Single Character Instance Detection in Cartographic Maps

Shishir Jakati
University of Massachusetts, Amherst
sjakati@umass.edu

04/22/2019

Project Advisor: Professor Erik Learned-Miller
Second Committee Member: Professor Liangliang Cao
Research Type: Project

1 Introduction

1.1 Project Description

Scene text detection is an actively developing field of research within computer vision, and within this field is the study of cartographic maps. My project goal, specifically, is to work directly with Prof. Erik Learned-Miller, Prof. Cao and Ph.D Student Archan Ray to improve the bounding box accuracy on individual character recognition predictions within cartographic maps. Major difficulties in text detection of maps arise when considering the various distractors that appear within maps themselves. Some of these distractors include bold geographical borders, rivers, lakes, train tracks, overlapping annotations, and even geographical features. These distractors make it particularly difficult for neural network detectors to discern between characters, and thus reduce the performance of detectors.

Word detections in neural networks can either be trained through understanding of words themselves, or through the understanding of their individual components. Characters play a large role as building blocks of words, especially in the context of scene text recognition. My contribution to Archan Ray's Map Text Detection project will useful in generating probabilistic indicators, which may then be compounded into sequential word detectors. Plainly, by preprocessing data using my proposed character detector, subsequent word detectors will have an augmented layer of input which may be used to implicitly verify whether or not a text detection is valid or invalid.

1.2 Objectives

Scene text problems may be generally categorized into two categories: detection or recognition. Pixelwise, detectors are used to classify regions of pixels as text or non-text. These detections are then generally converted to bounding boxes, or other actionable outputs, as results of detection. Recognition problems deal with the classification of text into different words or characters. Recognition problems usually result in strings of characters, or lexicon subsets, as results of recognition.

A major objective of this project is to utilize an existing scene text detector as a character instance detector. The training procedure which we have followed thus far seeks to solve problem of detection, while obtaining recognition for free. My detector can, generally, be described as a set of detectors which produce disjoint sets of detections for each character in our predefined alphabet. In this way, we attempt to solve both problems of detection and recognition. Our hypothesis being that single character text detection can be enhanced using a set of detectors which have their own inductive bias inclined to detect a specified character. A corollary to this main hypothesis is that these singular character detectors additionally provide a single character recognition pipeline, implicitly. To explore these hypotheses, I am utilizing the PixelLink Scene Text framework.

A tangential objective is to explore the Dilated Residual Network (DRN) framework. In short, DRNs provide the sought-after property of lossless spatial acuity. A major issue that arises when using convolutional neural networks is the constriction, and subsequent loss, of input resolution. Utilizing DRNs as a solution to this specific problem seeks to solve the resolution problem on maps and is wholly relevant for both false positive detection and true negative detections. True negative detections on maps may stem from various distractors, a pertinent example being a false detection of the letter "I" where a river is drawn. Similarly, a false positive detection may not be detected due to overlapping annotations. By dilating the network, a hypothetical detector should be able to accrue more contextual information and consequently detect text instances with higher performance.

1.3 Significance

Scene text, as a specific problem within computer vision, has its own facets and subproblems. Map text recognition being a significant problem in this space. Map text recognition can be regarded as having its own uncontrolled variables such as occlusion and text sparsity. In having a detector which is able to perform low-level character detections, a post-processing detector may be able to probabilistically make decisions based on those low-level character detections. Thus, improving the character detector's performance is an immediate step forward in generating word detections, and recognitions, with higher accuracy.

Utilizing PixelLink as a scene text recognizer enables the production of high accuracy detection results immediately. PixelLink, though, is not purpose built as a character recognizer. Exploring the use of its segmentation and connected

component framework is significant in testing our original hypothesis that a text detector may be taught the corrective inductive bias such that it is able to discern the correct character instances. The training procedure, in which we prepare a dataset of small character instances, is significant due to the relative size of character instances when compared to word instances.

Exploring the Dilated Residual Network framework is significant due to the fact that a discriminative model with more nuanced spatial understanding may be useful for text detection.

