Removing logos and stamp seals from images via deep network architectures

Hangyu Tian
ht2459@columbia.edu

Siyao Zhang
sz2963@columbia.edu

1 Introduction

Optical character recognition(OCR) is one of the earliest addressed computer vision task before the deep learning field rising, which makes people view it as a "solved" task. However, OCR technique yields acceptable results on some scenarios, the generalization of such technique is not as expected. The appearing of various types of logos and stamp seals on the document undesirably impacts the OCR result. Hence, effective methods for removing logos and stamp seals are needed urgently for further OCR performance improvement. The challenge underlying such improvement is the structure and color of the logos and stamp seals are embedded into the text. The task to simultaneously remove seals and preserve the informative text clearly is extraordinarily difficult and challenging.

To accomplish such problem, we implement and compare several methods for logos and stamp seals removal on both colored and gray-scale images. The gray-scale images are significantly harder comparing to colored images when the models are detecting logos and stamp seals from the text, especially using the classical machine learning segmentation method, such as KMeans and DBSCAN. Both KMeans and DBCAN require a prior knowledge on the color of logos, stamp seals and text for further filtering the text out after clustering, which can not be utilized on gray-scale images.

In this paper, we approach the task with a few methods including traditional machine learning, convolution neural network and residual attention network on 10,000 synthetic images and compare the results and running time on 12 real world test data.

1.1 Related Work

In[1], this paper proposed a new deep network architecture for removing rain streaks from individual images based on the convolution neural network.

- it used a lossless "negative residual mapping" in this paper to avoid gradient vanish, which was motivated by the observation that the residual of the rainy image **Y-X** has a significantly narrow range in pixel values. This skip connection can also propagate lossless information through the entire network. As a result, both color and object was preserved better by simply embedding the neg-mapping without the ResNet.
- instead of directly applying ResNet on the image, they exploited a prior knowledge by focusing on high frequency pixel value and used detail layer as the input. After removing the interference of background, the detail layer was more sparse than the image since most pixel values were zero.

In [2],they present a residual learning framework to ease the training of deeper networks by reformulating the layers as learning residual functions with reference to the layer inputs.

In [3], they propose “Residual Attention Network”, a convolution neural network using attention mechanism, achieved state-of-art in segmentation in complex tasks and attention residual learning to train very deep Residual Attention Networks.

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

1.2 Our Contributions

Detecting a seal and removing it on text is challenging, especially in gray-scale images. When images are of gray scale, the range of pixel values of seals are similar with text and background.

In this paper, we compare several methods for logo removing and evaluate them on the synthetic and real-world images. The contributions of this work are summarized as follows:

- We use the clustering methods to significantly reduce the number of features. By exploiting prior knowledge of seal color, we use a mask of right color range to extract the seal out, which is mainly inspired from face extraction problem.
- We propose a relatively simple convolution network architecture, simply stacking CONV-RELU layers, followed by dropout and batch normalization layers in order to keep the size of output image remain the same as the input image. We find that loss decreases epoch by epoch and the output changes epoch by epoch, indicating model learning progression. The model learns detects edge of all seals and text in first 10 epochs and gradually moves its attention from all text and seal to remove seals in the next 20 epochs. We also study the effect of different metrics for the network's loss layer for this stamp seal removal task. We compare l_2 loss against a state-of-art metric for image quality, structural similarity index(SSIM) and see which one can produce better results across training epochs.
- We adopt the attention residual learning for optimizing deep attention module layers that generate attention-aware features changing adaptively as layers are getting deeper, which leads to performance improvement. In this problem, we believe that the attention learning with appropriate design to avoid gradient vanishing will improve the result of stamp seal removing as the seal takes less than 10 percent of the image. Although the CNN of large parameter space can find position of stamp seals, it takes a lot of resource.

