Adapting Style and Content for Attended Text Sequence Recognition

Steven Schwarcz
University of Marlyand, College Park
schwarcz@umiacs.umd.edu [*]

Alexander Gorban
Google Research
gorban@google.com

Xavier Gibert Serra
Google Research
xgibert@google.com

Dar-Shyang Lee
Google Research
dsl@google.com

Abstract

In this paper, we address the problem of learning to perform sequential OCR on photos of street name signs in a language for which no labeled data exists. Our approach leverages easily-generated synthetic data and existing labeled data in other languages to achieve reasonable performance on these unlabeled images, through a combination of a novel domain adaptation technique based on gradient reversal and a multi-task learning scheme. In order to accomplish this, we introduce and release two new datasets - Hebrew Street Name Signs (HSNS) and Synthetic Hebrew Street Name Signs (SynHSNS) - while also making use of the existing French Street Name Signs (FSNS) dataset. We demonstrate that by using a synthetic dataset of Hebrew characters and a labeled dataset of French street name signs in natural images, it is possible to achieve a significant improvement on real Hebrew street name sign transcription, where the synthetic Hebrew data and real French data each overlap with different features of the images we wish to transcribe.

1. Introduction

There are eight alphabet groups in use today - Arabic, Aramaic, Armenian, Brahmi, Cyrillic, Georgian, Greek and Latin - each used by many languages in hundreds of dialects. For most of these languages it is hard to find skilled operators to label a large dataset at a reasonable price. Without a better way to train a system in novel languages, it would not be practical to build, for instance, text recognition systems for real word images, such as Google Street View, which can support non-Latin languages.

Most existing sequential OCR systems are trained using sequential models on a mix of synthetic and real data [18, 43]. For printed documents or books, the difference between synthetic and real data may be insignificant, and there are many ways to build an OCR model that is able to generalize. But for problems of text recognition on images in the wild, such as street name signs, the gap between synthetic text renderings and real images is far too large. Thus most existing OCR approaches are not able to generalize and require extensive labeling. The algorithm we present here is a solution to that problem, requiring no new manually labelling. Instead, we use a combination of trivial synthetic data and an existing dataset in an unrelated language.

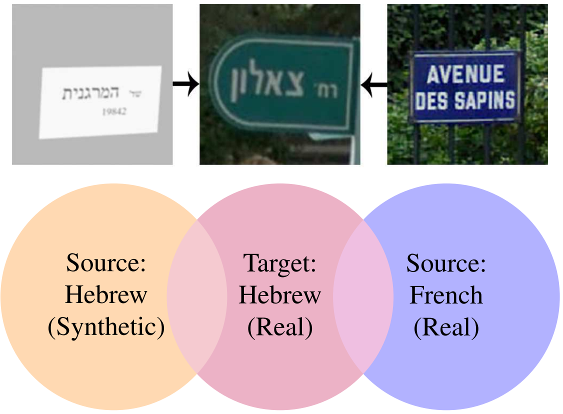

Figure 1. We seek to transcribe real images in some language (*e.g.* Hebrew) without access to any labeled training data by using a combination of synthetic data in the same language and labeled real data in a completely different language (*e.g.* French). The synthetic Hebrew data overlaps with the real Hebrew data in content, but not in style, while the real French data overlaps in style but not in content. Thus, the sources are complementary; they each overlap the target significantly, despite having very little overlap with each other.

Our experiments show that including another language during training actually alleviates the need for more realistic synthetic data. The neural network learns the ”content” of the first language from the synthetic data, while learning to

[*] This work was done while the first author was with Google Research

deal with the realistic ”style” of real images from the second language. We illustrate the effectiveness of this approach using Hebrew for our target language and French for our existing dataset, given the public availability of the French Street Name Signs (FSNS) dataset [30]. The synthetic data we generate is intentionally kept relatively minimalistic, to emphasize that the system is not using the synthetic data to learn anything stylistic, and because we believe our algorithm becomes more practical the less sophisticated the synthetic data.

Interestingly, significant learning happens despite the fact that Hebrew, an Aramaic language, shares no glyphs or characters with French, a Latin language. There is therefore nothing within our algorithm that is inherently language-specific: the French data should hypothetically be sufficient to train a system on almost any other language, all without the need for any manual labeling.

Finally, in order to ensure that our numbers are reproducible, we introduce and release the Hebrew Street Name Signs (HSNS) and Synthetic Hebrew Street Name Signs (SynHSNS) datasets, on which we perform all of our experiments.

2. Related work

2.1. Domain Adaptation

Within the field of computer vision, a large number of unsupervised and semi-supervised domain adaptation techniques have been invented and explored, especially in the context of image classification [29, 26, 25, 24, 23, 14], but also in other areas such as semantic segmentation [47, 27, 16], object pose recognition [2] and object detection [4, 17]. In all cases, the goal of these techniques is to match the distributions of some source domain to that of a target domain.

In some cases, this is achieved by attempting to explicitly match the moments of the two distributions. For example, Maximum Mean Discrepancy (MMD) [13] is a loss that explicitly minimizes the norm of the difference between two distributions’ means, and has been used to good effect in [37, 20, 3]. Alternatively, work such as [31] and [32] have made significant progress by explicitly aligning the second moments of the source and target domains.

In addition to explicit moment-matching techniques, another technique known as Gradient Reversal (GR) [8, 9] has emerged as a powerful paradigm for deep domain adaptation, serving a fundamental role in many deep domain adaptation systems [3, 4, 16]. GR has even been used effectively for problems completely outside the scope of computer vision, such as machine translation [18]. In the GR setting, a deep network is given an additional discriminator branch that uses deep features to classify samples as originating from either the source or the target domain. The network concurrently trains a feature extractor to fool the discriminator by flipping the sign of the gradient of the discriminator loss with respect to the feature extractor.

