Paper 2832-2018

Image Classification Using SAS® Enterprise Miner 14.1

Introduction

Character classification or image classification plays a vital role in many computer vision problems (for example, optical character recognition (OCR), license plate recognition, and so on), and therefore could be used in solving many business problems. The task is challenging because, in addition to dealing with the large number of levels necessary to classify each image, extensive data preparation is also required. We worked with a new publicly available data set, the Devanagari Handwritten Character Dataset available on the UCI Machine Learning Repository website, which contains 92 thousand images of 46 Devanagari characters. Our goal is to develop an image recognition system for Devanagari script.

Devanagari is an Indic script and forms the basis of over 100 languages spoken in India and Nepal, including Hindi, Marathi, Sanskrit, and Maithili to name a few. There is no capitalization of words, unlike in the English language. The learning model was trained over 80,000 image (32 *32 pixels). Each image can be classified into 46 different characters. One of the objectives of the image classification problem is to make a system where it can automatically detect the handwritten character and thus can be used in variety of applications like number plate detection or scanning of documents. The basic problem with character classification is it being a number of outcomes.

This work proposes to analyze character images and classify them into different characters using traditional machine learning methods as well as industry standard neural networks and measure the accuracy. The entire analysis has been done using python, but with the introduction of SAS VIYA we can easily pull the Jupyter notebook into SAS Viya and use the robust features of SAS.

Methodology

To classify the image into characters, both traditional and advance methods like CNN were adopted.

- ### Pixel level features to classify characters

  First, image compressing is done and to make sense out of those images, pixel values feature extraction was selected as the best method. Using open CV, pixel values were extracted and each image had 32*32 matrix across RGB values. This 2-dimensional matrix was converted into 1 dimensional matrix to test the assumption that different pixel value of an image could be used to classify an image. Support vector machine was used to classify an image into different characters. The accuracy of prediction after building a multi-class classification model was as low as 20%. Only color pixel values were not sufficient to predict the character.

- ### SIFT features to classify characters

  Each object in an image can have multiple interesting and important features. These features could be extracted to provide description of an object. This description can further be used to locate an object in an image. These feature descriptions are called SIFT features. Each image can have different local point or key point and each key point has a vector length of 128. The aim was to extract the key points from an image, with each key point having a descriptor length of 128, and then compare it with key points of a second image. If two images have similar features, they can have the same character. The idea behind using SIFT features was that characters written by different people would be of similar shape and these shapes can be used to determine the characters. Consider the following images:

These two are the same character written by two different people but the shape is similar. With the help of the sift features we can classify these images.

- Bag of Words

One of the challenges was to come up with same number of local points for each image. Every image has a different number of key points, thus making it difficult to apply any traditional modelling technique. To overcome this, Bag of Words was used to cluster key points from all images. Key points, with descriptor and vector length of 128, from all images were extracted and appended together. FLANN based matcher was used to optimize for nearest neighbors. With the use of FLANN base matcher, similar key points were calculated from appended data set and were assigned to one cluster. In total, 100 clusters were used to assign each key point one cluster value. Therefore, bag of words was calculated with 100 clusters. This process was then repeated for new image and sift features were calculated. Each key point was assigned to one cluster and finally, key points in each cluster for an image were calculated. This process was repeated for all 80000 images and the vector length of 100 was final result. In the end, an array of 80000*100 was produced. Prediction of the character of an image was made using 70 and 30 percent split into training and validation data. After getting the bag of words

