Sign Language Recognition Using Dictionary Learning

Nikhil Madaan
M.S. Electrical and Computer Engineering
nmadaan@andrew.cmu.edu

Rohan Panda
M.S. Electrical and Computer Engineering
rohanpan@andrew.cmu.edu

Sarthak Tandon
M.S. Computational Data Science
sarthakt@andrew.cmu.edu

Ning Cao
Electrical and Computer Engineering
ningc@andrew.cmu.edu

1 Introduction

Sign language (SL) uses visual ways such as gestures or hand movements to convey information. In the past, people with hearing or speaking disabilities always needed someone's help to interact with the world. In the current era of AI, several sign-language recognition (SLR) systems have been built to assist such people. Through this project, we seek to develop supervised dictionary learning-based SLR systems for robust sign language character recognition. We plan to investigate the performance of some of the existing supervised dictionary learning approaches in the context of SLR and further focus on the explainability of the learned representations. Finally we experiment with ensemble techniques to improve the performance of our models, and plan to extend our work to real-time SLR.

2 Related Work

We aim our focus on 3 subtopics, and found papers respectively:

Procedures of SLR: Elakkiya [3] summarized the challenges we face: a) feature extraction, b) feature classification, and c) modeling and recognition. **Improvements in dictionary learning algorithms**: Tuysuzoglu et al. [11] applied ensemble learning methods on Random Subspace Dictionary Learning (RDL) and Bagging Dictionary Learning (BDL). Smith and Elad [9] focused on improving MOD and K-SVD. They improved the dictionary update stage to find both the dictionary and the representations, and leverage known representations from the previous sparse-coding. Mensch et al. [7] proposed a more efficient factorization method when handling massive datasets. **Application of dictionary learning on SLR**: Yin et al. [12] proposed a dictionary based framework for SLR, in which they train dictionaries by considering the semantic constraints. Jiang et al. [4] applied label consistent K-SVD (LC-KSVD) to obtain dictionaries, and developed a new classification approach.

3 Datasets

Primarily, we will be using the ASL Alphabet dataset[1] which contains 87,000 200×200 training images over 29 output classes (3,000 images for each). The output labels cover the 26 alphabets of the English language (A-Z) along with SPACE, DELETE and NOTHING. The test set, however, contains a mere 29 images (1 image for each class). We will also be using the ASL MNIST dataset[2] which has 27,455 training images and 7,172 testing images of size 28×28 covering 24 classes (A-Z

[1] https://www.kaggle.com/grassknoted/asl-alphabet
[2] https://www.kaggle.com/datamunge/sign-language-mnist

Submitted to 34th Conference on Neural Information Processing Systems (NeurIPS 2020). Do not distribute.

(a) original image (b) SIFT visualization (c) HoG visualization

Figure 1: SIFT results

except J and Z). Currently we are using ASL MNIST for the code debugging and baselines, while our goal is to run it on ASL alphabet.

4 Methodology

4.1 Feature Extraction

4.1.1 Scale-invariant feature transform (SIFT)

Scale-invariant feature transform is an algorithm that extracts local features (keypoints) regardless of size and orientation[6]. We attempted to use SIFT in OpenCV to find the keypoints and their corresponding descriptors. Using cv2.SIFT().detectAndCompute() we can compute the feature keypoints with various attributes. Our experimentation on SIFT showed promising results (Fig. 1b) with sample images from the ASL Alphabet dataset. However, after we switched to ASL MNIST dataset, we encountered trouble when transforming SIFT keypoints to features for our learning algorithms. Test accuracy of baseline models on SIFT-related features are not competitive even with RAW features. We suspect that the smaller image size (28×28 in ASL MNIST; in comparison, 200×200 in ASL Alphabet) is the major reason of the unsatisfactory result.