An alternative but closely-related deep domain adaptation paradigm uses adversarial learning to minimize domain shift [36, 15, 2, 26, 27]. These techniques are closely related to Generative Adversarial Networks (GANs) [12] and also use a discriminator to push both feature distributions together.

Domain adaptation has also been used in computer vision for various text-related tasks. For example, domain adaptation techniques have been used to identify fonts in images [42, 41]. Domain adaptation has also been applied to problems involving natural language processing [6, 11, 5], a field related to OCR in its use of language modelling and sequential processing.

There has also been research that adapts style specifically, either for language problems [44, 45] or vision problems [35, 21], though none have been applied to the exact problem of sequential OCR in the wild. Finally, a variety of techniques have been used to train systems from incomplete data. For instance,[7] augment existing data to improve performance, while [48] use data from other languages for the purpose of machine translation.

2.2. Optical Character Recognition

Optical Character Recognition (OCR) is the task of identifying a string of characters in an image. Modern deep-learning-based approaches to OCR generally approach this using a system that first extracts features using a convolutional neural network (CNN) [18] and then extracts the text in a subsequent decoder layer [30, 43]. In particular [43] uses the first several layers of the InceptionV3 architecture [34] to extract features which are then fed through an LSTM with a special form of attention to produce a transcription.

Domain adaptation has also been exploited in the field of sequential OCR. When the target domain consists of a large corpus such as books, the style and linguistic consistency can be leveraged to fine tune a Gaussian based model under maximum likelihood or MAP criteria using Expectation-Maximization [28, 39]. This is also analogous to speaker adaptation using a speaker-independent HMM model [10]. In more recent works [46, 40], style and content separation have been effective in adapting digit recognition from MNIST to SVHN.

Finally, we note that while many of the image classification tasks discussed above demonstrate their effectiveness on the MNIST [19] and SVHN [22] datasets, it is important to emphasize that while this task certainly falls into the category of OCR, it is much simpler than the general task of sequential OCR. MNIST and SVHN both present a single digit at a time for classification, whereas we are concerned with images in which a variable-length series of characters must be identified and classified in the correct order. It is therefore nontrivial to directly apply the domain adaptation techniques discussed above to the task of sequential OCR. For example, the system on which we perform domain adaptation contains additional Recurrent Neural Network (RNN) and attention components that are not present in any of the non-sequential OCR architectures discussed above.

3. Method

We seek to design a system that can transcribe real images in a language for which no real labeled data exists. To do this, we approach the problem from two different sides simultaneously, by using two different datasets to address the ”style” and ”content” of the images in the dataset. Specifically, we use unsupervised domain adaptation to transfer knowledge about content (the language itself) learned from synthetic data while at the same time using a simple multi-task learning scheme to make the system robust to the style of real images.

We differentiate between three sets of images available to us at training time. The first set of source images $\mathbf{X}_C = \{x_1^C, x_2^C, \ldots, x_{N_C}^C\}$ is the ”content” dataset, representing synthetic images of text in some language while $\mathbf{Y}_C = \{\mathbf{y}_1^C, \mathbf{y}_2^C, \ldots, \mathbf{y}_{N_C}^C\}$ represents the associated labels. Here, each $\mathbf{y}_i^{S_C}$ is a sequence of integers in some alphabet $\mathcal{A}_C$. For concreteness we will refer to $\mathcal{A}_C$ as the Hebrew alphabet, since that is what we will use in our experiments, but our method could hypothetically work for any language. We will generally refer to $(\mathbf{X}_{S_C}, \mathbf{Y}_{S_C})$ as S_C or as the ”content source.” Similarly, the second set of source images $\mathbf{X}_S$ and labels $\mathbf{Y}_S$ represent the style dataset; images and labels for real images of text in some other language using a different alphabet which we denote $\mathcal{A}_S$. Again, for concreteness, we'll refer $\mathcal{A}_S$ as French, but any language, even one using different glyphs, is applicable to our method. We refer to $(\mathbf{X}_S, \mathbf{Y}_S)$ as S_S, or ”style source.” We will be using $\mathbf{X}_C$ for domain adaptation and $\mathbf{X}_S$ for multi-task training.

The third domain, the target domain T, contains only images $\mathbf{X}_T = \{x_1^T, x_2^T, \ldots, x_{N_T}^T\}$. The images in $\mathbf{X}_T$ are photographs of text in the same language as those in $\mathbf{X}_C$, *i.e.* text that uses $\mathcal{A}_C$ as its alphabet. A key feature in this setup is the assumption that the domain shift between T and each of S_C and S_S is not prohibitively large. That said, S_C and S_S have very little in common with each other, since they do not overlap in either style or content.

3.1. Base Architecture

We perform our experiments by extending the architecture introduced in [43]. At a high level, this architecture consists of three components: a CNN feature extractor G_f, a Recurrent Neural Network (RNN) G_r that recurrently outputs characters by processing the extracted visual features, and a spatial attention mechanism that guides the RNN component to look at salient features, which for the purposes of our discussion we fold into the RNN network G_r.

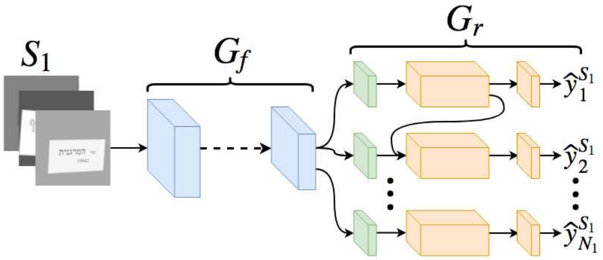

Figure 2. The baseline architecture, as described by Wojna *et al.* [43]. A feature extractor G_f is used to extract features, in this case from the content source S_C. These features are then fed into an RNN decoder G_r, which includes a spatial attention component.

