Identify characters from Google Street View Pictures

Ritesh Kumar
11602

Srikant Singh
12729

Problem Statement and Aim

- Recognising characters of the English character set from Google Street view pictures.
- The pictures mentioned above comprise of images of sign boards, hoardings, name of shops, etc.

Motivation

- Images coming from cameras and other such devices need better implementation for character recognition.
- A good solutions to this problem could improve the information present on google maps, with hand painted shop signs, etc. being easily recognised and appropriate image tagging done.

Past works

- In [2], the researcher basically used a Support Vector for detecting regions in image containing text. Text recognition phase consisted of performing Histogram processing was performed on the resulting followed by OCR for text recognition.

Past works (Contd.)

- In [3], the researchers used deep neural networks to train large, distributed neural networks on high quality images. The DistBelief[4] implementation of deep neural networks was used.

Our Approach

- We used Binarization technique for text region detection in images, followed by training the data using Random forests algorithm.
- We then use k-Nearest Neighbours (k-NN) algorithm with LeaveOneOut Fold Cross Validation (LOOF-CV) for measuring the performance of our model.

Binarization Algorithm

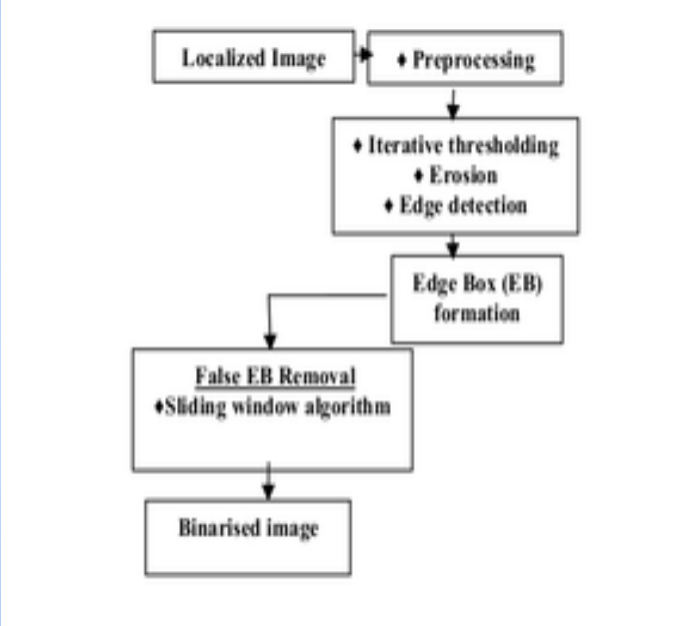
We propose the following algorithm[1] for text region detection in images -

- Preprocessing : To make foreground objects clearer than the background.
  - ➢ Conditional contrast enhancement
  - ➢ Smoothing
  - ➢ Grey scale extension

- Iterative thresholding and edge detection : Edges are detected from pre-processed images using an iterative thresholding algorithm and morphologic erode algorithm

Binarization Algorithm

- False Edge Removal and Binarization by Sliding Window Approach : Each edge box is checked for various characteristics of individual text character and edge boxes not fulfilling the criteria are removed.