  Inside each attention module, a bottom-up and top-down feedforward structure with skip connection in between is used. This soft mask module has applied to segmentation and human pose detection and achieved great success on solving these complex problems. The bottom-up structure produces a low-resolution feature map and then the top-down structure produces a dense feature map. We believe the attention learning with soft mask module will improve the seal removal result. The background attention mask diminishes the background response while the seals attention mask highlights the seal out of the background (!! add picture later).
- To learn the network, we create and use a synthetic dataset of 10000 images of documents with seals on the text for training from 110 different colored seals and 200 different black-white seals by first flipping and rotating seals and then randomly remixing them with 2000 background text. We test on 12 real world images. These images are of the form (width, height, channel).(hopefully!!! Although the network is trained on synthetic document data, we find that it generalizes very well to real-world images.)

2 Methods

image classification This is a significantly challenging task to simultaneously remove logos and preserve text since the structure and color of logos is very similar that of text, especially after we reduce the spatial resolution to decrease the data size.

Therefore, before that, we first start with a simpler problem, image classification problem which aims to classify whether each image is clean or with logos. The input of our logo detection problem is a document image either with or without a logo on the text.

2.1 Traditional Machine Learning

In image processing, feature extraction has a wide range of application and many methods proposed to extract feature, among which clustering method is one of most popular, especially in facial extraction.Hence, we first start with clustering method, to separate the colored logos from black text in a color image.

First, we convert images from BGR to LAB form and then reshape images from (width, height, channel) to (width $\times$ height, channel). In image processing, it is important to reduce dimension by feature extraction. When the input image is of high resolution and is too large for processing, the input data is transformed into a reduced representative set of features which is expected to capture the most relevant information to resolve the task in a dimension-reduced form. Here, we use KMeans and DBSCAN clustering method to segment each pixel to its closest cluster and then exploit the color information of text to apply a mask on the dimension-reduced by manually setting a color range to extract text out.

However, success is less noticeable compared to their application on face extraction. As this is a color-based segmentation method, it requires a prior knowledge of logo and text color to select the right feature representative for logo and text, which does not generalize well in images with various background color and is not even possible to work on gray-scale images.As shown in Figure(insert failure result in KMeans and DBSCAN), this implies logos are not totally detected and removed by KMeans and DBSCAN clustering method.

2.2 Convolution Neural Network(CNN)

Since CNN has achieved state-of-art in many challenging problems and deeper network architecture increases the capability to explore and detect the more complex and detailed image characteristics, we also use CNN to distinguish the logo from text and remove it from training.

Architecture We define a convolution neural network(CNN) that takes a (reduce width$\times$ reduced height) resized input. Our convolution neural network mainly follows the most common form of a CNN architecture that stacks a few CONV-RELU layers following as batch normalization(BN) layer and repeats several times. It is commonly believed that batch normalization is a standardization technique that prevents gradient vanish for our deeper CNN. It can also accelerate the learning process, which is crucial for another CNN model with large kernel size.

2.3 Residual Attention Network

Trunk Branch Trunk Branch performs feature processing and is solely composed of a stack of Residual Units. the number of Residual Units stacked, t, is the hyper-parameter.

Soft Mask Module Soft Mask Branch takes in the same input as Trunk Branch, and extracts the location information of certain features with the bottom-up top-down structure. The bottom-up structure is achieved by applying max pooling several times with r number of residual units in between. After this, 2*r number of residual units are applied to perform feature identification. The top-down structure, on the other hand, uses Upsampling layers with r number of residual units in between to expand the feature information. The number of Upsampling layers is the same as that of max pooling to keep the output size the same as the input feature map. The skip connection is added between bottom-up and top-down structure to ensure sufficient gradient backflow and avoid vanishing gradient problems.Finally, one layer of 200x200 convolution are applied before the output layer.

Residual Unit In our base network, we use bottleneck style Residual Unit from ResNet [4] as our basic unit. The Batch Normalization is repeated three times in residual mapping to allow less dependency between each layer, therefore encouraging higher learning rates and mitigating over-fitting issues. In addition, in order to address the challenge we identified before, where residual units behave differently throughout network proposed by the paper, we introduce a new type of Residual Units: Residual Unit-Downsampling. Compared to the regular Residual Unit, a stride of 2x2 is introduced to the intermediate convolution layer instead of 1x1 to achieve the halving effect described by the proposed design description. The Residual Unit-Downsampling is only applied to the network where the halving effect is identified. Unless specified, all Residual Units mentioned from now on refer to the regular Residual Unit.