4.1.2 Histogram of Oriented Gradients (HoG)

Histogram of gradients is a technique of generating features for object detection based on the assumption that an object's appearance and shape inside an image can be described well by the visualizing the distribution of the intensities of the pixel gradients. We use scikit image library (skimage.feature.hog()) to transform our input data into a histogram of gradients (1c.) To accommodate for smaller images in ASL MNIST, we only allowed 1 cell in each block and 2×2 pixels in each cell. It is noticed that HoG can generate features that performs better in baseline models, and we proceed with HoG as our feature extraction method. We suspect that the dominant reason HoG generates better results than SIFT is that HoG manages to depict the entire picture while SIFT is more focused on local features. Therefore, the performances of the two algorithms on datasets with different image sizes may vary.

4.2 Supervised Dictionary Learning

4.2.1 Entropy Dictionary Learning (EDL)

Let the given data be denoted by $X = [X1, X2, X3, ...]$ where $X_c = [x_1^c, x_2^c, .., x_{N_c}^c]$ (N = $\sum_c N_c$) denotes the data points of class c. Correspondingly we define, A and D to be K*N and D*K matrices respectively, each column of D represents an atom and each element of A (denoted by alpha) describes the contribution of atom k for the data point n. According to EDL[1], the dictionary atoms used in supervised DL can be seen as "sharing" atoms - atoms that help in reconstructing the data points

from multiple images- and "discriminative" - atoms that help in discrimination/classification for its support classes. What EDL tries to achieve is to increase the discriminability power of atoms so that a particular atom has one or the least number of supported classes. This is achieved by reducing the entropy of the distribution of a particular atom's contribution to data points of different classes. The entropy term can be described as:

$$H(d_{.i}) = -\sum_{c=1}^{L} p_{ic}.Log(p_{ic}) \tag{1}$$

$$p_{ic} = \frac{\|\alpha_{i.}^{c}\|_1}{\|\alpha_{i.}\|_1} \tag{2}$$

Now as we want to minimize this term, it can be added to the standard DL objective function to give us:

$$(\widehat{A}, \widehat{D}) = \underset{A,D}{argmin} \|DA - X\|_F^2 + \lambda_1 \sum_{j=1}^{N} \|\alpha_{.j}\| + \lambda_2 \sum_{i=1}^{k} H(d_{.i}) \tag{3}$$

Hence, as the value H(d_i) decreases the discrimination power of that atom increases. The training is performed using orthogonal matching pursuit(OMP[10] Once the training in complete the inference can be done using the membership function of the training data (denoted by pic) and the sparse code generated for the test signal ($\overline{A}$) by:

$$\widehat{c}(x_{.j}) = \underset{c}{argmax}\ p_{.c}^{T} \mid \overline{\alpha_{.j}} \mid \tag{4}$$

4.2.2 Support Vector Guided Dictionary Learning (SVGDL)

A discriminative dictionary learning (DDL) method formulates the model as sum of reconstruction error ($R(X, D, Z)$), p-norm regularizer ($\|Z\|_p^p$) and a discrimination term ($L(Z)$). It tries to maximize the similarity of pairs of coding vectors from the same class and the dissimilarity of pairs of coding vectors from different classes.

$$< D, Z >= \underset{D,Z}{argmin} R(X, D, Z) + \lambda_1 \|Z\|_p^p + \lambda_2 L(Z, w_{ij}) \tag{5}$$

$$L(Z, w_{ij}) = \sum_{i,j} ||z_i - z_j||_2^2 w_{i,j} \tag{6}$$

SVGDL [2] leverages the fact that some pairs of coding vectors might play more important roles than other pairs in learning a discriminative dictionary. So, rather than uniformly assigning weights $w_{i,j}$ for each pair, $w_{i,j}$ can be parameterized as a function with variable β. It formulates the discrimination term in the objective function as SVM dual formulation problem and adopts an adaptive weight assignment rather than assigning non-zero weights for all pairwise coding vectors. Since, support vectors are data points that are closer to the hyperplanes separating different classes, influencing the position and orientation of the hyperplane, hence, SVGDL assigns non-zero weights in the discrimination term to only support vectors.