2 Background

2.1 Overview

Scene text detection and recognition aim to automate the spotting and reading of text from natural images [1]. Thus, a major goal of research into the subject is focused on increasing accuracy and robustness to the set of challenges which arise in natural images. One factor, constant in most all natural scene images, is the noise and adversarial distractors which make it difficult for detectors and recognizers to operate. Datasets continue to grow in volume, containing labeled text instance within natural images originating from various sources [2, 3, 4]. These vast scores of data have led way to the use of deep learning techniques, which excel when provided extensive training examples.

Deep learning, as a supervised learning technique, is a data hungry methodology used to learn the complex representation of words in images. The use of deep learning, and consequently deep neural networks, is integral to the advancement of scene text detection and recognition [1]. Feature engineering is largely a problem handled by larger deep learning architectures, and thus allows for simpler methodologies and frameworks to be employed.

As neural network architectures continue to grow deeper, though, difficulty arises in form of the vanishing gradients and loss of resolution. The optimization of most neural network architectures is vulnerable to the degradation of gradients [5]. [5] aims to combat the degradation problem through the use of referral context. This alleviates the difficulties of training larger networks and allows for a deeper understanding of smaller features as they relate to the spatial structure of the input. Deeper architectures also exhibit the problem of a narrowed field of view [6]. The use of dilated layers aids in increasing the final resolution which a network operates on. In doing so, dilated layers are able to capture larger patches of features through an enlarged receptive field.

2.2 Datasets

Images are rich with aspect ratios, colors, and other various features; convolutional neural network models have achieved state of the art results in problems such as classification and segmentation by learning complex representations of these features. Imagenet [7] is a widely used benchmarking dataset due to its

vast size and large number of classification categories. The images are labeled according to the subject matter located within the images themselves. Significantly, many networks use backbones which were pre-trained on the ImageNet dataset [8, 9, 10]. While a subset of the images may contain distractors, most are focused on the label which they intend to represent.

Natural scene images contain a multitude of distractors which may hinder the effectiveness of any detection model. Thus, it is necessary to develop a large collection of text instances within natural scene images that may be used in a supervised training framework. A major dataset used as a benchmark in various models is the ICDAR 2015 text detection challenge [3]. These scene images are produced using Google glasses. The ICDAR challenge dataset is classified as an incidental scene text challenge, one where there is prior preparation regarding the orientation, quality, location, or clarity of the text within the image. Applications in wearable technology and large-scale captures were the primary focus in the creation of this dataset. Similarly, but with less frequency, the ICDAR 2013 text detection challenge [2] is used as a benchmark.

Words have the physical property of being long and narrow. Especially long when they are considered in terms of aspect ratios. Thus, a dataset comprised of text instances which are long is necessary to understand elongated text instances. The MSRA-TD500 [4] dataset is comprised of English and Chinese multi-oriented words.

2.3 Object Detection

In the following section we describe an object detector, trained to identify object instances within scene images. The study of object detection generalizes to the study of scene text detection due to the incidental nature of object and text instance respectively. Models need to be trained to produce localization results for the various entities which appear in the images they are predicting on.

2.3.1 Faster R-CNN

Faster R-CNN approaches the problem through the use of region proposal networks (RPNs) [9]. The main purpose of RPNs is to produce a collection of bounding boxes which presumably identify an object present within the image. To do this, the RPN is trained as a classifier, and doubly as a regressor, which operates on the feature map produced by the initial convolutional layers. By allowing RPNs share layers with the initial convolutional layers, computational overhead is reduced and object detection accuracy is increased. An integral component of RPN's is the use of translation invariant anchors. These anchors allow for objects to be detected in a multitude of image scenarios. In composing the different sibling components of the network, the authors of [9] provide an adaptable foundation for different object detection networks to be married to the novel RPNs.

Certain downsides in using the Faster RCNN, though, emerge initially. Mainly, the process of training can be complicated. To mitigate the problem of separate convergence between the RPN and detection network, Faster R-CNN is trained using an alternating training process. First the RPN is trained, which then produces proposals to train the detection network. This network is used to initialize the RPN and the process is continually repeated.

2.4 Scene Text Detection

Object detection frameworks have also been shown to work well as text detection methods [9, 11], however, specialized scene text detectors are able to generalize to natural images with greater performance [8, 12, 10]. One main objective of scene text detection methods is to detect instances of text using bounding boxes. Thus, many scene text detectors not only act as localization networks but also as bounding box regressors .

