Evaluation of LC-KSVD on UCF101 Action Dataset

Hyunjong Cho[1], Hyungtae Lee[1], Zhuolin Jiang[2]
[1]University of Maryland, College Park
[2]Noah's Ark Lab, Huawei Technologies
cho@cs.umd.edu, htlee@umd.edu, zhuolin.jiang@huawei.com

Abstract

The evaluation of action recognition algorithms on datasets with a large number of action classes taken under uncontrolled conditions reveals the insights and shortcomings of the algorithms, enabling fair comparisons across them. In this paper, we evaluate label consistent K-SVD (LC-KSVD), a discriminative dictionary learning algorithm for sparse representation of input signals, on the UCF101 action dataset. LC-KSVD incorporates a new label consistency constraint in addition to the reconstruction error and classification error in the objective function, so as to enforce discriminability during the dictionary learning process. UCF101 is currently the largest and most challenging action dataset, consisting of 101 action classes with 13320 clips. We obtained 66.1% overall classification accuracy of LC-KSVD1 and 66.3% of LC-KSVD2 on three standard UCF101 train/test partitions, which outperform the baseline result (43.9%) obtained using a naïve Bag-of-Words approach. Furthermore, we analyze the LC-KSVD classifier, learned jointly in the dictionary learning process, by comparison with an SVM classifier, and observe marginally better performance than the SVM classifier.

1. Introduction

Action recognition in the computer vision community has been widely researched but remains challenging. Several approaches have been introduced to tackle this topic, as comprehensively surveyed in recent papers [1], [2]. Despite the massive demands on action recognition from a variety of application areas such as user-uploaded online videos, surveillance, camera-equipped appliances, etc., it seems existing approaches are not yet ready to go into the wild since there is still a considerable gap between the performance on action recognition in research labs and the real world. The main reason for the difference is that many existing recognition algorithms are evaluated on datasets containing only a small number of action classes taken under controlled conditions with similar scenarios, e.g. KTH, Weizmann, IXMAS, which do not reflect the human actions in the real world.

Recently, larger and more realistic action video datasets have been released to make the evaluation processes of action recognition algorithms much more practical. They contain a large number of action classes and more clips for each class, uncontrolled recording conditions such as background cluttering, varying scale, camera movement, lighting, occlusion, etc. HMDB51 [6], UFC YouTube [7], UFC50, and UFC101 [8] are examples of this kind of dataset that include real human actions in the wild. Section 2 describes these datasets in detail.

In this paper, we evaluate label consistent K-SVD (LC-KSVD) [3], a dictionary learning algorithm for sparse representation of input signals, on the UCF101 dataset, which is currently the largest and most challenging action dataset. K-SVD [4] is an algorithm for producing overcomplete dictionaries for the sparse representation of input signals. K-SVD searches the best possible dictionary that represents each input signal as a sparse linear combination of dictionary atoms, while minimizing the reconstruction error. LC-KSVD aims to leverage the label information of input signals to design a discriminative as well as reconstructive dictionary by incorporating a new label consistency constraint called "discriminative sparse-code error" into the K-SVD objective function. In the previous work of Jiang *et at*. [3], LC-KSVD was evaluated on static image datasets such as Extended YaleB, AR face database, Caltech101, Caltech256, and 15 scene categories for image classification purpose. LC-KSVD was also evaluated on an action dataset, UCF Sports, in [3] but the dataset only contains 10 sport action classes, so it is necessary to evaluate the performance of the algorithm on action recognition task using a more realistic dataset.

Our contribution is twofold. First, we evaluate the LC-KSVD dictionary learning algorithm on the largest action dataset with detailed parameter setting. Also we analyze the LC-KSVD classifier by comparison with a SVM classifier in order to observe how efficient the LC-KSVD classifier is. This paper is structured as follows. Section 2 briefly summarizes the existing action datasets. The theory of K-SVD and LC-KSVD is described in Section 3. Experimental results and evaluation of the method are presented in Section 4. Finally, Section 5 draws conclusions.

2. Action Datasets

There have been many datasets on human action and activity recognition, each with their own attributes in terms of recording scenario, number of actions, camera movement, etc. Chaquet *et al.* [5] presents a detailed summary of a large number of action datasets.

