Denoising Dirty Documents

Project Final Report

2017 DS/NC/ESD 863 Machine Perception

Sriveda Reddy
IMT2013047

Udbhav Vats
IMT2013055

Simran Dokania
IMT2013044

Rishabh Manoj
IMT2013035

{SrivedaReddy.Chevuru, Udbhav.Vats, Simran.Dokania, Rishabh.Manoj
}@iiitb.org

IIIT-Bangalore

May 11, 2017

Contents

List of Figures

List of Tables

1 Problem Statement

Given a dataset of images of scanned text (synthetic images) that are “noisy” with stains and wrinkles, we propose to clean up the noise and help with the digitization process.

2 Motivation

Optical Character Recognition (OCR) is the process of getting typed or handwritten documents into a digitized format. The motivation of converting to a digitized format is to ensure security, accessibility, edit-ability and ease of searching and sharing. Also, digital documents don’t get dirty and cannot be ruined by coffee stains. [1]

Unfortunately, a lot of documents eager for digitization are being held back. Coffee stains, faded sun spots, dog-eared pages, and lot of wrinkles are keeping some printed documents offline and in the past. We were interested in speeding up this process and hence chose this topic.

3 Dataset

Kaggle provided a data-set which consists of two sets of images - train and test. These images contain various styles of text, to which synthetic noise has been added to simulate real-world, messy documents. The dirty images contain stains as well as creased paper. The training set also includes the cleaned up images of those found in the test file (train_cleaned) [1]. By clean, we mean black letters on a white background.

Additionally, a set of real images were procured, which contained stains and creases. We tested on these images to check if the algorithms developed using simulated data can be applied on the "real-world" messy documents.

Kaggle calculates the score based on the root-mean-squared-error (RMSE) value between each pixels of the generated output and the actual cleaned image.

4 Methods

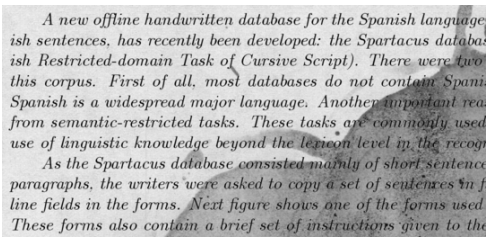
In [3], we tried out some image processing techniques, some of which worked well in removing the noises while others were not so efficient. Here, we propose methods that involve Machine Learning and Neural Networks.

4.1 Random Forest

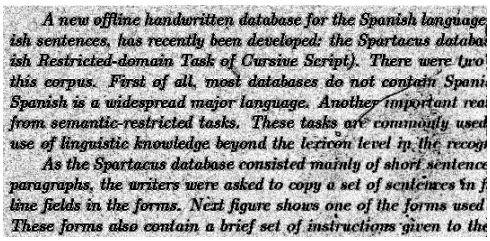
Here we propose a purely machine learning technique without any pre-processing whatsoever. The basic idea is to use a random forest regressor model to predict the pixel intensity based on neighbouring pixels.

Algorithm:

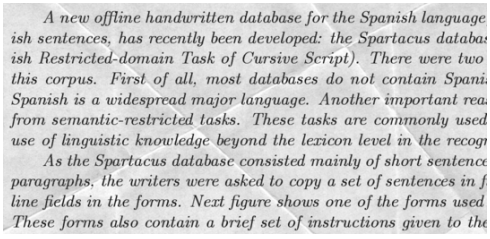
- Pad out each image by an extra 2 pixels (i.e) NxN becomes $(N+2)$x$(N+2)$.
- Run a 3×3 sliding window on the image. Please note that every pixel of the original image will at least become the center of the sliding window once.
- Use all 9 pixels within the sliding window as predictors for the pixel in the centre of the sliding window (i.e) All the pixels in the sliding window of the dirty image acts as a feature to predict the centre pixel of the window for the cleaned pixel.
- Use a Random Forest regressor model to predict the pixel brightness.



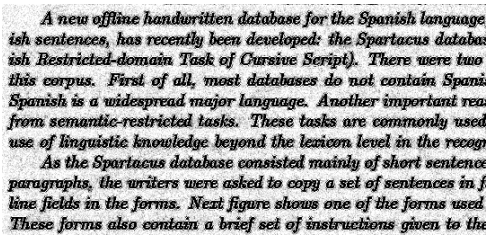
(a) Original Image



(b) Cleaned Image



(c) Original Image



(d) Cleaned Image

Figure 1: Using Random Forest a)

(a) Original Image (b) Cleaned Image

Figure 2: Using Random Forest b)

While this method succeeds in removing the stains [2], it does not work very well with dog-ears and creases [1], in fact random forest just makes it worse. It looks as if random forest takes the stain and sprinkle it across the entire image so that the stains are not concentrated in one particular spot but more milder but widespread. This, as one can see from the cleaned image, is not conducive for reading and thus will not help us in our goal of converting to a digitized format for future use.