$$< D, Z >= \underset{D,Z}{argmin} \|X - DZ\|_F^2 + \lambda_1 \|Z\|_p^p + \lambda_2 \max_{\beta}(4 \sum_{i=1}^{n} \beta_i - 2 \sum_{i,j} y_i y_j \beta_i \beta_j z_i^T z_j) \tag{7}$$

SVGDL replaces the subproblem of β with its primal SVM form, leading to the final support vector guided dictionary learning (SVGDL) model.

$$< D, Z, u, b >= \underset{D,Z,u,b}{argmin} \|X - DZ\|_F^2 + \lambda_1 \|Z\|_p^p + 2\lambda_2 \sum_{c=1}^{C} L(Z, y^c, u_c, b_c) \tag{8}$$

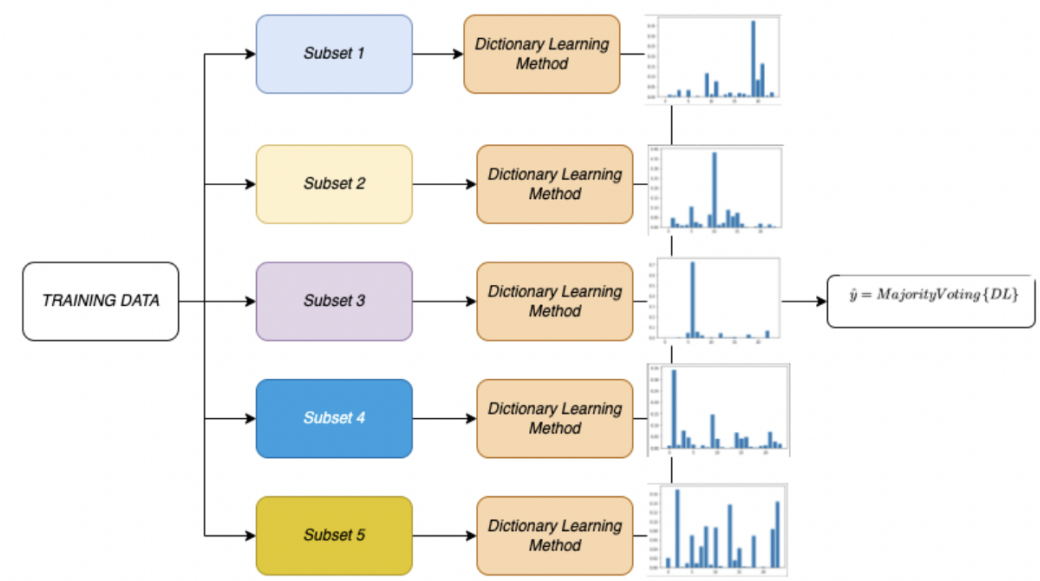


Figure 2: Ensemble Learning Pipeline

$$L(Z, y^c, u_c, b_c) = \|u\|_2^2 + \theta \sum_{i=1}^{n} l(z_i, y_i, u, b) \tag{9}$$

where $y^c = [y_1^c, y_2^c .., y_n^c]$, $y_i^c = 1$ if $y_i = c$, otherwise $y_i^c = -1$ and $l(z_i, y_i, u, b)$ is the hinge loss function.

Once the dictionary (D) and classifiers ($< U, b >$) are learned, for inference we get the coding vector of the test input and then use the C linear classifiers to get the confidence score. The final predicted label is given by:

$$y = \underset{c \epsilon 1,2,...C}{argmax} (u_c^T z + b_c) \tag{10}$$

4.3 Ensemble Dictionary Learning

To address the overfitting issue and sensitivity of Supervised Dictionary Learning methods to hyper-parameters, we used ensemble techniques such as Bagging, Boosting[8].

4.3.1 Bagging

This involved splitting the training data into multiple subsets and training dictionaries on each subset.

4.3.2 Random Forest

This involved taking subsets of features and training dictionaries on each subset.