KTH and Weizmann are two early video datasets created for action recognition. KTH has 600 videos of six human actions filmed by 25 actors in four different scenarios. Weizmann dataset includes 10 human actions performed by nine actors. Although KTH changes the clothing of actors and lighting to add variations, it is hard to expect them to reflect the human actions in a real world as they were filmed in controlled conditions with static or simple background, fixed viewpoint, similar action scenarios, etc.

HMDB51 [6] is one of the largest datasets that contains 6849 clips divided into 51 action classes. HMDB51 and UCF datasets were collected from various sources, e.g. movies, YouTube, and Google videos. UCF YouTube [7] includes 11 actions classes, each of which is grouped into 25 groups with more than four clips. UCF50 is an extension of UCF YouTube, consists of 50 actions over 6*K* clips collected in the same way as UCF YouTube dataset. UCF101 [8] is currently the largest dataset, containing 101 action classes with 13320 clips and 27 hours of footage. More details about this dataset are provided in section 4.1. Note that the videos in HMDB51 and UCF datasets were collected from various sources, all captured in uncontrolled situations such as background cluttering, varying camera movements, low quality, lighting changes, and various scales. This makes these datasets challenging to action recognition algorithms, and enables realistic analysis of these algorithms.

3. K-SVD and Label Consistent K-SVD

3.1. K-SVD

Aharon *et al.* [4] introduced K-SVD for learning an overcomplete dictionary for the sparse representation of input signals. Given a set of signals $\mathrm{Y} = \{y_i\}_{i=1}^{N} \in \mathbb{R}^{n\times N}$ that contains n-dimensional N signals, K-SVD searches the best possible dictionary $\mathrm{D} \in \mathbb{R}^{n\times K}$ that represents each signal of Y as a sparse linear combination of K prototype signal-atoms, $\{d_i\}_{i=1}^{K}$, of D. The dictionary D and the sparse representation X are obtained by minimizing the reconstruction error while preserving the sparsity constraint as below:

$$\min_{D,X} \|Y - DX\|_F^2 \ \text{s.t.}\ \forall i, \|x_i\|_0 \le T \qquad (1)$$

The goal of K-SVD is to minimize the reconstruction error, $\|Y - DX\|_F^2$, while preserving the sparsity constraint, $\|x_i\|_0 \le T$, that restricts the number of nonzero elements in each x_i becomes less than T. To this end K-SVD alternates sparse coding and dictionary update iteratively until convergence. It first finds the sparse coding matrix X with fixed D using an approximation pursuit method. Once X is calculated, K-SVD updates the dictionary D to obtain a better sparse representation. In particular, this step updates one signal-atom, d_k, and its corresponding coefficient values stored as a row in X at a time, aiming to reduce the mean square error. Updating a dictionary element and its coefficient values simultaneously results in faster convergence as the later updates can be based on the previously updated values. Singular value decomposition (SVD) is the straightforward solution for each update.

3.2. Label Consistent L-SVD

K-SVD designs reconstructive dictionaries that minimize the reconstruction error of the original signals Y. The learned dictionary is suited for signal reconstruction but is less likely to be useful for signal classification, as the dictionary learning process does not avail itself of input signal class label information.

Once a dictionary is obtained, sparse codings based on the dictionary can be used for learning classifiers (one-versus-all or pairwise), but the dictionary is likely suboptimal for this task. A better approach would be to jointly learn the classifier and dictionary at once by incorporating a classification term in the objective function, thus producing a dictionary that is more suited for classification. Jiang *et al.* [3] proposed Label Consistent K-SVD (LC-KSVD) that incorporates a new label consistency constraint and the classification term of the input signals into the objective function, which leads to a discriminative as well as reconstructive dictionary. This model jointly learns a dictionary and a classifier, which are both more appropriate for classification and reconstruction tasks. In [3], Jiang et al. introduced two LC-KSVD approaches, LC-KSVD1 and LC-KSVD2, that add 1) only a label consistency constraint and 2) a classification term in addition to the label consistency constraint into Eq. (1), respectively.