The RMSE score in Kaggle is 0.32492.

4.1.1 Challenges Faced

Fitting the training data to the model was gigantic task. We initially tried partial fitting but the results obtained were just random noises. The entire data-set had to be loaded simultaneously to get at least a proper output. Also training the model took around half an hour as we were unsure how to use GPU for this computation. To facilitate easier understanding we opted to go with IPython which is a very powerful interactive python shell. This helped us in saving the trained models and tracking variables without re-doing the entire thing.

4.2 Neural Network

We create a simple feed-forward neural network that de-noises one pixel at a time. This neural network has one hidden layer. Each layer contains a weight matrix W and a bias vector b and computes the function:

$$act(input * W + b)$$

where *act* is typically some sort of sigmoid function.

The activation function of the input layer is the *tanh* function, while the activation function for the hidden layer is the *clip* function of theano which clips the value

based on the given minimum and maximum value (i.e)

```
def clip(x,minx,maxx):
    if(x < min):
        return minx
    elif(x > max):
        return maxx
    return x
```

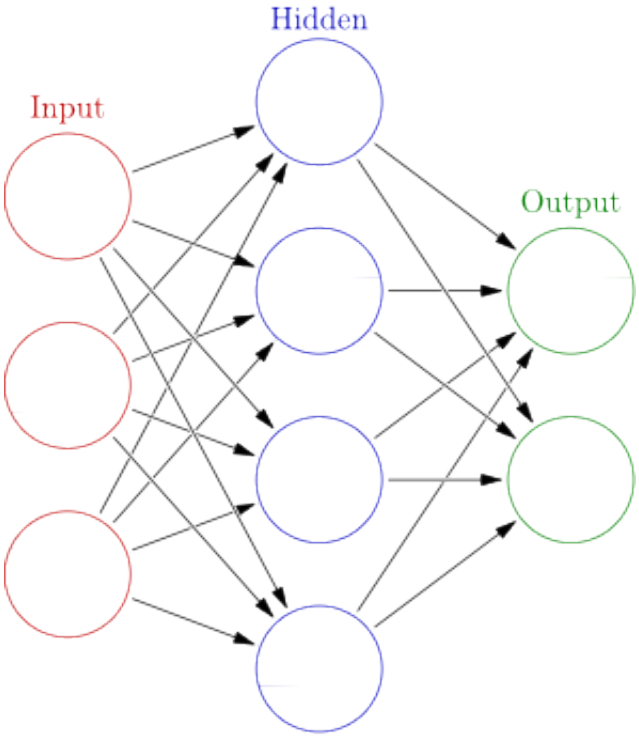


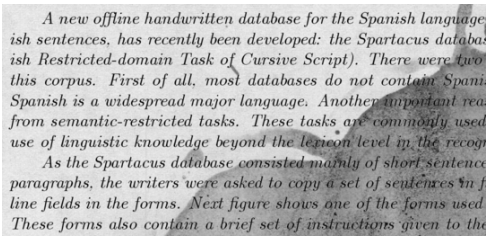
Figure 3: Artificial Neural Network [2]

The hidden layer contains 10 neurons, the no. of neurons for the input is 29 (which is the no. of feature vectors) and output layers has one neuron which is the pixel brightness.

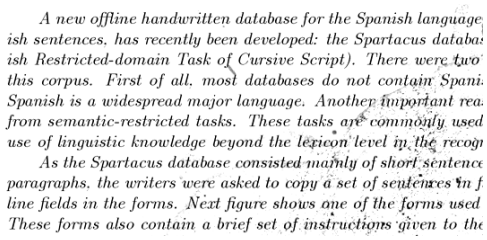
Before passing the images to the neural network, we first calculate the features of the image. We consider neighbouring pixels of the center pixel using a 5x5 window

as boundary as features. So for each pixel we have a feature vector containing 25 feature points. Also we do some initial image processing on these image and take the output as features for the neural network. We use median blur with kernel size 5 and kernel size 25. Using the *Sobel* operative we calculate the first and second derivative of the images. For each pixel of the image, we have 4 image processing outputs, the median blur with kernel size 5,the median blur with kernel size 25,first sobel derivtive and second derivative. These are then added to the already existing 25 feature points making the total to 29 feature points for each pixel. The feature vectors are combines together to create a feature matrix for the image and given to the neural network.