Following [43], we use the first several layers of the Inception V3 CNN architecture [34] for our visual feature extractor G_f; everything up to the ”Mixed5D” module. This mapping is fully convolutional, and we denote its output features as $f = G_f(x, \theta_f)$, where θ_f represents the vector of parameters for G_f. We denote the output of the RNN and spatial attention portions of the network in [43] as $\hat{\mathbf{y}} = G_r(\mathbf{f}, \theta_r) = (\hat{y}_1, \ldots, \hat{y}_n)$. We illustrate this architecture in Figure 2.

More precisely, to compute G_r at a specific step t, we first compute a spatial attention mask α_t over visual features f, after which we compute a context vector

$$u_{t,c} = \sum_{i,j} \alpha_{t,i,j} f_{i,j,c} \tag{1}$$

which is fed into the RNN as

$$\begin{aligned} \hat{x}_t &= W_c c_{t-1} + W_{u_1} u_{t-1} \\ (o_t, s_t) &= \text{RNNStep}(\hat{x}_t, s_{t-1}) \end{aligned} \tag{2}$$

where s_t and o_t denote the internal state and output of the RNN at time t, and c_{t-1} is a one-hot encoding of the previous letter, either from the ground truth during training or as predicted during inference.

Finally, we calculate the distribution over letters as

$$\hat{o}_t = \text{softmax}(W_o o_t + W_{u_2} u_t) \tag{3}$$

and assign

$$\hat{y}_t = \arg\max_c \hat{o}_t(c). \tag{4}$$

3.2. Style Adaptation

To learn the ”style” of real imagery, we utilize a simple multi-task learning procedure, training a single network which learns the tasks of transcribing synthetic Hebrew and real French simultaneously. The end result is a system that is significantly better at transcribing real Hebrew images by implicitly exploiting the style overlap between the real French and Hebrew data. Specifically, we train a single G_f to extract features from both synthetic Hebrew street signs $x^C \in \mathbf{X}_C$ and real French street signs $x^S \in \mathbf{X}_S$, as in Figure 3 (left). The output features f are then fed through two different Attention/RNN components, G_r^C and G_r^S, to produce two sets of outputs $\hat{\mathbf{y}}^C = G_r^C(\mathbf{f}, \theta_r^C) = (\hat{y}_1^C, \ldots, \hat{y}_n^C)$ and $\hat{\mathbf{y}}^S = G_r^S(\mathbf{f}, \theta_r^S) = (\hat{y}_1^S, \ldots, \hat{y}_m^S)$, respectively, where θ_r^C and θ_r^S are the parameters for G_r^C and G_r^S. We then train both sets of data separately according to their respective cross-entropy classification losses:

$$\begin{aligned}
&\mathcal{L}_C(\mathbf{X}_C, \mathbf{Y}_C) = \\
&-\mathbb{E}_{(x^C, \mathbf{y}^C)\sim(\mathbf{X}_C, \mathbf{Y}_C)}\left[\sum_{i=1}^{N_C}\sum_{j\in A_C}\mathbb{1}_{[j=y_i^C]}\log \hat{y}_i^C\right] \\
&\mathcal{L}_S(\mathbf{X}_S, \mathbf{Y}_S) = \\
&-\mathbb{E}_{(x^S, \mathbf{y}^S)\sim(\mathbf{X}_S, \mathbf{Y}_S)}\left[\sum_{i=1}^{N_S}\sum_{j\in A_S}\mathbb{1}_{[j=y_i^S]}\log \hat{y}_i^S\right]
\end{aligned} \tag{5}$$

In practice, we actually extend these losses to be autoregressive, as described in [33], where we pass in the ground truth labels as history when we perform training.

In order to learn to label the French images in X_S, the system must learn to ignore the realistic style of the French images and focus on the content. The realistic style of the French images overlaps heavily with the style of the images in X_T, and, as a result, we hypothesize that the system also learns to ignore the realistic style of the target images, even as it learns the content from the synthetic images in X_C.

3.3. Content Adaptation

While the system described in Section 3.2 still learns the content of the Hebrew language from the synthetic data, it does nothing to specifically enforce the similarities between the source domain S_C and target domain T; in fact, it does not use T at all during training. To address this, we use the techniques of unsupervised domain adaptation to explicitly adapt the synthetic Hebrew data to the real.

3.3.1 Gradient Reversal

We seek to improve our performance in the target domain in part by directly training our system to be robust to the domain shift between the synthetic and real Hebrew data. More specifically, we wish to reduce the divergence between the features of the source and target distributions. To this end Ben-David *et al.* [1] show that the $\mathcal{H}$-divergence between a source domain $S = (\mathbf{X}_{src}, \mathbf{Y}_{src})$ and and a target domain T can be computed as

$$\hat{d}_{\mathcal{H}}(S, T) = 2\left(1 - \min_{h\in\mathcal{H}}\left[\hat{\epsilon}_S(h) + \hat{\epsilon}_T(h)\right]\right) \tag{6}$$

where $\mathcal{H}$ is the set of binary classifiers that assign 1 to samples in the source domain and 0 to samples in the target, and $\hat{\epsilon}_S(h)$ and $\hat{\epsilon}_T(h)$ are the empirical classification errors on the source and target domains. It therefore follows that we can minimize the distance $\hat{d}_{\mathcal{H}}(S, T)$ between domains by maximizing the error of all classifiers that distinguish between the domains.

