Learning Surrogates via Deep Embedding

Yash Patel, Tomáš Hodaň, Jiří Matas

Visual Recognition Group, Czech Technical University in Prague
{patelyas,hodanto2,matas}@fel.cvut.cz

Abstract. This paper proposes a technique for training a neural network by minimizing a surrogate loss that approximates the target evaluation metric, which may be non-differentiable. The surrogate is learned via a deep embedding where the Euclidean distance between the prediction and the ground truth corresponds to the value of the evaluation metric. The effectiveness of the proposed technique is demonstrated in a post-tuning setup, where a trained model is tuned using the learned surrogate. Without a significant computational overhead and any bells and whistles, improvements are demonstrated on challenging and practical tasks of scene-text recognition and detection. In the recognition task, the model is tuned using a surrogate approximating the edit distance metric and achieves up to 39% relative improvement in the total edit distance. In the detection task, the surrogate approximates the intersection over union metric for rotated bounding boxes and yields up to 4.25% relative improvement in the F_1 score.

1 Introduction

Supervised learning of a neural network involves minimizing a differentiable loss function on annotated data. The differentiable nature of the loss function and the network architecture allows the model weights to be updated via backpropagation [53]. The performance on a wide range of computer vision tasks have significantly improved thanks to the progress in deep neural network architectures [30, 21, 56] and the introduction of large scale supervised datasets [8, 35]. As designing architectures often demands detailed domain expertise and creating new datasets is expensive, there has been a substantial effort in automating the process of designing better task-specific architectures [10, 54, 65] and employing self-supervised methods of learning to reduce the dependence on human-annotated data [12, 7, 14]. However, little attention has been paid to automate the process of designing the loss functions.

For many practical problems in computer vision, models are trained with simple proxy losses, which may not align with the evaluation metric. The evaluation metric may not always be differentiable, prohibiting its use as a loss function. An example of a non-differentiable metric is the visible surface discrepancy (VSD) [23] used to evaluate 6D object pose estimation methods. Another example is the edit distance (ED) defined by counting unit operations (addition, deletion, and substitution) necessary to transform one text string into another

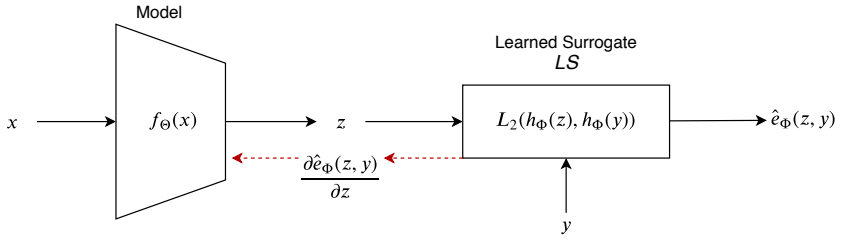

Fig. 1. For the input x with the corresponding ground-truth y, the model being trained outputs $z = f_\Theta(x)$. The learned surrogate provides a differentiable approximation of the evaluation metric: $\hat{e}_\Phi(z, y) = L_2(h_\Phi(z), h_\Phi(y))$, where h_Φ is a learned deep embedding model, and $h_\Phi(z)$ and $h_\Phi(y)$ are embedding representations for the prediction and the ground truth, respectively. Model $f_\Theta(x)$ for the target task (*e.g.* scene text recognition or detection) is trained with the gradients from the surrogate: $\frac{\partial(\hat{e}_\Phi(z,y))}{\partial z}$.

and is a common choice for evaluating scene text recognition methods [26, 27, 43]. Since ED is non-differentiable, the methods use either CTC [17] or per-character cross-entropy [3] as the proxy loss. Yet another popular non-differentiable metric is the intersection over union (IoU) used to compare the predicted and the ground truth bounding boxes when evaluating object detection methods. Although these methods typically resort to using proxy losses such as *smooth*-L_1 [50] or L_2 [49], Rezatofighi *et al.* [51] demonstrate that there is no strong correlation between L_n objectives and IoU. Further, Yu *et al.* [62] show that IoU accounts for a bounding box as a whole whereas regressing using an L_n proxy loss treats each point independently.

For popular metrics such as IoU, hand-crafted differentiable approximations have been designed [62, 51]. However, hand-crafting a surrogate is not scalable as it requires domain expertise and may involve task-specific assumptions and simplifications. The IoU-loss introduced in [62, 51] allows for optimization on the evaluation metric directly but makes a strong assumption about the bounding boxes to be axis-aligned. In numerous practical applications such as aerial image object detection [60], scene text detection [26] and visual object tracking [29], the bounding boxes may be rotated and the methods for such tasks revert to using simple but non-optimal proxy loss functions such as *smooth*-L_1 [40, 6, 2].

To address the aforementioned issues, this paper proposes to learn a differentiable surrogate that approximates the evaluation metric and use the learned surrogate to optimize the model for the target task. The metric is approximated via a deep embedding, where the Euclidean distance between the prediction and the ground truth corresponds to the value of the metric. The mapping to the embedding space is realized by a neural network, which is learned using only the value of the metric. Gradients of this value with respect to the inputs are not required for learning the surrogate. In fact, the gradients may not even exist, as is the case of the edit distance metric. Throughout this paper, we refer to the

proposed method for training with learned surrogates as "LS". Figure 1 provides an overview of the proposed method.

In this paper, the focus on a post-tuning setup, where a model that has converged on a proxy loss is tuned with LS. We consider two different optimization tasks: post-tuning with a learned surrogate for the edit distance (LS-ED) and the IoU of rotated bounding boxes (LS-IoU). To the best of our knowledge, we are the first to optimize directly on these evaluation metrics.

The rest of the paper is structured as follows. Related work is reviewed in Section 2, the technique for learning the surrogate and training with it is presented in Section 3, experiments are shown in Section 4 and the paper is concluded in Section 5.