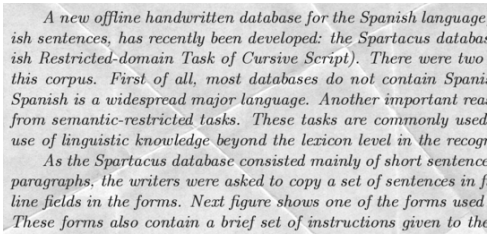
We train the neural network using a naive gradient descent learning algorithm with the entire data-set.



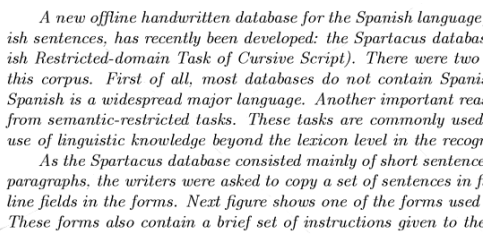
(a) Original Image



(b) Cleaned Image

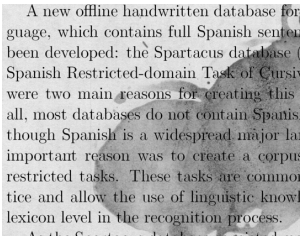


(c) Original Image

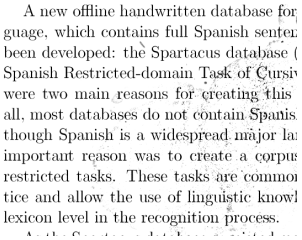


(d) Cleaned Image

Figure 4: Neural Network a)



(a) Original Image



(b) Cleaned Image

Figure 5: Neural Network b)

As you can see from [4] and [5] the creases are pretty much invisible to the eye while the stains are faded to the point that only faint patches are visible.

The RMSE score in Kaggle is 0.03363.

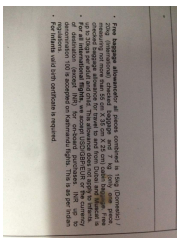
4.2.1 Challenges Faced

We could not use the entire training data as our RAM was too small for it. We used only half the training data for this method. Ideally we should have trained this for atleast 100 iterations(epochs) but due to low computational power we trained it only for 10 iterations(epochs) which took around 20 minutes in a GPU.

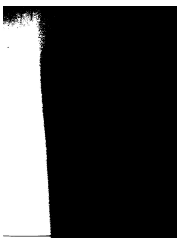
5 Experiments

We experimented these methods and methods mentioned in [3] with "real-world" data (i.e) actual images of text paper with stains. The results were pretty varied as you can see below

5.1 Fixed Thresholding



(a) Original Image



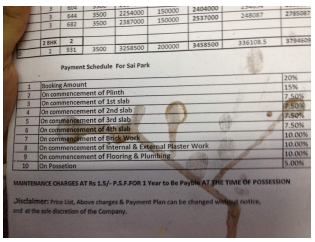
(b) Cleaned Image

Figure 6: Fixed Thresholding on Real World Image a)

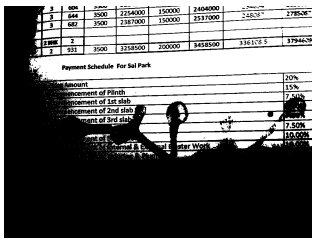
(a) Original Image

(b) Cleaned Image

Figure 7: Fixed Thresholding on Real World Image b)



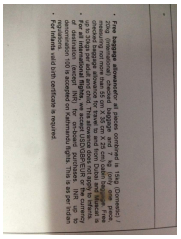
(a) Original Image



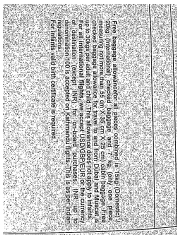
(b) Cleaned Image

Figure 8: Fixed Thresholding on Real World Image c)

5.2 Adaptive Thresholding



(a) Original Image



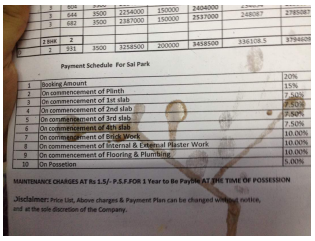
(b) Cleaned Image

Figure 9: Adaptive Thresholding on Real World Image a)

