OFFLINE HANDWRITTEN MATH EQUATION ANALYSIS

INFO (7390) SEC-4

ADVANCE DATA SCIENCE AND ARCHITECTURES

By: Raghav Avasthi [001873130]

ABSTRACT

Recognition of handwritten mathematical equations from non-interactive and time-invariant data sources such as images is a fascinating area of research in the field of computer vision and pattern recognition. Much research has been done in the field of online handwritten math equation analysis, but offline analysis still stays an area which is majorly untouched. The papers published before use incomparable datasets and results.

Thus, the purpose of this project is to test the hypothesis that Convolutional Neural Networks, specifically Fast-Regional Convolutional Neural Networks can be used to make an offline handwritten mathematical equation recognition system which recognizes individual symbols in math equations in such a robust fashion that it can be used as a dependable recognizer of math symbols from an equation. This problem in the field of computer vision presents a unique kind of difficulty which arises due to context, size and position-based meaning of symbols in the equation; high correlation among the symbols; and due to lack of annotated dataset for this task.

In this paper, Convolutional neural network (CNN)[2] and its derivatives such as Fast-Regional Convolutional Neural Networks (F-RCNN)[3] were tried and applied to the problem by using it to learn and classify math symbols. 3 custom datasets were created of different sizes and number of classes for the purpose of this project. Then, CNN of varying network structure and hyper parameters were applied on 2 custom datasets and their results were quantitatively compared. Finally, F-RCNN were used on them using the Tensorflow Abject Detection API and the results were qualitatively compared for 2 datasets.

INTRODUCTION

Handwritten character recognition has long been a hot topic in the field of machine learning and computer vision. Several algorithms and datasets have been produced and competitions have been conducted on this problem in past few decades. Handwritten math equation recognition is a subset of research in this field which has become very active lately. As handheld computer (alias Smartphones) are getting smarter and more powerful day by day, the demand of such solutions has seen a surge. Moreover, in the past few years various competitions and conferences have been held specifically on the problem of handwritten math equation recognition such as International Conference on Frontiers in Handwriting Recognition (ICFHR) [1] which has seen a rise in number of participating research groups, individuals and organizations.

The problem of online handwritten math equation recognition is well explored in the field of computer vision. It is mainly due to its possible applications for touch sensitive screens and writing pads for computer input, and due to the public release of CROHME (Competition on Recognition of Online Handwritten Mathematical Expressions) dataset in 2011. The field of offline handwritten math equation recognition is still considerably untouched because of the lack of dataset of the images of handwritten mathematical equations. Even if the CROHME dataset is used as a basis of offline handwritten math equation recognition by extracting math expression images from the dataset with their corresponding equations, the lack of symbol level annotation makes it hard to be used for training purposes. As a result, the number of papers published in the field of offline math expression analysis are very few and do not have practically implementable accuracy. Two of the satisfactory solutions found for this problem are developed and owned by 2 private companies i.e. Mathpix and VisionObjects. Thus, the public domain research still has a lot of improvement in this field. Yet, growing use of OCR (Optical character recognition) combined increase in computation power of hand-held devices indicates towards the huge potential for a solution in this field thus making it an interesting problem.

My project investigates into the task of recognizing all the mathematical and non-mathematical symbols used in a mathematical equation. It was done by compiling a dataset from various sources, of many kinds of symbols used in a mathematical equation. This dataset was then made uniform using data cleaning and generalization techniques such as basic image processing techniques such as gray conversion, erosion and dilation of images to pre-process images in the dataset complied. Then, convolutional neural networks and fast-regional convolutional neural networks were trained on this dataset for symbol level recognition.

BACKGROUND

Handwritten Symbol Recognition

Handwritten math equation recognition has been a central issue of many research papers in the field of computer vision and pattern recognition in a past few years. For the first part of the project i.e. math equation contemplation from a given image many papers have been published. Techniques ranging from recursive segmentation cut and semantic relationship analysis to modern RNN[5] and Long Short-Term Memory (LSTM) have been tried solving this problem.

Solutions proposing usage of traditional techniques such as Hidden Markov Models (HMM) or Adaboost and Principal Component Analysis (PCA) [6] have proven to be fast but have lacked accuracy and scope of generalization.

Solutions using neural networks such as Bidirectional Long Short-Term Memory (BLSTM) [7] or Support Vector Machines(SVM) have produced results with commendable accuracy but are time consuming.

