




Indian Sign Language Gesture Recognition

Group 11
CS365 - Project Presentation



Sanil Jain(12616)
Kadi Vinay Sameer Raja(12332)

Indian Sign Language

History

- ISL uses both hands similar to British Sign Language and is similar to International Sign Language.
- ISL alphabets derived from British Sign Language and French Sign Language alphabets.
- Unlike its american counterpart which uses one hand, uses both hands to represent alphabets.

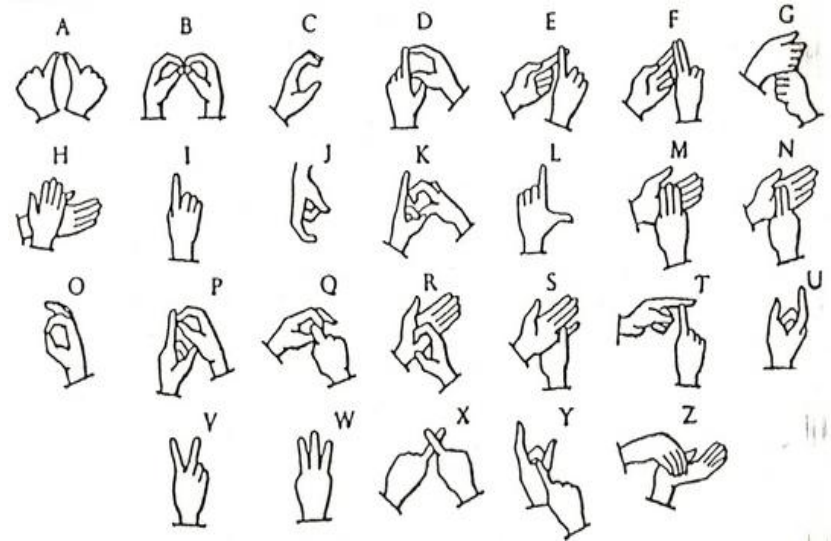
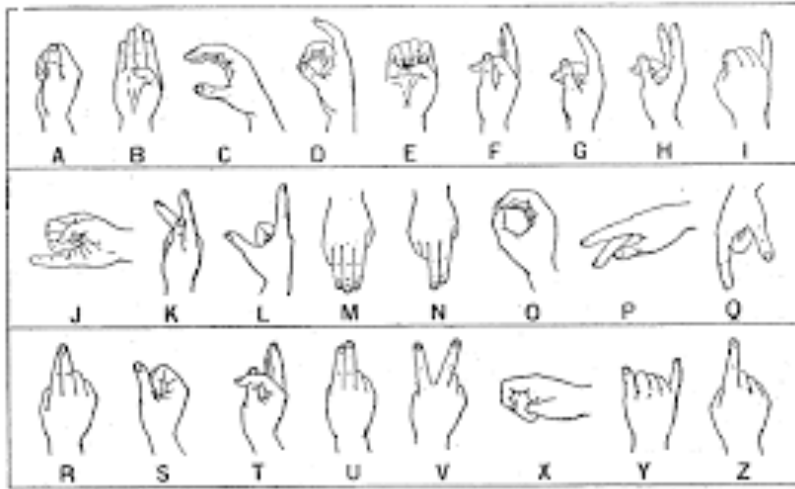