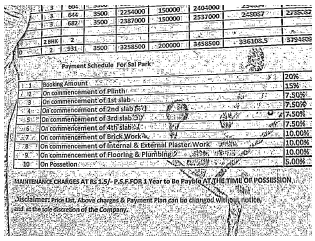
(a) Original Image

(b) Cleaned Image

Figure 10: Adaptive Thresholding on Real World Image b)



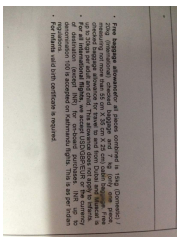
(a) Original Image



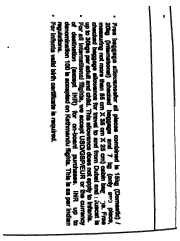
(b) Cleaned Image

Figure 11: Adaptive Thresholding on Real World Image c)

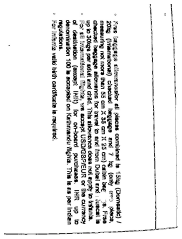
5.3 Canny Edge Detection and Morphology



(a) Original Image



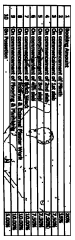
(b) Cleaned Image using Dilation



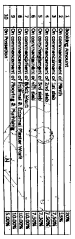
(c) Cleaned Image using Erosion

Figure 12: Canny Edge Detection and Morphology on Real World Image a)

(a) Original Image

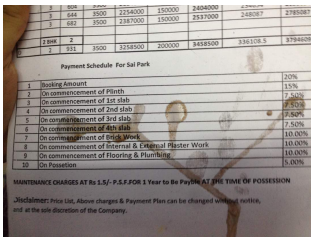


(b) Cleaned Image using Dilation

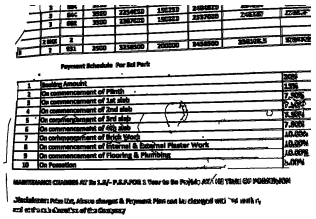


(c) Cleaned Image using Erotion

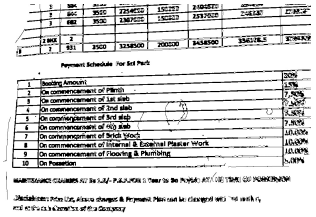
Figure 13: Canny Edge Detection and Morphology on Real World Image b)



(a) Original Image



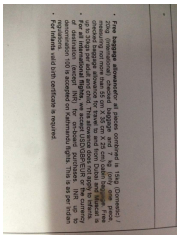
(b) Cleaned Image using Dilation



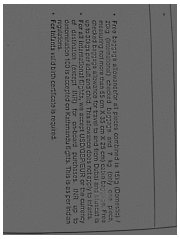
(c) Cleaned Image using Erosion

Figure 14: Canny Edge Detection and Morphology on Real World Image c)

5.4 Median Filtering



(a) Original Image



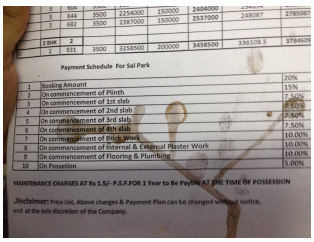
(b) Cleaned Image

Figure 15: Median Filtering on Real World Image a)

(a) Original Image

(b) Cleaned Image

Figure 16: Median Filtering on Real World Image b)



1	2 BHK	2					
2	501	3500	3258500	200000	3458500	336108.5	3794609

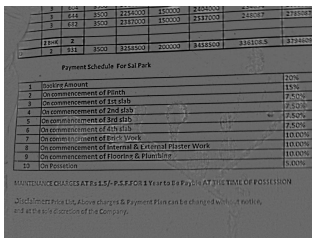
Payment Schedule For Sal Park

1	Booking Amount	20%
2	On commencement of Plinth	15%
3	On commencement of 1st slab	7.50%
4	On commencement of 2nd slab	7.50%
5	On commencement of 3rd slab	7.50%
6	On commencement of 4th slab	7.50%
7	On commencement of Brick Work	10.00%
8	On commencement of Internal & External Plaster Work	10.00%
9	On commencement of Flooring & Plumbing	10.00%
10	On Possesion	5.00%

MAINTENANCE CHARGES AT Rs 1.5/- P.S.F.FOR 1 Year to Be Paying AT THE TIME OF POSSESSION

Disclaimer: Price List, Above charges & Payment Plan can be changed without notice, and at the sole discretion of the Company.

