Restoring eroded text in tourist plates images using Deep Learning*

[1] Jia Guo
Gabelli School of Business
Fordham Univeristy
New York, New York, USA
jguo81@fordham.edu

[2] Hassia Gado Alzouma
Computer and Information Science Department
Fordham Univeristy
New York, New York, USA
hgadoalzouma@fordham.edu

***Abstract*—Since the inventions of writing, history has been recorded through different inscribed texts such as manuscripts, plate inscriptions, stone inscriptions, etc. However, many of these texts have been destroyed overtime causing a huge information loss. The goal of this work is to apply deep learning techniques such as Convolutional Neural Network (CNN) and Natural Language processing (NLP) to predict and input eroded text in images of plate inscriptions. Our contribution is to use deep learning techniques to recover information that will be otherwise lost forever.**

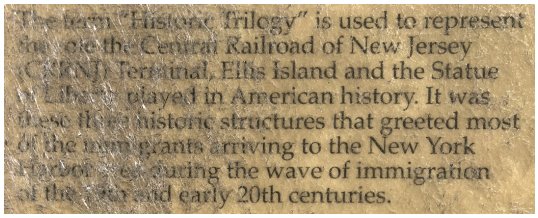
***keywords*—Natural Language Processing, Convolutional Neural Network, Deep Learning, text imputation, BERT**

I. Introduction

Many ancient history manuscripts have been degraded, spoiled, stained, torn or lost overtime. In such cases, we often found a large quantity of text and information missing in these manuscripts. In this paper, we will explore different deep learning methods that will help us recover the lost information. Our work is divided in two parts. First, we will use CNN techniques to extract the characters from the images and in the second part, we will use NLP to predict the missing text.

II. Data

Our Data is comprised of ancient plate inscriptions' images gathered from a New Jersey museum. As stated above, our work is divided in two parts: in the first part, We build a CNN model to extract the characters from the plates images. To achieve this goal, we will use the publicly available datasets "EMNIST" to train our model. Then, we will use our plate inscriptions images for testing and extraction of characters. In the second part, we use a pre trained NLP model to predict the missing text; we will use both a generated synthetic data and the text data extracted from the plates' images to test our model. As stated above, the plates' images contain eroded and/or stained text. Only a portion of the text can be read while the other portion is wiped out as shown in the picture below:



III. Problem Definition

Restoring text is a complex and daunting task. However, it is an important problem as it can help deal with large scale information reconstruction of lost heritage. It is important to recover lost information because it helps us understand better our history. History in return helps societies develop a better understanding of the world which can push more scientific discoveries in the future. Also, many organizations around the world still have their information stored in form of document papers; and these documents have been sometimes physically stained. With the advancing of new technologies, this information can be restored. Text restoration is also an important task in different fields such as speech transcription, autocorrection applications, social media, sentiment analysis, etc.

Our paper is divided in two parts. In the first part, we focus on recognizing and extracting the characters from the plates' images. In this section we will explore the different and recent works done in the character recognition field of computer vision. We will also explain our data acquisition, and methodology processes. Then we will discuss our experimental results. In the second part, we will concentrate on explaining our methodology as well as the data used to predict the missing text from the characters extracted from the plates' images.

IV. PART I

A. Related Work

Text recognition is a challenging task since the images usually suffer from background noise, blur, and/or distortion. Previously, methods for recognizing text have focused more on scanned documents but these methods are very limited as they are unable to efficiently deal with eroded and/or distorted text. In recent years, computer vision researchers