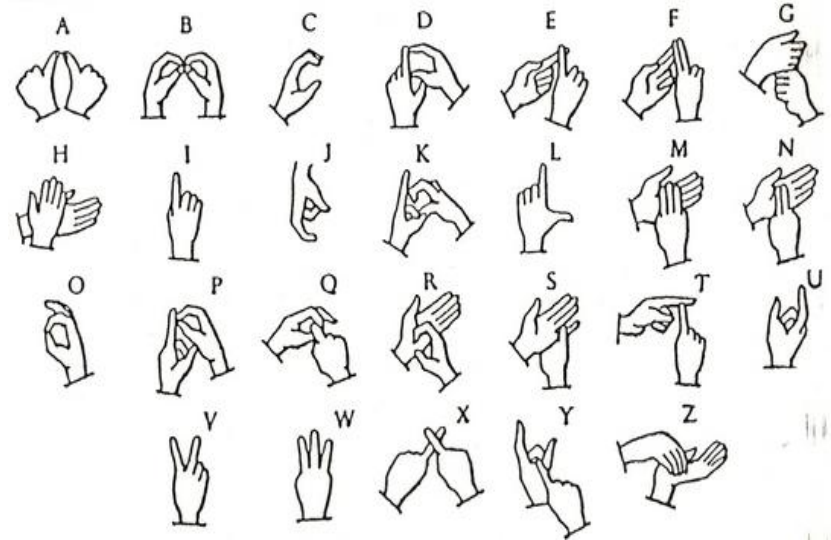
Image Src :<http://www.deaftravel.co.uk/signprint.php?id=27>

Indian Sign Language



American Sign Language

Image Src :<http://www.deaftravel.co.uk/signprint.php?id=26>



Indian Sign Language

Image Src :<http://www.deaftravel.co.uk/signprint.php?id=27>

Indian Sign Language

Previous Work

- Gesture Recognitions and Sign Language recognition has been a well researched topic for the ASL, but not so for ISL.
- Few research works have been carried out in Indian Sign Language using image processing/vision techniques.
- Most of the previous works found either analyzed what features could be better for analysis or reported results for a subset of the alphabets

Challenges

- No standard datasets for Indian Sign Language
- Using two hands leads to occlusion of features
- Variance in sign language with locality and usage of different symbols for the same alphabet by the same person.

Dataset Collection

Problems

- Lack of standard datasets for Indian Sign Language
- Videos found on internet for the same is of people describing how it looks like and not those who actually speak it
- The one or two datasets we found from previous works were created by a single member of the group doing the work..

Approach for collection of data

We went to **Jyoti Badhir Vidyalaya**, a school for deaf in a remote section of Bithoor. There for each alphabet, we recorded around 60 seconds of video for every alphabet from different students.

Whenever there were multiple conventions for the same alphabet, we asked for the most commonly used static sign for every alphabet.

Dataset Collection



A recollection of our time at the school
P.S. also a proof that we actually went there

Learning


A decorative network diagram in the top right corner, consisting of various sized circles (nodes) connected by thin lines (edges). Some nodes are solid grey, while others are hollow with a dashed border. The connections form a complex, branching structure.

Frame Extraction

Skin Segmentation

Feature Extraction

Training and Testing

A decorative network diagram in the bottom left corner, similar to the one in the top right. It features a cluster of nodes connected by lines, with some nodes highlighted in solid grey and others as dashed outlines.

Skin Segmentation

Initial Approaches

- Training on skin segmentation dataset
Tried machine learning models like SVM, random forests on the skin segmentation dataset from <https://archive.ics.uci.edu/ml/datasets/Skin+Segmentation>
Very bad dataset, after training on around 2,00,000 points, skin segmentation of hand images gave back almost black image(i.e. almost no skin detection)
- HSV model and constraints on values of H and S
Convert Image from RGB to HSV model and retain pixels satisfying $25 < H < 230$ and $25 < S < 230$
This implementation wasn't much effective and the authors in the report had used it along with motion segmentation which made their segmentation slightly better.

Skin Segmentation

Final Approach

In this approach, we transform the image from RGB space to YIQ and YUV space. From U and V, we get $\theta = \tan^{-1}(V/U)$. In the original approach, the author classified skin pixels as those with $30 < I < 100$ and $105^\circ < \theta < 150^\circ$.

Since those parameters weren't working that good for us, we somewhat tweaked the parameters and it performed much better than the previous two approaches.

