Active Learning Policies for Labeling Visual Inspection Data

Stanford | ENGINEERING
Computer Science

Masoud Charkhabi masoudc@stanford.edu

Problem Description

Optical Character Recognition (OCR) has advanced rapidly in domains with vast labeled training data. Images that contain signage required for visual inspection can be distorted, and labeling these data are time consuming and requires specialized skills. Scene Text Detection and Recognition (STDR) are advancements of OCR for these complex OCR tasks. STDR is more effective with character-level annotations than word-level annotations, which exacerbates the labeling challenge. Active Learning (AL) is a promising method to address label availability by accelerating learning. AL adds a data selection algorithm into the supervised learning workflow. We conduct experiments to evaluate two broad AL approaches to STDR on visual inspection: 1. Disagreement-Based Active Learning (DAL) and 2. Policy-based Active Learning (PAL). PAL involves a Reinforcement Learning (RL) framework. Our experiments show that PAL has performance advantages in some cases. Auxiliary benefits of PAL are when labeling professionals have unique skills that can be described in the RL states, and when transferring AL policy to a new but similar task.

The key components of our PAL framework are described below.

- **Agent and Environment:** we define the AL data selection algorithm as the RL agent, and the annotator as the environment. With this definition, rewards are the loss reduction of the STDR model on a test set. The workflow is shown below:

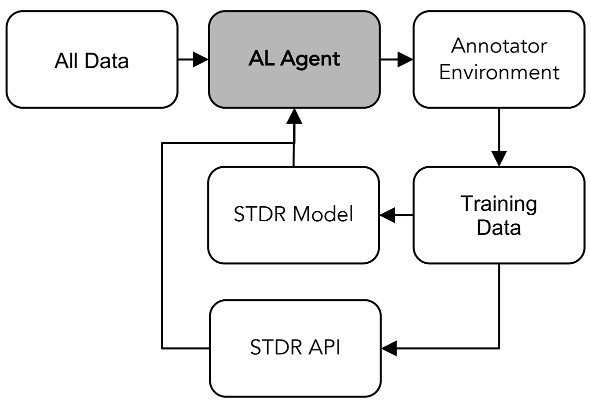

Figure 1. High-level flow of information between the components of the Active Learning system that is based on a Reinforcement Learning framework.

- **Sequential Process**: the decision of whether to select an example is temporal and sequential. Since errors are likely to come from complex images, the sequential presentation should distribute optimally from simple to complex, see Figure 2.
- **Decision Making:** the key decision in an AL problem is to present the example for labeling, or to model label it with the STDR model. This forms the action space of the RL framework.
- **Uncertainty:** the dynamics of the RL problem are model driven. The reward is determined by the change in Word Recognition Rate (WRR) of the STDR model on a test set, which involves uncertainty.

Figure 2: Top row: a sequence of images from simple to complex. Bottom row: a sequence of images with no pattern in complexity.

Datasets and Methods

Our Detection model is based on the Character Region Awareness For Text Detection (CRAFT) [3] State-Of-The-Art (SOTA) method for character level STD. Recognition is based the four stage STR [2] implementation. The models are combined in an STDR abstraction in our PAL framework. The data processing steps are shown in Figure 3.

Figure 3. Examples of the input and output behavior of the STDR model. The processing is in clock-wise orientation. Characters are first Detected, then cropped. The bottom right images are then fed into the Recognition system.

As a baseline we examine the performance of the stages of the STDR model on subsets of the training data with mini-batch. This is shown in Table 1. Once the STDR model is abstracted away as a function, a second API based STDR model (AWS Recognition) is used to examine disagreement for DAL. DAL uses a divergence in predictions over an ensemble to determine data selection. DAL and PAL should improve the learning trajectory over mini-batch training.

Level	Detection	Recognition	2-Stage
25	38.5	33.5	22.5
50	52.0	44.6	55.8
75	68.8	60.1	71.1
100	82.5	75.2	81.5

Table 1. Experiment Results for CRAFT on subsets of IC15. Metric is Top-1 Precision on Word Recognition Rate (WRR).

The full PAL procedure is described in Figure 4. The state, action, rewards of the RL problem are also described below the algorithm description.

