Utilizing Image Clustering for Sign Language Detection Research

Written By
Orya Shpigel, Asahel Cohen, Roni Harel, Matan-Ben Nagar

Guidance by
Professor Gil-Ben Artzi from Ariel University

Abstract

Communication is very crucial to human beings, as it enables us to express ourselves. We communicate through speech, gestures, body language, reading, writing, or through visual aids. speech is one of the most commonly used among them. Unfortunately, for the speaking and hearing impaired minority, there is a communication gap. Visual aids, or an interpreter, are used for communicating with them. However, these methods are rather cumbersome and expensive, and can't be used in an emergency. Sign Language chiefly uses manual communication to convey meaning. This involves simultaneously combining hand shapes, orientations, and movement of the hands, arms, or body to express the speaker's thoughts.

Sign Language consists of fingerspelling, which spells out words character by character, and word level association which involves hand gestures that convey the word's meaning. Fingerspelling is a vital tool in sign language, as it enables the communication of names, addresses, and other words that do not carry meaning in the word-level association. In spite of this, fingerspelling is not widely used as it is challenging to understand and difficult to use. Moreover, there is no universal sign language, and very few people know it, making it an inadequate communication alternative.

Introduction

Sign Language (SL) is the primary language for the speaking and hearing impaired. Each country has its own SL that is different from other countries. Each sign in a language is represented with variant hand gestures, body movements, and facial expressions.

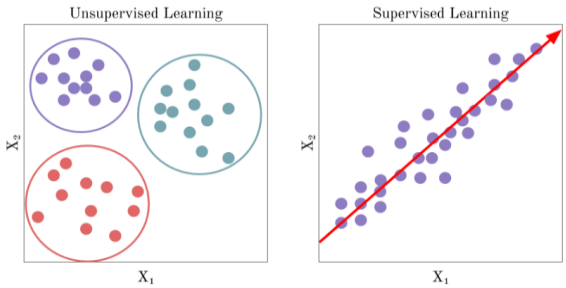
The Hebrew Sign Language is the communication method for Hebrew hearing-impaired people. Our goal is to improve the dynamic Hebrew Sign Language translation.

A system for sign language recognition that translates fingerspelling can solve this problem. Various machine learning algorithms are used and their accuracies are recorded and compared in this report.

Unsupervised Learning

We want to present a method to identify sign language using features learned by unsupervised techniques. *(Wilson, 2020)*

Unsupervised feature learning is how features are learned without any labeled data. In this user-independent model, classification machine learning algorithms are trained using a set of image data, and testing is done on a completely different set of data.



Unsupervised learning will often find subgroups or hidden patterns within the dataset that a human observer may not pick up on. This is shown in the figure above. With the given image, you can probably pick out the subgroups, but with a more complex dataset, these subgroups may not be so easy to find. This is where unsupervised learning can help us.

Algorithms used in this paper

The point of this research is to validate the most efficient algorithm to provide the highest-end image clustering technique. We have come across several relevant algorithms that we will explore. With each algorithm, we will use a few feature extraction techniques, in order to test the efficiency of the algorithms properly. The algorithms we used for clustering are K-Means and Gaussian Mixture.

Clustering

is the simplest and among the most common applications of unsupervised learning. Clustering aims to discover “clusters”, or subgroups within unlabeled data. Clusters will contain data points that are as similar as possible to each other, and as dissimilar as possible to data points in other clusters. Clustering helps find underlying patterns within the data that may not be noticeable to a human observer.

Types of Clustering

Broadly speaking, clustering can be divided into two subgroups:

-Hard Clustering: In hard clustering, each data point either belongs to a cluster completely or not. K-Means is a hard-clustering algorithm.

-Soft Clustering: In soft clustering, instead of putting each data point into a separate cluster, a probability or likelihood of that data point to be in those clusters is assigned. Gaussian Mixture is a soft clustering algorithm.

How do machines understand images?

Loading the image, reading them, and then processing them through the machine is difficult because the machine does not have eyes like us.

Machines see any images in the form of a matrix of numbers. The size of the matrix depends on the number of pixels of the input image.