2.4 Implementation

Loss function Loss function is used as measurement of how good a prediction model does in terms of being able to predict the expected outcome.

Network 1 Residual Attention Network(Attention 56)

Input: Image
Output: Image

Stage 1

Input: Image Tensor, Filter, residual unit type
Processing: ▷ Output Size
1. Residual Unit ▷ 800×800
2. Attention Module A * 1 ▷ 800×800

end Processing:

Stage 2

Input: Image Tensor, Filter, residual unit type
Processing: ▷ Output Size
1. Residual Unit-Downsampling ▷ 400×400
2. Attention Module B * 1 ▷ 400×400

end Processing:

Stage 3

Input: Image Tensor, Filter, residual unit type
Processing: ▷ Output Size
1. Residual Unit-Downsampling ▷ 200×200
2. Attention Module C * 1 ▷ 200×200

end Processing:

Ending Phase

Input: Image Tensor, Filter, residual unit type
Processing: ▷ Output Size
2. Residual Unit * 2 ▷ 200×200
3. Convolution Layer ▷ 200×200

end Processing:

Figure 1: RAN Architecture

By default, we use l_2 loss, the most commonly-used loss function in regression and then realize that l_2 loss suffers from an obvious limitation that it correlates poorly with image quality as perceived by us, as a human. For example, as the epochs increase, the loss is decreasing, however, the text on the predicted image is blurred, which makes us feel the result is getting worse. The perception is not positively correlated with the l_2 loss evaluation, therefore we seek the metric measure consistent with our human visual system(HSV) and use structural similarity index(SSIM) as metric to measure the same convolution network as l_2 norm.

We use two loss functions mentioned above on the same convolution network and compare the predicted results. Surprising, on the task we consider, l_2 outperform the SSIM, perceptually-motivated metric. The input of our seal removal problem is a document image **X** with seals on the text and the output is the clean image **Y**. The two loss function is defined as follows :

- The objective function for l_2 loss is defined as

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N} ||f(X_i; W, b) - Y_i||_F^2 \tag{1}$$

  where N is the number of training images, $f(\cdot)$ is the convolution neural network and W, b are training weights and biases.

- SSIM for pixel p is defined as

$$\begin{aligned} SSIM(p) &= \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \cdot \frac{2\sigma_{xy} + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \\ &= l(p) \cdot cs(p) \end{aligned} \tag{2}$$

The loss function for SSIM can written as

$$\mathcal{E}(p) = 1 - SSIM(p)$$

. The objective function for SSIM is

$$\mathcal{L}^{SSIM}(P) = \frac{1}{N}\sum_{p \in P} 1 - SSIM(p) \tag{3}$$

Training We resize the original input image to a fixed size $width_{reduced} \times height_{reduced}$ where $width_{reduced}$, $height_{reduced} \in \{32, 64, 128, 256\}$ to make sure the input of CNN model is of the fixed size. Another reason is that there is no sufficient computational resources to directly training on the original high-resolution images. We use Adam to minimizing the objective function in Equations 1 and 3.

3 Results

We first started from several classical machine learning methods, eg. KMeans, Meanshift and DBSCAN. There are several drawbacks of the classical machine methods.

We first tried several classical machine learning algorithms, eg. KMeans and DBSCAN.