Ganin *et al.* [9] achieve this goal with a technique known as gradient reversal (GR). Here, training is framed as a saddle point problem, where the system is broken into three parts. Features f are extracted using a feature extractor $f = G_f(x, \theta_f)$, and then fed into a task-specific classification branch $G_y(f, \theta_y)$ and a domain-discriminator branch $G_d(f, \theta_d)$. G_d attempts to classify the domain of any given sample as either source or target using the loss

$$\mathcal{L}_d = -\left(\sum_{x\in\mathbf{X}_S}\log G_d(x) + \sum_{x\in\mathbf{X}_T}\log(1 - G_d(x))\right). \tag{7}$$

In essence G_d is a classifier belonging to the hypothesis class $\mathcal{H}$ described above.

Thus, given some loss function $\mathcal{L}_y$ (*e.g.* cross-entropy) defined for S, we can then define an energy function

$$E(\theta_f, \theta_y, \theta_d) = \mathcal{L}_y(\mathbf{X}_{src}, \mathbf{Y}_{src}) - \lambda\mathcal{L}_d(\mathbf{X}, \mathbf{D}) \tag{8}$$

where d_i is a domain label that is equal to 1 if $x_i \in \mathbf{X}_{src}$ or 0 otherwise, $\mathbf{D} = (d_1, \ldots, d_n)$, and λ is a hyperparameter to control the trade-off between the two losses. $\hat{d}_{\mathcal{H}}(S, T)$ is then minimized at the saddle point

$$\begin{aligned}
(\hat{\theta}_f, \hat{\theta}_y) &= \arg\min_{\theta_f, \theta_y} E(\theta_f, \theta_y, \hat{\theta}_d) \\
\hat{\theta}_d &= \arg\max_{\theta_d} E(\hat{\theta}_f, \hat{\theta}_y, \theta_d).
\end{aligned} \tag{9}$$

Gradient reversal presents a simple way to optimize this saddle point problem using stochastic gradient descent. To do this, a special Gradient Reversal Layer (GRL) is added between G_f and G_d. On the forward pass of training, the GRL acts as an identity map, but on the reverse pass the GRL multiplies its gradient by -1. This effectively replaces $\frac{\partial\mathcal{L}_d}{\partial\theta_f}$ with $-\frac{\partial\mathcal{L}_d}{\partial\theta_f}$, which as [9] show is sufficient to achieve a saddle point of (8).

3.3.2 Adapting The Decoder

A naive way to apply the techniques of gradient reversal to the architecture described in Section 3.1 would be to treat

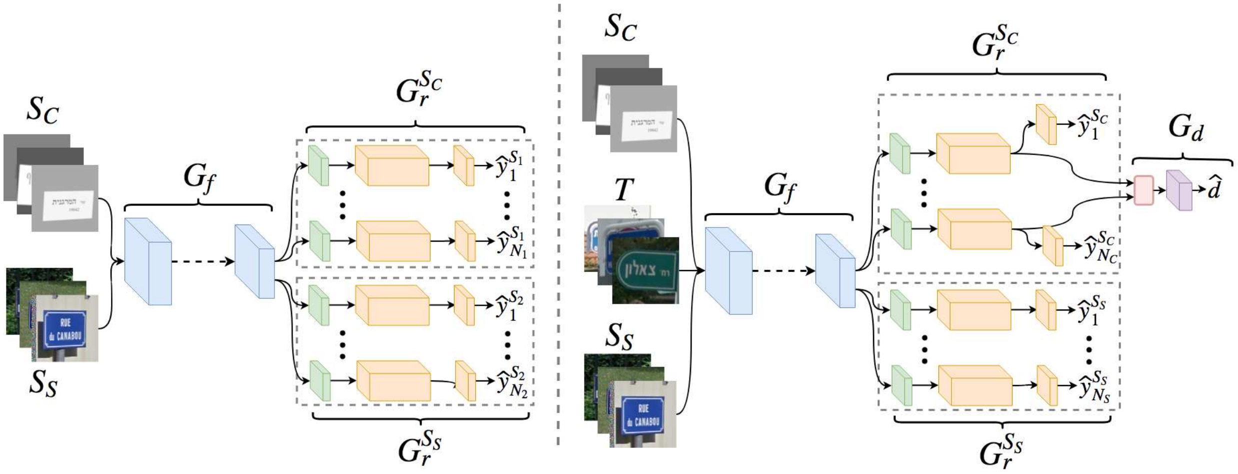

Figure 3. (Left) The configuration of the network for multi-task training. The same feature extractor G_f is used to extract features from both the content domain S_C and the style domain S_S. These features are then fed into separate RNN decoders G_r^C and G_r^S. (Right) We perform domain adaptation on the RNN decoder G_r^C by aggregating the intermediate RNN values s_t and using gradient reversal on a domain classifier that selects between S_C and the target domain T. We do not perform any adaptation with respect to S_S beyond what the network learns through multi-task training.

G_r^C the same we treated G_y in Section 3.3.1: as a simple classifier that acts on the features extracted by G_f. Informally, the intuition is that we would be adapting the visual features to become robust to the change in style between real and synthetic.

However, we explored multiple architectures using this approach, and we experimentally found that the main benefit of domain adaptation is in its ability to improve understanding of the content, and less so in its ability to build robustness to the style. Under this hypothesis, it makes more sense to perform domain adaptation in the RNN portion of the network, where the language structure is processed.

Thus, we introduce a method that directly adapts the RNN components of the system, which we illustrate in Figure 3. Specifically, we leave most of G_r unchanged, but for each RNN step t we introduce a new value

$$v = \text{GRL}(\max_t s_t) \tag{10}$$

where s_t is the internal state of the RNN, as introduced in Equation 2.We experimentally found that it was essential to aggregate the RNN output using maximization, as averaging or using a softmax attention-based aggregation did not result in a system that performed better than the baseline.

