Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Multiple Kernel Based Transfer Learning for the Few-Shot Recognition Task in Smart Home Scene

S. C. Chang, C. H. Zhao*

**State Key Laboratory of Industrial Control Technology, College of Control Science and Engineering, Zhejiang University, Hangzhou, 310027, China (Corresponding author*, email: chhzhao@zju.edu.cn)*

Abstract: With the development of artificial intelligence, smart home plays an increasingly important role in daily life. Since new objects may constantly appear at home and the collecting of enough training samples sometimes can be hard, the few-shot recognition task is essential and practical in smart home scene. MULticlass Transfer Incremental LEarning (MULTIpLE) is an effective algorithm that can perform transfer learning for class increment with few new samples based on Support Vector Machine (SVM). But the features of images are generally high-dimensional and the selection for kernel function affects the performance of MULTIpLE. In this paper, a new transfer learning algorithm based on multiple kernel learning, termed MULticlass Transfer Incremental LEarning based on Multiple Kernel Learning (MULTIpLE-MKL), is proposed for the few-shot recognition task in smart home scene. There are two main steps for the MULTIpLE-MKL, including the first multiple kernel learning stage and the second transfer learning stage. Specifically, multiple kernel learning is first applied in the construction of SVM models to optimize the selection of kernel function. When different kernels are calculated based on different features, the sparse kernel coefficients achieve the key feature selection. Second, the SVM models learn a new class from few samples by the virtue of the transfer learning algorithm, MULTIpLE. Compared with conventional methods, experiments based on the benchmark Caltech-256 dataset demonstrate that the proposed MULTIpLE-MKL not only maintains the good performance in original classes but also shows improving recognition ability for new classes only with few training samples.

Keywords: few-shot learning, scene recognition, multiple kernel learning, transfer learning, smart home scene

1. INTRODUCTION

Smart home (Jiang et al., 2004) is a kind of concept based on the residential platform that applies multiple technologies, like automatic control and network communication technology, to integrate residential smart devices. Through building an efficient management system for residential smart devices and family schedules, a safe, comfortable and green living environment can be created. At present, the smart home industry is in the early stage of development which needs further development and promotion for a larger consumption market. Its great potential also attracts increasing attention in this field. The research of scene recognition (Mei et al., 2013), voice recognition and understanding (Matić et al., 2017), internet of things (Paul et al., 2018), information fusion (Zhang et al., 2008) has been widely applied in the construction of smart home.

Scene recognition is an important task in smart home scene. For example, home robots often need to identify the objects at home first, and they can serve us better then. To guarantee the safety of home, the security system needs accurate face recognition technology to discriminate hosts and strangers. Besides, smart home system needs real-time location and recognition of human to provide specific service and help. Thus, the research of scene recognition is quite meaningful.

Generally, the conventional scene recognition methods based on images consist of two parts. First, various image features are extracted from the original images, including colour feature, texture feature, shape feature, etc. Next, machine learning models like decision tree (Ghoshal et al., 2011), SVM (Zheng et al., 2011), and neural networks (Zhang et al., 2010) are constructed for the scene recognition tasks. Zheng et al. (2017) proposed a superpixel based linear distance coding framework to implement food image recognition at home. In order to recognize the location and motion of human based on cameras at home, Nguyen et al. (2009) first adopted silhouette method to recognize and calculate human location, and then used SVM for human motion recognition. With the development of deep learning, the emergence of Convolutional Neural Networks (CNNs) (Bengio et al., 1995) provides a novel solution for scene recognition through extracting the deep features of an image. However, these scene recognition methods require sufficient training samples as basis. In smart home scene, new furniture and household products may constantly emerge in the market, so models are faced with increasing recognition demand. For a constructed model, it will be time-consuming to retrain with original and new training samples together. The storage of massive images is also a problem for smart mobile devices. In general, only few images of new products can be collected in a short time. How to reconstruct or modify the model effectively with few samples is a problem worthy of research.