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.275 & -0.321 \\ 0.212 & -0.528 & 0.311 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

$$\begin{bmatrix} Y' \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.14713 & -0.28886 & 0.436 \\ 0.615 & -0.51499 & -0.10001 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Skin Segmentation

Final Approach

In this approach, we transform the image from RGB space to YIQ and YUV space. From U and V, we get $\theta = \tan^{-1}(V/U)$. In the original approach, the author classified skin pixels as those with $30 < I < 100$ and $105^\circ < \theta < 150^\circ$.

Since those parameters weren't working that good for us, we somewhat tweaked the parameters and it performed much better than the previous two approaches.



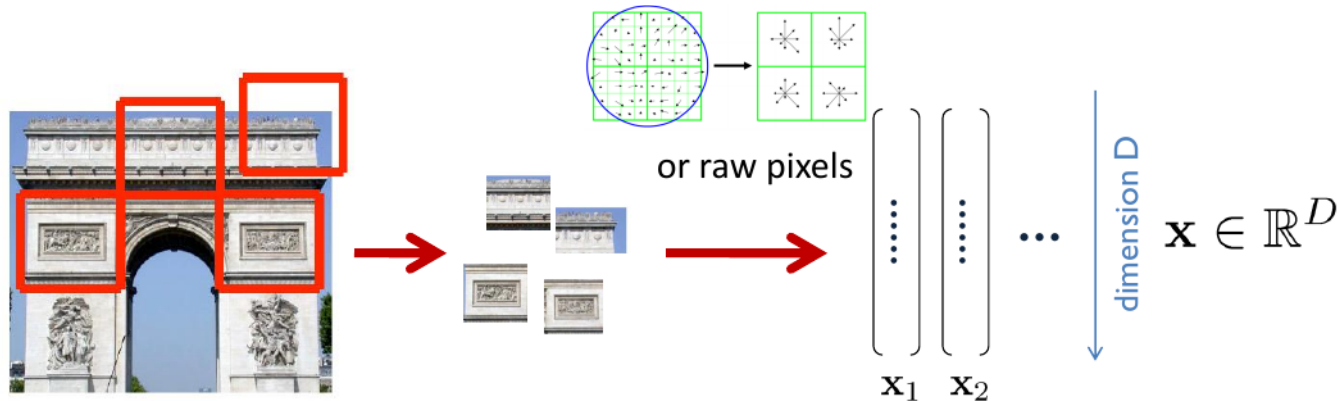
Bag of Visual Words

Each image abstracted by several local patches.

These patches described by numerical vectors called feature descriptors.

One of the most commonly used feature detector and descriptor is SIFT (Scale Inverse Feature Transformation) which gives a 128 dimensional vector for every patch.

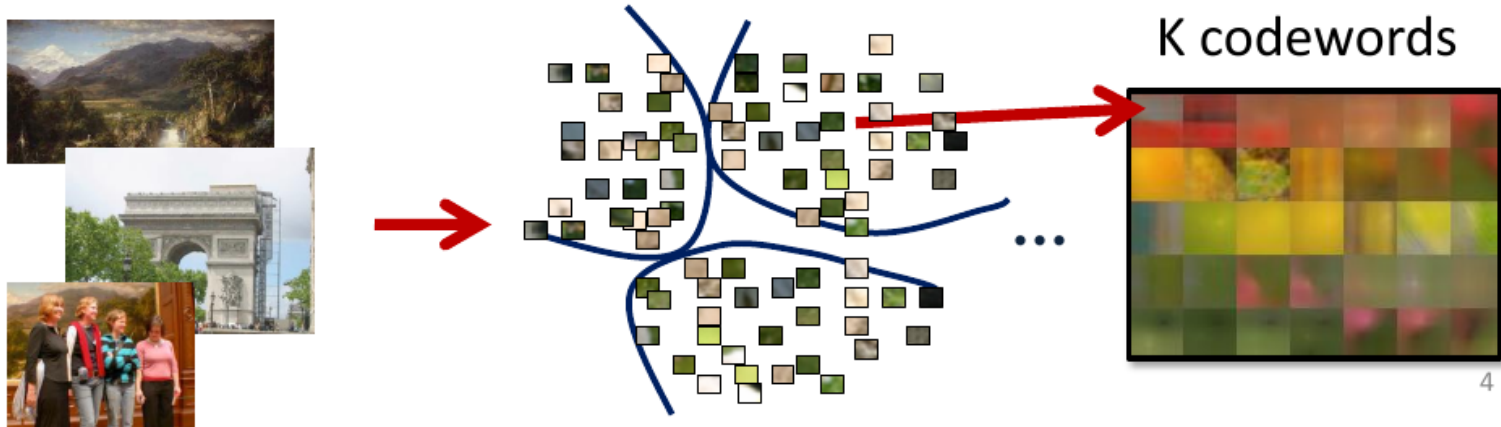
The number of patches can be different for different images.