We then use a domain-discriminator G_d on the output, which we calculate as

$$\begin{aligned} w &= W_{d_2}\text{relu}(W_{d_1}v + b_{d_1}) + b_{d_2} \\ \hat{d} &= \text{softmax}(W_{d_3}w) \end{aligned} \tag{11}$$

where $W_{d_1}, W_{d_2}, W_{d_3}, b_{d_1}$, and b_{d_2} are all parameters learned by the network.

We can then define $\mathcal{L}_d$ as it was defined in Equation 7, and our final content energy function becomes

$$E(\theta_f, \theta_r, \theta_d) = \mathcal{L}_C(\mathbf{X}_C, \mathbf{Y}_C) - \lambda\mathcal{L}_d(\mathbf{X}, \mathbf{D}). \tag{12}$$

This modification is essential for success once data from S_S is added to the system, since it performs adaptation on a portion of the network that is not directly enhanced by the additional data.

When combined with multi-task learning, our final energy function becomes

$$\begin{aligned} E(\theta_f, \theta_r^S, \theta_r^C, \theta_d) &= \\ \mathcal{L}_S(\mathbf{X}_S, \mathbf{Y}_S) + \mathcal{L}_C(\mathbf{X}_C, \mathbf{Y}_C) &- \lambda\mathcal{L}_d(\mathbf{X}, \mathbf{D}). \end{aligned} \tag{13}$$

During each step of training, we optimize all three components of this loss in a single batch. The complete architecture with all components and unsupervised domain adaptation applied to the decoder is illustrated in Figure 3. When training, we use $\lambda = 0.5$, a value which we determined experimentally.

4. Experiments

The setup we suggest is both unique and highly specific, so in order to properly evaluate it we introduce two new datasets containing real and synthetic images of Hebrew street name signs. Used in conjunction with the existing FSNS street name dataset, we illustrate the effectiveness of both our domain adaptation technique and a simple multi-task learning approach. We then demonstrate that using both techniques together performs better than using only a single technique, and provide a detailed empirical analysis of our results.

Trained on:	SynHSNS	FSNS	HSNS	SynHSNS Acc.	FSNS Acc.	**HSNS Acc.**
FSNS Baseline		✓		*N/A*	64.34%	*N/A*
Baseline	✓			94.68%	*N/A*	18.49%
Fine-Tuning	✓	✓		89.81%	64.34%	29.56%
Multi-Task Training (MT)	✓	✓		94.68%	64.26%	36.54%
Domain Adaptation (DA)	✓		✓	93.57%	*N/A*	38.64%
DA + MT	✓	✓	✓	91.47%	63.39%	**50.16%**

Table 1. Full-sequence accuracy on the test data of each dataset for the various systems we discuss in this paper. Check marks indicate which datasets were available during training for each experiment. The most important accuracy results are those of HSNS, the target dataset for our system. We also report performance on the SynHSNS and FSNS datasets, though we note that optimizing performance on these datasets is not the goal of our system. Still, results indicate our system does not completely destroy performance on these secondary tasks, a fact which may be useful in building a more general system.

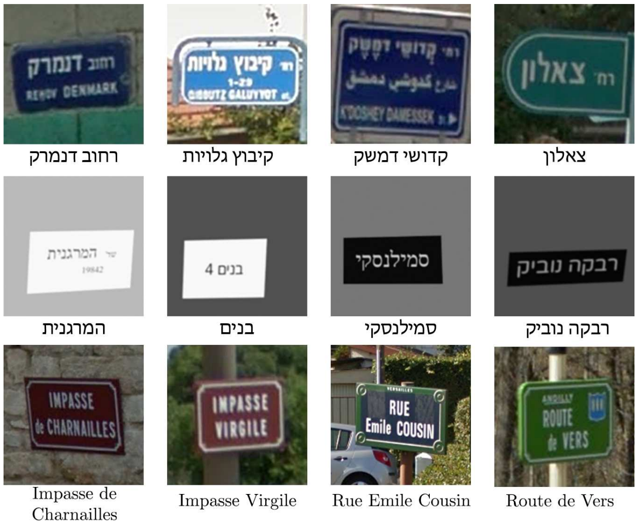
רחוב דנמרק
קיבוץ גלויות
קדושי דמשק
צאלון
המרגנית
בנים
סמילנסקי
רבקה נוביק
Impasse de Charnailles
Impasse Virgile
Rue Emile Cousin
Route de Vers

Figure 4. Sample images from the HSNS, SynHSNS, and FSNS datasets, displayed on the top, middle, and bottom rows respectively.

Following [43], the metric which we report for all of the following techniques is full sequence accuracy, wherein a sample is considered correctly classified only if every character is predicted correctly.

Unfortunately, in the absence of an alternate yet reliable means of performing hyperparameter optimization, we follow [3] and perform out experiments directly on a small set of validation data. We understand that this is not optimal, as the argument can be made that any labeled data available at training time should be used during training, and we therefore hope that in the future the research community will present an alternative means for validation in the unsupervised domain adaptation scenario. For now, we leave the development of such a metric to future work.

4.1. Datasets

4.1.1 Hebrew Street Name Signs

For our target dataset, we collected approximately 92,000 cropped images of Hebrew street name signs from Israel. We divided these into three different splits of 89,936 test images, 899 validation images and 903 test images, of which only the validation and test images have labels. When splitting the dataset we maintained a geographic distance of at least 100 meters between the location of any training/validation and test images, to ensure that the system does not have exposure to any test signs while training or performing validation. All of these images are 150 × 150 resolution.

Many Hebrew street signs include certain prefixes that translate to words such as ”street,” ”road,” ”avenue,” *etc*. More often than not, these words are written in a much smaller font than the rest of the sign, making them illegible at 150 × 150 resolution. Since many Israeli map services don’t include these prefixes, we also decided to exclude them from the transcriptions.