Recently, the emergence of transfer learning provides a kind of solution for the problem in smart home scene. Transfer learning (Pan et al., 2010) can deal with the difference of feature space or distribution between original data (referred to as source domain) and future data (referred to as target domain). By virtue of the idea of transfer learning, *N*-class and *N+1*-class objects can be respectively regarded as source domain and target domain. Generally, transfer is feasible when source and target domain are related. As the knowledge of source domain model can be transferred into the discrimination for original class and new class, transfer learning can quickly implement class increment based on source domain model and few samples in target domain. Many researches have already applied transfer learning algorithm to implement class increment. TrAdaboost algorithm (Dai et al., 2007) is a typical instance-transfer method based on Adaboost algorithm. In every iteration, the algorithm adjusts the weight of each sample according to the prediction of samples from different domains. Therefore, useful source domain samples for target domain can be selected through several iterations. But the drawback of this algorithm is that the samples of original classes need to be kept during transfer. Li et al. (2003) proposed a one-shot transfer algorithm based on Bayesian classifier to implement class increment. The information of source domain samples is represented by the parameters of probability density function, which act as the prior probability density function of target domain model. Then the posterior probability of target domain model is learned with new samples. Transfer learning algorithm based on neural networks (Vapnik et al., 2017) can also implement class increment. The shallow layer parameters of source domain network can be directly transferred to target domain network, while other parameters are randomized and updated by new samples. Although this algorithm can implement transfer, more new samples are needed compared with other methods. Tommasi et al. (2010) proposed a transfer algorithm based on SVM. This algorithm supposes the new classifier to discriminate original classes and new class should be as close as possible to the linear combination of original classifiers. Then, the original classifiers together with the new classifier can implement class increment. Based on this algorithm, Kuzborskij et al. (2013) proposed an improved algorithm, termed MULTIpLE. This algorithm supposes not only the new classifier should be as close as possible to the linear combination of original classifiers but also the original classifiers should have tiny modification. Compared with the algorithm proposed by Tommasi et al. (2010), MULTIpLE can balance between transferring to new class and preserving what has already been learned in original classes. As kernel selection is quite important in kernel-based methods and may influence the performance a lot, MULTIpLE only applies single kernel function, which may be a limitation.

Despite these advances, most of the transfer learning algorithms mainly focus on how to transfer knowledge while neglecting the importance of different features in different classification tasks. Obviously, colour features and texture features should be paid more attention rather than shape features for the classification of horse and zebra. It will be disadvantageous for discrimination if all features are treated equally. Meanwhile, image features are generally of high dimension and they will cost more in storage and calculation. Therefore, how to select the key features is of importance.

In this paper, a new transfer learning algorithm, termed MULTIpLE-MKL, is proposed based on multiple kernel learning for the few-shot recognition task in smart home scene. This algorithm mainly consists of two steps, the first multiple kernel learning stage and the second transfer learning stage. First, multiple kernel learning is applied in the construction of original SVM models. The main idea of multiple kernel learning is to implement more complex nonlinear high-dimensional mapping through the linear combination of multiple kernels. Kernel can be optimized through learnable coefficients. When different kernels are calculated based on different features, sparse coefficients can implement key feature selection at the same time. Second, based on the idea of MULTIpLE, the original SVM models learn one new class from few samples through model modification and addition. Thus, the introduction of multiple kernel learning can address the kernel optimization and feature selection problems for performance improvement.

The contributions of this paper are summarized as below:

- Multiple kernel learning is introduced into the transfer learning algorithm, MULTIpLE, to implement kernel function optimization and performance improvement.
- Different kernels are calculated by different features so that kernel selection implements key feature selection.

2. THE PROPOSED MULTIpLE-MKL METHOD

The method mainly consists of two parts, the construction of multiple kernel learning model and the transfer of model. The details of each part are introduced below.

2.1 The construction of multiple kernel learning model

SVM is a powerful and famous classification model. It constructs binary classification hyperplane based on margin maximization. Generally, the objective function of linear SVM can be expressed in (1).

$$\min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{m} loss(y_i, \boldsymbol{w}^T\boldsymbol{x}_i + b) \tag{1}$$

$\boldsymbol{w}$ and b are normal vector and bias of hyperplane. $\boldsymbol{x}_i$ and $\boldsymbol{y}_i$ respectively denote the feature vector and label of sample *i*. *m* and *C* are respectively the number of training samples and the trade-off coefficient. $loss(\cdot,\cdot)$ is the loss function with several forms.

SVM can obtain pretty good performance with few samples and can be easily extended to linear-inseparable problem by kernel trick. Thus, SVM gets wide applications in many scenarios. But the kernel selection can influence the performance to some extent and single kernel function may not be the best choice for kernel-based methods.