Convolutional Neural Networks

Convolutional Neural Networks are pretty much like regular neural networks with a few added layers such as the pooling layer, convolutional layer and application of weight sharing.

Popularized by Prof. Yan LeCun in 2006, Convolutional Neural Networks have proven to be the backbone of many modern computer vision algorithms and solutions.

My approach relies heavily on the CNNs and uses its augmentation techniques to generalize training. I also used other techniques to avoid overfitting such as a dropout layer in my network.

APPROACH

My approach consists of three steps. 1) Data set creation. 2) Symbol level recognition using Convolutional Neural Networks 3) Symbol level recognition using F-RCNN using Tensorflow Object Detection API.

Data Set Creation

Three datasets were created for this project. These datasets were compiled from various other datasets available publicly on the internet. The first dataset named created has 80970 images of size 45x45 pixels with single channel. These images encompass 80 classes i.e. 80 kinds of symbols used in mathematical equations. Second dataset created has 28,782 images of size 32x32 pixels with 3 channels in it i.e. RGB, it is derived from 29 classes of math symbols. The third dataset being the smallest has 16565 single channel images of size 45x45 which are derived from 24 categories of math symbols.

All these datasets were formed by combining various datasets having other varied parameters such as size, contrast etc. Thus, to combine these datasets for creating new datasets various techniques were used. First all images were applied with morphological filter for erosion effect to dilate the black ink mark of symbols in images and make it more prominent. It also helped in preserving the outline of the data when resizing images. Then images were zero padded to make them square shaped images accordingly. Images were then resized to the need size i.e. 45x45 and 32x32 in this case. Then contrast filters were used to even the contrast on them. Speckle noise was removed from images using median filter. Images were then turned into single channel binary images as this preserve most of the data in our case and reduces the computation needs for this project. Thereby, reducing time need to train the algorithms.

	Images	**Channel**	**Size**	**Classes**
Dataset 1	80970	1	45 x 45	80
Dataset 2	28782	3	32 x 32	29
Dataset 3	16565	1	45 x 45	24

Below is the flow chart for dataset creation

[collect data] -> [compile data] -> [erosion] -> [zero padding] -> [resize] -> [contrast adjustment] -> [noise removal] -> binarizing images]

After the image dataset was created, 2 kinds of labels were created in the form of a .csv file for these datasets

Symbol level recognition using CNN (Quantitative)

A quantitative approach was taken while using CNNs in this project. The convolutional neural network used for this project has the following structures.

Network 1: [convolution – activation – convolution – activation – max pool – dropout – flatten – dense – activation – dropout – dense – activation]

Network 2: [convolution – activation – convolution – activation – max pool – convolution – activation – max pool – dropout – flatten – dense – activation – dropout – dense – activation]

The table for their hyperparameter is given below.

	Network 1	**Network 2**
Activation Function	ReLU	ReLU
Cost Function	Categorical cross-entropy	Categorical cross-entropy
Epoch	20	20
Gradient Estimation	Adam	Adadelta

CNNs were applied to dataset 1 and Dataset 2.

Symbol level recognition using F-RCNN (Qualitative)

A qualitative approach was taken while using Fast Regional Convolutional Neural Networks. Tensorflow's Object detection API was used for implementing this algorithm on our custom Dataset 1 and Dataset 3 for 850 epochs.

This was done by creating .csv files for labels and following the instructions given with the API.

RESULTS

Results for applying CNN on the 2 datasets are given below.

Epoch = 20

	Dataset 1	**Dataset 2**
Network 1	Did not converge	84.82%
Network 2	Did not converge	87.46%

Increasing the epochs to 350 for both the datasets

Epoch = 350

	Dataset 1	**Dataset 2**
Network 1	Did not converge	92.82%
Network 2	Did not converge	94.46%