We will be releasing this data as the Hebrew Street Sign Names (HSNS) dataset. Samples from this dataset are shown in Figure 4. Although the images are collected in full RGB color and will be released as such, in all of the tests that follow we convert each image into greyscale so as to maintain consistency with our synthetic images, which we describe below.

4.1.2 Synthetic Hebrew Street Name Signs

We elected to use a relatively simple scheme for generating synthetic data. This decision is motivated both by the difficulty of generating more sophisticated natural-looking synthetic data, and by the observation that the synthetic data need only contain the same content as the target data to be useful, as we can use other methods to address the style.

Our synthetic images therefore consist of only straightforward text rendering, a box placed behind the text, a perspective transform, and some slight blur. When rendering the text, we randomly select from one of nineteen different Hebrew fonts. In some cases, we randomly add English text or numbers below or above the Hebrew, which we don’t include in the ground truth transcriptions. The size and placement of the text, the parameters of perspective transform, and the amount of blur are all selected randomly. The actual text itself is selected from a list of real Israeli street names.

To better match the text distribution of HSNS, we also randomly add small font prefixes which translate to the Hebrew words for ”street”, ”road”, ”avenue”, *etc*. We found that these prefixes were essential for performance, since they are often included in real images but are often too small to read, and including them in the synthetic data signals to the system that they do not need to be transcribed. We generate all images at 150×150 resolution.

In order to simplify the text generation process further, all synthetic images are generated in greyscale. This greatly simplifies the generation process by making it much easier to produce images in a realistic color range. The exact colors for each image are selected randomly, though we do enforce a minimum amount of contrast between the text and the box behind it. We used a solid color for the background, because preliminary tests using more complicated backgrounds (*e.g*. Gaussian noise) did not yield any differences in performance.

We generate roughly 430,000 synthetic images for training, and another 10,000 each for evaluation and testing. For sample images, see Figure 4. We release this data along with HSNS as the Synthetic Hebrew Street Name Signs dataset (SynHSNS).

4.1.3 French Street Name Signs

In addition to the two Hebrew-language datasets above, we also use the existing French Street name Signs (FSNS) dataset [30] for our multi-task learning. FSNS contains roughly 1 million training, 20,000 evaluation, and 16,000 test samples of French street name signs, each containing between one and four views of the same sign at 150×150 resolution. For consistency with HSNS and SynHSNS, we only use one of these views during training, taking whichever view is listed first. Similarly, we maintain consistency with SynHSNS by converting each image into greyscale. Sample images from the original FSNS dataset can be seen in Figure 4.

4.2. Implementation Details

With the exception of the fine-tuning experiment described in Section 4.3.2, all of the training is performed with a learning rate of 0.0047 using Stochastic Gradient Descent with a momentum value of 0.75. We train for 800,000 steps with a batch size of 15 for each domain actually used during training. When domain adaptive components are present, we turn them on starting at 20,000 steps and compute the loss in Equations 12 and 13 with the value $\lambda = 0.5$. All input images are 150×150 resolution, consistent with the resolution of the data in all three datasets.

4.3. Domain Adaptation and Joint Training

4.3.1 Baselines

In order to show the efficacy of our system, we need to demonstrate that our methods perform better than a naive approach. We therefore define our baseline to be the test performance of HSNS on a system trained exclusively on the SynHSNS data. Results of this experiment are reported in Table 1 as “Baseline”.

Table 1 also includes for reference the performance of a system trained exclusively on the version of FSNS used in all experiments, listed as “FSNS Baseline”. As described above, our usage of FSNS differs from the standard usage in that we have only used one of the up to four possible views for each sign, and we have removed all color from the image. Therefore, while the number we report here for FSNS is lower than the number [43] reported for essentially the same system, it is important to note that the two experiments were not performed on precisely the same data. We would also like to emphasize that our goal is not to optimize performance on FSNS, but rather on HSNS, and therefore these numbers are included only for reference.

4.3.2 Multi-Task Learning Baselines

We report results on the multi-task learning scheme described in Section 3.2, where we train on both the SynHSNS and the FSNS datasets simultaneously. We report this in Table 1 as “Multi-Task Training (MT)”. As with the baselines described above, HSNS data is not seen during training, yet we still achieve 36.54% accuracy on the HSNS test set. Thus, simply by learning to parse real French images, the model achieves an 18 point improvement when parsing real Hebrew images, supporting our hypothesis that the system is better able to understand the realistic style of the Hebrew data just from seeing the real French data.

In addition to the joint training scheme described above, we also evaluate our method on a simple fine-tuning scheme, listed in Table 1 as “Fine-Tuning”. In this scheme, we first train the entire system on FSNS alone for 800,000 steps. Then we replace $G_r^{S_S}$ with $G_r^{S_C}$ and train the network for an additional 66,000 steps at a reduced learning rate of 0.002 (additional steps of training did not increase the performance on HSNS). Performance results of both methods are reported in Table 1. We see that multi-task learning is superior to fine-tuning, probably because the additional training phase reduces some of the benefits gained by the French data in the first phase.

4.3.3 Domain Adaptation

To evaluate the effectiveness of gradient reversal, we again perform two experiments, both based on the RNN-centric domain adaptation described in Section 3.3.2.

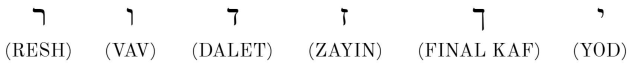

Figure 5. Examples of Hebrew letters that are visually hard to distinguish.

The first of these experiments, denoted in Table 1 as "Domain Adaptation (DA)", performs domain adaptation on the RNN portion of the network $G_r^{S_C}$, explicitly optimizing the loss in Equation 12 using only HSNS and SynHSNS as input, *i.e*. the architecture shown in Figure 3 (right) with $G_r^{S_S}$ and the FSNS input removed.