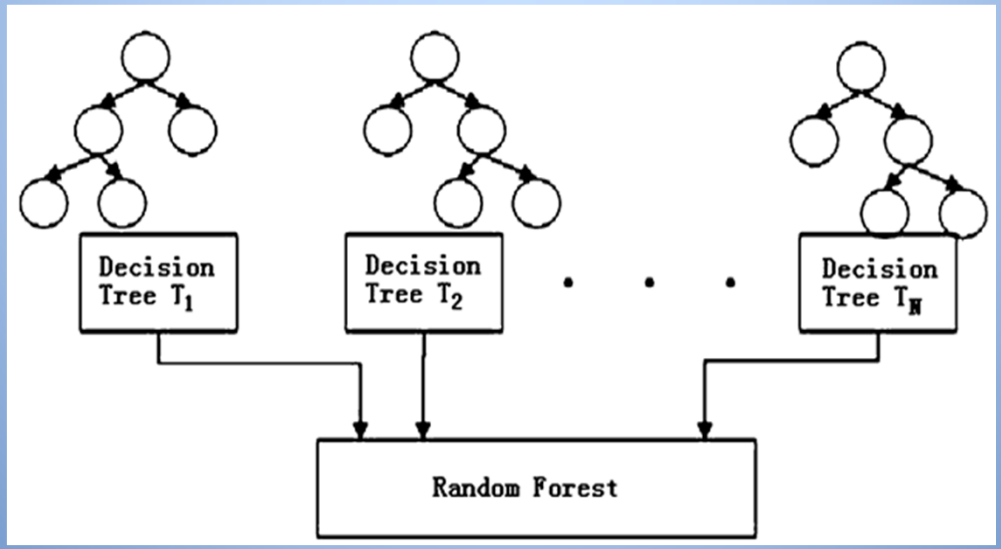
Flow chart showing the proposed algorithm

Random forest Algorithm

- Random forests are decision trees with ensemble learning incorporated.
- If decision trees grow very deep, they learn irregular patterns and show overfitting.
- This happens as they have because they have low bias and high variance.
- Random Forests use averaging between multiple decision trees, to avoid this problem.

Random forest Algorithm

- We have to set 3 parameters for this model - the number of features to choose at each split, the number of trees, and the ratio of subsampling.



Random forest Algorithm

- The number of features to try at each split is usually chosen to be -

$$\sqrt{no.of\ features}$$

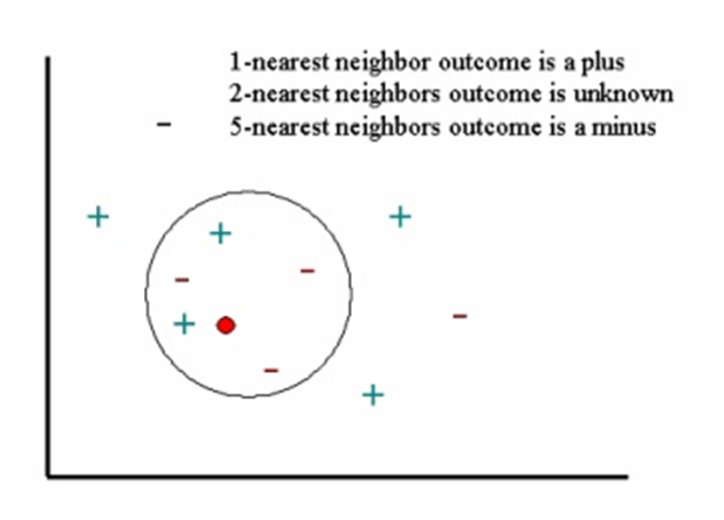
- The numbers of trees is arbitrary and the ratio of subsampling is usually chosen to be 1.
- We can also chose different values for the above mentioned parameters and then choose the one with the best results.

Validation and performance measure

- In any k-fold cross validation, the training data is split in k partitions.
- The model is tested on one of the k folds, with others for training the data.
- To remove overfitting and improve time cost, Leave-One-Out-Fold-Cross-Validation (LOOF-CV) is used
- The model is tested on each individual data point after being trained with the remaining points in the data.
- As only one point is left out in training the data, the problem of bias is reduced

Validation and performance measure

- We first use k-nn to find the k nearest neighbours to the given point.



A simple depiction of k-NN algorithm

- LOOF-CV is applied by excluding the test point and only that given point.
- A label is assigned to our test point, from the selected neighbours.
- This is then carried out for each point in the training data.
- By comparing our predictions with the true labels, we judge the accuracy of the model.

References -

1) *Sliding window approach based Text Binarization from Complex Textual images*, Chitrakala Gopalan, D.Manjula, Chennai, India.
2) *Recognizing Text in Google Street View Images*, James Lintern, University of California, San Diego.
3) *Multidigit 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.
4) *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

Thank You

Any Questions?