Algorithm 1 PAL algorithm

procedure PAL(X, E)
  $i \leftarrow 0$
  $M \leftarrow (s_0, a_0, r_0, s_1)$
  while $i \leq n$ **do**
    $s_i = (p_{A_(Xi)}, p_{S_(Xi)}, M, E)$
    $a_i = \arg\max Q^{\phi}(s_i, a)$
    if $a_i = 1$ **then**
      $y_i = Annotate_(X_i, E)$
      $X \leftarrow X + (x_i, y_i)$
      update $STDR$
    end if
    $r_i = \delta(TestErr(i, i-1))$
    if $B(E) > B_{max}(E)$ **then**
      $M \leftarrow M + (s_i, a_i, r_i, s_{i+1})$
      Break
    end if
    $s_{i+1} = (p_{A_(i+1)}, p_{S_(i+1)}, M, E)$
    $M \leftarrow M + (s_i, a_i, r_i, s_{i+1})$
    update ϕ
    $i \leftarrow i + 1$
  return ϕ

X: image training examples
E: environment (annotator)
M: memory for alpha-greedy sampling
s, a, r: state, action, reward
B: annotator budget function
$STDR$: Scene Text Detection and Recognition Model
P_A: API model prediction
P_S: STDR model prediction
y_i: annotated example
ϕ: RL (AL) learned policy

Figure 4. PAL pseudo-code for a single episode of our PAL implementation.

Experiments and Results

The PAL approach is examined against the DAL implementation and the mini-batch baseline. The experiments are conducted with a group of two annotators in a simulation. The annotators are represented in the state of our RL with the environment variable E (see Figure 4.). We find that with E=1 the PAL approach has lower word recognition error, however with E=2 the results are worse than the baseline (Figure 5.)

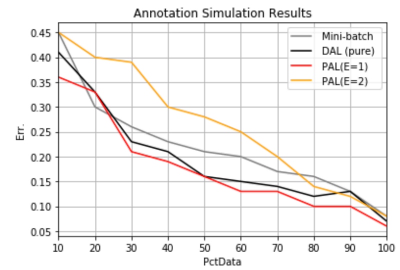

Figure 5. Annotation simulation with two users E=1 and E=2 compared to the mini-batch and DAL baselines.

With error analysis on common failure modes in STDR (difficult fonts, occlusion, low resolution, etc, [2]); we see that complex examples were presented to E=2 early in the training regime (Table 2.). In large samples this issue will be less problematic. Our simulation was run with N=100 examples. Besides the common failure modes presented in [2], we discover that text on glass surfaces are a failure mode that is common across all simulations (Figure 6.).

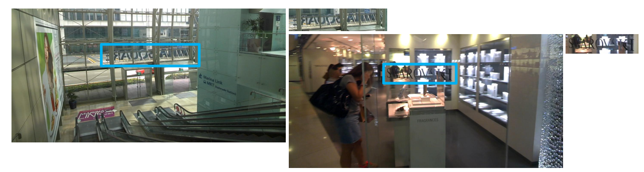

Figure 6. Common failure model of trained model is text on glass.

Level	Complexity E=1	Complexity E=2
20	.03	.08
50	.09	.17
70	.15	.22
100	.25	.25

Table 2. Distribution of example complexity across data.

Conclusions and Future Work

Our PAL implementation improved over the baseline. In order to create a more stable improvement, a better distribution of complexity across annotators is required. A content variable in the state, as in [4], could achieve this. Also, it could be used to estimate the difficulty of the labeling example. This can be estimated passively by recording the time spent on each example, as in activity recognition [1], or from the above mentioned content variable. Examples of complexity are shown in Figure 7. The simulation highlights that the state environment variable E is useful, since it captured the task complexity by extension. With large samples PAL can be transferred to similar tasks, however DAL and mini-batch cannot learn beyond their STDR model.

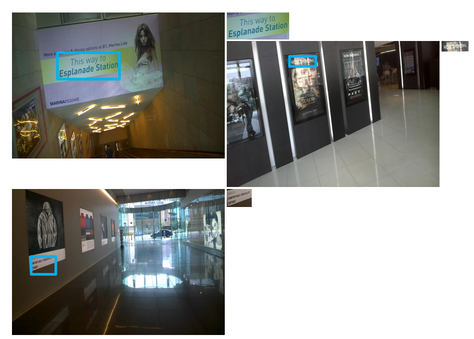

Figure 7. Example complexity simple to complex, clock-wise.

Stanford | ENGINEERING
Computer Science

Active Learning Policies for Labeling Visual Inspection Data

Masoud Charkhabi masoudc@stanford.edu

References

[1] H. Doughty, W. Mayol-Cuevas, D. Damen. The Pros and Cons: Rank-Aware Temporal Attention for Skill Determination in Long Videos. 06 2019.

[2] J. Baek, G. Kim, J. Lee, S. Park, D. Han, S. Yun, S. J. Oh, and H. Lee. What is wrong with scene text recognition
model comparisons? dataset and model analysis, 04 2019.

[3] Y. Baek, B. Lee, D. Han, S. Yun, and H. Lee. Character region awareness for text detection. 04 2019.

[4] M. Fang, Y. Li, and T. Cohn. Learning how to active learn: A deep reinforcement learning approach. 08 2017.