Design Document

Mobile Optical Character Recognition of Hindi

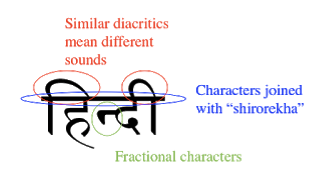
Laikh Tewari

May 22, 2018

1 Problem Statement:

* Tourists in India can’t read Hindi so they get lost
  + Menus, street signs, stores
* Current Hindi OCR methods are inaccurate, not mobile, and slow/require large processors

2 Scope:

* 2.1 Big picture:
  + Design a mobile app that allows a user to point their phone at Hindi text and quickly translates the text into English
* 2.2 Constraints:
  + Hindi OCR is more difficult than English (Phoenician) OCR
    - English, French, and Spanish use few discrete characters with maybe a few diacritics, Hindi forms words with conjoined characters, many diacritics, and fractional character elements.
    - 
* 2.3 Goals:
  + Develop and train a deep convolutional neural network to recognize discrete Hindi characters
  + Develop iOS app with camera functionality
  + Integrate ConvNet using CoreML for efficiency and speed
  + Web scrape Google Translate using extracted text for translation
* 2.4 Future work:
  + Preprocess image of full words to separate conjoined words into individual characters and “read” right to left
  + Implement real time translation by augmenting camera view with translated text to make interface more intuitive and friendlier for users
  + Use natural language processing techniques such as a recurrent neural network improve accuracy by looking at context of characters
  + Test accuracy of ConvNet against other models such as Extra Random Decision Forests

3 Software Design:

* 3.1 Overall structure of iOS app:



* 3.2 Dependencies
  + Developed using Xcode IDE
  + Libraries used include UIKit, AVFoundation, AFNetworking, SwiftyJSON, CoreML
    - CoreML implicitly takes advantage of Accelerate, BNNS, Metal performance shaders
  + Google Translate HTTP web API
* 3.3 Features for Goals
  + Asynchronous camera delegation to prevent blocking of main queue
  + Potential to convert into real-time translation
    - Could be slower
    - More intuitive for user
    - Implements augmented reality natively using ARkit by replacing Hindi text with English text in camera view

4 Theory of Operation

* 4.1 Convolutional Neural Networks
  + 4.1.1 Perceptron model
  + 4.1.2 Activation functions
    - Sigmoid
    - ReLU
    - Leaky ReLU
  + 4.1.3 Learning
    - Gradient Descent
    - Stochastic Gradient Descent
    - Backpropagation
    - Cost Function
  + 4.1.4 Derivations
    - Error in output layer
    - Error in layer in terms of subsequent layers
    - Partial derivative of cost with respect to weights
    - Partial derivative of cost with respect to biases
* 4.2 CoreML
  + 4.2.1 Motivation
  + 4.2.2 Dependencies
    - Basic Neural Network Subroutines (BNNS)
    - Accelerate
    - Metal performance shaders
* 4.3 iOS Development
  + 4.3.1 App production
  + 4.3.2 AVFoundation

5 Results

* 5.1 Accuracy
* 5.2 Speed
* 5.3 Memory