LC-KSVD1 adds a label consistency constraint that encourages signals of the same class to have similar sparse codings. The objective function of LC-KSVD1 can be defined as

$$\min_{D,X,A} \|Y - DX\|_F^2 + \alpha\|Q - AX\|_F^2 \quad \text{s.t.}\ \forall i, \|x_i\|_0 \le T \qquad (2)$$

where $Q \in \mathbb{R}^{K\times N}$ is a label matrix that contains the ideal sparse coding distribution where the class relations between each sparse coding in X and the dictionary elements in D are represented, and $A \in \mathbb{R}^{K\times K}$ is a transformation matrix that approximates the matrix Q from the (yet undetermined) sparse codings X. The parameter α controls the trade-off

between the reconstruction and label consistency terms. In detail, a column $q_i \in \mathbb{R}^K$ in Q, has nonzero values only where the corresponding dictionary elements are from the same class as the *i*th signal y_i. For example, if y_i belongs to class *c*, then only the values in q_i that corresponding to the *c*-class dictionary elements are nonzero. By incorporating the label consistency constrain $\|Q - AX\|_F^2$ that enforces the transformed sparse coding AX to be similar to the discriminative sparse coding matrix Q, the dictionary D becomes more discriminative than the dictionary learned in Eq. (1).

LC-KSVD2 adds a classification term in addition to the label consistency constraint into the objective function as another way to learn a discriminative dictionary for the classification task. The objective function for LC-KSVD2 can be defined as

$$\min_{D,X,A,W} \|Y - DX\|_F^2 + \alpha\|Q - AX\|_F^2 + \beta\|H - WX\|_F^2 \text{ s.t. } \forall i, \|x_i\|_0 \leq T \quad (3)$$

where $H \in \mathbb{R}^{m \times N}$ contains the class labels of the input signals Y, each of which belongs to one of m classes, W is the classifier matrix, and α and β control the contributions of the terms. A column $h_i \in \mathbb{R}^m$ in H is a label vector that has one nonzero value where it denotes the class of the corresponding input signal y_i. The constraint that WX should approximate the truth classification result H enables the dictionary designed by LC-KSVD2 to be more suited for the classification task.

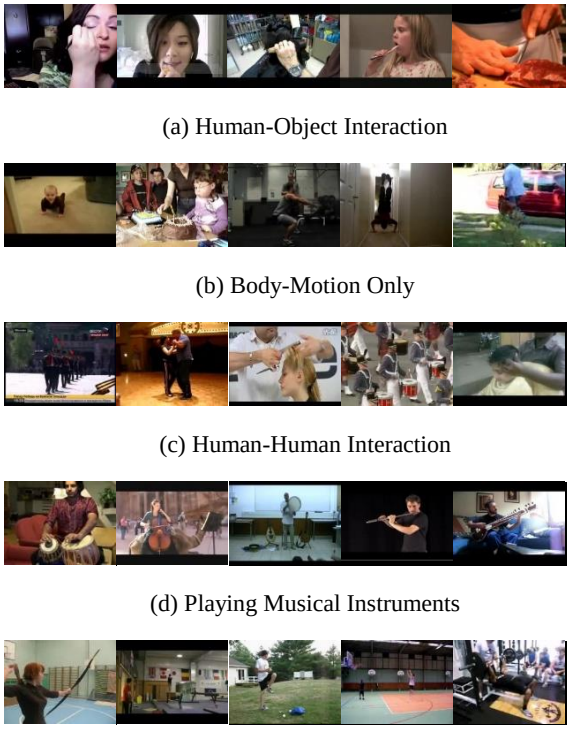
Note that both LC-KSVD1 and LC-KSVD2 learn a single dictionary and a multiclass classifier, which is in contrast to some dictionary learning approaches where multiple dictionaries or classifiers (pairwise or one-versus-all) are computed[9], [10]. LC-KSVD2 learns a dictionary and a classifier simultaneously, avoiding local optima that might be obtained if the classifier and the dictionary were learned separately. The classifier for LC-KSVD1 is learned separately using the ridge regression model [11]. Our experiments in section 4.5 compare the classification performance between the LC-KSVD classifier and a SVM classifier which is learned using the sparse codings computed using the LC-LSVD dictionary.

4. Experiments

We evaluate LC-KSVD on the UCF101 action dataset through several parameter optimizing steps. We also conduct an experiment to see how discriminative the LC-KSVD classifier is compared to other classifiers that are based on separate classifier and dictionary learning steps. To this end, we learn an SVM classifier using the sparse codings of a training set computed based on the LC-KSVD dictionary.