2 Related Work

Training machine learning models by directly minimizing the evaluation metric has been shown effective on various tasks. For example, the state-of-the-art learned image compression [33, 4] and super-resolution [32, 9] methods directly optimize the perceptual similarity metrics such as MS-SSIM [59] and the peak signal-to-noise ratio (PSNR). Certain compression methods optimize on an approximate of human perceptual similarity, which is learned in a supervised manner using annotated data [44, 45]. Image classification methods [30, 21, 56] are typically trained with the cross-entropy loss that has been shown to align well with the misclassification rate, *i.e.* the evaluation metric, under the assumption of large scale and clean data [5, 31].

When designing evaluation metrics for practical computer vision tasks, the primary goal is to fulfil the requirements of potential applications and not to ensure the metrics being amenable to an optimization approach. As a consequence, many evaluation metrics are non-differentiable and cannot be directly minimized by the currently popular gradient-descent optimization approaches. For example, the visible surface discrepancy [23], which is used to evaluate 6D object pose estimation methods, was designed to be invariant under pose ambiguity. This is achieved by calculating the error only over the visible part of the object surface, which requires a visibility test that makes the metric non-differentiable. Another example is the edit distance metric [26, 15], which is used to evaluate scene text recognition methods and is calculated via dynamic programming, which makes it infeasible to obtain the gradients.

There have been efforts towards approximating non-differentiable operations in a differentiable manner to enable end-to-end training. Kato *et al.* [28] proposed a neural network to approximate rasterization, allowing for a direct optimization on IoU for 3D reconstruction. Agustsson *et al.* [1] proposed a soft-to-hard vector quantization mechanism. It is based on soft cluster assignments during backpropagation, which allows neural networks to learn tasks involving quantization, *e.g.* the image compression. Our work differs as we propose a general approach to approximate the evaluation metric, instead of approximating task-specific building blocks of neural networks.

Another line of research has focused on hand-crafting differentiable approximates of the evaluation metrics, which either align better with the metrics or enable training on them directly. Prabhavalkar *et al.* [46] proposed a way of optimizing attention based speech recognition models directly on word error rate. As mentioned earlier, [62, 51] proposed ways for directly optimizing on intersection-over-union (IoU) as the loss for the case of axis-aligned bounding boxes. Rahman *et al.* [48] proposed a hand-crafted approximation of IoU for semantic segmentation.

Learning task-specific surrogates has been attempted. Nagendra *et al.* [42] demonstrated that learning the approximate of IoU leads to better performance in the case of semantic segmentation. However, the method requires custom operations to estimate true and false positives, and false negatives, which makes the learning approach task-specific. Engilberge *et al.* [11] proposed a learned surrogate for sorting-based tasks such as cross-modal retrieval, multi-label image classification and visual memorability ranking. Their results on sorting-based tasks suggest that learning the loss function could outperform hand-crafted losses.

More closely related to our work is the direct loss method by Hazan *et al.* [20] where a surrogate loss is minimized by embedding the true loss as a correction term. Song *et al.* [57] extended this approach to the training of neural networks. However, it assumes that the loss can be disentangled into per-instance sub-losses, which is not always feasible, *e.g.* the F_1 score [16] involves two non-decomposable functions (recall and precision). An alternative is to directly learn the amount of update values that are applied to the parameters of the prediction model. The framework proposed in [34] includes a controller that uses per-parameter learning curves comprised of the loss values and derivatives of the loss with respect to each parameter. The method suffers from two drawbacks that prohibit its direct application to training on evaluation metrics: a) for large networks, it is computationally infeasible to store the learning curve of every parameter, and b) no gradient information is available for non-differentiable losses.

Our work is similar to the approach by Grabocka *et al.* [16], where the evaluation metric is approximated by a neural network. Their approach differs as the network learning the surrogate takes both the prediction and the ground truth as the input and directly regresses the value of the metric. Since we formulate the task as embedding learning and train the surrogate such that the L_2 in the embedded space corresponds to the metric, our method ensures that the gradients are smaller when the prediction is closer to the ground truth. Furthermore, as illustrated in Section 3, we learn the surrogate with an additional gradient penalty term to ensure that the gradients obtained from our learned surrogate are bounded for stable training.

3 Learning Surrogates via Deep Embedding

Say that the supervised task is being learned from samples drawn uniformly from a distribution $(x, y) \sim P_D$. For a given input x and an expected output y,

a neural network model outputs $z = f_\Theta(x)$ where Θ are the model parameters learned via backpropagation as:

$$\Theta_{t+1} \leftarrow \Theta_t - \eta \frac{\partial l(z, y)}{\partial \Theta_t} \tag{1}$$

where $l(z, y)$ is a differentiable loss function, t is the training iteration, and η is the learning rate.

The model trained with loss $l(z, y)$ is evaluated using metric $e(z, y)$. When metric $e(z, y)$ is differentiable, it can be directly used as the loss. The technique proposed in this paper addresses the cases when metric $e(z, y)$ is non-differentiable by learning a differentiable surrogate loss denoted as $\hat{e}_\Phi(z, y)$. The learned surrogate is realized by a neural network, which is differentiable and is used to optimize the model. The weight updates are:

$$\Theta_{t+1} \leftarrow \Theta_t - \eta \frac{\partial \hat{e}_\Phi(z, y)}{\partial \Theta_t} \tag{2}$$

3.1 Definition of the Surrogate

The surrogate is defined via a learned deep embedding h_Φ where the Euclidean distance between the prediction z and the ground truth y corresponds to the value of the evaluation metric:

$$\hat{e}_\Phi(z, y) = \|h_\Phi(z) - h_\Phi(y)\|_2 \tag{3}$$

3.2 Learning the Surrogate

Learning the surrogate, *i.e.* approximating the evaluation metric, with a deep neural network is formulated as a supervised learning task requiring three major components: a model architecture, a loss function, and a source of training data.

Architecture. In this paper, the architecture is designed manually, such that it is suitable for the nature of the inputs z and y (details are in Section 4). Modern approaches for architecture search, *e.g.* [10, 54, 65], could yield better results but are computationally expensive.