2.4.1 EAST

EAST [10] proposes an adaptation to the network presented in [13]. [10] outputs two robust geometries to be used in detection: rotated bounding boxes and arbitrarily oriented quadrangles. All aggregated geometries produced by the network are subsequently sent through a non-max suppression procedure to yield final outputs. The adaptable framework presented in [10] is able to produce word level or line level predictions, which significantly outperform previous methods.

A discussion of the novel loss functions used in [10] is presented here, as they the loss functions may be useful in producing geometries from segmentation maps. The loss functions used in [10] take into account the practical considerations associated with problems in scene text detection; specifically, the ratio of non-text pixels to text pixels.

We must first define the two separate classes of geometries: Rotated Bounding Boxes (RBOX) and Arbitrarily Oriented Quadrangles (QUAD). Axis Aligned Bounding Boxes (AABB), are used in the creation of the Rotated Boxes. They are defined as such

$$AABB = R = \{d_i | i \in \{1, 2, 3, 4\})\}; RBOX = \{R, \theta\}$$

Here θ is the angle of rotation, and d_i are the distances from the top left pixel location. It is easy to see that the AABB and RBOX are represented by 4 and 5 channels respectively. The other outputted geometry is a quadrangle, QUAD. It is are defined as such

$$QUAD = Q = \{(\Delta x_i, \Delta y_i) | i \in \{1, 2, 3, 4\}\}$$

Here each point $(\Delta x_i, \Delta y_i)$ represents the coordinate shift from the top left pixel location. Since there are 4 such points, there are a total of 8 channels in each quadrangle $\mathbf{Q}$.

We may now define the loss functions as follows:

$$L = L_s + \lambda_g L_g$$

where L_s is the score map loss, and L_g is the geometry loss. In this case the λ_g constant represents the importance between the two losses. The loss on the score map is simply a balanced cross-entropy term used to mitigate any bias the network may develop in terms of negative classification of text regions. The loss term is defined as follows

$$L_s = -\beta Y^* log(\hat{Y}) - (1-\beta)(1-Y^*) log(1-\hat{Y})$$

where

$$\beta = 1 - \frac{\sum_{y^* \in Y^*} y^*}{|Y^*|}$$

In this case, $\hat{Y}$ are the predicted text/non-text labels, and Y^* are the ground truth text/non-text labels. Using balanced cross entropy is a common theme shared between multiple scene text detection schemes, thus this technique may be useful in training a sufficient detector when image resolution and text instance surface area is less than ideal.

The loss on geometries is more involved. Due to the nature of scene text, the loss function itself my balance large and small text instances. For this reason, the geometry loss is defined to be scale-invariant. Additionally, both the rotated textboxes and quadrangles have separate losses.

Rotated Textbox The rotated textbox loss is defined as follows

$$L_g = L_{AABB} + \lambda_\theta L_\theta$$

Intuitively, the loss function is composed of the loss on axis-aligned bounding boxes and the loss on the angle of rotation. By practical consideration, there is a constant to maintain how important the rotation loss is, this was set to 1 in the original experiments. The loss on the axis-aligned bounding boxes is simply a negative log functin of the Intersection Over Union (IoU) of box predictions. It is defined as follows

$$L_{AABB} = -log(|\frac{|\hat{R} \cap R^*|}{|\hat{R} \cup R^*|})$$

Here R^* is the predicted AABB geometry and $\hat{R}$ is the corresponding ground-truth label. The specific implementation of the IoU on bounding boxes is that the width and height of the intersected rectangle is computed and used as the numerator of the apparent fraction, while the sum of the two bounding boxes respectively less the intersection is used as the denominator.

The loss on rotation angles is quite simple, and is defined as follows

$$L_\theta = 1 - cos(\hat{\theta} - \theta^*)$$

Here $\hat{\theta}$ is simply the predicted angle of rotation, and θ^* is the corresponding ground truth rotation angle.