4.1. Dataset and Feature Descriptors

UCF101 is currently the largest dataset, containing101 action classes in 13320 clips compiled from YouTube. See Fig. 5 for the whole list of the 101 actions included. Each of the 101 action classes belongs to one of five class types: Human-Object Interaction, Body-Motion Only, Human-Human Interaction, Playing Music Instruments, and Sports. Fig. 1 shows some sample frames of this dataset.



(a) Human-Object Interaction

(b) Body-Motion Only

(c) Human-Human Interaction

(d) Playing Musical Instruments

(e) Sports

Figure 1. Example snapshots of five action class types from the UCF101 dataset. Note that the objects of the videos are all in different scales and the backgrounds are uncontrolled.

UCF101 is one of the most challenging action datasets compared to others in terms of scale and its uncontrolled conditions. It includes 101 action classes which is currently the largest number with 13*K* user-uploaded clips recorded under unconstrained realistic environments covering camera motion, cluttered background, various lighting, occlusion, low quality, etc.

We represent each clip by three types of feature descriptor which are most commonly used in action recognition: SIFT [12], STIP [13], and DTF [14]. In particular, DTF represents a clip by computing HOG, HOF, MGH, and trajectory descriptors along the dense motion trajectories. Each of the six descriptors, SIFT, STIP and four DTF descriptors, is represented using a standard bag-of-features approach with 4,000 visual words [22]. We first conduct L_1-normalization for each feature descriptor, then

concatenate them in a various combinations to form a discriminative feature descriptions. The concatenated feature descriptors of a clip form a high dimensional vector ($4000d$ ~ $24000d$), thus PCA is performed to reduce the feature dimension. Section 4.2 compares the different combinations of descriptors. The dataset is partitioned into three train/test sets, following the guideline of the standard evaluation setup [22].

4.2. Feature Descriptor Comparison

The combination of three feature descriptors form a high dimensional vector (< $24000d$) if concatenated. As shown in Fig. 2, we first compare individual descriptor, then combine them to find the most discriminative and complementary combinations for the preference on low dimensional video representation. Compared to SIFT and STIP features, DTF shows much higher discrimination power for the classification task. The feature concatenations along with DTF slightly change the classification accuracy. Finally, we use DTF+STIP concatenation for the representation of videos in the following experiments.

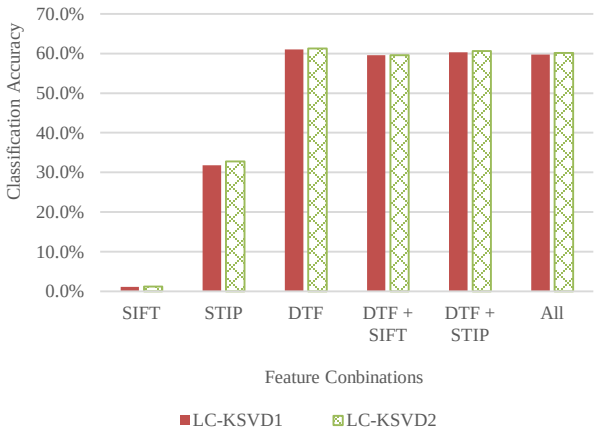


Figure 2. Classification accuracy of individual feature descriptors, and some combinations across them.

4.3. α and β:Label Consistency and Classification constraints

α and β control the contributions of the label consistency and classification terms along with the reconstruction error, respectively. Fig. 3 shows the classification accuracy for the selections of α and β; while we perform a coarser search over a larger range of α and β values, we only show the most interesting range where performance is optimal. We observe $\alpha = 0.012$ and $\beta = 0.001$ yield optimal performance on UCF101.

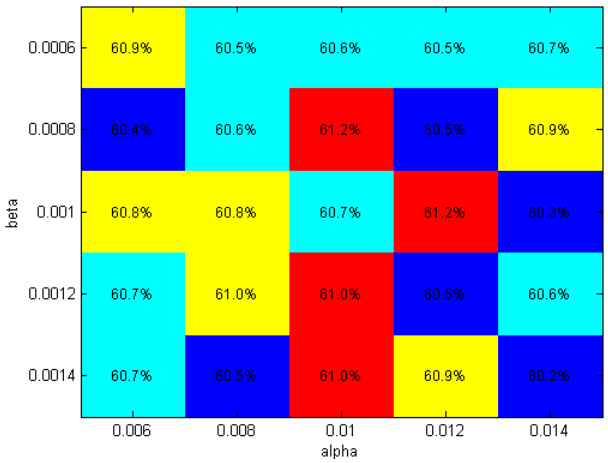


Figure 3. Classification accuracy on the UCF101 dataset with different α and β.