Therefore, Multiple Kernel Learning (MKL) (Sonnenburg et al., 2006) is proposed to apply the linear combination of multiple kernels as the kernel function of model. The linear combination of multiple kernels still satisfies the semi-definite property, which proves to be a qualified kernel function. MKL can be applied in SVM as well as other kernel-based methods. Equation (2) is the mathematical expression of MKL.

$$\begin{aligned} &K(\boldsymbol{x},\boldsymbol{x}')=\sum_{m=1}^{M} d_m K_m(\boldsymbol{x},\boldsymbol{x}') \\ &s.t. \quad d_m \geq 0, \quad \sum_{m=1}^{M} d_m = 1 \end{aligned} \tag{2}$$

$\boldsymbol{x}$ and $\boldsymbol{x'}$ respectively denote different samples. d_m represents the combination coefficient (referred to as kernel coefficient below) of kernel function $K_m(\cdot,\cdot)$, and M is the number of kernels. The kernel coefficients satisfy the constraints of non-negativity and unit summation.

When different kernels are calculated based on different features, the kernel coefficients can represent the importance of features. Because of the unit summation constraint, the learned kernel coefficients will be sparse for feature selection. MKL can simultaneously learn the model parameters and kernel coefficients, so it belongs to embedded feature selection method.

Therefore, we construct SVM classification model for original classes based on MKL to implement kernel optimization and feature selection. As for the multi-class classification problem of N-class household objects, the one-versus-rest strategy can transform N-class classification problem into N binary classification problems. SimpleMKL (Rakotomamonjy et al., 2007) is applied to learn N multiple kernel SVM models. The predicted label of a sample is assigned by the classifier that yields the maximum value. Equation (3) is the mathematical form of prediction.

$$\hat{y} = \underset{n\in\{1,2,\ldots,N\}}{argmax} \sum_{i=1}^{m} \alpha_{ni} y_i K(\boldsymbol{x},\boldsymbol{x}_i) + b_n \tag{3}$$

α_{ni} is the Lagrange Multiplier corresponding to classifier n and sample i. b_n denotes the bias of classifier n. $\boldsymbol{x}_i$ and y_i are respectively the feature vector and label of training sample i where m is the number of training samples. $K(\cdot,\cdot)$ denotes the kernel function learned by MKL. $\boldsymbol{x}$ is the feature vector of test sample and $\hat{y}$ is the predicted label.

2.2 The transfer of model

After the construction of multiple kernel SVM model, N binary classification hyperplanes with the same kernel function are learned. The normal vectors of them can be denoted by $\boldsymbol{W'}$=[$\boldsymbol{W}_1'$,…, $\boldsymbol{W}_N'$]. Each hyperplane discriminates its class and other classes. To implement class increment, it is necessary to add one new hyperplane to discriminate original classes and one new class. Therefore, with few new samples, the aim is to get $N+1$ hyperplanes after transfer. In the process of transfer, the hyperplanes of original classes are updated and one new hyperplane for new class is generated. The normal vectors of original classes and new class after transfer can be represented by $\boldsymbol{W}$=[$\boldsymbol{W}_1$,…, $\boldsymbol{W}_N$] and $\boldsymbol{\omega}_{N+1}$.

MULTIpLE is a transfer learning algorithm that can balance between transferring to new class and preserving what has been learned in original classes. But some transfer learning algorithms mainly focus on the performance in new class, while in the smart home scene, original classes are as important as new class after transfer. Therefore, MULTIpLE is applied in this paper for class increment and the kernel function learned by MKL is used. The transfer learning idea of MULTIpLE supposes that the N hyperplanes of original classes should keep as invariant as possible to main the performance in original classes, and the new hyperplane should be as close as possible to the linear combination of the N original hyperplanes. The knowledge of original model can be transferred to new class through linear combination. The objective function in (4) is established based on this transfer idea. $\boldsymbol{\beta}$ is the linear combination coefficient of N original hyperplanes. M and C are respectively the number of new samples and trade-off coefficient. $\phi(\cdot)$ denotes the mapping from feature space to high-dimensional space, which is implicitly defined by kernel function. $\boldsymbol{b}$ is the bias of the model after transfer. $\boldsymbol{Y} \in \mathbb{R}^{M\times(N+1)}$ is the label matrix of new samples and Y_{in} equals to 1 if sample i belongs to class n or else -1.