[1] Jia Guo is responsible for NLP (Part II) in this paper
[2] Hassia Gado Alzouma is responsible for CNN (Part I) in this paper

have been focusing more on text recognition in natural scenes. [Jaderberg et al., 2016] are among the first to propose a CNN based model for text recognition to classify words into a predefined set of words in the dictionary. The methods first detect each character using binarization or sliding window operation, then the characters are formed together and recognized as a word. More recent works have then followed which addressed Jaderberg's limitations. They generally do not require a dictionary and treat scene text recognition as a labeling problem where text is represented as a sequence of characters. [Zhou et al., 2017] proposed a fully-convolutional neural network adapted for text detection which is trained to directly predict the existence of text instances and their geometries from full images. This eliminated intermediate steps such as candidate proposal, text region formation and word partition. Similarly, [Yin et al. 2017] proposed an implicit segmentation method which overcomes the difficulty of character segmentation by sliding window and the underlying CNN character model can be learned end-to-end. [Ren et al. 2017] also proposed another CNN based model called the attention convolutional neural network. It is composed of an attention feature encoder, a convolutional sequence modeling module and a CTC module. Finally, [Borisyuk et al. 2019] from Facebook also proposed a system based on Faster-RCNN model. The system is responsible for detecting regions of the image that contain text. After that, a fully-convolutional character-based recognition model is used to process the detected locations and recognize the text they contain.

B. Data Acquisition and Preprocessing

For our training, we use the EMNIST publicly available dataset from the NIST Special Database 19. There are six different splits provided in this dataset but we will only use one which is the "EMNIST Letters". This split is comprised of 145,600 character images with 26 balanced classes. Each class represents a letter in the alphabet. For example, class 1 represents the letter "A", class 2 represents the letter "B", etc. Each image in the dataset is of size 28x28 and grayscale. Sample images from the EMNIST dataset is shown below:

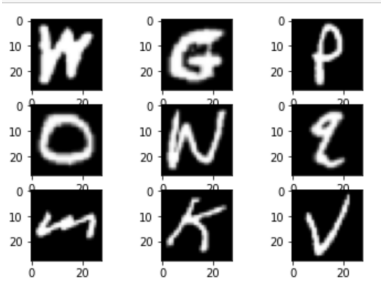


Fig. 1. Different letters from the EMNIST dataset

We used 100,000 images out of the 145,600 from the dataset for the training and validation.

We also set aside 20,000 images to test our model and predict how well it will do on images it has never seen before. We also test the model on character images from the plates. The testing process and results will be discussed later in another section.

C. Method

To train our data, we build a CNN model similar to the one described in [3]. We build a 15-layer CNN model. Our input shape is 28x28 which is the shape of each image in the EMNIST dataset. The filters of the convolutional layers are of size 3X3 and the stride is fixed to one. The feature map is increased from 50 to 400. Spatial pooling is after every 3 convolutional layers and succeeded with max pooling of stride 2. Each convolutional has a "relu" activation, a "uniform" kernel initializer and a padding "same". At the end of the model, the feature maps are flattened, followed by 2 fully-connected layers of 900 and 200 units respectfully. Finally, we have a SoftMax layer at the end of the model.

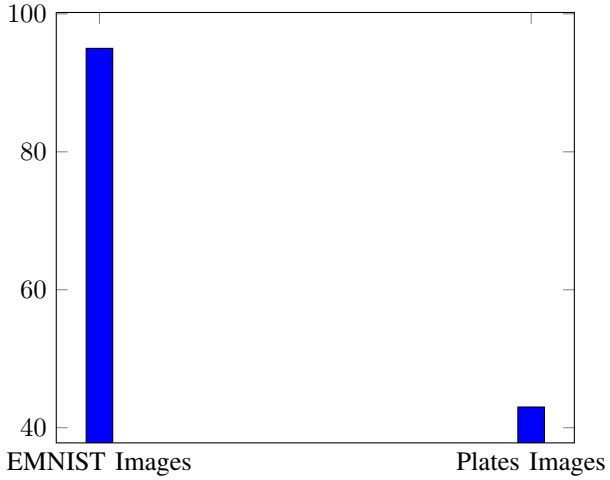
Type	Configuration
Conv2D	maps:50, k:3*3, s:1, p:1, dropout:0.0
Batch Normalization	
Conv2D	maps:100, k:3*3, s:1, p:1, dropout:0.1
Conv2D	maps:100, k:3*3, s:1, p:1, dropout:0.1
Batch Normalization	
Max-pooling	Windows:2*2, s:2
Conv2D	maps:150, k:3*3, s:1, p:1, dropout:0.2
Batch Normalization	
Conv2D	maps:200, k:3*3, s:1, p:1, dropout:0.2
Conv2D	maps:200, k:3*3, s:1, p:1, dropout:0.2
Batch Normalization	
Max-pooling	Windows:2*2, s:2
Conv2D	maps:250, k:3*3, s:1, p:1, dropout:0.2
Batch Normalization	
Conv2D	maps:300, k:3*3, s:1, p:1, dropout:0.3
Conv2D	maps:300, k:3*3, s:1, p:1, dropout:0.3
Batch Normalization	
Max-pooling	Windows:2*2, s:2
Conv2D	maps:350, k:3*3, s:1, p:1, dropout:0.4
Batch Normalization	
Conv2D	maps:400, k:3*3, s:1, p:1, dropout:0.4
Conv2D	maps:400, k:3*3, s:1, p:1, dropout:0.4
Batch Normalization	
Max-pooling	Windows:2*2, s:2
Fully Connection	hidden Units:900, dropout:0.5
Fully Connection	hidden Units:200, dropout:0.0
Softmax	Output Units:27