4.4. Dictionary Size

Each of three standard train/test partitions consists of over $9K$ train clips and over $3K$ test clips, there are about 90 clips per action in each train set. As already shown in [3], LC-KSVD does not degrade much when using a portion of training clips compared to using the entire training set, so we test different sizes of dictionaries with 5, 10… 35, 40 random clips per each action category, yielding 505, 1010… 3525, 4040 elements dictionaries. It is also expected that having a compact dictionary will reduce computation time, although some approaches [15], [16] have been introduced for fast sparse coding. Fig. 4 shows the effects of different dictionary sizes on the classification accuracy.

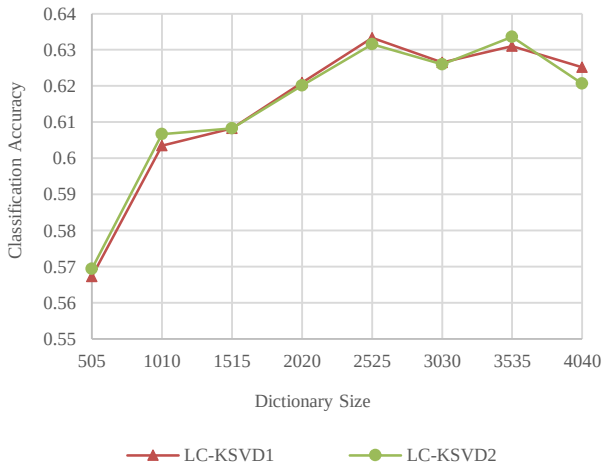


Figure 4. Classification accuracy variations along with the dictionary size. We select dictionary size 2525 as the size is relatively compact but preserves the classification accuracy.

4.5. Overall Performance

We measure the overall performance of LC-KSVD using three standard train/test partitions. In addition, the LC-KSVD classifier is compared to a SVM classifier learned using the sparse codes of same train/test sets computed using the LC-KSVD dictionary. Table 1 summarizes the experiment results.

	LC-KSVD1/ LC-KSVD1+SVM	LC-KSVD2/ LC-KSVD2+SVM
Set1	64.9% / **65.6%**	**65.3%** / 65.1%
Set2	**66.1%** / 65.9%	**65.9%** / 65.5%
Set3	**67.3%** / 67.0%	**67.7%** / 67.1%
Overall	66.1% / **66.2%**	**66.3%** / 65.9%

Table 1. Classification accuracy of three algorithms on three train/test partitions. Note that the LC-KSVD classification results are compared to a SVM classifier.

Overall accuracies of both LC-KSVD1 (66.1%) and LC-KSVD2 (66.3%) outperform the baseline accuracy 43.9% obtained using standard bag of words approach. To the best our knowledge, no action recognition paper has been published using UCF101 yet, thus direct comparison with any state-of-the-art results is not available. Instead, we compare our results with methods evaluated on UCF50 to get an abstract analysis by indirect comparison. State-of-the-art performances on UCF50 are distributed in 72.6%~81.03% [17], [18], [19], [20], which are higher than LC-KSVD. We find the reason of this in the scale of the datasets. UCF101 contains 51 classes more than UCF50 which includes 50 classes with a total of 6676 clips. It is not a surprising result since more action types lead to confusion and performance degradation. Reddy and Shah [17] studied the effect of increasing the number of action classes on the UCF YouTube by adding new actions from UCF50 one at a time. The performances on the initial 11 actions dropped 13.18% in average, once 39 new actions from UCF50 added. Moreover, standard train/test partitions of UCF101 for direct comparison across different algorithms are for 3-fold group-wise cross validation. It is already known that video wise cross-validation would yield higher performance than group wise validation. This is because a set of clips belonging to a group is just a sequential division of a long video, thus separating clips from a same group into train and test sets as done in video wise cross-validation would give higher performance. For example, Sadanand and Corso [19] obtained 76.4% accuracy on UCF50 under video wise cross–validation, but observed 57.9% accuracy when group wise validation is performed. Table 2 and Table 3 show the per-class accuracy and per-type accuracy of LC-KSVD2 on Set 3, respectively.

