# Identify characters from Google Street View Pictures

Ritesh Kumar (11602)  Srikant Singh (12729)
Professor Amitabha Mukherjee
March 16, 2014

## Introduction

The objective of this project is to identify characters from Google Street view pictures. In this project we using images of sign boards, hoardings and name of shops, etc. obtained from Google Street view. Our aim is to first detect the English character set (by the English character set we mean English alphabets of Latin script and Hindi – Arabic numerals) and then correctly identify the different characters within it.

## Motivation

Character recognition is a classic pattern recognition problem for which researchers have worked since the early days of computer vision. For Latin script, this is largely considered a solved problem in constrained situations, such as images of scanned documents containing common character fonts and uniform background. However, images obtained with popular cameras and handheld devices still pose a formidable challenge for character recognition. With increased use of services like Google street view, etc., there arises a need for an efficient implementation for such images.

With a proper implementation and enough dataset, a good solution for this problem can be provided, with widespread applications, such as tagging images with names of shops using images of sign boards and hoardings, thus adding more information to google maps. A count on the number of advertisements using images of hoardings can also be maintained and thus can track various advertisements throughout cities.

## Dataset

Our data has been taken from a data set referred to as "*Chars74K dataset*" [1], which contains characters of both English character set as well as of the Kannada script. We however have based our project solely on the English character set. Our dataset as such consists of the following -

- 64 classes (0-9, A-Z, a-z)

- 7705 characters obtained from natural images
- 3410 hand drawn characters using a tablet PC
- 62992 synthesised characters from computer fonts

This gives a total of over 74K images as is evident in the name of the dataset.

## Related works

There has been much research done in the field of character recognition from images. One such method employs deep convolutional neural network operating directly on the image pixels. In this method, the DistBelief [2] implementation of deep neural networks was used in order to train large, distributed neural networks on high quality images. [3]

Another interesting approach involved using images from the surrounding area of the concerned image (the camera was translated up and down the street and rotated in all directions to give a view from all angle and lighting conditions). The task of image text recognition was done in two different phases – which were referred to as "text detection" and "word recognition. A support Vector Machine based on locally aggregated statistical features was employed for text detection. The Word recognition implementation took these to actually determine the text. [4]

## Our Approach

We may define our approach to the problem in two parts. The first part involves the description of the language we are using and the second involves the algorithm we are going to use. We are going to use "*Julia*", a relatively new programming language for data science. It is quite a good language for technical computing that attempts to combine the strengths of other popular programming languages. For the algorithm, we are going to implement the K Nearest Neighbours algorithm, with additional features such as parallelization and speed.

Using "*Julia*", we implement the K-Nearest Neighbours (referred to as k-NN henceforth) algorithm with Leave-One-Out-Fold Cross Validation (referred to as LOOF-CV). LOOF CV is cross validation technique is which out of n observations, we leave one of the observations as the validation set and the remaining observations as used the training set. This is repeated on the original sample with n different validation and training sets (as is evident). We customize the implementation of the algorithm to use this property. This results is a more efficient method involving tuning of the parameter k (the number of neighbours). For cases where a custom implementation is necessary, Julia is an attractive language choice because of easy to write prototypes and no need of unnecessary external coding. We will also incorporate parallelization in Julia, which allows us to speed up the program.

## References

[1] T. E. de Campos, B. R. Babu and M. Varma, *Character recognition in natural images*, Proceedings of the International Conference on Computer Vision Theory and Applications (VISAPP), Lisbon, Portugal, February 2009.

[2] *Large Scale Distributed Deep Networks*, Jeffrey Dean, Greg S. Corrado, Rajat Monga, Kai Chen, Matthieu Devin, Quoc V. Le, Mark Z. Mao, Marc'Aurelio Ranzato, Andrew Senior, Paul Tucker, Ke Yang, Andrew Y. Ng

[3] *Recognizing Text in Google Street View Images*, James Lintern, University of California, San Diego.

[4] *Multi-digit Number Recognition from Street View Imagery using Deep Convolutional Neural Networks*, Ian J. Goodfellow, Yaroslav Bulatov, Julian Ibarz, Sacha Arnoud, Vinay Shet Google Inc., Mountain View, CA.