D. Experimental Results

1) Results on EMNIST Dataset: As briefly mentioned above, we set aside 20,000 images from the EMNIST dataset to test our model. These images are first preprocessed before

being fed into the model. Using the model.evaluate() function in Keras, we got an accuracy of 95% which proves that our model does well in recognizing letters.

2) Results on plates character images: To predict each character in the plates images, we first built a sliding window based python program where each window contains a specific character. The original window size is 32x32x3 but it is resized to 28x28 to meet our input shape requirements. We then feed the window content to the model for the letter prediction. Unfortunately most of the characters could not be recognized by the model and since the model is not doing better than random guess, we decided to not use its results in the second part of our problem. Instead we used online Optical Character Recognition software to extract the text from the plates images. The histogram below shows the performance of the model on the EMNIST dataset vs plates images



V. PART II

A. Methodology

In this second section, we try to predict the missing (eroded) text from the plates inscriptions images. We are using a modification of the BERT model developed by Google AI in 2019 (Devlin et al., 2019). BERT has a natural advantage to recover missing words using their surrounding words as hints. For the purpose of this paper, we made some modifications to the softmax layer of the model to improve our accuracy.

1) Introduction to BERT model: Since its implementation, BERT has become the new state-of-the-art model in the Natural Language Processing field. It is a pre-trained model, which means it can be applied to multiple language tasks such as document classification and question answering. Thanks to its excellent performance, it has made new records on 11 NLP tasks from the time when it was released.

The architecture of BERT is based on a multi-layer bidirectional Transformer encoder (Vaswani et al., 2017). In more details, there are two types of BERT model. The first one is BERT Base where the number of Transformer blocks, hidden size, and number of self-attention heads are 12, 768, and 12 respectively. The second one is the BERT Large which includes 24 Transformer blocks, 1024 hidden states and 16 self-attention heads. Also, the word embedding list for BERT has 30,000 tokens, and the maximum length of input sentence should be 512 tokens.

BERT has two main unsupervised pre-training tasks. Masked Language Model is the first one and it is used to predict a masked (missing) word in a sentence through deep bi-directional architecture. The second task is Next Sentence Prediction. It is based on two input sentences, and the model will predict whether one sentence is immediately after another one. Depending on the task at hand, BERT's parameters are updated effectively.

For this paper, BertTokenizer, BertModel and BertForMaskedLM are imported from pytorch-pretrained-bert. Further modifications would be conducted on the bert-large-uncased extracted from Masked Language Model as explained below.

2) Modification on BERT: BERT is a word-based model. After the two pre-training tasks mentioned above, a list of 30,000 words is generated. Using the Masked Language Model as the downstream task, BERT can predict a masked word (unknown word) in a sentence using the words before and after it as clues (hints). However, BERT does not use any information about the masked word itself as clues to make its prediction. Yet, in many real world scenarios such as ours, some eroded words are not completely wiped out. We can still extract some information such as few letters left in the word and/or its length. Using this information and combining both the word length and the letters left out as hints in the Masked Language Model, the prediction accuracy can improve.