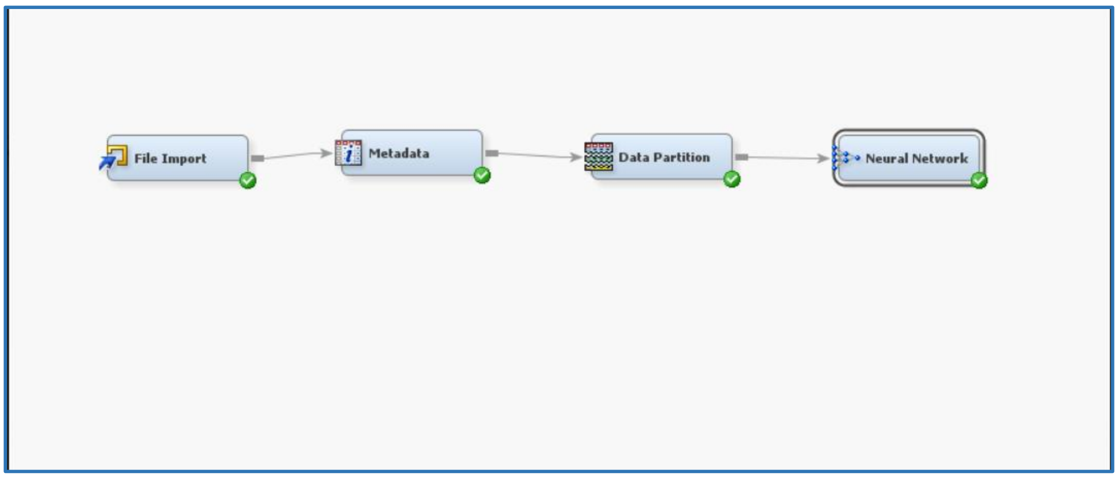
Following two method were implemented in python and SAS respectively.

- Support Vector machine in python

Since the target has more than two levels, SVM classifier was used to classify the images into characters. Different variations were tried by changing the hyper-parameters like kernel, gamma and C value. In the end classifier with linear kernel was chosen for the analysis. The accuracy obtained was about 45% which was quite high as compared to the previous method.

- Neural networks in SAS

The data obtained from bag of words method was then taken into SAS Enterprise miner. Below is the diagram of SAS EM:



Misclassification rate of about 60 percent was achieved using neural networks in SAS.

- CNN based feature extraction to classify characters

In practice, training an entire Convolutional Network from scratch with random initialization is difficult. The reasons being it is rare to have a dataset of sufficient size and the computational power and time required to train a large enough dataset is very high. Instead, the practice is to pre-train a ConvNet on a very large dataset (e.g. ImageNet, which contains 1.2 million images with 1000 categories), and then use the ConvNet either as an initialization or a fixed feature extractor for the task of interest. For the research a ConvNet VGG16 was used to extract features. Natural images are 'stationary', meaning that the statistics of one part of the image are the same as any other part. This suggests that the features that is suggested at one part of the image can also be applied to other parts of the image, and same features can be used at all locations. Neural networks generalize well over these key points, VGG16 from Visual Geometry Group at Oxford University was used, one of the most popular methods, called the VGG-net. The VGG features of an image were extracted.

The Deep Learning Model

The CNN model that was developed, has 2 hidden layers between the VGG features and the final output layer. The simple neural network, an image goes into this network, the first layer's output is 512 dimensions. This hidden layer output undergoes a pointwise RELU activation. The second layer's output dimensions are 256, with another RELU activation. This output gets transformed into the output layer of 46 dimensions. It goes through a SoftMax. By squashing the score of each of the 47 output labels between 0 and 1, SoftMax lets us interpret their scores as probabilities.

Accuracy Score

The accuracy score provides a metric based on which it can be said that probability of the object extracted by VGG16 CNN are in the class in which new CNN was trained on. In our case, the probability of object, identified by the VGG net, in an image to be of A is 60%.

Results

Following are the results of the various techniques that we discussed

MODEL	**ACCURACY (%)** **(1-Misscalssification Rate)**
Pixel Model	<20
Sift with bag of words	40-45
VNN(VGG16)	60

Conclusions

Based on the analysis we can conclude that, deep learning models are becoming an effective way of classifying images. With the help of deep learning techniques, we are now able to solve various computer vision problems like image classification, sound recognition etc. A simple character recognition model could be used effectively for number plate or handwriting detection.

Reference:

[Batik Image Classification Using SIFT Feature Extraction, Bag of Features and Support Vector Machine Ryfial Azhara, , Desmin Tuwohingidea,b, Dasrit Kamudia,b, Sarimuddina, Nanik Suciati]

[Linear Spatial Pyramid Matching Using Sparse Coding for Image Classification: - Jianchao Yang, Kai Yu]

[Convolutional Neural Networks for Image Classification: - Stanford]

http://ufldl.stanford.edu/tutorial/supervised/FeatureExtractionUsingConvolution/

https://towardsdatascience.com/devanagari-script-character-recognition-using-machine-learning-6006b40fa6a9

Contact Information

Akshay Arora

Ph:972-672-7058

Email:Akshay.arora@okstate.edu