The pixel values for each of the pixels stand for or describe how bright that pixel is, and what color it should be. So In the simplest case of binary images, the pixel value is a 1-bit number indicating either foreground or background. So pixels are the numbers or the pixel values which denote the intensity or brightness of the pixel. Smaller numbers that are closer to zero help to represent black, and the larger numbers which are closer to 255 denote white.

For the case of a colored image, we have three Matrices or the channels Red, Green, and Blue. So in these three matrices, each of the matrices has values between 0-255 which represents the intensity of the color of that pixel.

This is how a computer can differentiate between the images.

Dimensionality Reduction

Dimensionality reduction (Brownlee, 2020), or dimension reduction, is the transformation of data from a high-dimensional space into a low-dimensional space so that the low-dimensional representation retains some meaningful properties of the original data, ideally close to its intrinsic dimension.

In many datasets, we find that the number of features is very large and if we want to train the model it takes more computational cost. To decrease the number of features we can use Principal component analysis (PCA) or feature extraction.

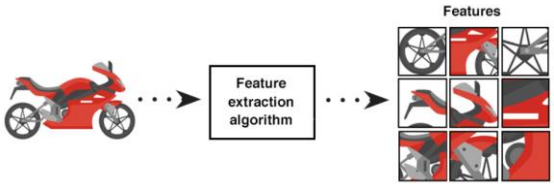
Principal component analysis (PCA)

PCA (Brownlee, 2018) decreases the number of features by selecting the dimension of features that have most of the variance.

Reducing the number of variables of a data set naturally comes at the expense of accuracy, but the trick in dimensionality reduction is to trade a little accuracy for simplicity. Because smaller data sets are easier to explore and visualize and make analyzing data much easier and faster for machine learning algorithms without extraneous variables to process.

So, to sum up, the idea of PCA is simple — reduce the number of variables of a data set, while preserving as much information as possible.
To get the best results, in every algorithm we used we trained the original training data and also the training data after implementing PCA. This allowed us to explore whether the PCA algorithm improved the performance of the algorithms.

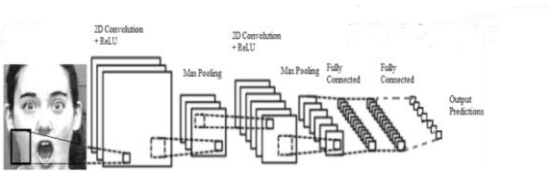
Feature Extraction



Feature extraction refers to the process of transforming raw data into numerical features that can be processed while preserving the information in the original data set. It yields better results than applying machine learning directly to the raw data.

It is part of the dimensionality reduction process, in which an initial set of raw data is divided and reduced to more manageable groups, so when you want to process it will be easier. The most important characteristic of these large data sets is that they have a large number of variables. These variables require a lot of computing resources to process. So Feature extraction helps to get the best feature from those big data sets by selecting and combining variables into features, thus, effectively reducing the amount of data. These features are easy to process, but still able to describe the actual data set with accuracy and originality. In our research, we implemented a few feature extraction methods, VGG16, VGG19, and RestNet50.

VGG 16



VGG16 (G, 2021) is object detection and classification algorithm which is able to classify 1000 images of 1000 different categories with 92.7% accuracy. It is one of the popular algorithms for image classification and is easy to use with transfer learning.

When performing deep learning feature extraction, we treat the pre-trained network as an arbitrary feature extractor, allowing the input image to propagate forward, stopping at the pre-specified layer, and taking the outputs of that layer as our features.

VGG 19

The concept of the VGG19 model (also VGGNet-19) is the same as the VGG16 except that it supports 19 layers. The “16” and “19” stand for the number of weight layers in the model (convolutional layers). This means that VGG19 has three more convolutional layers than VGG16.

Resnet50

ResNet-50 (Rastogi, 2020) is a convolutional neural network that is 50 layers deep. You can load a pre-trained version of the network trained on more than a million images from the ImageNet database. The pre-trained network can classify images into 1000 object categories, such as keyboard, mouse, pencil, and many animals.

Algorithms

K-means clustering

K-means clustering (Bhandari, 2020) is an approach for vector quantization. In particular, given a set of n vectors, k-means clustering groups them into k clusters. in such a way that each vector belongs to the cluster with the closest mean.

