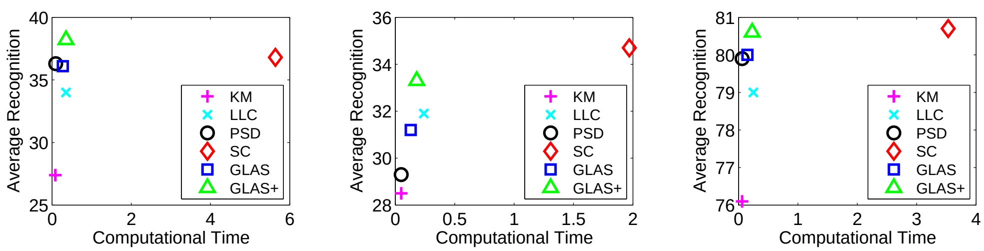
SIFT (128 Dim.) [15]

Methods	KM	LLC [22]	PSD [11]	SC [24]	**GLAS**	**GLAS+**
15 Train	22.7±0.4	28.1±0.5	30.4±0.6	30.7±0.4	30.4±0.4	32.1±0.4
30 Train	27.4±0.5	34.0±0.6	36.3±0.5	36.8±0.4	36.1±0.4	38.2±0.4

Local Self-Similarity (30 Dim.) [20]

Methods	KM	LLC [22]	PSD [11]	SC [24]	**GLAS**	**GLAS+**
15 Train	23.7±0.4	26.3±0.5	24.3±0.6	28.7±0.5	26.0±0.5	27.6±0.6
30 Train	28.5±0.4	31.9±0.5	29.3±0.5	34.7±0.4	31.2±0.5	33.3±0.5

Table 2: Recognition accuracy on Caltech-256. The dictionary sizes are all set to 2048 for SIFT and 1024 for Local Self-Similarity.



(a) (b) (c)

Figure 3: Plotting computational time vs. average recognition. (a) and (b) are SIFT and Local-Self Similarity respectively evaluated on Caltech-256 with 30 training images. The dictionary size is set to 2048. (c) is SIFT evaluated on 15 Scenes. The dictionary size is set to 1024.

4.3 15 Scenes

The 15 Scenes [13] dataset contains 4485 images divided into 15 scene classes ranging from indoor scenes to outdoor scenes. 100 training images per class are used for training and the rest for testing. We used SIFT to learn 1024 dictionary bases for each method. The results are plotted with computational time taken in Figure 3 (c). The result of GLAS+ (80.6%) are very similar to that of SC (80.7%), yet the former is significantly faster. In summary, we show that our approach works well on three different challenging datasets.

5 Conclusion

This paper has presented an approximation of ℓ_1 sparse coding based on the generalized lasso called GLAS. This is further extended with the post-refinement procedure to handle mutual inhibition between bases which are essential in an overcomplete setting. The experiments have shown competitive performance of GLAS against SC and achieved significant computational speed up. We have also demonstrated that the effectiveness of GLAS on two local descriptor types, namely SIFT and Local Self-Similarity where LLC and PSD only perform well on one type. GLAS is not restricted to only approximate ℓ_1 sparse coding, but should be applicable to other variations of sparse coding in general. For example, it may be interesting to try GLAS on Laplacian sparse coding [6] that achieves smoother sparse codes than ℓ_1 sparse coding.

Acknowledgment

NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program.