Training Loss. The surrogate is learned with the following objectives:

1. The learned surrogate corresponds to the value of the evaluation metric:

$$\hat{e}_\Phi(z, y) \approx e(z, y) \tag{4}$$

2. The first order derivative of the learned surrogate with respect to the prediction z is close to 1:

$$\left\| \frac{\partial \hat{e}_\Phi(z, y)}{\partial z} \right\|_2 \approx 1 \tag{5}$$

Both objectives are realized and linearly combined in the training loss:

$$\text{loss}(z, y) = \left\| \left(\hat{e}_{\Phi}(z, y) - e(z, y) \right\|_2^2 + \lambda \left(\left\| \frac{\partial \hat{e}_{\Phi}(z, y)}{\partial z} \right\|_2 - 1 \right)^2 \right. \tag{6}$$

Bounding the gradients (Equation 5) has shown to enhance the training stability for Generative Adversarial Networks [18] and has shown to be useful for learning the surrogate. Parameters Φ of the embedding model h_{Φ} are learned by minimizing the loss (Equation 6).

Source of Training Data. Source of the training data for learning the surrogate determines the quality of the approximation over the domain. The model $f_{\Theta}(x) = z$ for the supervised task is trained on samples obtained from a dataset D. Let us assume that R is a random data generator providing examples for the learning of the surrogate, sampled uniformly in the range of the evaluation metric (see Section 4 for details). Note that R is independent of $f_{\Theta}(x)$.

Three possibilities for the data source are considered:

1. *Global approximation*: $(z, y) \sim P_R$.
2. *Local approximation*: $(z, y) \sim P_{f_{\Theta}(x)}$, where $(x, y) \sim P_D$.
3. *Local-global approximation*: $(z, y) \sim P_{f_{\Theta}(x) \cup R}$.

The local-global approximation yields a high quality of both the approximation and gradients (Section 4.1) and is therefore used in the main experiments.

3.3 Training with the Learned Surrogate

The learned surrogate is used in a post-tuning setup, where model $f_{\Theta}(x)$ has been pre-trained using a proxy loss. This setup ensures that $f_{\Theta}(x)$ is not generating random outputs and thus simplifies post-tuning with the surrogate. The parameters of the surrogate Φ are initialized randomly.

Learning of the surrogate $\hat{e}_{\Phi}$ and post-tuning of the model $f_{\Theta}(x)$ are conducted alternatively. The surrogate parameters Φ are updated first while the model parameters Θ are fixed. The surrogate is learned by sampling (z, y) jointly from the model and the random generator. Subsequently, the model parameters are trained while the surrogate parameters are fixed. Algorithm 1 demonstrates the overall training procedure.

4 Experiments

The efficacy of LS is demonstrated on two different tasks: post-tuning with a learned surrogate for the edit distance (Section 4.2) and for the IoU of rotated bounding boxes (Section 4.3). This section provides details of the models for these tasks, design choices for learning the surrogates and empirical evidence showing the efficacy of LS. Unless stated otherwise, the results were obtained using the local-global approximation setup as elaborated in Algorithm 1.

Algorithm 1 Training with LS *(local-global approximation)*

Inputs: Supervised data D, random data generator R, evaluation metric e.
Hyper-parameters: Number of update steps I_a and I_b, learning rates η_a and η_b, number of epochs E.
Objective: Train the model for a given task that is $f_{\Theta}(x)$ and the surrogate ,*i.e.*, e_{Φ}.

Initialize $\Theta \leftarrow$ pre-trained weights, $\Phi \leftarrow$ random weights.
for epoch = 1,...,E **do**
  for i = 1,...,I_a **do**
    sample $(x, y) \sim P_D$, sample $(z_r, y_r) \sim P_R$
    inference $z = f_{\Theta^{epoch-1}}(x)$
    compute loss $l_{\hat{e}} = loss(z, y) + loss(z_r, y_r)$ (Equation 6)
    $\Phi^i \leftarrow \Phi^{i-1} - \eta_a \frac{\partial l_{\hat{e}}}{\partial \Phi^{i-1}}$
  end for
  $\Phi \leftarrow \Phi^{I_a}$
  for i = 1,...,I_b **do**
    sample $(x, y) \sim P_D$
    inference $z = f_{\Theta^{i-1}}(x)$
    compute loss $l_f = \hat{e}_{\Phi^{epoch}}(z, y)$ (Equation 3)
    $\Theta^i \leftarrow \Theta^{i-1} - \eta_b \frac{\partial (l_f)}{\partial \Theta^{i-1}}$
  end for
  $\Theta \leftarrow \Theta^{I_b}$
end for

4.1 Analysing the Learned Surrogates

The aspects considered for evaluating the surrogates are:

1. The quality of approximation $\hat{e}_{\Phi}(z, y)$.
2. The quality of gradients $\frac{\partial(\hat{e}_{\Phi}(z,y))}{\partial z}$.

Both the quality of the approximation and the gradients depend on three components: an architecture, a loss function, and a source of training data (Section 3.2). Given an architecture, the choices for the loss function to learn the surrogate and the training data are justified subsequently.

Quality of approximation. The quality of the approximation is judged by comparing the value of the surrogate with the value of the evaluation metric, calculated on samples obtained from model $f_{\Theta}(x)$. When learning the surrogate, higher quality of approximation is enforced by the mean squared loss between $e(z, y)$ and $\hat{e}_{\Phi}(z, y)$ (the first term on the right-hand side of Equation 6). Figure 2 (left) shows the quality of the approximation measured by the L_1 distance between the learned surrogate and the edit distance. It can be seen that the surrogate approximates the edit distance accurately (the L_1 distance drops swiftly below 0.2, which is negligible for the edit distance).