4.3.3 Boosting

This involved training a sequence of dictionaries, where each dictionary is trained on misclassified training instances of previous dictionaries. We had to limit the number of dictionaries in the cascade to 3 since the number of misclassified training instances beyond this were insufficient for learning a dictionary.

Using ensemble techniques in particular Bagging we were able to address the overfitting issue and dictionary learning became less sensitive to hyperparameters. At the inference time, the label is determined based on the majority vote of the ensemble of dictionaries.

5 Evaluation Method

To evaluate the performance of dictionary learning for sign-language recognition, we are comparing the accuracy anf F1 Score of dictionary learning models with traditional ML classifiers that we have used for baseline estimation.

6 Experiments

1. **Feature Engineering**

Our initial experiments were on understanding how different features play a vital role in the whole process. To explore this, we tried out using the raw image as a flattened out feature vector, using PCA (50 Principal components) on this raw feature, using descriptors outputted from SIFT, and HOG as described in Section4.1. For HOG and SIFT descriptors we also use PCA in two different ways, first we append all the descriptors end-to-end to get a very long feature vector whose dimension is reduced using PCA, and in the second way, we stack the descriptors of one image side by side and perform pca on the collection of descriptors and use the 1st principal component. The first approach worked better for both HOG and SIFT, since SIFT did not perform better than chosen baselines, they are omitted from the results table.

2. **Hyperparameter Tuning**

One of the challenges faced during training the models was selecting proper hyperparameters, we noted that both methods were very sensitive to hyperparameters and hence, selecting a proper set of values was vital. We initially tried out the values based on the original work, but those values were not suitable for the selected dataset and task. For SVGDL, we were able to develop a method to tune the hyperparameters based on the intermediate loss decrements.[Insert the exact tuning strategy]. However, this strategy did not work for EDL as it starts to diverge if the hyperparameters are not ideal, hence manual tuning based on chosen sparsity was conducted.

3. **changing SVM to other classifiers**

An interesting extension to the existing method used in SVGDL was to use a different classifier instead of SVM. The basic criterion required by the replaced classifier was that the model learns a hyperplane which can then be utilized for inference. We therefore, tried replacing SVM with logistic regression and found that we were able to attain similar results to SVM, but the performance does not increase from using the standard SVM.

4. **Mini-batch Randomized Accelerated Coordinate Descent**

EDL suffered from large training times owing to the coordinate gradient descent used. To improve the training time while making sure the process is not hurt, we tried out two different strategies. Firstly, we tried using Accelerated mini-batch randomized block coordinate descent method (MBRBCD) as described in [13]. This method helped in greatly reducing the training time while making sure that the model converges to its best representations.

5. **Learning bases using the Lagrange dual**

In SVGDL, one of the steps in the optimization of the non-convex objective involves finding Dictionary atoms that minimize $\|X - DZ\|_F^2$ s.t $\|d_i\|_2^2 \leq 1, \forall d_i$ (where d_i are dictionary atoms). We started with projected gradients to find the optimal D, i.e. found the optimal D using the closed-form solution of the minima while ignoring the constraint, followed by normalization of the atoms. However, we found that the objective function was not converging because of this step, hence, we used

Lagrange dual method[5] which gives a more efficient solution to the problem. This method helped in addressing the divergence problem we were facing while using projected gradients.

6. **Noisier Data**

Now that our model has been debugged and the initial results have been obtained, we plan to train the models on the ASL alphabet dataset, which is noiser, has illumination differences, and position shifts in the data points. This would be a tougher problem for the system to solve owing to the variability in the data. Our initial results on ASL alphabets have been around 75% for SVGDL and 76% for EDL respectively. The performance could be improved by further hyperparameter tuning and rigorous experimentation on this dataset.

7. **L1 and L2 norms**

In both SVGDL and EDL, we find a sparsity constraint which is denoted by the L1 norm of the sparse codes. This term in the objective can be replaced with a L2 norm instead to achieve similar results. We tried swapping the L1 norm with L2 norm, and noted that L1 norm gave better results than L2 norm. Furthermore, we also tried using both L1 and L2 norm on the sparse codes, to tackle overfitting, however we did not see any significant improvement in the performance or reduction in overfitting.