Our second experiment, denoted "DA+MT", uses all three datasets as input, and is a test of our full system as illustrated in Figure 3 (right). This experiment stands out as the only one to make use of all three datasets available to us.

From these experiments, we see that domain adaptation alone, just between HSNS and SynHSNS, is enough to yield a performance increase from 18.49% to 38.64%. What's perhaps more interesting is that combining this with multi-task learning takes the performance to 50.16%. In particular, the marginal increase from "DA" to "DA+MT" (about 11 points) is not trivial. Similarly, the jump from from "MT" to "DA+MT" (about 14 points) is also quite substantial.

We believe that this supports our hypothesis that domain adaptation targets the content while multi-task learning targets the style, because it suggests that the improvements provided by each technique are mostly disjoint, *i.e*. domain adaptation is helping for a different reason than multi-task learning. If these techniques weren't complementary, and both "DA" and "MT" improved performance by addressing the same features of the target, then we might expect to see a smaller marginal improvement when we used both of them together, since it would suggest that there is very much "overlap" between the techniques.

4.3.4 Analysis of Errors

The Hebrew alphabet is a challenging set of characters - it has multiple characters which are hard to distinguish both for humans (untrained or non-native Hebrew speakers) and computers, such as those illustrated in Figure 5. There are several others, but just these account for 22.7% (1596 out of 7013) of all printable characters of our validation set. It is interesting to note that all model configurations confuse these characters and the rate of confusion does not change drastically from one configuration to another (for instance, the "MT" model confuses VAV for YOD 40/894 times, and "MT+DA" confuses VAV for YOD 41/894 times).

Another interesting observation is the way that the network learns to represent the features for white space, specifically the NULL character (which terminates a sequence) and the SPACE character. Table 6 shows t-SNE plots [38] for the character embedding of the "MT" and "DA+MT" architectures. We observe that as the network's performance increases, the NULL and SPACE characters develop clusters that are more separate from the other clusters. We see this confusion when we look at performance numbers as well: "MT" classifies SPACE as NULL 88/620 times, while "MT+DA" makes this mistake only 45/620 times. We believe this observation can be explained by looking at the area around characters.

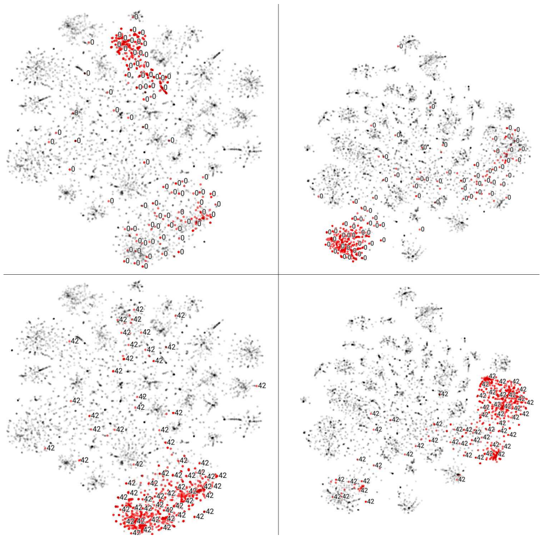

Figure 6. Visualization of individual character predictions in a network with only multi-task learning (left) and with multitask and DA (right). The numbers refer to clusters of individual characters in the Hebrew alphabet. Points in red on the top correspond to the SPACE character, while points in red on the bottom correspond to the NULL (end of sequence) character.

We posit that the main difference in visual appearance between synthetic and natural images is the way regions without characters look. In a tight crop around any character there will not be that much difference between real and synthetic images, but our model operates on a large context where the area around the text may be too distracting for the model to easily ignore. The area without characters is exactly the area to which NULL and SPACE refer.

5. Conclusion

In this paper, we have explored different approaches to teaching a system to perform sequential OCR on photos of street name signs in a language for which there exists no labeled data. To do this, we introduce two new datasets: the SynHSNS dataset of synthetic Herbrew street sign names, and the HSNS dataset of real unlabeled Hebrew street sign names. Ultimately, we demonstrate that our approach, which leverages existing data in other languages and easily-generated synthetic data in the same language, can be used to greatly improve performance in the target domain by transferring information about both style and content.

References

[1] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan. A theory of learning from different domains. *Machine Learning*, 2010. 4

[2] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 95–104, July 2017. 2

[3] K. Bousmalis, G. Trigeorgis, N. Silberman, D. Krishnan, and D. Erhan. Domain separation networks. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 343–351. Curran Associates, Inc., 2016. 2, 6

[4] Y. Chen, W. Li, C. Sakaridis, D. Dai, and L. V. Gool. Domain adaptive faster r-cnn for object detection in the wild. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2

[5] C. Chu and R. Wang. A survey of domain adaptation for neural machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1304–1319. Association for Computational Linguistics, 2018. 2

[6] H. Daume III. Frustratingly easy domain adaptation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 256–263. Association for Computational Linguistics, 2007. 2

[7] M. Fadaee, A. Bisazza, and C. Monz. Data augmentation for low-resource neural machine translation. In *ACL*, 2017. 2

[8] Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pages 1180–1189. JMLR.org, 2015. 2

[9] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. Domain-adversarial training of neural networks. *J. Mach. Learn. Res.*, 17(1):2096–2030, Jan. 2016. 2, 4

[10] J. . Gauvain and C.-H. Lee. Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains. *IEEE Transactions on Speech and Audio Processing*, 2(2):291–298, April 1994. 2

[11] X. Glorot, A. Bordes, and Y. Bengio. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, pages 513–520, USA, 2011. Omnipress. 2

[12] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014. 2

[13] A. Gretton, A. Smola, J. Huang, M. Schmittfull, K. Borgwardt, and B. Schölkopf. *Covariate shift and local learning by distribution matching*, pages 131–160. MIT Press, Cambridge, MA, USA, 2009. 2