Bag of Visual Words

Now we convert these vector represented patches to codewords which produces a codebook (analogous to dictionary of words in text).

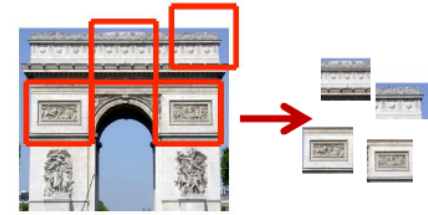
The approach we use now is Kmeans clustering over all the obtained vectors and get K codewords (clusters). Each patch (vector) in image will be mapped to the nearest cluster. Thus similar patches are represented as the same codeword.



Bag of Visual Words

So now for every image, the extracted patch vectors are mapped to the nearest codeword, and the whole image is now represented as a histogram of the codewords.

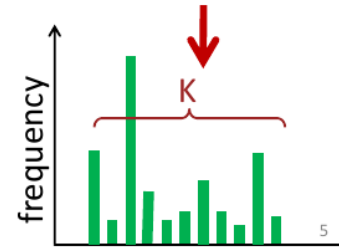
In this histogram, the bins are the codewords and each bin counts the number of words assigned to the codeword.



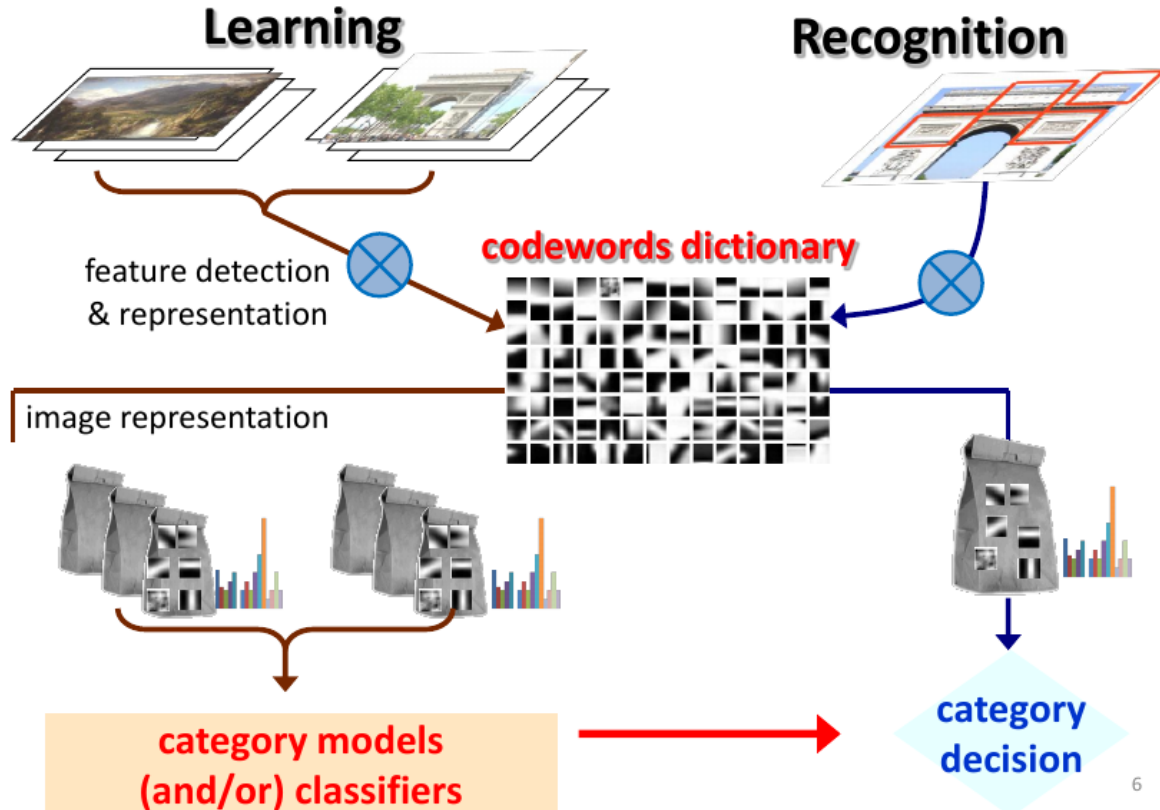
Nearest
Neighbour
Matching



**“bag of
words”**



Bag of Visual Words



Results Obtained for Bag of Visual Words

We took 25 images per alphabet from 3 person each for training and 25 images per alphabet from another person for testing.

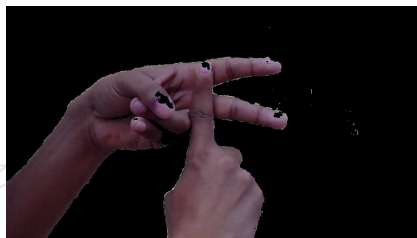
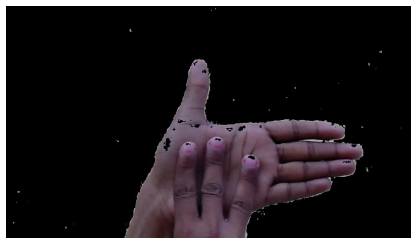
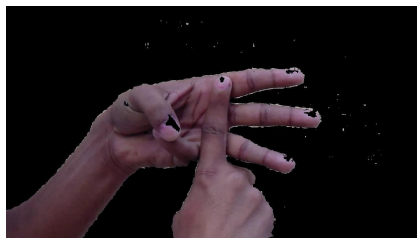
So training over 1950 images, we tested for 650 images and obtained the following results :-

Train Set Size	Test Set Size	Correctly Classified	Accuracy