Quality of gradients. Judging the quality of gradients is more complicated. When learning the surrogate, the gradient-penalty term attempts to make the

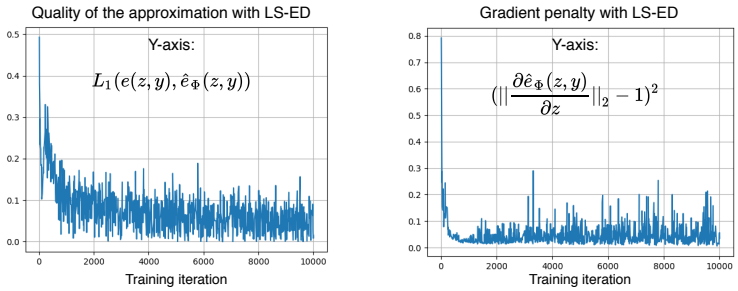

Fig. 2. Left: The error in approximation for the first $10K$ training iterations. The error is obtained by computing the L_1 distance between the true edit distance values and the LS-ED predictions and dividing by the batch size. Note that the edit distance can only take non-negative integer values, thus the error in the range of $0 - 0.2$ is fairly low. **Right**: The gradient penalty term from the optimization of the LS-ED model (Equation 6).

gradients bounded, *i.e.* to make the training stable (second term on the right-hand side of the equation 6). However, this is not sufficient if the gradients do not optimize $f_\Theta(x)$ on the evaluation metric. We rely on the improvement or the decline in the performance of the model $f_\Theta(x)$ to judge the quality of the gradients. Table 3 shows that the local-global approximation leads to the largest improvements when optimizing on IoU for rotated bounding boxes.

Choice of training data. Figure 3 shows the quality of approximation with different choices of training data for learning the surrogate. These empirical observations suggest that using global approximation leads to a low quality of the approximation. This can be accounted to the domain gap between the data obtained from the random generator and the model. Using the local approximation leads to a higher quality of the approximation, however, the gradients obtained from the surrogate are not useful to train $f_\Theta(x)$ (Table. 3), *i.e.* although the quality of the approximation is high, the quality of gradients is not. This can be attributed to surrogate over-fitting on samples obtained from the model and losing generalization capability on samples outside this distribution. Finally, it was observed that using the local-global approximation leads to both properties – high quality of approximation and high quality of gradients.

4.2 Post-Tuning with a Learned Surrogate for ED (LS-ED)

It is experimentally shown that LS can improve scene text recognition models (STR) on edit distance (ED), which is a popularly used metric to evaluate STR methods [26, 27, 43]. The empirical evidence shows that post-tuning STR models with LS-ED lead to improved performance on various metrics such as accuracy, normalized edit distance, and total edit distance [15].

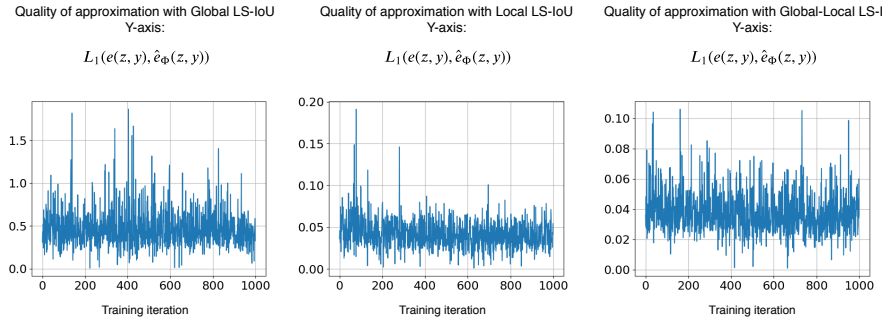

Fig. 3. The error in the approximation of the IoU for rotated bounding boxes is shown for the first $1K$ iterations of the training with LS-IoU. Error is measured by the L_1 distance between IoU and the surrogate. It can be seen that the error is high for the global and low for the local and global-local approximation variants.

Scene Text Recognition (STR). Given an input image of a cropped word, the task of STR is to generate the transcription of the word. The state-of-the-art architectures for scene text recognition can be factorized into four modules [3] (in this order): (a) transformation, (b) feature extraction, (c) sequence modelling, and (d) prediction. The feature extraction and prediction are the core modules of any STR model and are always employed. On the other hand, transformation and sequence modelling are not essential but have shown to improve the performance on benchmark datasets. Post-tuning with LS-ED is investigated for two different configurations of STR models.

The transformation module attempts to rectify the curved or tilted text, making the task easier for the subsequent modules of the model. It is learned jointly with the rest of the modules, and a popular choice is thin-plate spline (TPS) [55, 25, 36]. TPS can be either present or absent in the overall STR model.

The feature extraction module maps the image or its transformed version to a representation that focuses on the attributes relevant for character recognition, while the irrelevant features are suppressed. Popular choices include VGG-16 [56] and ResNet [21]. It is a core module of the STR model and is always present.

The features are the input of the sequence modelling module, which captures the contextual information within a sequence of characters for the next module to predict each character more robustly. BiLSTM [22] is a popular choice.

The output character sequence is predicted from the identified features of the image. The choice of the prediction module depends on the loss function used for training the STR model. Two popular choices of loss functions are CTC [17] (sigmoid output) or attention [55] (per-character softmax output).