Quadrangles The loss on quadrangles is simply a smoothed L1 term with a normalization constant which accounts for the fact that text detections generally have one side with a dominating length. It is defined as follows

$$L_g = L_{QUAD}(\hat{Q}, Q^*)$$

and

$$L_{QUAD} = \min_{\hat{Q} \in P_{Q^*}} \sum_{c_i \in C_Q, \widetilde{c_i} \in C_{\widetilde{Q}}} \frac{smoothed_{L1}(c_i - \widetilde{c_i})}{8 \times N_{Q^*}}$$

Here, N_{Q^*} is the normalization term, which is the shorted edge length of the quadrangle. C_Q is the ordered set of coordinates of some quadrangle Q.

2.4.2 TextBoxes

Characters, and words to a larger extent, exhibit aspect ratios and surface areas which are not characteristic of other arbitrary object instances. Thus, engineering features such that they account for the longer aspect ratios attains better performance [12, 8, 10]. The TextBoxes [12] architecture attempts to solve this problem through the use of rectangular receptive fields, fully convolutional networks, and predefined aspect ratios better suited for word instance detections. Adapting the VGG-16 architecture [14] as a backbone, TextBoxes replaces the fully connected portions of the original architecture with convolutional layers and introduces text-box layers [12].

Text-box layers are inserted at intermediate layers to produce and regress text boxes for suspected word instances. Each location of an input to the network is encoded with default boxes of varying aspect ratios and horizontal offsets. The text-box layers produce offsets from each of these default input locations as well as a confidence level of an apparent text instance. These default boxes are engineered to deal with the varying shapes and aspect ratios exhibited by words in scene images. As such, they have the following aspect ratios: 1, 2, 3, 5, 7, 10. These horizontally dominant aspect ratios are further confirmed by the rectangular receptive fields used by the convolutional layers in [12].

2.4.3 PixelLink

PixelLink [8] abandons the need for bounding box regression and approaches the problem of labeling text instances using instance segmentation. PixelLink utilizes a VGG [14] backbone for the text detection tasks and has two separate network headers for segmentation and localization respectively.

The first step in the PixelLink pipeline is to generate two separate feature maps from the input image. These two are generated from the two network headers: one for text/non-text prediction, the other for link prediction. The text prediction network is self-explanatory. The feature map is the activated output of locations the network believes there is a text instance. We must then established, for sake of further discussion, that each pixel has 8 neighbors. These are: left, left-down, left-up, right, right-down, up, down. Each of these

neighbors are useful in generating connected components between pixels. For every pixel that is predicted as positive, a link between it and its neighbors is created as being part of a connected component. One of the two network headers is specifically responsible for predicting these links. This linking process is useful for refining the original text/non-text prediction that is performed by the other network header. While text/non-text prediction is useful for generating larger instance segmentation, the pixel linking routine is useful in separating between different words.

Here we include discussion of the PixelLink training procedure. To account for the relative size of different text instances, the PixelLink training routine defines a central weight for each text instance. Given N text instances, all text instances are assigned the central total weight B_i. Within each text instance, every pixel is assigned the weight $w_i = B_i/S_i$, where S_i is the area of some text instance. B_i is trivially defined to be the total number of pixels within text instances divided by the number of separate text instances.

The training loss is thus defined as

$$L = \lambda L_{pixel} + L_{link}$$

The loss function is separated between the two tasks, text/non-text prediction and link prediction. The loss function for text/non-text prediction is defined as

$$L_{pixel} = \frac{1}{(1+r)S} W L_{pixel\_CE}$$

Here, r is defined as the negative-positive ratio, as used in Online Hard Example Mining (OHEM). In this specific case, is used to select negative text predicted pixels with the top-K loss. It was found best to set r to 3. S is defined to be the area of all text instance areas. W is the weight matrix of the pixels being considered, the weights are assigned as prescribed above. Finally, L_{pixel_CE} is defined to be the Cross-Entropy loss matrix on text/non-text prediction.

