ASL Fingerspelling Interpretation

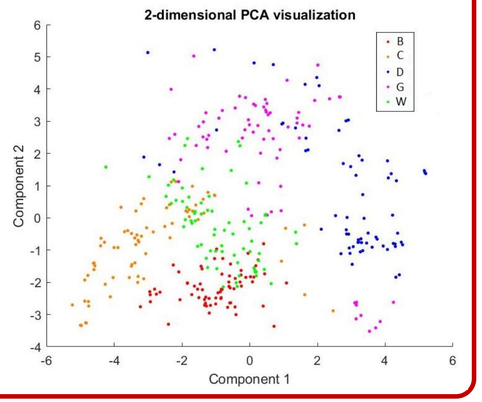
Hans Magnus Ewald, Ishan Patil, Shalini Ranmuthu
Department of Electrical Engineering, Stanford University

Motivation and Goals

- Develop modern and mobile method of sign language interpretation
- Concepts can be expanded to real-time gesture recognition
- Goal: Real-time implementation on Android app

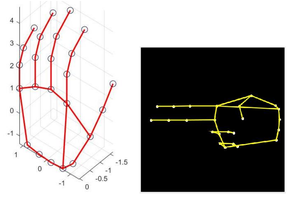
Visualization Using PCA

- Visualize first 2 principal components of 1074-dim data
- Early insight: PCA may help reduce overfit from Gabor filters



Future Work - Hand Model

- Feature extraction that fits a 20-DoF hand model to image
- Optimization problem, solution with learning method
- Symbol classification in the model parameter space

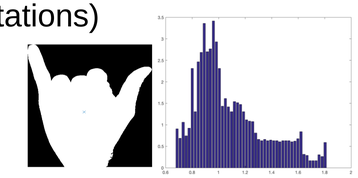


Status:

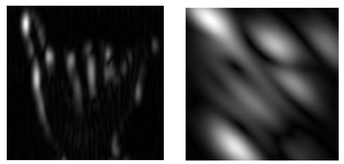
- Model and projection defined
- Gradient Descent considered for hand model fit

Feature Extraction

- **Histogram of centroid distances (HOCD):** distance from centroid to each edge pixel in the hand mask
- **Gabor filters:** flexible gradient operators (reflect different scales and orientations)



HOCD from mask



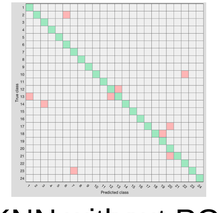
Gabor Filter results

Method(s)	Generalization Error*	Accuracy on Real Images
HOCD	0.1940	40%
Gabor	0.0090	50%
HOCD + Gabor	0.0107	**80%**

*Using Fine KNN Classifier and 5-fold cross validation.

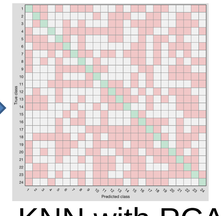
Classification and Error Analysis

- Methods: Fine KNN, SVM with RBF Kernel, Linear Discriminant Analysis, Random Decision Forest
- Performance also analyzed using PCA for dimensionality reduction (from 1074 to 34) to retain 95% variance.
- Precision and Recall values are averaged over each class.



KNN without PCA

Reason for dropping PCA in implementation



KNN with PCA

Classifier*	Precision	Recall	Accuracy on Real Images
RandForest	0.9844	0.9840	60%
SVM	0.9931	0.9928	50%
LDA	0.9503	0.9498	40%
KNN	**1**	**1**	**80%**

*Using combination of HOCD and Gabor Filter features.

Prediction Pipeline and Implementation

Capture image on Android app & send to server

Android app

Extract Gabor features

Extract HOCD features

Classify using pre-trained model

Server code

Display result on Android App

Android app