Baek *et al.* [3] provides a detailed analysis of STR models and the impact of different modules on the performance. Following [3], LS-ED is investigated with the state-of-the-art performing configuration, which is *TPS-ResNet-BiLSTM-Attn*. To demonstrate the efficacy of LS-ED, results are also shown with *ResNet-BiLSTM-Attn*, *i.e.*, the transformation module is removed. Note that the CTC

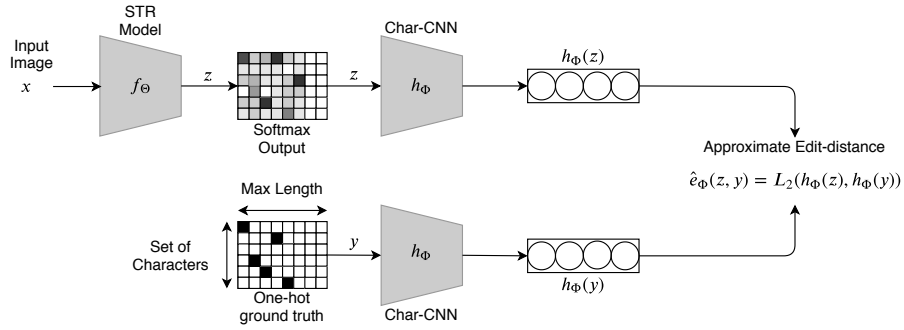

Fig. 4. Training scene text recognition (STR) models with LS-ED. The output of the STR model $z_{|A|\times L}$ and the ground-truth $y_{|A|\times L}$ (L is the maximum length of the word and A is the set of characters) are fed to the Char-CNN embedding model to obtain embedding vectors, $h_\Phi(z)$ and $h_\Phi(y)$ respectively. The approximate edit distance value is obtained by computing $\hat{e}_\Phi(z, y) = L_2(h_\Phi(z), h_\Phi(y))$.

based prediction has been shown to consistently perform worse compared to the attention counter-part [3], and thus the analysis in this paper has been narrowed down to only the attention-based prediction.

Similar to [3], the STR models are trained on the union of the synthetic data obtained from MJSynth [24] and SynthText [19] resulting in a total of 14.4 million training examples. Furthermore, following the standard setup of [3], there is no fine-tuning performed in a dataset-specific manner before the final testing. Let us say that the STR model is $f_\Theta(x)$, such that $f_\Theta : \mathbb{R}^{100\times32\times1} \rightarrow \mathbb{R}^{|A|\times L}$. The dimensions of the input cropped word image x is fixed to $100\times32\times1$ (gray-scale). The output for attention based prediction module is a per-character softmax over the set of characters. Here L is the maximum length of characters in the word and $|A|$ is the number of characters. During inference, argmax is performed at each character location to output the predicted text string. The ground truth y is represented as a per-character one-hot vector.

The STR models are first trained with the proxy loss, *i.e.*, cross-entropy for $300K$ iterations with a mini-batch size of 192. The models are optimized using ADADELTA [63] (same setup as [3]). Once the training is completed these models are tuned with LS-ED on the same set of 14.4 million training examples for another $20K$ iterations. The models trained purely on the synthetic datasets are tested on a collection of real datasets - IIIT-5K [41], SVT [58], ICDAR'03 [38], ICDAR'13 [27], ICDAR'15 [26], SVTP [47] and CUTE [52] datasets.

LS-ED architecture. Char-CNN architecture [64] is used for learning the deep embedding h_Φ. It consists of five $1D$ convolution layers equipped with LeakyReLU activation [61] followed by two fully connected layers. The embedding h_Φ maps the input such that $h_\Phi : \mathbb{R}^{|A|\times L} \rightarrow \mathbb{R}^{1024}$. Note that since h_Φ constitutes of convolution and fully-connected layers, it is differentiable and allows

for backpropagation to the STR model. In feed-forward, the two embeddings for the ground-truth y (one-hot) and the model prediction z (softmax) are obtained by performing feed-forward through h_Φ and an approximate of edit distance is computed by measuring the L_2 between the two vectors (Figure 4).

Post-tuning with LS-ED. A random generator is designed for this task, which generates a pair of words (z_r, y_r) and ensures uniform sampling in the range of the true error. It was observed that the uniform sampling is essential to avert over-fitting of the learned surrogate on a certain range of the true metric. For the edit distance metric $e(z, y) \in \{0, ..., b\}$ (b being the maximum possible value), the generator samples a word randomly from a text corpus and distorts the words by performing random addition, deletion, and substitution operations.

The post-tuning of the STR model $f_\Theta(x)$ with LS-ED follows Algorithm 1. For the case of the edit distance, there is a significant domain gap between the samples obtained from the STR model (z) and the random generator (z_r). This is because the random generator operates directly on the text string, *i.e.*, z_r is one-hot representation. Thus, using the global approximation setting yields a low quality of the approximation. Further, it was observed that training the surrogate purely with the data generated from the STR model, *i.e.*, local approximation, leads to a good approximation but does not lead to an improvement in the performance of the STR model, which indicates a low quality of gradients.

Finally as described in Algorithm 1, the local-global approximation is used. The quality of approximation and the gradient penalty from post-tuning with LS-ED are shown in Figure 2. Note that the edit distance value is a whole number and the surrogate attempts to approximate it, thus the error in approximation as shown in Figure 2 is low. The quality of the gradients can be seen by improvement in the performance of the STR models. Thus the local-global approximation guides to a high quality of both the approximation and gradients.

The results for the two configurations of STR models, *i.e.*, *ResNet-BiLSTM-Attn* and *TPS-ResNet-BiLSTM-Attn*, are shown in Table 1 and Table 2, respectively. It can be observed that LS-ED improves the performance of the STR models on all metrics. The most significant gains are observed on total-edit distance (TED) as the surrogate attempts to minimize its approximation.

4.3 Post-Tuning with a Learned Surrogate for IoU (LS-IoU)

It is experimentally demonstrated that LS can optimize scene text detection models on intersection-over-union (IoU) for rotated bounding boxes. IoU is a popular metric used to evaluate the object detection [49, 50] and scene text detection models [40, 6, 37, 26, 15]. Gradients for IoU can be hand-crafted for the case of axis-aligned bounding boxes [62, 51], however, it is complex to design the gradients for rotated bounding boxes. The learned surrogate of IoU allows backpropagation for rotated bounding boxes. For the task of rotated scene text detection on ICDAR'15 [26], it is shown that post-tuning the text detection model with LS-IoU leads to improvement on recall, precision, and F_1 score.