$$\begin{aligned} \min_{\boldsymbol{W},\boldsymbol{\omega}_{N+1},\boldsymbol{b}} &\frac{1}{2}\|\boldsymbol{W}\text{-}\boldsymbol{W'}\|_F^2 + \frac{1}{2}\|\boldsymbol{\omega}_{N+1}\text{-}\boldsymbol{W'\beta}\|_F^2 \\ &+\frac{C}{2}\sum_{i=1}^{M}\sum_{n=1}^{N+1}(\boldsymbol{W}_n^T\phi(\boldsymbol{x}_i)+b_n-Y_{in})^2 \end{aligned} \tag{4}$$

The first term in this objective function makes the N hyperplanes of original classes keep as invariant as possible, and the second term makes the new hyperplane be as close as possible to the linear combination of the original N hyperplanes. Error loss is denoted by the third term.

The objective function above can be solved by Lagrange Multiplier based on the assumption of least square SVM (Suykens et al., 1999), which is more efficient than SVM. The model parameters after transfer can be represented in matrix form in (5)-(7), and the process of calculation is shown in (8)-(11).

$$\boldsymbol{W}_n = \boldsymbol{W}_n' + \sum_{i=1}^{M} A_{in}\phi(\boldsymbol{x}_i), n = 1,\ldots,N \tag{5}$$

$$\boldsymbol{\omega}_{N+1} = \boldsymbol{W}'\boldsymbol{\beta} + \sum_{i=1}^{M} A_{i(N+1)}\phi(\boldsymbol{x}_i) \tag{6}$$

$$\boldsymbol{b} = \boldsymbol{b}' - \begin{bmatrix} \boldsymbol{b}'' \\ \boldsymbol{b}''^T\boldsymbol{\beta} \end{bmatrix} \tag{7}$$

$$\boldsymbol{A} = \boldsymbol{A}' - \begin{bmatrix} \boldsymbol{A}'' & \boldsymbol{A}''\boldsymbol{\beta} \end{bmatrix} \tag{8}$$

$$\begin{bmatrix} \boldsymbol{A}' \\ \boldsymbol{b}'^T \end{bmatrix} = \boldsymbol{M}\begin{bmatrix} \boldsymbol{Y} \\ \boldsymbol{0} \end{bmatrix} \tag{9}$$

$$\begin{bmatrix} \boldsymbol{A}'' \\ \boldsymbol{b}''^{T} \end{bmatrix} = \boldsymbol{M} \begin{bmatrix} \phi(\boldsymbol{X})^{T} \boldsymbol{W}' \\ \boldsymbol{0} \end{bmatrix} \tag{10}$$

$$\boldsymbol{M} = \begin{bmatrix} \phi(\boldsymbol{X})^{T} \phi(\boldsymbol{X}) + \dfrac{1}{C}\boldsymbol{I} & \boldsymbol{1} \\ \boldsymbol{1} & 0 \end{bmatrix}^{-1} \tag{11}$$

$\boldsymbol{X} \in \mathbb{R}^{d\times M}$ is the feature matrix of new samples where d is the dimension of feature vector. Prediction is calculated by (3) while the only difference is that one new class is learned.

C and $\boldsymbol{\beta}$ are two hyper-parameters in this algorithm. C is assigned the same as the one in the construction of original model. $\boldsymbol{\beta}$ can be optimized by projected subgradient descent method (Kuzborskij et al., 2013).

Through the construction of MKL model and the transfer of model, class increment is implemented based on feature selection and kernel optimization.

3. EXPERIMENT

Smart home scene is the background of practical problem in this study, so we experiment based on 5106 images of 42 kinds of family objects from Caltech-256 (Gehler et al., 2009) which is a standard dataset composed of 256 categories common objects in life. There are about 100~200 samples in each class, and some samples are shown in Figure 1.

Fig. 1. Some samples in dataset

Pyramid histogram of oriented gradient, region covariance, and local binary patterns, totally 14 group features of 6465 dimensions, are extracted as image features. 10 objects (backpack, baseball-glove, bathtub, binoculars, breadmaker, computer-monitor, head-phones, ladder, mattress, and T-shirt) with the most samples are used in this experiment. In each time experiment, one object is selected as new class, and the rest are taken as original classes. Thus, there will be totally 10 times experiments. Prediction accuracy is used as the performance evaluation criterion. 80 and 20 samples from each original class consist of training set and validation set. 1, 3, 5, 10 samples from each original and new class consist of 4 kinds of new training sets to compare transfer performance with different size of new training set. 20 samples from each original and new class consist of test set.

