ layer
$\mathbf{D^U}$, descriptors in $\mathbf{D^U}$ layer
Output :
$\theta^U\{\alpha, \beta, \gamma\}$, rotation angle for robot arm

$L\_D^c \leftarrow$ maxLikelihood($d^{in}$, $\mathbf{D^C}$)
$L\_D^U \leftarrow$ maxLikelihood($d^{in}$, $\mathbf{D^U}$)
if $L\_D^U > L\_D^U$ then
        $d\_target \leftarrow$ maxLikelihood($d^{in}$, MB in $\mathbf{D^U}$)
        $\theta^U \leftarrow$findmax_Coclass($d^{in}$, $d\_target$)
    while t < max_t || maxLikelihood(t)> threshold do
        if maxLikelihood(t) > maxLikelihood(t-1) then
            $t++$
            $\theta^U \leftarrow \theta^U + Step$
        else
            $Break$
        end if
    end while
else
        $d\_target \leftarrow$ maxLikelihood($d^{in}$, MB in $\mathbf{D^C}$)
        $\theta^U \leftarrow$findmax_Coclass($d^{in}$, $d\_target$)
end if
Return $\theta^U\{\alpha, \beta, \gamma\}$
```

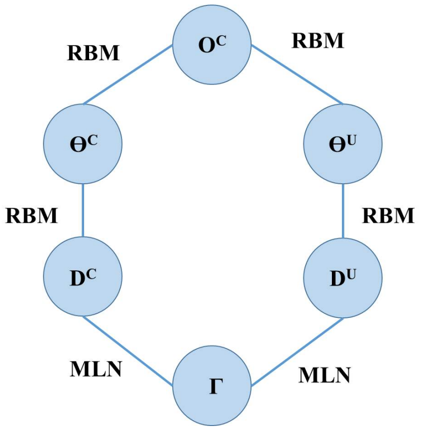


Figure 4. Structure configuration of each layer in the proposed model

where w_r and b are the weights and bias. μ is penalty factor which decreases the probability while the inference is failed. μ is depended on log likelihood of θ_i^U which can lead to lower activation probability if inference result failed several times, and avoid system derives wrong results over again. On the other hand, for both $\mathbf{\Theta^U}$ and $\mathbf{\Theta^C}$ layer, if results are correct, the model will be retrained by greedy layer-wise training. If validated result is derived from left part of Fig. 4, the generative model is defined by the joint distribution of top layers $P(\mathbf{O^C}, \mathbf{\Theta^C})$. If the result is derived from right part, the generative model is defined by $P(\mathbf{O^C}, \mathbf{\Theta^U})$. By doing so, the relation between prior and observations can be self-taught from numerous random unlabeled

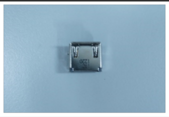
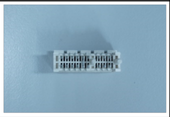
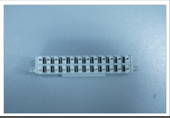
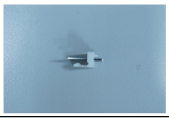
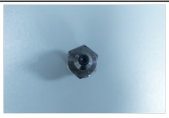
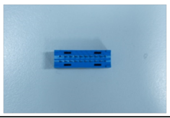
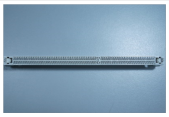
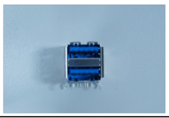
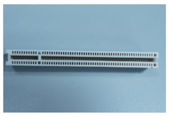
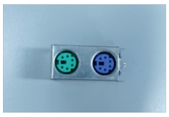
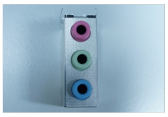
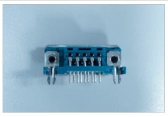
Table 2. Comparisons on **Caltech − 101**

Methods	Accuracy
MLN − based	**74.6**
LLC[37]	73.1
P-LLC[38]	78.75
P-FV[38]	80.1
M-HMP[39]	82.5
ImageNet-pretrained convnet[40]	86.5

Table 3. Comparisons on **Caltech − 256**

Methods	45	60
MLN − based	**66.7**	**69.6**
LLC	45.3	47.7
P-LLC	44.9	48.0
P-FV	44.9	52.6
M-HMP	54.8	58
ImageNet-pretrained convnet	72.7	74.2

Table 4. Three classes of testing work pieces for experiments

WP1		WP2		WP3	
WP1.1	WP1.5	WP2.1	WP2.5	WP3.1	WP3.5
					
WP1.2	WP1.6	WP2.2	WP2.6	WP3.2	WP3.6
					
WP1.3	WP1.7	WP2.3		WP3.3	WP3.7
					
WP1.4		WP2.4		WP3.4	
					

inputs, and infer possible relationship without any manual intervention.