Test Data	Loss Function	↑ Acc.	↑ NED	↓ TED
IIIT-5K	Cross-Entropy	84.300	0.954	945
IIIT-5K	LS-ED	86.300 +2.37%	0.953 −0.10%	837 +11.42%
SVT	Cross-Entropy	84.699	0.940	229
SVT	LS-ED	86.399 +2.00%	0.947 +0.74%	196 +14.41%
ICDAR'03	Cross-Entropy	92.558	0.972	151
ICDAR'03	LS-ED	94.070 +1.63%	0.977 +0.51%	119 +26.89%
ICDAR'13	Cross-Entropy	89.754	0.949	260
ICDAR'13	LS-ED	91.133 +1.53%	0.960 +1.15%	157 +39.61%
ICDAR'15	Cross-Entropy	71.452	0.889	1135
ICDAR'15	LS-ED	74.655 +4.48%	0.899 +1.12%	1013 +10.74%
SVTP	Cross-Entropy	74.109	0.891	424
SVTP	LS-ED	77.519 +4.60%	0.901 +1.22%	381 +10.14%
CUTE	Cross-Entropy	68.293	0.838	285
CUTE	LS-ED	71.777 +5.10%	0.868 +3.57%	234 +17.89%

Table 1. ResNet-BiLSTM-Attn: The models are evaluated on IIIT-5K [41], SVT [58], ICDAR'03 [38], ICDAR'13 [27], ICDAR'15 [26], SVTP [47] and CUTE [52] datasets. The results are reported using accuracy **Acc.** (higher is better), normalized edit distance **NED** (higher is better) and total edit distance **TED** (lower is better). Relative gains are shown in green and relative declines in red.

Scene Text Detection. Given a natural scene image, the objective is to obtain precise word-level rotated bounding boxes. The method proposed by Ma *et al.* [40] is used for the task. It extends Faster-RCNN [50] based object detector to incorporate rotations. This is achieved by adding angle priors in anchor boxes to enable rotated region proposals. A sampling strategy using IoU compares these proposals with the ground truth and filter the positive and the negative proposals. Only the filtered proposals are used for the loss computation.

The positive proposals are regressed to fit precisely with the ground truth. Through rotated region-of-interest (RROI) pooling, the features corresponding to the proposals are obtained and used for text/no-text binary classification. The overall loss function for training in [40] is defined as a linear combination of classification loss (negative log-likelihood) and regression loss (*smooth*-L_1).

The publicly available implementation of [40, 39] is used with the original hyper-parameter settings – the model is trained for $140K$ iterations using the SGD optimizer and batch-size of 1. The model is trained on a union of ICDAR'15 [26] and ICDAR-MLT [43] datasets, providing 6295 training images.

LS-IoU architecture. The embedding model for LS-IoU consists of five fully-connected layers with ReLU activation [13]. A rotated bounding box is repre-

Test Data	Loss Function	↑ Acc.	↑ NED	↓ TED
IIIT-5K	Cross-Entropy	87.500	0.961	722
IIIT-5K	LS-ED	87.933 +0.49%	0.963 +0.20%	645 +10.66%
SVT	Cross-Entropy	87.172	0.952	180
SVT	LS-ED	86.708 −0.53	0.954 +0.21%	163 +9.44%
ICDAR'03	Cross-Entropy	94.302	0.979	110
ICDAR'03	LS-ED	94.535 +0.24%	0.981 +0.20%	99 +10.00%
ICDAR'13	Cross-Entropy	92.020	0.966	137
ICDAR'13	LS-ED	92.299 +0.30%	0.979 +1.34%	108 +21.16%
ICDAR'15	Cross-Entropy	78.520	0.915	868
ICDAR'15	LS-ED	78.410 −0.14%	0.915 ±0.00%	837 +3.57%
SVTP	Cross-Entropy	78.605	0.912	346
SVTP	LS-ED	79.225 +0.78%	0.913 +0.10%	333 +3.75%
CUTE	Cross-Entropy	73.171	0.871	224
CUTE	LS-ED	74.216 +1.42%	0.875 +0.45%	219 +2.23%

Table 2. TPS-ResNet-BiLSTM-Attn: The models are evaluated on IIIT-5K [41], SVT [58], ICDAR'03 [38], ICDAR'13 [27], ICDAR'15 [26], SVTP [47] and CUTE [52] datasets. The results are reported using accuracy **Acc.** (higher is better), normalized edit distance **NED** (higher is better) and total edit distance **TED** (lower is better). Relative gains are shown in green and relative declines in red.

sented with six parameters, two for the coordinates of the centre of the box, two for the height and the width and two for *cosine* and *sine* of the rotation angle. The centre coordinates and the dimensions of the box are normalized with image dimensions to make the representation invariant to the image resolution.

The embedding model maps the representation of a positive box proposal and the matching ground-truth into a vector as $h_\Phi : \mathbb{R}^6 \rightarrow \mathbb{R}^{16}$. The approximation of the IoU between two bounding boxes is computed by the L_2 distance between the two vector representations.

Post-tuning with LS-IoU. The random generator for LS-IoU samples rotated bounding boxes from the set of training labels and modifies the boxes by changing the centre locations, dimensions, and rotation angle within certain bounds to create a distorted variant. Since uniform sampling over the range of IoU is difficult, we store roughly 3 million such examples along with the IoU values and sample from this collection.

Note that since the overall loss for training [40] is a combination of a regression loss and a classification loss, LS-IoU only replaces the regression component (*smooth-*L_1) with the learned surrogate for IoU. For post-tuning with LS-IoU, the results are shown for all three setups, that is, global approximation, local

Loss Function	↑ **Recall**	↑ **Precision**	↑ F_1 **score**
Smooth-L_1	71.21%	84.71%	77.37%
LS-IoU (global)	66.97% −5.95%	84.71% ±0.00%	74.81% −3.30%
LS-IoU (local)	70.92% −0.40%	86.60% +2.23%	77.98% +0.78%
LS-IoU (local-global)	76.79% +7.83%	84.93% +0.25%	80.66% +4.25%

Table 3. RRPN-ResNet-50 [40, 39]: Evaluations on Incidental Scene Text ICDAR'15 [26]. Relative gains are shown in green and relative declines in red.

approximation and global-local approximation (Algorithm 1). For each of these, the model trained with proxy losses is post-tuned with LS-IoU for $20K$ iterations. The quality of the approximations for the first $1K$ iterations of the training is shown in Figure 3. Since the range of IoU is in $[0, 1]$, it can be seen that the error is high for the global approximation. For both local and global-local, the quality of the approximation is significantly better (roughly 10 times lower error).