To validate the superiority of MKL and transfer learning, five contrast models are designed, namely non-transfer model, reconstruct model, linear transfer model, Radial Basis Function (RBF) kernel transfer model and average kernel transfer model. Non-transfer model only uses new training set to learn a new SVM classifier together with the original multiple kernel SVM model to implement class increment. Reconstruct model learns multiple kernel SVM model from scratch with new training set only. Linear, RBF kernel and average kernel transfer model respectively uses linear, RBF and average kernel function to learn original SVM model and transfer through MULTIpLE. The penalty coefficient C, and the bandwidth of Gaussian kernel function in RBF and average kernel transfer model are optimized in $[10^{-1}, 10^{0}, \ldots, 10^{6}]$. Every kernel in MULTIpLE-MKL is Gaussian kernel, and totally 56 kernels including 4 kinds of bandwidth (10,100,1000,10000) are used.

As shown in Figure 2 and 4, the prediction accuracy curves in original classes and new class with the size of new training set are respectively plotted. Figure 3 is the local zoom of the red rectangular in Figure 2. Dotted and solid line denote each model before and after transfer. Because original models cannot predict new class, in other words, the prediction accuracy in new class is zero, the prediction accuracy curves in new class for original models are not plotted.

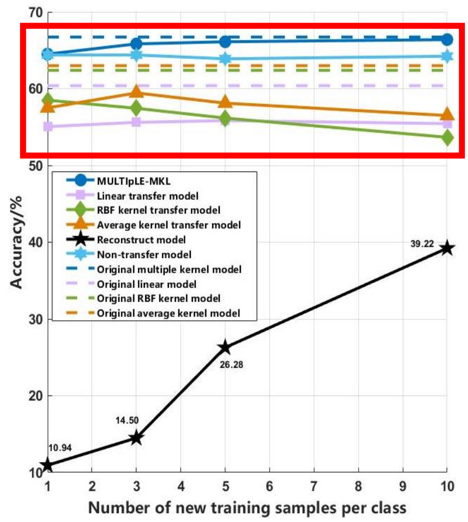

Fig. 2. Prediction accuracy curves in original classes with the number of new training samples per class

From Figure 2 and 3, it is obvious that original multiple kernel model performs better than other ordinary kernel function models. Because RBF and average kernel function are calculated in a similar way, their performances also show some similarities and are superior to linear model. Compared with original models, models after transfer all show some decline, while only MULTIpLE-MKL gradually improves as the size of new training set increases, and almost reaches the level of the original model. The performances of other ordinary kernel function models after transfer nearly stay the same or even decline a little. Non-transfer model keeps the classifiers for original classes, so its performance is a bit worse than MULTIpLE-MKL but still better than other ordinary kernel function transfer models. Reconstruct model shows poor performance and severe under-fitting phenomenon due to the lack of training samples.

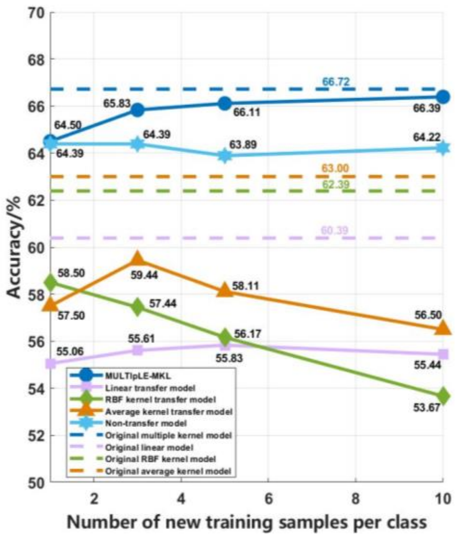

Fig. 3. Local zoom of the red rectangular in Figure 2

From Figure 4, it is clear that reconstruct model outperforms other models when the new training set gets larger. The main reason for this is that the larger new training set to some extent alleviate the severe under-fitting problem, so its performance shows a significant improvement. In addition, reconstruct model is learned from scratch without the original model, so there is no inclination to original classes or new class. But models based on the original models will have obvious inclination to original classes, reflecting in different performance in original classes and new class. Besides, other models perform similarly, and all promote with the increment of the new training set. Among them, MULTIpLE-MKL performs the best which demonstrates that the high-dimensional space constructed by SVM for original classes is appropriate to discriminate new class.