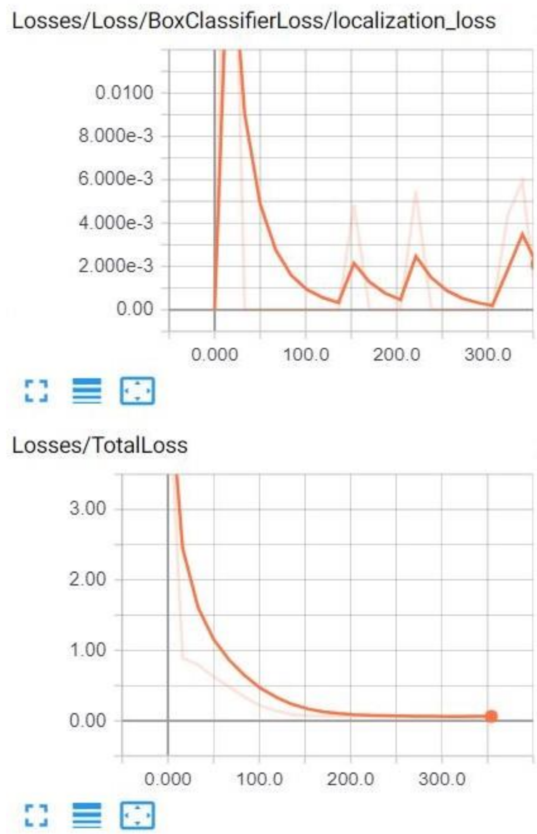


Fig 1: The graphs for Total Loss and localization loss of Dataset 2 with Network 1, for 350 epochs is given below

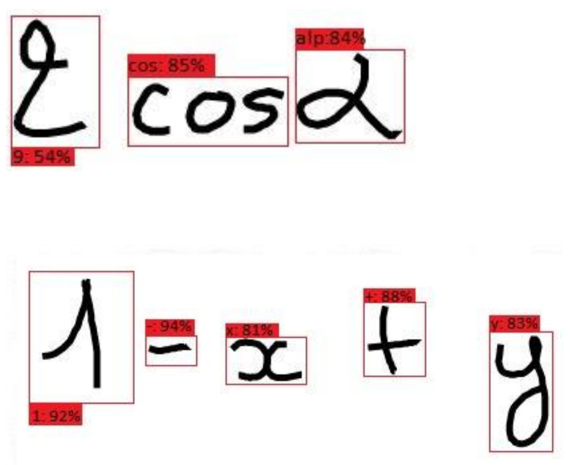


Fig 2: Qualitative results for F-RCNN after 850 epochs on Dataset 3 is given below.

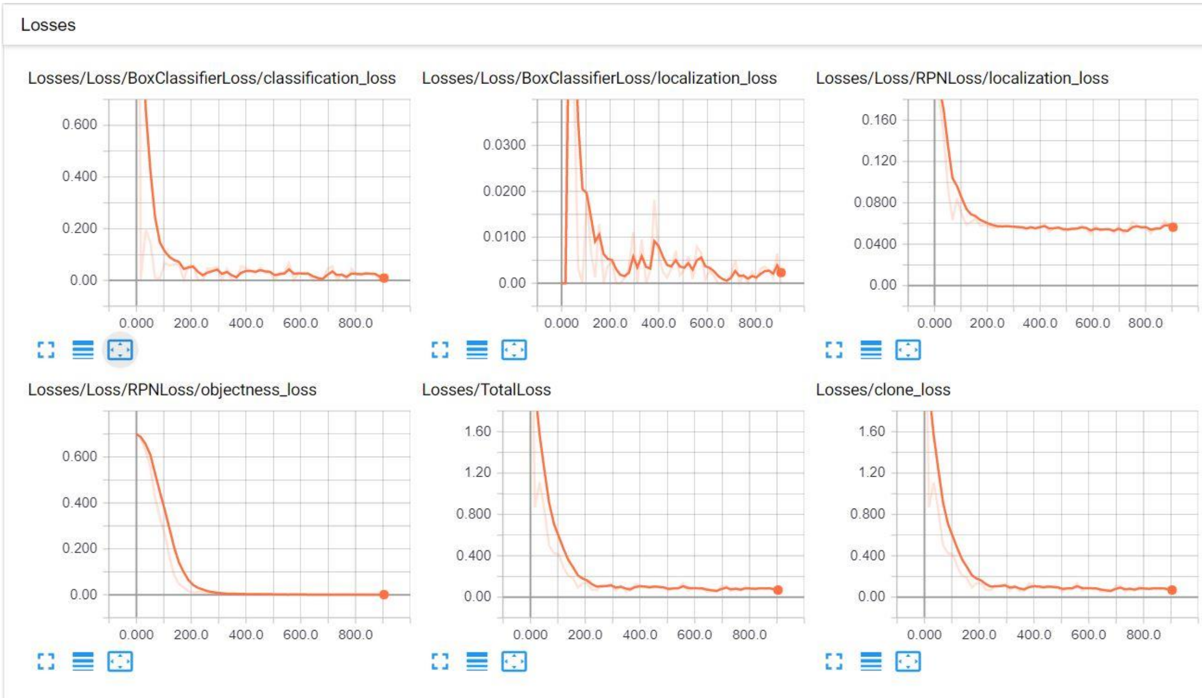


Fig 3: The graphs for all sorts of losses for F-RCNN on Dataset 3 for 850 epochs

Qualitative results for F-RCNN applied on Dataset 1 are not available since the algorithm didn't converge in the needed time and so gives no results i.e. no bounding box on images recognizing symbols on a math equation.