As mentioned earlier, the quality of gradients is judged by the improvement or deterioration of the model ($f_\Theta(x)$) post-tuned with LS-IoU. The results for scene text detection on the ICDAR'15 [26] dataset are shown in Table 3. It is observed that post-tuning the detection model with LS-IoU (global) leads to deterioration. Post-tuning with LS-IoU (local) improves the precision but makes recall worse. Finally, LS-IoU (local-global) from Algorithm 1 improves both the precision and recall, boosting the F_1 score by relative 4.25%.

5 Conclusions

A technique is proposed for training neural networks by minimizing learned surrogates that approximate the target evaluation metric. The effectiveness of the proposed technique has been demonstrated in a post-tuning setup, where a trained model is tuned on the learned surrogate. Improvements have been achieved on the challenging tasks of scene-text recognition and detection. By post-tuning, the model with LS-ED, relative improvements of up to 39% on the total edit distance has been achieved. On detection, post-tuning with LS-IoU has shown to provide a relative gain of 4.25% on the F_1 score.

Acknowledgement

The authors thank R. Manmatha, Dmytro Mishkin, Michal Bušta, Klára Janoušková, Viresh Ranjan and Abhijeet Kumar for the feedback. This research was supported by Research Center for Informatics (project CZ.02.1.01/0.0/0.0/-16019/0000765 funded by OP VVV) and CTU student grant (SGS OHK3-019/20).

References

1. Agustsson, E., Mentzer, F., Tschannen, M., Cavigelli, L., Timofte, R., Benini, L., Gool, L.V.: Soft-to-hard vector quantization for end-to-end learning compressible representations. In: NeurIPS (2017)
2. Azimi, S.M., Vig, E., Bahmanyar, R., Körner, M., Reinartz, P.: Towards multi-class object detection in unconstrained remote sensing imagery. In: ACCV (2018)
3. Baek, J., Kim, G., Lee, J., Park, S., Han, D., Yun, S., Oh, S.J., Lee, H.: What is wrong with scene text recognition model comparisons? dataset and model analysis. ICCV (2019)
4. Ballé, J., Minnen, D., Singh, S., Hwang, S.J., Johnston, N.: Variational image compression with a scale hyperprior. ICLR (2018)
5. Berrada, L., Zisserman, A., Kumar, M.P.: Smooth loss functions for deep top-k classification. ICLR (2018)
6. Bušta, M., Patel, Y., Matas, J.: E2e-mlt-an unconstrained end-to-end method for multi-language scene text. In: ACCV (2018)
7. Caron, M., Bojanowski, P., Joulin, A., Douze, M.: Deep clustering for unsupervised learning of visual features. In: ECCV (2018)
8. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: CVPR (2009)
9. Dong, C., Loy, C.C., He, K., Tang, X.: Image super-resolution using deep convolutional networks. TPAMI (2015)
10. Elsken, T., Metzen, J.H., Hutter, F.: Neural architecture search: A survey. arXiv preprint arXiv:1808.05377 (2018)
11. Engilberge, M., Chevallier, L., Pérez, P., Cord, M.: Sodeep: A sorting deep net to learn ranking loss surrogates. In: CVPR (2019)
12. Gidaris, S., Singh, P., Komodakis, N.: Unsupervised representation learning by predicting image rotations. ICLR (2018)
13. Glorot, X., Bordes, A., Bengio, Y.: Deep sparse rectifier neural networks. In: AISTATS (2011)
14. Gomez, L., Patel, Y., Rusiñol, M., Karatzas, D., Jawahar, C.: Self-supervised learning of visual features through embedding images into text topic spaces. In: CVPR (2017)
15. Gomez, R., Shi, B., Gomez-Bigorda, L., Neumann, L., Veit, A., Matas, J., Belongie, S.J., Karatzas, D.: ICDAR2017 robust reading challenge on coco-text. In: ICDAR (2017)
16. Grabocka, J., Scholz, R., Schmidt-Thieme, L.: Learning surrogate losses. arXiv preprint arXiv:1905.10108 (2019)
17. Graves, A., Fernández, S., Gomez, F., Schmidhuber, J.: Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In: ICML (2006)
18. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.C.: Improved training of wasserstein gans. In: NeurIPS (2017)
19. Gupta, A., Vedaldi, A., Zisserman, A.: Synthetic data for text localisation in natural images. In: CVPR (2016)
20. Hazan, T., Keshet, J., McAllester, D.A.: Direct loss minimization for structured prediction. In: NeurIPS (2010)
21. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)

22. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation (1997)
23. Hodan, T., Michel, F., Brachmann, E., Kehl, W., GlentBuch, A., Kraft, D., Drost, B., Vidal, J., Ihrke, S., Zabulis, X., et al.: Bop: Benchmark for 6d object pose estimation. In: ECCV (2018)
24. Jaderberg, M., Simonyan, K., Vedaldi, A., Zisserman, A.: Synthetic data and artificial neural networks for natural scene text recognition. CoRR (2014)
25. Jaderberg, M., Simonyan, K., Zisserman, A., Kavukcuoglu, K.: Spatial transformer networks. In: NeurIPS (2015)
26. Karatzas, D., Gomez-Bigorda, L., Nicolaou, A., Ghosh, S., Bagdanov, A., Iwamura, M., Matas, J., Neumann, L., Chandrasekhar, V.R., Lu, S., et al.: Icdar 2015 competition on robust reading. In: ICDAR (2015)
27. Karatzas, D., Shafait, F., Uchida, S., Iwamura, M., i Bigorda, L.G., Mestre, S.R., Mas, J., Mota, D.F., Almazan, J.A., De Las Heras, L.P.: Icdar 2013 robust reading competition. In: ICDAR (2013)
28. Kato, H., Ushiku, Y., Harada, T.: Neural 3d mesh renderer. In: CVPR (2018)
29. Kristan, M., Matas, J., Leonardis, A., Felsberg, M., Pflugfelder, R., Kamarainen, J.K., Cehovin Zajc, L., Drbohlav, O., Lukezic, A., Berg, A., et al.: The seventh visual object tracking vot2019 challenge results. In: ICCV Workshops (2019)
30. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NeurIPS (2012)
31. Lapin, M., Hein, M., Schiele, B.: Loss functions for top-k error: Analysis and insights. In: CVPR (2016)
32. Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., et al.: Photo-realistic single image super-resolution using a generative adversarial network. In: CVPR (2017)
33. Lee, J., Cho, S., Beack, S.K.: Context-adaptive entropy model for end-to-end optimized image compression. ICLR (2019)
34. Li, K., Malik, J.: Learning to optimize neural nets. arXiv preprint arXiv:1703.00441 (2017)
35. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: ECCV (2014)
36. Liu, W., Chen, C., Wong, K.K., Su, Z., Han, J.: Star-net: A spatial attention residue network for scene text recognition. In: BMVC (2016)
37. Liu, X., Liang, D., Yan, S., Chen, D., Qiao, Y., Yan, J.: FOTS: fast oriented text spotting with a unified network. In: CVPR (2018)
38. Lucas, S.M., Panaretos, A., Sosa, L., Tang, A., Wong, S., Young, R.: Icdar 2003 robust reading competitions. In: ICDAR (2003)
39. Ma, J.: RRPN in pytorch. https://github.com/mjq11302010044/RRPNpytorch (2019)
40. Ma, J., Shao, W., Ye, H., Wang, L., Wang, H., Zheng, Y., Xue, X.: Arbitrary-oriented scene text detection via rotation proposals. IEEE Transactions on Multimedia (2018)
41. Mishra, A., Alahari, K., Jawahar, C.: Scene text recognition using higher order language priors. In: BMVC (2012)
42. Nagendar, G., Singh, D., Balasubramanian, V.N., Jawahar, C.: Neuro-iou: Learning a surrogate loss for semantic segmentation. In: BMVC (2018)
43. Nayef, N., Patel, Y., Busta, M., Chowdhury, P.N., Karatzas, D., Khlif, W., Matas, J., Pal, U., Burie, J.C., Liu, C.l., et al.: Icdar2019 robust reading challenge on multi-lingual scene text detection and recognition–rrc-mlt-2019. arXiv preprint arXiv:1907.00945 (2019)

44. Patel, Y., Appalaraju, S., Manmatha, R.: Deep perceptual compression. arXiv preprint arXiv:1907.08310 (2019)
45. Patel, Y., Appalaraju, S., Manmatha, R.: Hierarchical auto-regressive model for image compression incorporating object saliency and a deep perceptual loss. arXiv preprint arXiv:2002.04988 (2020)
46. Prabhavalkar, R., Sainath, T.N., Wu, Y., Nguyen, P., Chen, Z., Chiu, C.C., Kannan, A.: Minimum word error rate training for attention-based sequence-to-sequence models. In: ICASSP (2018)
47. Quy Phan, T., Shivakumara, P., Tian, S., Lim Tan, C.: Recognizing text with perspective distortion in natural scenes. In: ICCV (2013)
48. Rahman, M.A., Wang, Y.: Optimizing intersection-over-union in deep neural networks for image segmentation. In: ISVC (2016)
49. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: CVPR (2016)
50. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: NeurIPS (2015)
51. Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., Savarese, S.: Generalized intersection over union: A metric and a loss for bounding box regression. In: CVPR (2019)
52. Risnumawan, A., Shivakumara, P., Chan, C.S., Tan, C.L.: A robust arbitrary text detection system for natural scene images. Expert Systems with Applications (2014)
53. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning representations by back-propagating errors. Nature (1986)
54. Ryoo, M.S., Piergiovanni, A., Tan, M., Angelova, A.: Assemblenet: Searching for multi-stream neural connectivity in video architectures. NeurIPS (2019)
55. Shi, B., Wang, X., Lyu, P., Yao, C., Bai, X.: Robust scene text recognition with automatic rectification. In: CVPR (2016)
56. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
57. Song, Y., Schwing, A., Urtasun, R., et al.: Training deep neural networks via direct loss minimization. In: ICML (2016)
58. Wang, K., Babenko, B., Belongie, S.: End-to-end scene text recognition. In: ICCV (2011)
59. Wang, Z., Simoncelli, E.P., Bovik, A.C.: Multiscale structural similarity for image quality assessment. In: ACSSC (2003)
60. Xia, G.S., Bai, X., Ding, J., Zhu, Z., Belongie, S., Luo, J., Datcu, M., Pelillo, M., Zhang, L.: Dota: A large-scale dataset for object detection in aerial images. In: CVPR (2018)
61. Xu, B., Wang, N., Chen, T., Li, M.: Empirical evaluation of rectified activations in convolutional network. CoRR (2015)
62. Yu, J., Jiang, Y., Wang, Z., Cao, Z., Huang, T.: Unitbox: An advanced object detection network. In: ACM-MM (2016)
63. Zeiler, M.D.: ADADELTA: an adaptive learning rate method. CoRR (2012)
64. Zhang, X., Zhao, J.J., LeCun, Y.: Character-level convolutional networks for text classification. In: NeurIPS (2015)
65. Zoph, B., Le, Q.V.: Neural architecture search with reinforcement learning. arXiv preprint arXiv:1611.01578 (2016)