(a) Original Image



1	2 BHK	2					
2	501	3500	3258500	200000	3458500	336108.5	3794609

Payment Schedule For Sal Park

1	Booking Amount	20%
2	On commencement of Plinth	15%
3	On commencement of 1st slab	7.50%
4	On commencement of 2nd slab	7.50%
5	On commencement of 3rd slab	7.50%
6	On commencement of 4th slab	7.50%
7	On commencement of Brick Work	10.00%
8	On commencement of Internal & External Plaster Work	10.00%
9	On commencement of Flooring & Plumbing	10.00%
10	On Possesion	5.00%

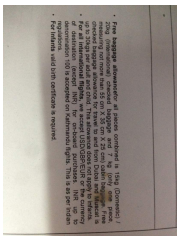
(b) Cleaned Image

Figure 17: Median Filtering on Real World Image c)

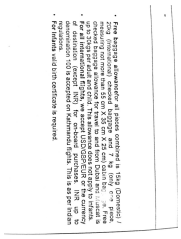
5.5 Random Forest Regression

Unfortunately the regressor model outputs all zeroes when given real world image as input

5.6 Artificial Neural Network



(a) Original Image



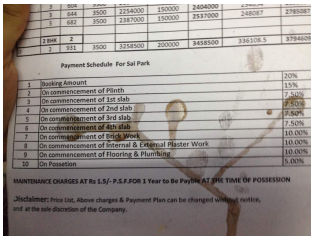
(b) Cleaned Image

Figure 18: Artificial Neural Network on Real World Image a)

(a) Original Image

(b) Cleaned Image

Figure 19: Artificial Neural Network on Real World Image b)



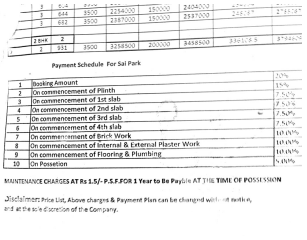
Payment Schedule For Sai Park

1	Booking Amount	20%
2	On commencement of Plinth	15%
3	On commencement of 1st slab	7.50%
4	On commencement of 2nd slab	7.50%
5	On commencement of 3rd slab	7.50%
6	On commencement of 4th slab	10.00%
7	On commencement of Brick Work	10.00%
8	On commencement of Internal & External Plaster Work	10.00%
9	On commencement of Flooring & Plumbing	10.00%
10	On Possetion	5.00%

MAINTENANCE CHARGES AT Rs 1.5/- P.S.F.FOR 1 Year to Be Payble AT THE TIME OF POSSESSION

Disclaimer: Price List, Above charges & Payment Plan can be changed without notice, and at the sole discretion of the Company.

(a) Original Image



Payment Schedule For Sai Park

1	Booking Amount	20%
2	On commencement of Plinth	15%
3	On commencement of 1st slab	7.50%
4	On commencement of 2nd slab	7.50%
5	On commencement of 3rd slab	7.50%
6	On commencement of 4th slab	10.00%
7	On commencement of Brick Work	10.00%
8	On commencement of Internal & External Plaster Work	10.00%
9	On commencement of Flooring & Plumbing	10.00%
10	On Possetion	5.00%

MAINTENANCE CHARGES AT Rs 1.5/- P.S.F.FOR 1 Year to Be Payble AT THE TIME OF POSSESSION

(b) Cleaned Image

Figure 20: Artificial Neural Network on Real World Image c)

6 Conclusion

Comparing the results of all the methods listed we find that *ANN* works the best. It removes the stains & crevices and it is readable!! While the other methods remove stains, the text is quite hard to decipher as it is blurred or the ink is too thin. The

RMSE values of the test data using our methods and the original image are listed in the table below:

Methods/Score	RMSE (%)
Fixed Thresholding	35.173%
Adaptive Thresholding	42.228%
Canny Edge (Dilation)	51.638%
Canny Edge (Erotion)	36.547%
Median Blur	55.096%
Random Forest Regressor	32.492%
Artificial Neural Network	3.363%

Table 1: Table with methods and their RMSE scores

7 Future Work

References

[1] Kaggle - denoising dirty documents. http://tinyurl.com/z4ukatx. Accessed: 2017-04-30.

[2] Glosser.ca. Artificial neural network. https://commons.wikimedia.org/w/index.php?curid=24913461. Accessed: 2017-05-10.

[3] Dokania S. Reddy V. Manoj R., Vats U. Denoising dirty documents, 2017.