DISCUSSION / CONCLUSION

From these results I can conclude that Convolutional Neural Networks may be suitable for symbol level detection but cannot be used for expression level detection as the accuracy achieved for symbol level classification is comparable to industry standards and can be used for practical purposes but the accuracy in recognizing a handwritten math equation from non-interactive offline sources such as images still has a lot of potential of improvement. As the accuracy in this case is

$$\text{Acc.} = (928/1000)^n \times 100$$

This accuracy for recognizing offline handwritten math equation drops below 50% if there are 10 or more symbols in an equation. It is worth noticing that it goes below 80% for 3 or more symbols in an equation.

In case of F-RCNN, it can be concluded that F-RCNNs can be used for symbol level and expression level detection of offline handwritten mathematical equations but only if the number of classes used to train are conveniently less. It is shown from the result that F-RCNN failed to recognize symbols and failed to converge with the Tensorflow Object detection API with Dataset 1 as it has 80 classes while they were giving satisfactory results when trained with dataset 3 which has just 24 classes.

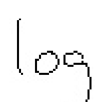
Therefore, it can be said that the hypothesis failed and was wrong.

The reason for such high inaccuracy can be attributed to high correlation among symbols in mathematics which can only be resolved by taking the size, position of a symbol in an equation and the context in which it is written.

(a)

(b)

(c)



(d)

Fig.4: Image(a) can easily be misunderstood as a subset of image (b). ‘1’ in image(c) has high resemblance with ‘L’ in image(d).

From my work on this project I can also conclude that applying a few preprocessing techniques such as padding, resizing, erosion and contrast enhancement help in making the data consistent and uniform thereby increasing the accuracy on the CNN algorithm and reducing time in training it.

Changing the images to 1 channel, black and white image from a 3-channel RGB images also increases the speed in training and reduces the memory requirements.

Using techniques such as Data Augmentation and Dropout layer further avoids overfitting and improve the accuracy of the algorithm.

FUTURE WORK / IMPROVEMENTS

Several improvements can be made in the future works of this project.

Differentiating between compound and simple symbols and training on them separately can help as that would reduce the overlapping as shown in figure 4. Using tree-based techniques or other deep learning algorithms such as YOLO [4] may also help. Training the data on a better GPU for a longer duration will also improve the results.

REFERENCES

[1] http://icfhr2018.org/ as of 9th February 9, 2018

[2] Wang, Fenglei, et al. “Convolutional recurrent neural networks with hidden Markov model bootstrap for scene text recognition.” *IET Computer Vision*, vol. 11, no. 6, Jan. 2017, pp. 497–504., doi:10.1049/iet-cvi.2016.0417.

[3] Ren, Shaoqing, Kaiming He, Ross Girshick, and Jian Sun. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks." [1506.01497] Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. January 06, 2016. Accessed February 09, 2018. https://arxiv.org/abs/1506.01497.

[4] Redmon, Joseph, Santosh Divvala, Ross Girshick, and Ali Farhadi. "You Only Look Once: Unified, Real-Time Object Detection." [1506.02640] You Only Look Once: Unified, Real-Time Object Detection. May 09, 2016. Accessed February 09, 2018. https://arxiv.org/abs/1506.02640?context=cs.

[5] F. Á lvaro, J.-A. Sánchez, and J.-M. Benedí, "Classification of Online Mathematical Symbols with Hybrid Features and Recurrent Neural Networks," in International Conference on Document Analysis and Recognition, 2013, pp. 1012-1016.

[6] A. Graves, M. Liwicki, S. Fernández, R. Bertolami, H. Bunke, and J. Schmidhuber, "A novel connectionist system for unconstrained handwriting recognition," Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 31, no. 5, pp. 855-868, 2009.

[7] Alvaro, Francisco, Joan-Andreu Sanchez, and Jose-Miguel Benedi. "Recognition of on-line handwritten mathematical expressions using 2d stochastic context-free grammars and hidden markov models. " Pattern Recognition Letters (2012).