Within the link loss, losses for positive and negative links are calculated separately and on positive pixels only. This loss is defined as

$$L_{link} = \frac{L_{link\_pos}}{rsum(W_{pos\_link})} + \frac{L_{link\_neg}}{rsum(W_{neg\_link})}$$

Here,

$$L_{link\_pos} = W_{pos\_link} L_{link\_CE}$$

and

$$L_{link\_neg} = W_{neg\_link} L_{link\_CE}$$

Where L_{link_CE} is the previously defined Cross-Entropy loss on the link prediction, and W_{pos_link} and W_{neg_link} are defined to be the weight of the pixel conditioned on whether or not the pixel is predicted to be a link or not.

PixelLink is optimized using SGD with momentum 0.9, and weight decay 5 * 10^{-4}. Additionally, when training the learning rate is set to 10^{-3} for the first 100 iterations, and then increased to 10^{-2} for the remaining iterations.

2.5 Alleviating Deficiencies in Deep Networks

We discuss two specific issues which emerge when engineering deep neural networks; degradation of gradients and loss of resolution. As completely differentiable systems, neural networks rely on the backward propagation algorithm and the use of gradients to determine parameter updates. Repeated use of these gradients of through backpropagation causes the widely documented "vanishing gradient problem", the loss of direct influence on the parameter update. The issue of resolution loss is due to the nature of contiguous receptive fields, as they are used in traditional convolutional layers. By constricting the input of a layer to a decreased size, layers remove much of the spatial information contained in the original input.

2.5.1 U-Net

U-Net [13] approaches the problem of loss of spatial acuity through the use two distinct network paths. A constricting path crushes the input to a small dimension, while an expansive path up-samples the input such that the resolution is only slightly diminished. The use of these two complementary paths allows for the U-Net architecture to capture high-level image context as well as low-level localization detail.

The constricting path of the network can be considered a traditional fully convolutional network. The input size is fixed at 572 * 572, which allows for tiles of large images to be used as inputs. The successive constriction is performed by traditional 3x3 convolutional layers, followed by the ReLU non-linearity. Immediately following 2 convolutional constricting operations is maximum pooling layer with kernel size 2x2 with stride two. The expansive path of the network consists of upsampled features and copied crops from intermediate layers of the constricting path. Each block within the expansive path is composed of traditional convolutional layers followed by the ReLU non-linearity, and 2x2 upconvolutional layers with a stride of 2. These upconvolutional layers, when used with the crop-copy operation increase the resolution of the output.

2.5.2 Dilated Residual Networks

ResNet [5] was proposed to combat the problem of gradient degradation through the utilization of "skip connections." These connections provided an identity mapping to the original output from preceding layers, reducing the effects of vanishing gradients. Dilated residual networks (DRNs) [6] take this one step further and attempt to combat both problems presented above. Skip connections as presented in [5] have been shown to greatly improve both the speed and efficiency of training through backpropagation but have no additional effect on the loss of spatial acuity which occurs in a traditional convolutional network. Dilated blocks, as used in [6], exponentially increase the receptive field of a convolutional layer by introducing nonadjacent kernels. In doing so, [6] does not crush the dimensions of the input. DRNs subsequently achieved

higher performance on the ImageNet Challenge [7] compared to their non-dilated counterparts with no modifications to model depth or complexity.

As an apparent disadvantage of DRNs, gridding artifacts may be caused through the use of dilated convolution operations. These are caused by an imbalance of content frequency between the dilated operator and the input itself. Due to the noncontiguous nature of dilated operators, the output from an active pixel may result in a corresponding grid pattern as output. Methods of alleviating these artifacts include augmenting the network with additional convolutional layers with progressively less dilation, removing max pooling layers which may aggravate effects of gridding, and removing residual connections that may directly exacerbate the effects of gridding through the identity mapping.

3 Methodology

3.1 Preliminary Work

Preliminary work has focused on obtaining character specific bounding boxes using the PixelLink [8] architecture trained on individual character annotations. Data augmentation techniques are used to enhance rotated bounding box detections. Using the training split data, we have produced a set of detectors from lowercase letters 'a' through 'i'. Similarly, we have trained a detector on the uppercase letter 'A'.