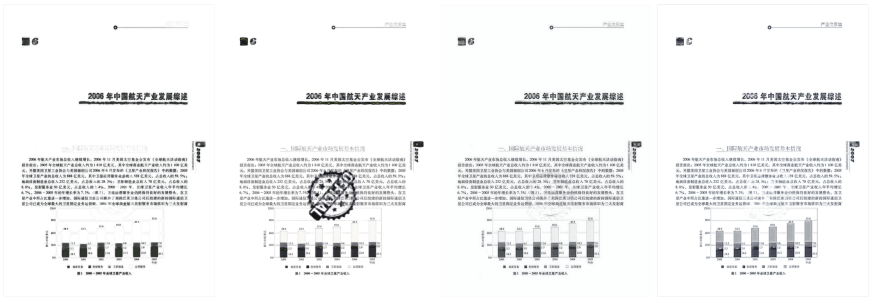
- KMeans turns out to be the most time-efficient method with relatively good performance. However, priori knowledge such as whether the logo is on a gray-scale image, whether the input image has background pattern, etc. is required to obtain such result, which means that KMeans clustering does not generalize well from images to images.
- DBSCAN is a density-based method, which is optimized to find somewhat amorphous shapes in image that may not necessarily have a single clear centroid. It is expected to perform better in colored image than KMeans, however, it does not. One point worth noting is that the DBSCAN algorithm implemented in scikit-learn has a limited memory that it does not perform well on large-size images. Therefore, we resize the input image from its original size around 1200×1700 to size 32×32, losing plenty of information, which is probably the main reason that it does not outperform KMeans clustering as we does not resize the input for KMeans.

3.1 The effect of loss function

The loss function is a measure of how good a prediction model does in terms of predict the expected outcomes. There is not a single loss function that can work perfectly in every scenario. The impact of the loss function on the performance of neural networks did not attract much attention at beginning. We replace l_2 loss by the SSIM, simply because we observe that it is hard for us as a human to track the model improvement indicated by l_2 loss as the number of epochs increases. We expected the SSIM would perform better in capturing the detailed differences such as a stamp seal on the text, while it does not. The prediction results of CNN model using the two different loss functions are shown as Figures 2 and 3.(add more description).

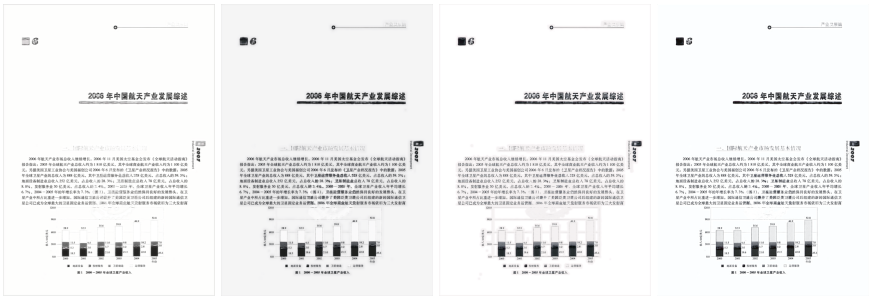
After researching on l_2 and SSIM, we found that this seal removal task exactly follows the assumption of l_2 that the impact of noise is independent of the local characteristics of the image. In our case, stamp seals added on text are independent of the local characteristic of the clean image.

- Running Time: The running time of the same CNN model using l_2 and SSIM takes 218s for one epoch.
- Prediction Outcome: SSIM prediction outcome for a medium difficulty level test image is more stable over epochs whereas l_2 prediction outcome gives more clear detail of the text with the stamp seals on the input image, shown in Figure 2 and 3