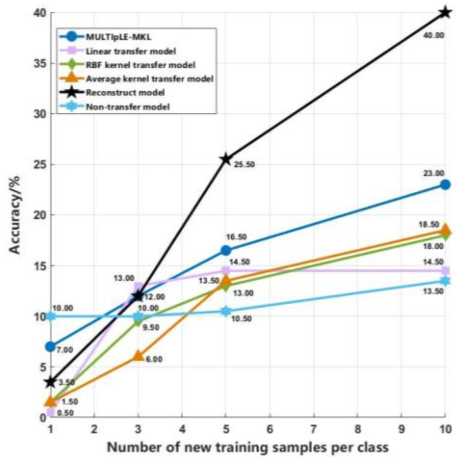

Fig. 4. Prediction accuracy curves in new class with the number of new training samples per class

From the experimental results above, it demonstrates that MULTIpLE-MKL shows better performance both in original classes and new class when single kernel function cannot find a proper high-dimensional mapping. In addition, transfer models outperform non-transfer and reconstruct model which proves the effectiveness of MKL and transfer learning. With MKL, MULTIpLE-MKL just needs to use 5~7 group features of 14 group features and improves its performance. But high computational complexity is a limitation for MKL. Thus, MULTIpLE-MKL will be limited by low efficiency when original training set is dozens of times larger than here.

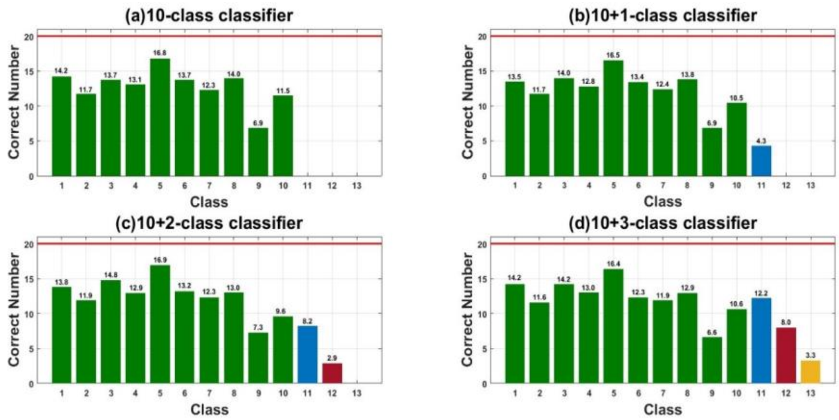

Fig. 5. The performance of MULTIpLE-MKL in each class after multiple times transfers

The experiment above only considers increasing one new class. But in practical application, it is often necessary to constantly transfer model to increase multiple new classes. Therefore, to show the continuous transfer ability of the proposed model, another experiment is set only for MULTIpLE-MKL that the 10 classes in the above experiment are used as original classes, and baseball-bat, bonsai, laptop are used as new classes. Model transfer is successively performed three times and model learns one new class each time, while other settings are the same as before. 4 samples from each class are provided for each time transfer and the experiment is repeated 10 times. The averaged results are shown in Figure 5. Figure 5(a) shows the number of correctly predicted test samples in each class of original model. Figure 5(b)-(d) respectively show the number of correctly predicted test samples in each class after each time transfer. The horizontal red line represents the number of test samples in each class. It can be seen that MULTIpLE-MKL can maintain the performance in original classes and gradually elevate the performance in new classes as the model constantly transfers. The performance in class 11 after three times transfers nearly reaches the performance in original classes while only 3*4=12 training samples of class 11 are used. It demonstrates that MULTIpLE-MKL can fully exert the power of the original model to implement class increment with few samples.

4. CONCLUSION

In this study, a new transfer learning algorithm, termed MULTIpLE-MKL, is proposed based on MKL for few-shot recognition task in smart home scene. The new algorithm can not only address the problem of few samples for new objects in smart home scene but also implement feature selection and kernel optimization to improve computational efficiency and transfer performance. Experimental results demonstrate the effectiveness of transfer learning and MKL. In addition, multiple times transfers can constantly implement class increment while gradually improving the performance in new classes and maintaining the performance in original classes as well, which demonstrates the practical value of the proposed method. However, MULTIpLE-MKL also has its limitation, like the high computational complexity. Besides, the new training set used for model transfer includes both original and new class samples, so future work can extend the method to scenarios when only new class samples accessible.