In the Masked Language Model, the last layer is a softmax, which will output a predicted word with maximum probabilities. Given some letters in the masked word and its length as screening conditions, if the predicted word with the maximum probability does not satisfy those screening conditions, we will compare the word with the second maximum probability with the hints and verify if the conditions are matched. In an iterative loop, if any word meets our given conditions, that word will be the final output at this masked location in the sentence. Based on this modification on the softmax layer, a more accurate prediction is generated.

B. Data

1) Synthetic Data: In order to justify the performance of our modified version of BERT, it is necessary to test it at scale. As the texts on the plates are not domain specific, the 20 newsgroups text dataset in Scikit-Learn is introduced here, which includes 18,000 short news. For each news, two words are masked and substituted as [MASK] sign randomly, and our modified BERT model is used to predict the words at the places of the [MASK] sign. When taking the letter clues into consideration, two or three random letters are used as hints in the softmax layer.

For the synthetic data generated from the 20 newsgroups text dataset, the modified BERT is tested in two scales. The first scale has 1,911 masked words (Figure 1), and another

one has 5705 masked words (Figure 2). Meanwhile, each test is divided into two parts. When predicting the masked words, the clues about the letters and the word length will be taken into consideration for fifty percent of samples in one part (the columns with the top as "50%" in Figure 1 and Figure 2), and the clues will be used for all the samples in another part (the columns with the top as "100%" in Figure 1 and Figure 2). In terms of letter clue, if too many letters (more than three) in a masked word are known, it is then easy to quickly predict the word. However, if only one letter is used as a clue, the prediction becomes harder. Thus, to make the situation close to reality, a synthetic masked word is only allowed to have two or three letters as a letter clue (hint).

2) Authentic Data: For the sake of applying the modified BERT into a realistic scenario, nine pates images are used. Since the texts on those plates are eroded to some extent, the eroded parts are marked as [MASK] like in the synthetic data. In addition, not all the eroded words are fully distorted, thus some hints can be extracted. In this case, if some letters and the length of eroded words can be recognized, we will be used as the clues for prediction.

C. Experimental Results

1) Synthetic Data: Figure 1 and Figure 2, show the prediction accuracy for 1,911 and 5,705 masked words respectively from the synthetic data using the modified BERT. Firstly, regarding the word length as clues, it is evident that word length could improve prediction accuracy to a large extent. When there is no letter clue applied in the prediction, the accuracy rate only increases from around 40% to 57%. This shows that other than the word length, the letter clue in a masked word plays a more important role. Comparing with employing no letter clue, the accuracy rate climbs to about 67% when two or three letter clues are used. If we combine both the letter clue and the word length, the prediction accuracy is between 70% and 73%, which proves the efficacy of the modified BERT. However, there is no obvious difference when applying two or three letter clues. The reason may be that two letter clues may be just enough and ideal for the model. If with two letter clues, the predicted word is incorrect, adding a third letter clue will not improve the accuracy, thus the model will still not predict the correct word. In addition, the length of the token list in BERT is only about 30,000, which concludes that some uncommon words are not included in the list. In this case, no matter how many letter clues are used, the modified BERT will not be able to predict the correct word if it is among those uncommon words.