[14] P. Haeusser, T. Frerix, A. Mordvintsev, and D. Cremers. Associative domain adaptation. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2784–2792, Oct 2017. 2

[15] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. Efros, and T. Darrell. CyCADA: Cycle-consistent adversarial domain adaptation. In J. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1989–1998, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR. 2

[16] J. Hoffman, D. Wang, F. Yu, and T. Darrell. Fcns in the wild: Pixel-level adversarial and constraint-based adaptation. *CoRR*, abs/1612.02649, 2016. 2

[17] N. Inoue, R. Furuta, T. Yamasaki, and K. Aizawa. Cross-domain weakly-supervised object detection through progressive domain adaptation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012. 1, 2

[19] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pages 2278–2324, 1998. 2

[20] M. Long, Y. Cao, J. Wang, and M. I. Jordan. Learning transferable features with deep adaptation networks. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pages 97–105. JMLR.org, 2015. 2

[21] A. Mohammadian, H. Aghaeinia, F. Towhidkhah, and S. Seyyedsalehi. Subject adaptation using selective style transfer mapping for detection of facial action units. *Expert Systems with Applications*, 56, 03 2016. 2

[22] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011. 2

[23] K. Saito, Y. Ushiku, and T. Harada. Asymmetric tri-training for unsupervised domain adaptation. In D. Precup and Y. W. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 2988–2997, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR. 2

[24] K. Saito, K. Watanabe, Y. Ushiku, and T. Harada. Maximum Classifier Discrepancy for Unsupervised Domain Adaptation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2

[25] K. Saito, S. Yamamoto, Y. Ushiku, and T. Harada. Open set domain adaptation by backpropagation. In *The European Conference on Computer Vision (ECCV)*, September 2018. 2

[26] S. Sankaranarayanan, Y. Balaji, C. D. Castillo, and R. Chellappa. Generate to adapt: Aligning domains using generative adversarial networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2

[27] S. Sankaranarayanan, Y. Balaji, A. Jain, S. Lim, and R. Chellappa. Unsupervised domain adaptation for semantic segmentation with gans. *CoRR*, abs/1711.06969, 2017. 2

[28] P. Sarkar and G. Nagy. Style-consistency in isogenous patterns. In *Proceedings of Sixth International Conference on Document Analysis and Recognition*, pages 1169–1174, Sept 2001. 2

[29] R. Shu, H. Bui, H. Narui, and S. Ermon. A DIRT-t approach to unsupervised domain adaptation. In *International Conference on Learning Representations (ICLR)*, 2018. 2

[30] R. Smith, C. Gu, D.-S. Lee, H. Hu, R. Unnikrishnan, J. Ibarz, S. Arnoud, and S. Lin. End-to-end interpretation of the french street name signs dataset. In *ECCV Workshops*, 2016. 2, 7

[31] B. Sun, J. Feng, and K. Saenko. Return of frustratingly easy domain adaptation. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, pages 2058–2065. AAAI Press, 2016. 2

[32] B. Sun and K. Saenko. Deep coral: Correlation alignment for deep domain adaptation. In G. Hua and H. Jégou, editors, *Computer Vision – ECCV 2016 Workshops*, pages 443–450, Cham, 2016. Springer International Publishing. 2

[33] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. 4

[34] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826, 2016. 2, 3

[35] C. Thomas and A. Kovashka. Artistic object recognition by unsupervised style adaptation. In C. V. Jawahar, H. Li, G. Mori, and K. Schindler, editors, *Computer Vision – ACCV 2018*, pages 460–476, Cham, 2019. Springer International Publishing. 2

[36] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. Adversarial discriminative domain adaptation. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2962–2971, 2017. 2

[37] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell. Deep domain confusion: Maximizing for domain invariance. *CoRR*, abs/1412.3474, 2014. 2

[38] L. van der Maaten and G. E. Hinton. Visualizing data using t-sne. 2008. 8

[39] S. Veeramachaneni and G. Nagy. Adaptive classifiers for multisource ocr. *Document Analysis and Recognition*, 6(3):154–166, Mar 2003. 2

[40] R. Volpi, P. Morerio, S. Savarese, and V. Murino. Adversarial feature augmentation for unsupervised domain adaptation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2

[41] Z. Wang, J. Yang, H. Jin, E. Shechtman, A. Agarwala, J. Brandt, and T. S. Huang. Real-world font recognition using deep network and domain adaptation. *CoRR*, abs/1504.00028, 2015. 2

[42] Z. Wang, J. Yang, H. Jin, E. Shechtman, J. B. Aseem Agarwala, and T. S. Huang. Decomposition-based domain adaptation for real-world font recognition. 2

[43] Z. Wojna, A. N. Gorban, D.-S. Lee, K. Murphy, Q. Yu, Y. Li, and J. Ibarz. Attention-based extraction of structured information from street view imagery. *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, 01:844–850, 2017. 1, 2, 3, 6, 7

[44] Z. Yang, Z. Hu, C. Dyer, E. P. Xing, and T. Berg-Kirkpatrick. Unsupervised text style transfer using language models as discriminators. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 7287–7298. Curran Associates, Inc., 2018. 2

[45] X.-Y. Zhang and C.-L. Liu. Writer adaptation with style transfer mapping. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35:1773–1787, 2013. 2

[46] Y. Zhang, W. Cai, and Y. Zhang. Separating style and content for generalized style transfer. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2

[47] Y. Zhang, P. David, and B. Gong. Curriculum domain adaptation for semantic segmentation of urban scenes. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 2

[48] B. Zoph, D. Yuret, J. May, and K. Knight. Transfer learning for low-resource neural machine translation. pages 1568–1575, 01 2016. 2