5. Experiments

The experiments for proposed model are separated into two parts. Firstly, we evaluate the performance of MLN-based descriptor by standard object recognition datasets: **Caltech − 101** [31] and **Caltech − 256** [32]. Results shown in Table II are comparisons with recently published papers. The images in the datasets are rescaled into five different scales for training proposed MLN-based descriptor. For **Caltech − 101**, we follow general procedure and randomly selecting 30 images for each class. For **Caltech − 256**, select 45 and 60 images for each class, and trained by pseudo-log likelihood. Although, for the proposed model, the result does not outperform in **Caltech − 101**, but the accuracy in **Caltech − 256** is slightly behind ImageNet-pretrained model. In the other scopes, the result shows that MLN-based descriptor keep well performance even increasing categories. Most of recently published methods get dramatically performance decreasing while categories increase from 101 to 256.

For the second part of experiment, we implement the proposed system in real industrial application. The prior knowledge is target faces of assigned objects, and there are twenty kinds of assigned objects in our experiment. Table IV shows twenty target faces for each assigned object. The experiment is implemented based on several assumptions: The input objects are not occluded, and not adjacent with each other. Hereafter, the inputs are randomly chosen from assigned objects with random face on top.

The testing objects are classified into three classes in Table IV. For class **WP1**, the work pieces are featureless and relative small, so it's hard to construct robustness descriptor even building relational model for entire model. For class **WP2**, all work pieces acquire similar shapes and size, so this kind of object is easily mismatched in the matching process. The work pieces in **WP3** are matched group of this experiment. These work pieces acquire sufficient features for descriptor, and have plenty of information for identifying and constructing descriptor. In the first stage of experiment, we compare the performance of proposed system between different classes in different lighting conditions. The results of different classes are shown in Fig.5. In Fig.5(a), the environment lighting is controlled by on-axis lighting source, so the information of object are more complete and distinct than images without lighting control in Fig.5(b). The accuracy is average of 100 times repeatedly testing.

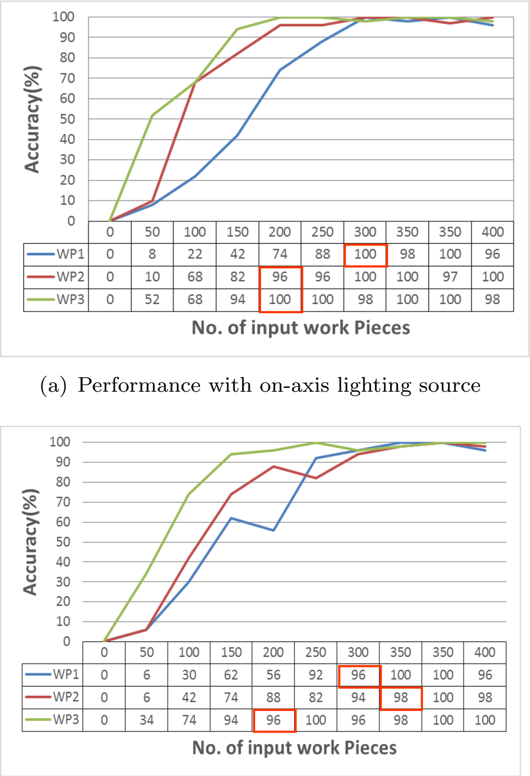
The system is considered convergence while accuracy is over 95%, and stop learning approach. If the accuracy is under 95% again, the learning approach would be re-excuted. Comparing the results, in both cases, class **WP3** could be convergent with least input sample, and convergent time of class **WP2** is slowest. The results show that the efficiency of learning could be slightly improved by environment constrain, but the accuracy is not effected, and always keep over 95% after learning approach stopped.