Billiards (5)	100.0	PlayingDhol (4)	80.5	TennisSwing (5)	57.8
BodyWeightSquats (2)	100.0	Biking (5)	80.0	CricketBowling (5)	57.1
Bowling (5)	100.0	FloorGymnastics (5)	78.4	HulaHoop (1)	57.1
Punch (5)	100.0	PoleVault (5)	77.8	LongJump (5)	55.9
HammerThrow (5)	97.2	HandstandPushups (2)	76.2	Rafting (5)	55.9
HorseRace (5)	97.2	ParallelBars (5)	75.0	CricketShot (5)	55.8
TrampolineJumping (2)	97.1	SoccerPenalty (5)	74.3	BreastStroke (5)	55.2
PlayingTabla (4)	96.8	CliffDiving (5)	73.7	Typing (1)	54.8
BenchPress (5)	95.2	BlowDryHair (1)	73.5	Drumming (4)	54.2
BasketballDunk (5)	95.0	Surfing (5)	71.8	HeadMassage (3)	53.7
JumpingJack (2)	93.6	Fencing (5)	71.4	PlayingViolin (4)	53.6
IceDancing (5)	92.9	Skiing (5)	71.1	BrushingTeeth (1)	52.9
Skijet (5)	92.9	Rowing (5)	70.3	BaseballPitch (5)	52.5
VolleyballSpiking (5)	92.9	TableTennisShot (5)	67.5	BlowingCandles (2)	50.0
SkyDiving (5)	91.2	FrontCrawl (5)	66.7	TaiChi (2)	50.0
BoxingPunchingBag (5)	91.1	SumoWrestling (5)	66.7	WalkingWithDog (2)	50.0
HorseRiding (5)	90.9	YoYo (1)	66.7	ApplyLipstick (1)	48.5
PlayingPiano (4)	90.9	UnevenBars (5)	65.6	CuttingInKitchen (1)	46.4
SalsaSpin (3)	90.9	BandMarching (3)	65.1	RopeClimbing (2)	42.9
WritingOnBoard (1)	90.5	FrisbeeCatch (5)	64.9	ShavingBeard (1)	41.7
PlayingSitar (4)	89.1	RockClimbingIndoor (2)	64.9	PizzaTossing (1)	41.4
JumpRope (1)	88.4	BalanceBeam (5)	64.5	Archery (5)	39.5
Knitting (1)	88.2	Mixing (1)	64.5	Hammering (1)	35.7
Diving (5)	86.8	JugglingBalls (1)	64.3	BabyCrawling (2)	33.3
BoxingSpeedBag (5)	86.5	GolfSwing (5)	62.5	Basketball (5)	31.0
Swing (2)	86.1	MilitaryParade (3)	62.2	Nunchucks (1)	29.7
PommelHorse (5)	85.7	ThrowDiscus (5)	61.1	MoppingFloor (1)	28.1
PullUps (2)	85.7	Kayaking (5)	60.9	Lunges (2)	27.3
StillRings (5)	83.9	PushUps (2)	60.7	JavelinThrow (5)	24.2
CleanAndJerk (5)	83.3	FieldHockeyPenalty (5)	60.0	HandstandWalking (2)	22.6
PlayingCello (4)	83.0	PlayingDaf (4)	60.0	HighJump (5)	22.6
SkateBoarding (1)	82.4	PlayingFlute (4)	59.5	Haircut (3)	11.4
PlayingGuitar (4)	81.3	ApplyEyeMakeup (1)	59.5	Shotput (5)	11.4
WallPushups (2)	81.0	SoccerJuggling (1)	58.1		

Table 2. Per-class classification accuracy of LC-KSVD sorted in descending order (class type: 1=Human-Object Interaction, 2=Body-Motion Only, 3=Human-Human interaction, 4=Playing Musical Instruments, 5=Sports.)

Action class type	Average accuracy
Human-Object Interaction	58.6%
Body-Motion Only	63.8%
Human-Human Interaction	56.7%
Playing Musical Instruments	74.9%
Sports	71.3%

Table 3. Classification accuracy of L on three train/test partitions.