K-means clustering can be used to group an unlabeled set of inputs into k clusters, and then use the [*] *centroids* of these clusters to produce features.

* *Centroids: In mathematics and physics, the centroid or geometric center of a plane figure is the arithmetic mean position of all the points in the figure. Informally, it is the point at which a cutout of the shape could be perfectly balanced on the tip of a pin. The same definition extends to any object in n-dimensional space*

In other words, the K-means algorithm identifies k number of centroids, and then allocates every data point to the nearest cluster, while keeping the centroids as small as possible.

The 'means' in the K-means refers to averaging of the data; that is, finding the centroid. (Bhandari, 2020)

* *Mean: The Mean (aka the arithmetic Mean, different from the geometric mean) of a dataset is the sum of all values divided by the total number of values. It's the most commonly used measure of central tendency and is often referred to as the "average."*

How the K-means algorithm works

To process the learning data, the K-means algorithm in data mining starts with the first group of randomly selected centroids, which are used as the beginning points for every cluster, and then performs iterative (repetitive) calculations to optimize the positions of the centroids. It halts creating and optimizing clusters when either:

1. The centroids have stabilized — there is no change in their values because the clustering has been successful.

2. The defined number of iterations has been achieved.

Gaussian Mixture

Gaussian mixture (MIT, 2017) models (GMM) are composed of k multivariate normal density components, where k is a positive integer. Each component has a d-dimensional mean (d is a positive integer), a d-by-d covariance matrix, and a mixing proportion. Mixing proportion j determines the proportion of the population composed by component j, j = 1,...,k.

Gaussian Mixture models work based on an algorithm called Expectation-Maximization, or EM. When given the number of clusters for a Gaussian Mixture model, the EM algorithm tries to figure out the parameters of these Gaussian distributions in two basic steps.

- The E-step makes a guess of the parameters based on available data. Data points are assigned to a Gaussian cluster and probabilities are calculated that they belong to that cluster.

- The M-step updates the cluster parameters based on the calculations from the E-step. The mean, covariance, and density are calculated for clusters based on the data points in the E step.

The process is repeated with the calculated values continuing to be updated until convergence is reached.

Our Process

Due to limited computational resources we decided to focus our research on the Hebrew alphabet signs, instead of the all Hebrew sign language.

In order to test the performances of the algorithms we chose, we started training with 4 different labels (the first 4 letters of the Hebrew alphabet). We wanted to make sure we can obtain good results in learning a small number of signs before we moved on to a higher number of labels.

When using our algorithms to train a 4-labeled dataset we found out that Gaussian Mixture can only yield results on the PCA data. Attempts to run the Gaussian Mixture Model on the data without PCA always give an out-of-memory error.

Results

The next part of our research was to test our algorithm with 10 labels. We used the first 10 letters of the Hebrew Alphabet, for each word we used 200 pictures to train our algorithms. This time we only used the algorithms that proved efficiency in the first stage (on 4 labels).

Our goal was to train the entire Hebrew Alphabet, but we realized that with the limited resources we have it wasn't possible for our computers to process that much data, and achieve generalized results.

	K-Means		Gaussian Mixture	
	PCA	No PCA	PCA	No PCA
VGG16	-	0.5385 9061	-	Doesn't work without PCA
VGG19	0.686 78592	0.6866 9530	-	
ResNet50	Results were poor on 4 images Decided not to continue with this feature extraction algorithm			

Conclusions

This paper demonstrates how to construct, use, and evaluate a high-performance unsupervised ML system for classifying images.

We used a few feature extraction methods, which are built as convolutional neural networks, pre-trained on the ImageNet dataset of natural images to extract feature representations for each micrograph. After applying principal component analysis to extract signals from the feature descriptors, we used k-means and Gaussian Mixture clustering to classify the images without needing labeled training data.

The best algorithm we found is the K-Means algorithm, with VGG19. The algorithm gave very similar results with and without using PCA.