1950	650	220	33.84%

Results Obtained for Bag of Visual Words

Observations

- Similar looking alphabets misclassified amongst each other
- One of the persons among the 3 persons was left handed and gave laterally inverted images for many alphabets.



Future Work

Obtain HOG(Histogram of Oriented Gradient) features from scaled down images and use Gaussian random projection on them to get feature vectors in a lower dimensional space. Then use the feature vectors for learning and classification.

Apply the models in a hierarchical manner e.g :- classify them as one and two handed alphabets and then do further classification.

References

1. <http://mi.eng.cam.ac.uk/~cipolla/lectures/PartI/old/IB-visualcodebook.pdf>
2. <https://github.com/shackenberg/Minimal-Bag-of-Visual-Words-Image-Classifer/blob/master/sift.py>
3. <http://en.wikipedia.org/wiki/YIQ>
4. <http://en.wikipedia.org/wiki/YUV>
5. <http://cs229.stanford.edu/proj2011/ChenSenguptaSundaram-SignLanguageGestureRecognitionWithUnsupervisedFeatureLearning.pdf>
6. http://en.wikipedia.org/wiki/Bag-of-words_model_in_computer_vision
7. Neha V. Tavari, P. A. V. D., Indian sign language recognition based on histograms of oriented gradient, *International Journal of Computer Science and Information Technologies* 5, 3 (2014), 3657-3660