Fig. 6 shows the result while all twenty kinds of assigned object are involved in the same time. The result shows that system need more inputs to converge while more kinds of objects are involved, but the system still slightly converge, and accuracy is all keeping over 95% for both conditions. In brief, these two experiments verify proposed system is competent to learn the HD model automatically. Although the learning rate is dragged by the number of assigned objects, the learning rate still can be convergent by reasonable number of inputs.

The experiment in Fig. 5 and 6 testified the performance of proposed system can meet our requirements. We compare the performance of proposed system with other advanced approaches. Since none of similar systems could handle this issue in our literature survey results, so the comparisons are done by dividing our system into two parts. One is 2D descriptors for each face of objects, and the other is machine learning approach for learning relational model.

For the descriptor part, four kinds of other descriptors are chosen to compare with proposed system. B-SIFT[35]and Edge-SIFT[36] are modified versions of SIFT approach which enhanced the accuracy of feature point registration. ***Binary Robust Invariant Scalable Keypoints(BRISK)***[15] descriptor is constructed based on binary robust invariant scalable key points, and ***Zernike Moment (ZM)***[13] phase-based descriptor is a moment-based descriptor which use the phase information of signal. All of these descriptors are representative methods in relative field recent years, and had been testified by many researchers. To compare the robustness and accuracy, the performance is testified by two conditions as shown in Table V. One is relationship of each face is prior of system, and the descriptors only provide information for object matching. The experiments are implemented by the same learning approach which is proposed in the previous section. The other is no prior for learning approach that information of descriptor need to be used for inferring the relational model. The ZM descriptor has the best performance in the condition with prior, but results of descriptors are close. In condition, without prior, the MLN-based descriptor acquires best performance which testified MLN-based descriptor is suited for handling large amount of unlabeled data.

Hereafter, the performances of different learning methods are further discussed. The other three kinds of transfer learning approaches: ***Locally Weighted Ensemble approach(LWE)***[25], ***Transductive SVM(TSVM)***[26], and self-taught learning[21] are chosen to compare with proposed method. Similarly, the experiments are divided into two parts as shown in Table VI. The



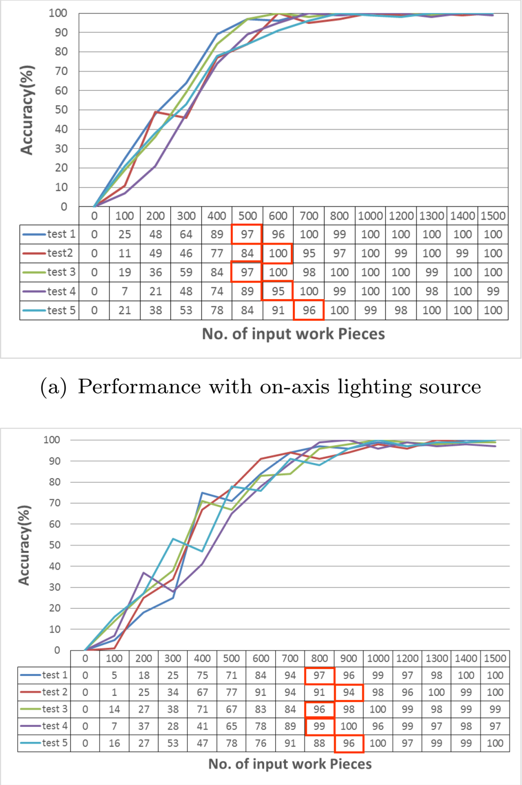
	0	50	100	150	200	250	300	350	350	400
WP1	0	8	22	42	74	88	100	98	100	96
WP2	0	10	68	82	96	96	100	100	97	100
WP3	0	52	68	94	100	100	98	100	100	98

(a) Performance with on-axis lighting source

	0	50	100	150	200	250	300	350	350	400
WP1	0	6	30	62	56	92	96	100	100	96
WP2	0	6	42	74	88	82	94	98	100	98
WP3	0	34	74	94	96	100	96	98	100	100

(b) Performance without on-axis lighting source

Figure 5. Experimental Results of different classes in different lighting conditions



	0	100	200	300	400	500	600	700	800	1000	1200	1300	1400	1500
test 1	0	25	48	64	89	97	96	100	99	100	100	100	100	100
test2	0	11	49	46	77	84	100	95	97	100	99	100	99	100
test 3	0	19	36	59	84	97	100	98	100	100	100	99	100	100
test 4	0	7	21	48	74	89	95	100	99	100	100	98	100	99
test 5	0	21	38	53	78	84	91	96	100	99	98	100	100	100

(a) Performance with on-axis lighting source

	0	100	200	300	400	500	600	700	800	900	1000	1200	1300	1400	1500
test 1	0	5	18	25	75	71	84	94	97	96	99	97	98	100	100
test 2	0	1	25	34	67	77	91	94	91	94	98	96	100	99	100
test 3	0	14	27	38	71	67	83	84	96	98	100	99	98	99	99
test 4	0	7	37	28	41	65	78	89	99	100	96	99	97	98	97
test 5	0	16	27	53	47	78	76	91	88	96	100	97	99	99	100

(b) Performance without on-axis lighting source