7 Results

Method	Feature	Accuracy	F1 Score
KNN	RAW	79.40%	77.23%
	HOG	91.00%	90.68%
SVM	RAW	78.20%	76.16%
	HOG	89.00%	88.42%
SVGDL	RAW	87.68%	86.86%
	HOG	90.97%	90.33%
	HOG (Ensemble)	93.29%	92.82%
EDL	RAW	86.73%	84.23%
	HOG	92.02%	88.60%
	HOG (Ensemble)	92.12%	85.54%

Table 1: Results of KNN, SVM (baseline) and the supervised dictionary learning methods experimented.

Table 1 presents our learnt dictionaries' classification results in terms of accuracy and F1 score. As we can see, using ensemble methods improves the accuracy scores of classifiers in the case of EDL and SVGDL. It is noted that the ensemble results correspond to results obtained when bagging is used. Our experiments show that boosting does not improve the results from simple training while bagging produces the results shown in Table1. The highest accuracy is achieved using an ensemble SVGDL followed by an ensemble EDL approach. For both raw and HOG features, EDL and SVGDL perform better compared to baseline estimates from KNNs and SVMs. Further, with raw features, dictionary Learning methods achieve a better F1 score than traditional ML methods. It is worth noting that both KNN and SVMs perform well on HOG features but are outperformed by dictionary learning methods.

8 Discussion

8.1 Classification vs Interpretability

When it comes to supervised dictionary learning algorithms, one of the main concerns for a model is to be complex enough to provide good classification performance, while giving interpretable results. In the case of supervised DL methods, the dictionary atoms act as the interpretable components of

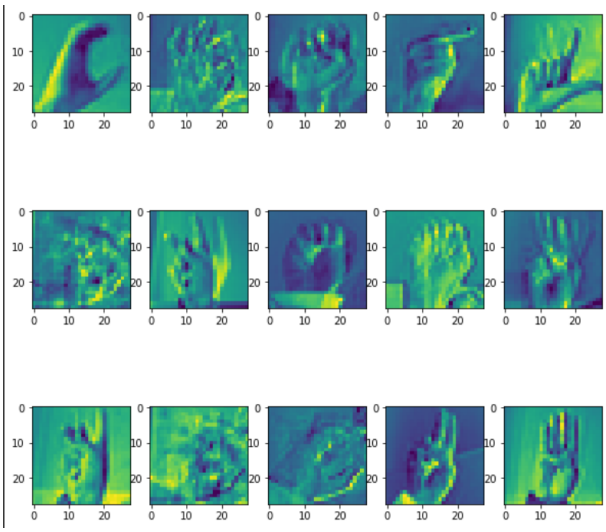
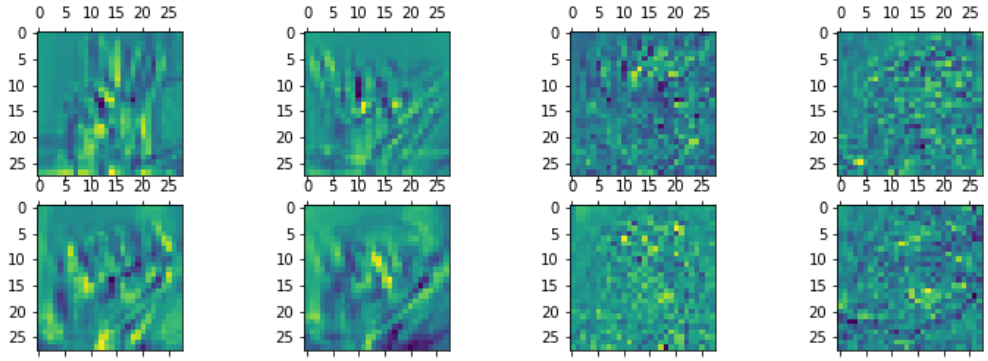


Figure 3: "Learnt Dictionary atoms using EDL"



(a) Atoms in earlier epochs

(b) Final Learnt atoms

Figure 4: SVGDDL atoms

the model,hence we try to visualize and make sense of the learnt dictionary items to explore if the model tries to retain semantics of the data during its training phase.