Table 1 also shows the comparison of classification results between LC-KSVD and a SVM classifier. We learn a SVM classifier for each set of LC-KSVD1 and LC-KSVD2 using liblinear [21]. In most case, LC-KSVD performs slightly better than the SVM classifier. Note that the LC-KSVD2 classifier is learned jointly during the dictionary learning process, thus no need to have additional classifier learning process as some of other dictionary learning approaches do.

5. Conclusion

In this paper, we evaluated LC-KSVD, a discriminative dictionary learning algorithm for sparse representation of

input signals, on UCF101 which is currently the largest and challenging action dataset. LC-KSVD incorporates a new label consistency constraint in addition to the reconstruction error and classification error in the objective function, so as to enforce discriminability during the dictionary learning process. We obtained 66.1% overall classification accuracy using LC-KSVD1 and 66.3% using LC-KSVD2 on three standard UCF101 train/test partitions, both of which outperform the baseline result (43.9%) obtained using a naïve Bag-of-Words approach. Furthermore, we analyzed the LC-KSVD classifier, learned jointly in the dictionary learning process, by comparison with a SVM classifier, and observe marginally better performance than the SVM classifier.

References

[1] R. Poppe. A survey on vision-based human action recognition. Image and Vision Computing, 28(6):976-990, 2010.

[2] D. Weinland, R. Ronfard, and E. Boyer. A survey of vision-based methods for action representation, segmentation and recognition. Computer Vision Image Understanding, 115(2):224-241, 2010.

[3] Z. Jiang, Z. Lin, and L. Davis. Label Consistent K-SVD: Learning a discriminative dictionary for recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(11):2651-2664, November 2013.

[4] M. Aharon, M. Elad, and A. Bruckstein. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. IEEE Transactions on Signal Processing, 54(11):4311-4322, November 2006.

[5] J. M. Chaquet, E. J. Carmona, and A. Fernandez-Caballero. A survey of video datasets for human action and activity recognition. Computer Vision Image Understanding, 117:633-659, 2013.

[6] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: A large video database for human motion recognition. In ICCV, 2011.

[7] J. Liu, J. Luo, and M. Shah. Recognizing realistic actions from videos in the wild. In CVPR, 2009.

[8] K. Soomro, A. R. Zamir, and M. Shah. UCF101:A dataset of 101 human action classes from videos in the wild. CRCV-TR-12-01, November, 2012.

[9] J.C. Yang, K. Yu, and T. Huang. Supervised Translation-Invariant Sparse coding. In CVPR, 2010.

[10] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Supervised Dictionary Learning. In NIPS, 2009.

[11] G. H. Golub, P. C. Hansen and D. P. O'Leary "Tikhonov regularization and total least squares", SIAM J. Matrix Anal. App., 21(1):185 -194, 1999.

[12] P. Scovanner, S. Ali, and M. Shah. A 3-dimensional SIFT descriptor and its application to action recognition. In ACM Multimedia, 2007.

[13] H. Wang, M. Ullah, A. Klaser, I. Laptev, and C. Schmid. Evaluation of local spatio-temporal features for action recognition. In BMVC, 2009.

[14] H. Wang, A. Klaser, C. Schmid, and C.-L. Liu. Action Recognition by Dense Trajectories. In CVPR, 2011.

[15] H. Lee, A. Battle, R. Raina, and A. Y. Ng. Efficient sparse coding algorithms. In NIPS, 2006.

[16] K. Gregor and Y. LeCun. Learning fast approximations of sparse coding. In ICML, 2010.

[17] K. K. Reddy and M. Shah. Recognizing 50 human action categories of web videos. Machine Vision and Applications Journal, 24:971-981, September, 2012. 1.

[18] O. Kliper-Gross, Y. Gurovich, T. Hassner, and L. Wolf. Motion interchange patterns for action recognition in unconstrained videos. In ECCV, 2012.

[19] S. Sadanand and J. Corso. Action bank: A high-level representation of activity in video. In CVPR, 2012.

[20] S. Todorovic. Human activities as Stochastic Kronecker Graphs. In ECCV, 2012.

[21] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A library for large linear classification. JMLR, June 2008.