Figure 6. Experimental Results of all work pieces in different lighting conditions

Table 5. Comparisons of different descriptors

		Descriptor				
		MLN-based	**B-SIFT**	**Edge-SIFT**	**BRISK**	**ZM**
With prior	WP1	**0.9781**	0.9664	0.8384	0.9556	0.9788
	WP2	0.9630	**0.9766**	0.9233	0.9676	0.9523
	WP3	0.9901	**0.9963**	0.9454	0.9899	0.9949
	All	0.9594	0.8982	0.8066	0.9432	**0.9634**
Without prior	WP1	**0.9611**	0.7688	0.7544	0.8103	0.7787
	WP2	**0.9505**	0.7043	0.7123	0.7197	0.7979
	WP3	**0.9718**	0.8044	0.7963	0.8243	0.8231
	All	**0.9543**	0.6431	0.6144	0.6741	0.7570

Table 6. Comparisons of different learning approaches

		Learning approach			
		Proposed	**LEW**	**TSVM**	**Self-taught**
With prior	WP1	0.9781	**0.9802**	0.9511	0.9603
	WP2	0.9630	**0.9763**	0.9690	0.9701
	WP3	**0.9901**	0.9899	0.9799	0.9807
	All	0.9594	0.9677	0.9567	**0.9701**
Without prior	WP1	**0.9611**	0.5601	0.6443	0.6213
	WP2	**0.9505**	0.6543	0.7158	0.7128
	WP3	**0.9718**	0.7188	0.7799	0.7846
	All	**0.9603**	0.6553	0.7497	0.7081

result shows LEW acquires the best accuracy in the condition with priors, and proposed learning approach acquire greatest performance in condition without prior. Although the performances between different methods are quite close when priors are provided, the accuracy of the other methods goes down in no prior condition. It seems that the results are not only affected by descriptor, but also learning approach. The proposed system is only one method which can automatically learn and recognize object without prior knowledge of 3D model.

6. Comparisons of related works

The proposed system is a HD model with parallel branches. Conventionally, the descriptors [13, 15, 35-39] and learning model [27-28] are separated, so that the learning approaches only learn the distribution of descriptors but cannot adjust the distribution of descriptors. According to our surveys, there are no such descriptors construction can fit every case without manual intervention. Therefore, we propose to establish a learning approach which can adjust the distribution of descriptors to make the MLN-based descriptor can be refined during learning stage. This approach is proved by experimental results as shown in Table V and VI. The descriptors have to be constructed without manual intervention under the condition of without prior knowledge.

Moreover, the other distinct feature of our proposed system is that the learning knowledge in both image and rotation angle can be done by just only one model. Most of HD models [8-11] are focus on learning knowledge in the same domains, e.g. handwriting [8], text categorization[9], Speech Recognition [10], images[11]. To handle the knowledge in different domains, we involve the technique of self-taught clustering and parallel structure model, and the experimental results also support the feasibility of learning knowledge in different domains by our proposed HD model.

7. Conclusion

An automatic learning system for vision system is an important part in assembling production line with small-volume, large-variety components. In this work, we reverse the concept of traditional vision system. The robustness of feature points and descriptor is not main concerns. Instead, the relation between input and output is the most essential.

To learn the relationship between input and output, we propose a HD model which combines the concept of deep learning, transfer learning, and Markov logic network. The model acquires self-taught ability which can infer relational model and self-supervised the performance of learning results. Being an automatic system, tackling large amount of unlabeled data and inferring relation with labeled data is necessary. The relation between features in different levels can be represented as a discriminative distribution by the model. Through the discriminative distributions, KL divergence is further involved to find the max co-cluster between labeled and unlabeled data, and makes system stable under growing database.