Future work

User-dependent model using pre-training:

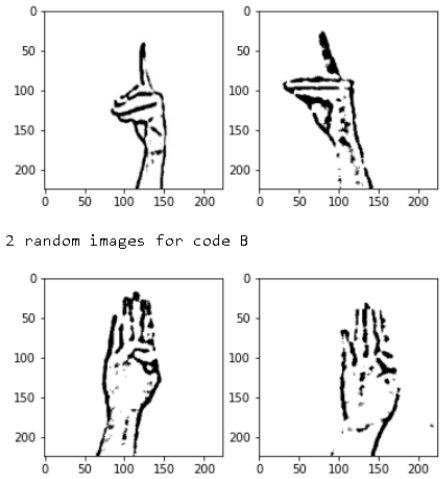
Pre-training the model on a larger dataset (e.g. ILSRVC), which consists of around 14,000 classes, and then fine-tuning it with ISL dataset, so that the model can show good results even when trained with a small dataset. For user-dependent, the user will give a set of images to the model for training, so it becomes familiar with the user. This way the model will perform well for a particular user.

Given that sign language are under-resourced, unsupervised feature learning techniques are the right tools and our results show that this is realistic for sign language identification. Future work can extend this work in two directions:

1) By increasing the number of signs and signers, to check the stability of the learned feature activations and to relate these to iconicity and signer differences.

2) By comparing our method with deep learning techniques. In our experiments, we used a single hidden layer of features, but it is worth researching deeper layers to improve performance and gain more insight into the hierarchical composition of features. Other questions for future work. How good are human beings at identifying sign languages? Can a machine be used to evaluate the quality of sign language interpreters by comparing them to a native language model?

Data



The data we used for our research is in this OneDrive folder - https://1drv.ms/u/s!Aqmah9OMflvlgZgjmS9m1ZKQU_n22g?e=hfjwVl

The compressed file contains 3 folders: test, train, and validation. All images are images of the Hebrew Alphabet, post-processing. To simplify the input images, a binary mask is applied, and the hand's edges are highlighted. The binary mask consists of gray-scaling, blurring, and applying thresholding.

We would like to thank Romansko/SignLanguageRecognition: Hebrew sign language real time recognition using CNN, Keras & OpenCV. (github.com) for making Hebrew SL data available.

Acknowledgment

This work was created as part of the Final Project for Bs.c in Computer Science and Mathematics at Ariel University. We would like to thank our

supervisor Dr. Gil Ben Artzi for guiding us and assisting us throughout the process.

References

1. Bhandari, P. (2020, October 9). *What Is the Mean | How to Find It & Examples*? Scribbr. Retrieved August 16, 2022, from https://www.scribbr.com/statistics/mean/

2. Brownlee, J. (2018, March 2). *How to Calculate Principal Component Analysis (PCA) from Scratch in Python*. Machine Learning Mastery. Retrieved August 16, 2022, from https://machinelearningmastery.com/calculate-principal-component-analysis-scratch-python/

3. Brownlee, J. (2020, May 6). *Introduction to Dimensionality Reduction for Machine Learning*. Machine Learning Mastery. Retrieved August 16, 2022, from https://machinelearningmastery.com/dimensionality-reduction-for-machine-learning/

4. Ecosystem, E. (2018, September 12). *Understanding K-means Clustering in Machine Learning | by Education Ecosystem (LEDU)*. Towards Data Science. Retrieved August 16, 2022, from https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1

5. G, R. (2021, September 23). *Everything you need to know about VGG16 | by Great Learning*. Medium. Retrieved August 16, 2022, from https://medium.com/@mygreatlearning/everything-you-need-to-know-about-vgg16-7315defb5918

6. Rastogi, A. (2020, 3 14). *ResNet50. ResNet-50 is a convolutional neural… | by Aditi Rastogi*. Dev Genius. Retrieved August 16, 2022, from https://blog.devgenius.io/resnet50-6b42934db431

7. MIT UNI, M. (2017, 3 8). *Gaussian Mixture Models*. Gaussian Mixture Models. Retrieved August 16, 2022, from https://lost-contact.mit.edu/afs/inf.ed.ac.uk/group/teaching/matlab-help/R2016b/stats/gaussian-mixture-models-1.html

8. Wilson, A. (2020, December 7). *A Brief Introduction to Unsupervised Learning | by Aidan Wilson*. Towards Data Science. Retrieved August 16, 2022, from https://towardsdatascience.com/a-brief-introduction-to-unsupervised-learning-20db46445283