Firstly, we present the the dictionary atoms of the learnt dictionary using EDL in Figure3. It is worth mentioning that the dictionary was initialized with random datapoints from each class of the training data. We see that most of the atoms are recognizable and retain their original form while some of the atom undergo expected changes to accommodate the discriminability required for classification

Next, we take a look at the atoms learnt in SVGDL. Compared to EDL, SVGDL has less-interpretable atoms, which can be owing to high discrimination power that the atoms possess. It was also noted that as we increased the contribution of the discrimination error in the main objective of SVGDL, atoms kept getting less interpretable until the atoms could not be recognized by the human eye, as shown in Fig4

9 Conclusion

Based on the work presented earlier, we can conclude that Ensemble SVGDL and EDL both outperform standard machine learning methods. It was also observed that raw and HOG features are more

useful for sign language recognition than SIFT features. In addition, as is the case with most learning paradigms, in supervised dictionary learning as well, we observed a tradeoff between classification and interpretability wherein learnt atoms become less interpretable with increasing complexity.

10 Future Work

Future work in this domain should focus on increasing the robustness of this approach to different types of backgrounds and foregrounds in the images. Real-time usage analysis of such approaches is also important to cement their applicability. As ensemble techniques proved useful in combating overfitting, it is prudent to explore this in more detail with approaches combining bagging and boosting.

References

[1] Arash Abdi, Mohammad Rahmati, and Mohammad M. Ebadzadeh. Entropy based dictionary learning for image classification. *Pattern Recognition*, 110:107634, 2021.

[2] Sijia Cai, Wangmeng Zuo, Lei Zhang, Xiangchu Feng, and Ping Wang. Support vector guided dictionary learning. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 624–639, Cham, 2014. Springer International Publishing.

[3] R Elakkiya. Machine learning based sign language recognition: A review and its research frontier. *Journal of Ambient Intelligence and Humanized Computing*, 12(7):7205–7224, 2021.

[4] Zhuolin Jiang, Zhe Lin, and Larry S Davis. Label consistent k-svd: Learning a discriminative dictionary for recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(11):2651–2664, 2013.

[5] Honglak Lee, Alexis Battle, Rajat Raina, and Andrew Y Ng. Efficient sparse coding algorithms. In *Advances in neural information processing systems*, pages 801–808, 2007.

[6] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[7] Arthur Mensch, Julien Mairal, Bertrand Thirion, and Gaël Varoquaux. Dictionary learning for massive matrix factorization. In *International Conference on Machine Learning*, pages 1737–1746. PMLR, 2016.

[8] J. R. Quinlan. Bagging, boosting, and c4.s. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence - Volume 1*, AAAI'96, page 725–730. AAAI Press, 1996.

[9] Leslie N Smith and Michael Elad. Improving dictionary learning: Multiple dictionary updates and coefficient reuse. *IEEE Signal Processing Letters*, 20(1):79–82, 2012.

[10] Joel A Tropp and Anna C Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Transactions on information theory*, 53(12):4655–4666, 2007.

[11] Goksu Tuysuzoglu, Nazanin Moarref, and Yusuf Yaslan. Ensemble based classifiers using dictionary learning. In *2016 International Conference on Systems, Signals and Image Processing (IWSSIP)*, pages 1–4, 2016.

[12] Fang Yin, Xiujuan Chai, Yu Zhou, and Xilin Chen. Semantics constrained dictionary learning for signer-independent sign language recognition. In *2015 IEEE International Conference on Image Processing (ICIP)*, pages 3310–3314, 2015.

[13] Tuo Zhao, Mo Yu, Yiming Wang, Raman Arora, and Han Liu. Accelerated mini-batch randomized block coordinate descent method. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.