Moreover, proposed system includes image features, descriptors and rotation angles for robot arm. These different domain features are impossible to be learned simultaneously by traditional single layer model, but the experimental results prove proposed HD model can transfer and learn different domain knowledge, and recognize 3D object without manual intervention. We believe this system is practical in real industrial assembling production line, and save labor cost.

References

[1] Torgny Brogrdh, "Present and future robot control development - An industrial perspective," *Annual Reviews in Control*, Vol. 31, no. 1, pp. 69-79, 2007.

[2] Ebrahim Mattar, "Robotics Arm Visual Servo: Estimation of Arm-Space Kinematics Relations with Epipolar Geometry, Robotic Systems - Applications, Control and Programming,"*Dr. Ashish Dutta (Ed.)*, ISBN: 978-953-307-941-7, InTech, DOI: 10.5772/25605.

[3] So-Youn Park ,Yeoun-Jae Kim ,Ju-Jang Lee ,Byung Soo Kim, and Khalid A.Alsaif, "Controlling robot arm manipulator using image-based visual servoing without pre-depth information," in *Proc. IEEE Int. Conf. Ind. Ele.*, 2011, pp. 3157-3161.

[4] K. Deguchi, H. Sakurai, and S. Ushida, "A Goal Oriented just-in-time visual servoing for ball catching robot arm," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2008, pp. 3034-3039.

[5] J. Baker, L. Deng, J. Glass, S. Khudanpur, Chin hui Lee, N. Morgan, and D. OShaughnessy, "Developments and directions in speech recognition and understanding, part 1," *IEEE Signal Processing Magazine*, vol. 26, no. 3, pp. 75-80, May. 2009.

[6] S. Furui, 'Digital Speech Processing, Synthesis," in *Marcel Dekker*, 2000.

[7] Tong Simon, and Daphne Koller, "Support vector machine active learning with applications to text classification," *The Journal of Machine Learning Research*, pp. 45-66, 2002.

[8] Hinton, G. E., Osindero, S. and Teh, Y, "A fast learning algorithm for deep belief nets," *Neural Computation*, pp. 1527-1554, 2006.

[9] Srivastava, N., Salakhutdinov, R. R. and Hinton, G. E., "Modeling Documents with a Deep Boltzmann Machine," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, 2013.

[10] Graves, A., Mohamed, A. and Hinton, G. E., "Speech Recognition with Deep Recurrent Neural Networks," in *Proc. Int. Conf. Acoustics, Speech and Signal Processing*, 2013, pp. 6645-6649.

[11] Ranzato, M., Mnih, V., Susskind, J. and Hinton, G. E., "Modeling Natural Images Using Gated MRFs," *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2013.

[12] H. Bay, A. Ess, T. Tuytelaars, and L. Gool, "SURF: Speeded up robust features," *Comput. Vis. Image Understand.*, vol. 110, no. 3, pp. 346-359, Mar. 2008.

[13] Zen Chen and Shu-Kuo Sun, "A Zernike Moment Phase-Based Descriptor for Local Image Representation and Matching," *IEEE Trans. Image Process*, vol. 19, no. 1,pp. 205219, Jan. 2010.

[14] A. Alahi, R. Ortiz, and P. Vandergheynst, "Freak: Fast retina keypoint," in *Proc. Computer Vision and Pattern Recognition*, 2012.

[15] S. Leutenegger, M. Chli, and R. Siegwart, "Brisk: Binary Robust Invariant Scalable Keypoints," in *Proc. IEEE Int. Conf. Computer Vision*, 2011.

[16] Vijay Chandrasekhar, Gabriel Takacs, David Chen, Sam S. Tsai, Jatinder Singh, and Bernd Girod, "Transform coding of image feature descriptors," *Visual Communications and Image Processing*, 2009.

[17] Matthew Richardson and Pedro Domingos, "Markov logic networks," *Journal of Machine Learning*, Vol. 62, no. 1, pp. 107-136, Feb. 2006.

[18] L. Mihalkova, T. Huynh, and R.J. Mooney, "Mapping and Revising Markov Logic Networks for Transfer Learning," in *Proc. Conf. Ass. Adv. Arti. Int.*, 2007, pp. 608-614.