(a) epoch 5 (b) epoch 10 (c) epoch 15 (d) epoch 20

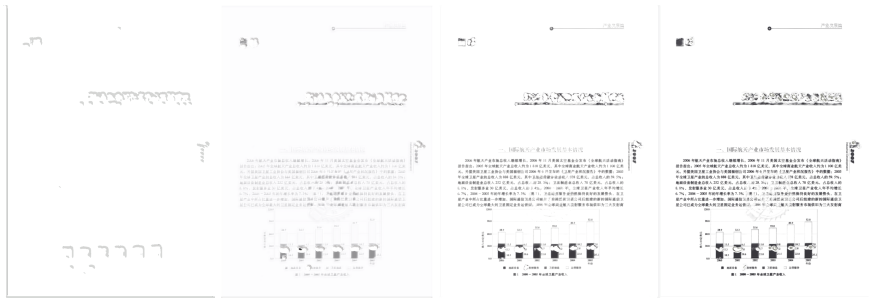
Figure 2: l_2



(a) epoch 5 (b) epoch 10 (c) epoch 15 (d) epoch 20

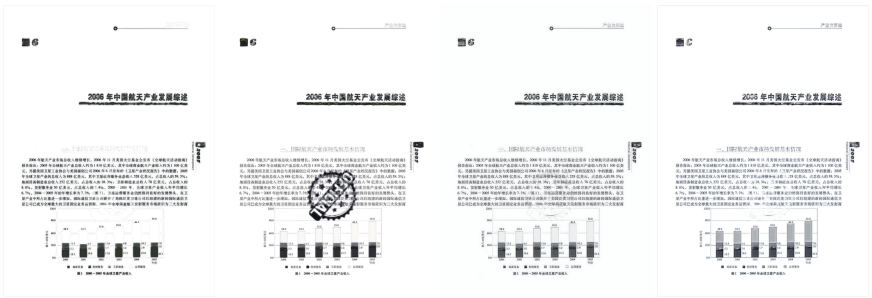
Figure 3: SSIM

3.2 Depth VS Breadth



(a) epoch 5 (b) epoch 10 (c) epoch 15 (d) epoch 20

Figure 4: Depth



(a) epoch 5 (b) epoch 10 (c) epoch 15 (d) epoch 20

Figure 5: Breadth

There are two ways to increase the parameter capacity: one is to increase the depth of the structure by stacking more layers and the other is to increase the breath of the network by using larger kernel size. In our task, we train the depth and breadth on 1500 resized training data of resolution 64×64 and test on 200 test data for 20 epochs. From Figure4 and 5, Increasing the kernel size performs better result on the first 20 epochs, which is supported by [5] that deeper structure does not work well on low-level image task. The performance of deeper network may also impact by gradient vanishing as batch normalization does not perform a strict regularization.

3.3 Residual Attention Network

Although we expected that Residual Attention Network would perform better than the deeper CNN with help of attention module and residual mechanism. However, we are limited by the computing resources. The network contains over 350 layers and 4.2 million parameters, it takes in a full size image 800x800 in most of the cases. The batch size has to be less than 2 while training the model, each epoch consumes over 350 seconds. We initialized the model with different optimizer like Adam and SGD, the default learning rate is set to 1e-1, decay 1e-4 with momentum 0.9. The training process takes over 300 epochs to achieve an acceptable loss, we are limited by the computational power and time cost while implementing Residual Attention Network on this specific task.

4 Discussion

We begin this stamp seal removal task by classical clustering algorithm, a color-based segmentation method, which does not even work on gray-scale images. Our goal is to remove the stamp seals on both color and gray-scale images. Although CNN performs good prediction result at some epoch, it varies a lot by difficulty of removing it and the number of epochs. We believe that this is because a simple CNN structure does not have a sufficiently smart strategy during learning to detect and remove a small difference,taking less than 10 percent in an image. RAN is chosen after researching because of the ResNet's ability of preventing gradient vanishing and stopping learning for deeper network and attention module's good performance on segmentation on complex task, however, it does not provide with nice prediction result with the restricted computational resources we had, even though the loss keeps decreasing at a stable rate.

References

[1] X. Fu, J. Huang, D. Zeng, Y. Huang, X. Ding and J. Paisley, "Removing Rain from Single Images via a Deep Detail Network," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, 2017, pp. 1715-1723, doi: 10.1109/CVPR.2017.186.

[2] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90.

[3] F. Wang et al., "Residual Attention Network for Image Classification," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, 2017, pp. 6450-6458, doi: 10.1109/CVPR.2017.683.

[4] H. Zhao, O. Gallo, I. Frosio and J. Kautz, "Loss Functions for Image Restoration With Neural Networks," in IEEE Transactions on Computational Imaging, vol. 3, no. 1, pp. 47-57, March 2017, doi: 10.1109/TCI.2016.2644865.

[5] C. Dong, C. L. Chen, K. He, and X. Tang. Image super-resolution using deep convolutional networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 38(2):295–307, 2016.