Figure 1: Prediction results for 1,911 masked words

1911 Words	without word length		with word length	
	50%	100%	50%	100%
0 letter	0.4019	0.4019	0.5683	0.5683
2 letters	0.5358	0.6719	0.5620	0.7211
3 letters	0.5374	0.6677	0.5515	0.7007

Figure 1

Figure 2: Prediction results for 5,705 masked words

5705 Words	without word length		with word length	
	50%	100%	50%	100%
0 letter	0.4040	0.4040	0.5767	0.5767
2 letters	0.5401	0.6806	0.5648	0.7306
3 letters	0.5495	0.6738	0.5571	0.7015

Figure 2

2) Authentic Data: After justifying the effectiveness of the modified BERT model on synthetic data, 55 masked words on nine plate images are used in the model to recover the eroded words. All the hints of the eroded words are extracted from the plates based on their situation. When not considering any hint of the masked word, the prediction accuracy is only 45%. However, this result goes up to 67% when considering only the word length as clue. Interestingly, if only the letter clues are used, the result would be 76%. Finally taking both the world length and the letters as hints, the final accuracy rate reaches 81%. Two eroded plates and their prediction results are presented below. For the image plate 1 (Figure 3), it is eroded to some extent, and "represent", "New", "Jersey", "most", "immigrants" and "immigration" are the eroded words that should be recovered (Figure 4). Comparing the prediction results given by the original BERT and the modified BERT, which takes the word clues into consideration, the modified predicts three more words than the original BERT – "describe", "America" and "many".

Figure 3: Real plate 1

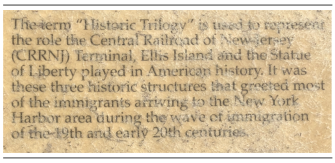


Figure 3

Figure 4: Prediction results on real plate 1

Eroded word	Predicted word by BERT	Predicted word with modified BERT
represent	describe	represent
New	america	america
Jersey	america	jersey
most	many	most
immigrants	immigrants	immigrants
immigration	immigration	immigration

Figure 4

For the image plate 2 (Figure 5), if using the original BERT and not using any information about the masked word as hint, three words are predicted incorrectly (Figure 6). However, two of these words are predicted correctly after using information about the masked word as clues into the modified BERT. It is therefore apparent that the modified BERT with hints about the masked word is able to improve prediction results significantly.

Figure 5: Real plate 2

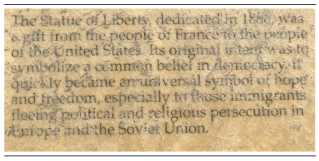


Figure 5

Figure 6: Prediction results on real plate 2

Eroded word	Predicted word by BERT	Predicted word with modified BERT
Liberty	liberty	liberty
dedicated	erected	erected
gift	gift	gift
people	people	people
intent	purpose	intent
belief	belief	belief
universal	important	universal
political	political	political
Europe	europe	europe

Figure 6

VI. Conclusion

In this paper, we first attempted to extract characters from plates images. We proposed a text recognition model based on previous work done in the field. The model was trained on a publicly available character dataset. The experiments were done on both the EMNIST dataset images and our plates images. However, even though the model achieved a good accuracy on the EMNIST dataset, it could not do better than random guess when tested on plates images. For this reason, it was better to use public Optical Character Recognition software online to extract the characters in order to proceed to the second part of our work. In the second and part of the paper, we investigated a modified version of the BERT model that predicts eroded text in our plates images. The proposed method achieves superior performance for our task comparing to the original BERT model.

In the future, we will train our CNN model on more challenging datasets. We will add more sophisticated preprocessing methods such as histogram matching and Gaussian noise.

REFERENCES

M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman. Reading text in the wild with convolutional neural networks. IJCV, 2016.

X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, and J. Liang. EAST: an efficient and accurate scene text detector. In CVPR, 2017.

F. Yin, Y. Wu, X. Zhang, and C. Liu. Scene text recognition with sliding convolutional character models. CoRR, abs/1709.01727, 2017.

S. Ren, K. He, R. Girshick, and J. Sun: Towards real-time object detection with region proposal networks. TPAMI, 2017.

F. Borisyuk, A. Gordo, V. Sivakumar: Rosetta: Large scale system for text detection and recognition in images. TPAMI, 2019.

J. Devlin, M. Chang, K. Lee, K. Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Google AI Language, 2019.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, L. Kaiser, and I. Polosukhin.. Attention is all you need. In Advances in Neural Information Processing Systems, pages 6000–6010, 2017.