ACKNOWLEDGEMENT

This work is supported by NSFC-Zhejiang Joint Fund for the Integration of Industrialization and Informatization (No. U1709211), and Zhejiang Key Research and Development Project (2019C03100).

REFERENCES

Bengio, Y., LeCun, Y., Nohl, C. and Burges, C. (1995). LeRec: A NN/HMM hybrid for on-line handwriting recognition. *Neural computation*, 7(6), 1289-1303.

Dai, W., Yang, Q., Xue, G.R. and Yu, Y. (2007). Boosting for transfer learning. *In Proceedings of the 24th international conference on Machine learning*, 193-200.

Gehler, P. and Nowozin, S. (2009). On feature combination for multiclass object classification. *In 2009 IEEE 12th International Conference on Computer Vision*, 221-228.

Ghoshal, R., Roy, A., Bhowmik, T.K. and Parui, S.K. (2011). Decision tree based recognition of Bangla text from outdoor scene images. *International Conference on Neural Information Processing*, 538-546.

Jiang, L., Liu, D.Y. and Yang, B. (2004). Smart home research. *In Proceedings of 2004 International Conference on Machine Learning and Cybernetics*, 2, 659-663.

Kuzborskij, I., Orabona, F. and Caputo, B. (2013). From N to N+1: Multiclass transfer incremental learning. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3358-3365.

Li, F.F., Fergus, R. and Perona, P. (2003). A Bayesian approach to unsupervised one-shot learning of object categories. *In Proceedings Ninth IEEE International Conference on Computer Vision*, 1134-1141.

Matić, M., Stefanović, I., Radosavac, U. and Vidaković, M. (2017). Challenges of integrating smart home automation with cloud based voice recognition systems. *In 2017 IEEE 7th International Conference on Consumer Electronics-Berlin (ICCE-Berlin)*, 248-249.

Mei, K., Feng, S., Li, G., Chou, W. and Zhang, L. (2013). A virtual touch event method using scene recognition for digital television. *IEEE Transactions on Consumer Electronics*, 59(1), 214-219.

Nguyen, Q.C., Shin, D. and Kim, J. (2009). Real-time human tracker based on location and motion recognition of user for smart home. *In 2009 Third International Conference on Multimedia and Ubiquitous Engineering*, 243-250.

Pan, S.J. and Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10), 1345-1359.

Paul, C., Ganesh, A. and Sunitha, C. (2018). An overview of IoT based smart homes. *In 2018 2nd International Conference on Inventive Systems and Control (ICISC)*, 43-46.

Rakotomamonjy, A., Bach, F., Canu, S. and Grandvalet, Y. (2007). More efficiency in multiple kernel learning. *In Proceedings of the 24th international conference on Machine learning*, 775-782.

Sonnenburg, S., Rätsch, G., Schäfer, C. and Schölkopf, B. (2006). Large scale multiple kernel learning. *Journal of Machine Learning Research*, 7(Jul), 1531-1565.

Suykens, J.A. and Vandewalle, J. (1999). Multiclass least squares support vector machines. *IJCNN*, 99, 900.

Tommasi, T., Orabona, F. and Caputo, B. (2010). Safety in numbers: Learning categories from few examples with multi model knowledge transfer. *In 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 3081-3088.

Vapnik, V. and Izmailov, R. (2017). Knowledge transfer in SVM and neural networks. *Annals of Mathematics and Artificial Intelligence*, 81(1-2), 3-19.

Zhang, L., Leung, H. and Chan, K.C. (2008). Information fusion based smart home control system and its application. *IEEE Transactions on Consumer Electronics*, 54(3), 1157-1165.

Zhang, Y., Pan, W. and Chen, X.X. (2010). Scene Recognition Based on Partially Connected Neural Network. *Journal of Xiamen University (Natural Science)*, 4, 10.

Zheng, J., Wang, Z.J. and Zhu, C. (2017). Food image recognition via superpixel based low-level and mid-level distance coding for smart home applications. *Sustainability*, 9(5), 856.

Zheng, M.X., Xue, J. and Zhang, T. (2011). Recognition of Outdoor Scene Image Features Based on SVM. *Computer Simulation*, 4.