[19] Kok, Stanley and Domingos, Pedro, "Learning the Structure of Markov Logic Networks," in *Proc. Int. Conf. Machine Learning*, 2005, pp. 441-448.

[20] Parag Singla and Pedro Domingos, “Discriminative training of Markov logic networks,” in *Proc. Int. Conf. Articial Intelligence*, 2005.

[21] Wenyuan Dai, Qiang Yang, Gui-Rong Xue and Yong Yu, “Self-taught Clustering,” in *Proc. Int. Conf. Machine Learning*, 2008.

[22] Saso Dzeroski, “Multi-relational Data Mining: An Introduction,” *SIGKDD Explore Newsletter*, Vol. 5, No. 1, Jul. 2003, pp.1-16.

[23] Sinno Jialin Pan, and Qiang Yang, “A Survey on Transfer Learning,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp.1345-1359, Oct. 2010.

[24] T. Dietterich, L. Getoor, and K. Murphy, “Statistical Relational Learning and its Connections to Other Fields,” in *Proc. Int. Conf. Machine Learning*, 2004.

[25] Jing Gao and Wei Fan and Jing Jiang and Jiawei Han, ‘Knowledge Transfer via Multiple Model Local Structure Mapping,” in *Proc. Int. Conf. Knowledge Discovery and Data Mining*, 2008, pp. 283-291.

[26] T. Joachims, “Making large-scale svm learning practical., advances in kernel methods - support vector learning,” MIT-Press, 1999.

[27] A. J. Carlson, C. M. Cumby, J. L. R. Nicholas D. Rizzolo, and D. Roth, “Snow learning architecture,” *Technical report UIUCDCS*, 1999.

[28] Cover, T. M. and Thomas, J. A., “Elements of information theory,” Wiley-Interscience.

[29] Yoshua Bengio, Pascal Lamblin, Popovici Dan, and Larochelle Hugo, “Greedy Layer-Wise Training of Deep Networks,” *Advances in Neural Information Processing Systems*, 2007.

[30] Fei-Fei L., R. Fergus, and P. Perona, “Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories,” in *Proc. Computer Vision and Pattern Recognition*, 2004, pp. 178-178.

[31] Grifn, G. Holub, and P. Perona, “The Caltech 256,” *Caltech TechnicalReport.*

[32] Karthikeyan Vaiapury, Anil Aksay and Ebroul Izquierdo, “GrabcutD: Improved Grabcut Using Depth Information,” in *Proc. Int. Conf. ACM*, 2008, pp. 57-62.

[33] Z. Zivkovic, “GrabcutD: Improved Grabcut Using Depth Information,” in *Proc. Int. Conf. Pattern Recognition*, 2004, pp. 28-31.

[34] Fei Sha and Fernando Pereira, “Shallow parsing with conditional random elds,” in *Proc. Int. Conf. North American Chapter of the Association for Computational Linguistics on Human Language Technology*, 2003.

[35] Yanning Zhang,Zhi-Hua Zhou, Changshui Zhang and Li, Ying, “B-SIFT: A Highly Efficient Binary SIFT Descriptor for Invariant Feature Correspondence,” in *Proc. The Second Sino-foreign-interchange Conference on Intelligent Science and Intelligent Data Engineerin*, pp.426-433, 2012.

[36] S. Zhang, Q. Tian, K. Lu, Q. Huang and W. Gao, “Edge-SIFT: Discriminative binary descriptor for scalable partial-duplicate mobile search,” *IEEE Transactions on Image Process*, vol. 22, no. 7, pp.2889-2902, Jul. 2013.

[37] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Guo., “Locality-constrained Linear Coding for Image Classication,” in *Int. Conf. CVPR*,pp. 3360-3367, 2010.

[38] L. Seidenari, G. Serra, AD. Bagdanov, and A. Del Bimbo, “Local Pyramidal Descriptors for Image Recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 5,pp.1033-1040, May,2014.

[39] Liefeng Bo, Xiaofeng Ren, and D. Fox, “Multipath Sparse Coding Using Hierarchical Matching Pursuit,” in *Proc. Computer Vision and Pattern Recognition*, 2013, pp. 660-667.

[40] Matthew D Zeiler, and Rob Fergus, “Visualizing and Understanding Convolutional Networks,” in *Proc. ECCV*, 2014, pp. 818-833.