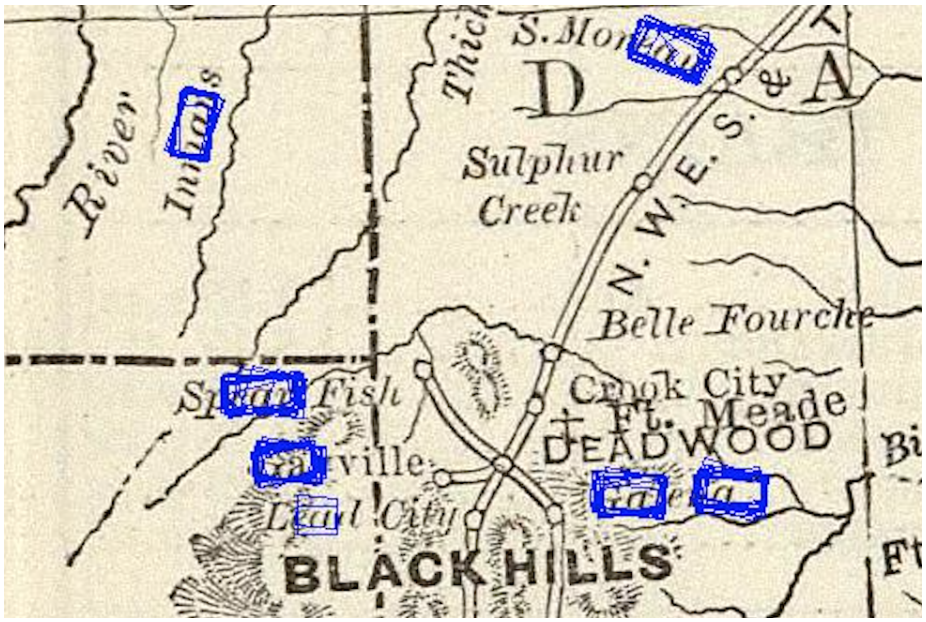

Figure 1: Rotated Lowercase 'a' Detections

A set of trained detectors may be used to produce character detections for each separate character, and thus implicitly obtain detection and recognition.

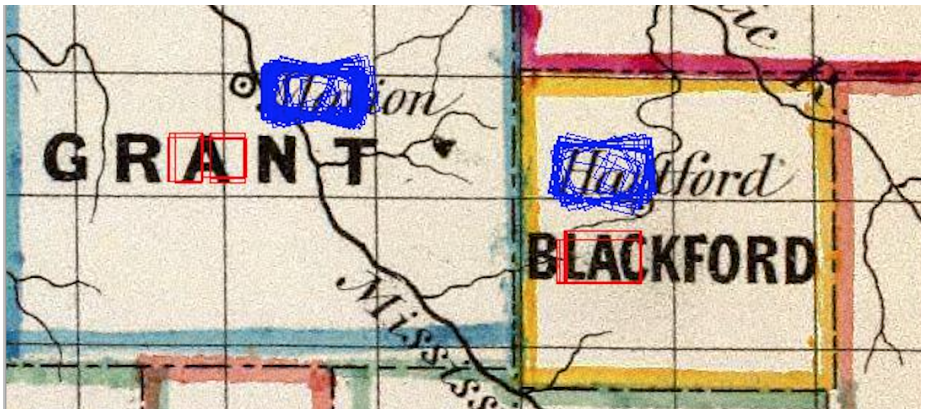

Figure 2: Rotated Lowercase 'a' Detections and Horizontal Uppercase 'A' Detections

3.2 Coding

In the preliminary stage, coding was focused mainly on creating a library of software used to wrangle the dataset of images and character annotations. Additionally, modifications were made to the configuration files of the original PixelLink repository. My personal work may be found at this link: PixelLinkCharacter Link. These files make heavy use of the OpenCV and PIL Image libraries for Python 3.6.

Focus for the coming semester is on perfecting these PixelLink annotations with additional data. Specifically, a complete set of alphabetic detectors must be trained in a similar fashion to the ones that currently exist. This will be done by training both uppercase and lowercase letter detectors on horizontal and rotated annotation data. A unified detection pipeline must be composed of these separate alphabetic detectors; one which produces rotation invariant annotations of all detected character instances. Non-Maximum Suppression (NMS) should also be achieved in a similar timeframe. NMS will allow for better annotation predictions, and consequently a more accurate predictor.