[22] Y.-G. Jiang, J. Liu, A. Roshan Zamir, I. Laptev, M. Piccardi, M. Shah, and R. Sukthankar, THUMOS Challenge: Action Recognition with a Large Number of Classes, 2013, http://crcv.ucf.edu/ICCV13-Action-Workshop

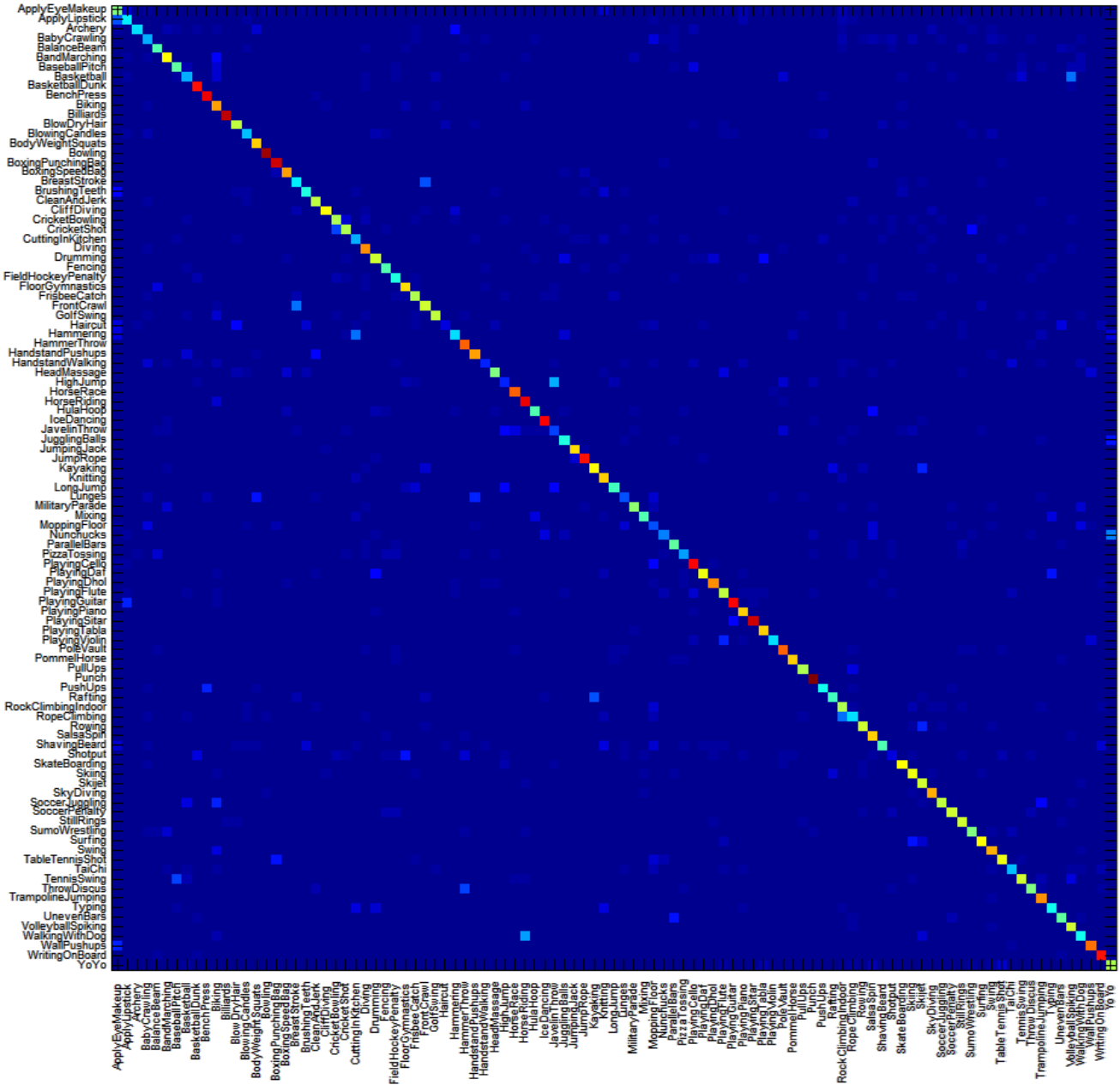


Figure 5. Confusion Matrix for the UCF101 dataset