A parallel goal is to explore the DRN framework and attempt to produce a character annotation pipeline. Specifically, I would produce a detector which is capable of detecting single character instances through use of a segmentation map and a subsequent localization procedure. One approach would be to utilize an immediately available Dilated Residual Network as provided by [6]; this model is focused on producing a high fidelity segmentation map. This module may then be coupled with the label geometry generation procedure from [10] to produce accurate annotation geometries. Both frameworks are available as Tensorflow implementations on the Github repositories here and here.

Once a working model has been established, then it may be used to perform the same benchmarking exercises as will be performed with the previously mentioned PixelLink detectors. I describe the benchmarking in the following sections.

3.3 Data Collection

The dataset used to train the PixelLink detectors can be found here. The images are pictures of cartographic maps from the David Rumsey Collection. Of this collection, we maintain a curated list of 31 maps, from nine different atlases. Maps are all from US regions.

Additionally, I have augmented the dataset by producing crops and rotated crops. These crops have original dimensions of 512 * 512, and are further separated into training and testing splits of 80% and 20% respectively.

We may additionally produce a segmentation map training dataset from the existing cropped dataset. Simply, we must generate segmentation masks where text instances are labeled as positive and all other pixel instances are labeled as negative. Using a package such as Numpy or OpenCV, we may label all pixels contained within the ground truth bounding boxes as positive and the rest as negative.

3.4 Evaluation of Work

We will perform our experiments on the previously mentioned David Rumsey Map Collection. We maintain a split of training and testing examples of the crops I have created. This will then be used for 5-Fold Cross-Validation. We will report the F-Score of the respective pipelines, where a positive detection is a bounding box over a reasonable area of the text instance.

4 Evaluation

4.1 Milestones

The Milestones are as follows:

- Complete Training Procedure on All Character Annotations Using PixelLink
- Collection of the PixelLink Results
- Creation of Segmentation Dataset
- Modification of a Dilated Residual Network Model to be Used with Loss Function Based on [10]
- Collection of the DRN with EAST Results
- Completion of Manuscript and Oral Defense

In the following sections we will describe in detail the above milestones, and their criterion for evaluation.

4.1.1 PixelLink Training

My PixelLink detectors, as described in my preliminary work section, are trained to detect specific characters within bounding boxes. These bounding boxes are axis invariant, i.e. can detect rotated instances, but we employ the data augmentation procedure of rotation to create more training examples. Thus, further work will be completed by training the remaining character detectors, lowercase letters 'j' through 'z' and uppercase letters 'B' through 'Z'.

Evaluation of this milestone will be based on the timely completion of the trained detectors. I will present the trained models, and their annotation outputs to Professor Miller. We will discuss their results and modifying the training procedure as necessary.

4.1.2 PixelLink Results

Evaluation of this milestone is dependent on completion of the training procedures mentioned in the previous milestone. I will collect the trained models, generate predicted bounding boxes from the testing split, and calculate the F-Score using an IoU metric. Additionally, I will conduct an analysis of these results, discussing the predictions and my analysis in a 1-2 page report.

4.1.3 Creation of Segmentation Dataset

Evaluation of this milestone is dependent on the conversion of our current annotation into a set of segmentation maps, to be used in training the subsequent DRN model.

4.1.4 DRN Model Modification

Similar to the PixelLink training milestone, the ultimate goal of this milestone will be to obtain annotation predictions from the model. To create the model, though, I anticipate on marrying a dilated residual network with a localization scheme, as is presented in EAST [10]. To do this, I will modify an existing implementation of either ResNet [5] as provided in Keras [15], or the DRN implementation provided in [6].

Following the successful construction of this model, I will proceed to train the model on the training split of the map data .Evaluation of this milestone will be based on the timely completion of the trained detectors. I will present the trained models, and their annotation outputs to Professor Miller. We will discuss their results and modifying the training procedure as necessary.

4.1.5 DRN Model Results

Evaluation of this milestone is dependent on completion of the modification and training procedures mentioned in the previous milestone. I will collect the trained models, generate predicted bounding boxes from the testing split, and calculate the F-Score using an IoU metric when applicable. Additionally, I will conduct an analysis of these results, discussing the predictions and my analysis in a 1-2 page report.

4.1.6 Manuscript and Oral Defense

The report is to describe the separate milestones. The first draft will cover a majority of the results, especially my notes and analysis of the prospective results I receive from my detectors. Additionally, I will include a succinct literature review in the form of a related works section. After submission of the first draft, I will continue to append my final findings and incorporate feedback I receieve from Professor Miller and Professor Cao. The second draft will be a refined report in regards to content and incorporated analysis. The final draft, submitted to CHC Paths, will be an edited and complete draft.

Evaluation of this report will be based on the timely manner in which it is completed. The oral defense will be evaluated on my comprehensibility of the subject, my ability to explain my own implementation, and my ability to answer questions about the project.

5 Communication

I plan on meeting with Professor Miller bi-weekly for an hour. These meetings will occur from the first week of classes until the last week of classes. I will meet with Professor Cao as needed, but preferably at least once a month for an hour.

The meetings with Professor Miller will be focused on me sharing my current progress and results, while Professor Miller will be able to give immediate feedback. Additionally, we will discuss my progress towards completing the milestones as listed above.

I will be receiving 6 credits for my work, as such I plan to work on average 12 hours per week.

6 Timeline

The proposed timeline is as follows:

1. 09/13/2019 Uppercase and Lowercase Horizontal and Rotated PixelLink Models are Trained
2. 09/20/2019 Validation Results Obtained of PixelLink Models

3. 10/27/2019 Creation of Segmentation Dataset
4. 10/9/2019 Creation of DRN for Bounding Box Localization
5. 10/16/2019 Validation Results Obtained of DRN Models
6. 11/27/2019 1st Draft of Project Manuscript Submitted to Advisor (2 Weeks Before Last Day of Classes)
7. 12/04/2019 2nd Draft of Project Manuscript Submitted to Advisor After Necessary Revisions (1 Week Before Last Day of Classes)
8. 12/11/2019 Oral Defense (By Last Day of Classes)
9. 12/11/2019 Submission to CHC (By Last Day of Classes)

I do not anticipate any changes to the proposed timeline, but I will notify Professor Miller for necessary amendments.

References

[1] S. Long, X. He, and C. Yao, "Scene text detection and recognition: The deep learning era," 2018.

[2] D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, L. G. i. Bigorda, S. R. Mestre, J. Mas, D. F. Mota, J. A. Almazàn, and L. P. de las Heras, "Icdar 2013 robust reading competition," in *Proceedings of the 2013 12th International Conference on Document Analysis and Recognition*, ICDAR '13, (Washington, DC, USA), pp. 1484–1493, IEEE Computer Society, 2013.

[3] D. Karatzas, L. Gomez-Bigorda, A. Nicolaou, S. Ghosh, A. Bagdanov, M. Iwamura, J. Matas, L. Neumann, V. R. Chandrasekhar, S. Lu, F. Shafait, S. Uchida, and E. Valveny, "Icdar 2015 competition on robust reading," in *Proceedings of the 2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, ICDAR '15, (Washington, DC, USA), pp. 1156–1160, IEEE Computer Society, 2015.

[4] C. Yao, X. Bai, W. Liu, Y. Ma, and Z. Tu, "Detecting texts of arbitrary orientations in natural images," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1083–1090, June 2012.

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.

[6] F. Yu, V. Koltun, and T. Funkhouser, "Dilated residual networks," 2017.

[7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR09*, 2009.

[8] D. Deng, H. Liu, X. Li, and D. Cai, "Pixellink: Detecting scene text via instance segmentation," 2018.

[9] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," 2015.

[10] X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, and J. Liang, "East: An efficient and accurate scene text detector," 2017.

[11] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," 2015.

[12] M. Liao, B. Shi, X. Bai, X. Wang, and W. Liu, "Textboxes: A fast text detector with a single deep neural network," 2016.

[13] O. Ronneberger, P.Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, vol. 9351 of *LNCS*, pp. 234–241, Springer, 2015. (available on arXiv:1505.04597 [cs.CV]).

[14] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014.

[15] F. Chollet *et al.*, "Keras." `https://keras.io`, 2015.