STRIVE: Scene Text Replacement In Videos

Vijay Kumar B G *[1], Jeyasri Subramanian[2], Varnith Chordia[3], Eugene Bart[2], Shaobo Fang[4], Kelly Guan[5], and Raja Bala[3]

[1]NEC Laboratories, America
[2]Palo Alto Research Center
[3]Amazon
[4]Work done at PARC
[5]Stanford University

Abstract

We propose replacing scene text in videos using deep style transfer and learned photometric transformations. Building on recent progress on still image text replacement, we present extensions that alter text while preserving the appearance and motion characteristics of the original video. Compared to the problem of still image text replacement, our method addresses additional challenges introduced by video, namely effects induced by changing lighting, motion blur, diverse variations in camera-object pose over time, and preservation of temporal consistency. We parse the problem into three steps. First, the text in all frames is normalized to a frontal pose using a spatio-temporal transformer network. Second, the text is replaced in a single reference frame using a state-of-art still-image text replacement method. Finally, the new text is transferred from the reference to remaining frames using a novel learned image transformation network that captures lighting and blur effects in a temporally consistent manner. Results on synthetic and challenging real videos show realistic text transfer, competitive quantitative and qualitative performance, and superior inference speed relative to alternatives. We introduce new synthetic and real-world datasets with paired text objects. To the best of our knowledge this is the first attempt at deep video text replacement.

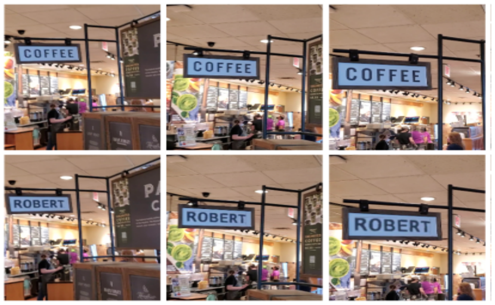

Figure 1: Our method replaces the scene text in the original video (”COFFEE” in upper row) with a personalized string (”ROBERT” in bottom row) while preserving the original geometry, appearance, and temporal properties.

1. Introduction

We address the problem of realistically altering scene text in videos. Our primary application is to create personalized content for marketing and promotional purposes. An example would be to replace a word on a store sign with a personalized name or message, as shown in Figure 1. Other applications include language translation, text redaction for privacy, and augmented reality. For research purposes, the ability to realistically manipulate scene text also enables augmentation of datasets for training text detection, recognition, tracking, erasure, and adversarial attack detection. Traditionally text editing in images is performed manually by graphic artists, a process that typically entails a long painstaking process to ensure the original geometry, style and appearance are preserved. For videos, this effort would be considerably more arduous.

Recently several attempts have been made to automate text replacement in still images based on principles of deep style transfer ([22], [1], [31], [26], [29]). We leverage this progress to tackle the problem of text replacement in videos. In addition to the challenges faced in the still image case, video text replacement must respect temporal consistency and model effects such as lighting changes, blur induced by camera and object motion. Furthermore, over the course of the video, the pose of the camera relative to the text object can vary widely, and hence text replacement must be able to handle diverse geometries. A logical method to solving the problem would be to train an image-based text style trans-

*Corresponding author: bg.vijay.k@gmail.com fer module on individual frames, while incorporating temporal consistency constraints in the network loss. The problem with such an approach is that the network performing text style transfer (an already non-trivial task) is now additionally burdened with handling geometric and motion-induced effects encountered in video. We show in experiments that current still-image text replacement techniques such as [26, 13] do not robustly handle such effects. We therefore take a different approach. We first extract text regions of interest (ROI) and train a spatio-temporal transformer network (STTN) to frontalize the ROIs in a temporally consistent manner. We then scan the video and select a reference frame with high text quality, measured in terms of text sharpness, size, and geometry. We perform still-image text replacement on this frame using a state-of-art method SRNet [26] trained on video frames. We then transfer the new text onto other frames with a novel *text propagation module* (TPM) that takes into account changes in lighting and blur effects with respect to the reference frame. TPM takes as input the reference and current frame from the original video, infers an image transformation between the pair, and applies it to the altered reference frame generated by SRNet. Crucially, TPM takes temporal consistency into account when learning pairwise image transforms. Our framework, dubbed STRIVE (**S**cene **T**ext **R**eplacement **I**n **V**id**E**os) is summarized in Figure 2.

To our knowledge this is the first attempt at replacing scene text in videos. We make the following contributions:

1) A modular pipeline that disentangles the problem of text replacement in a single reference frame from modeling the flow of the replaced text within the scene over time. Such a parsing of the problem into simpler subtasks serves to both simplify training and reduce computations during inference.

2) A learned parametric differential image transformation that captures temporal photometric changes between a pair of aligned ROIs in the original video, and applies it to ROIs in the text-altered video. The transform consists of a learnable blur/sharpness operator, and is trained on synthetic data and fine-tuned via self-supervision on real-world images.

3) New synthetic and real-world datasets comprising a diversity of annotated scene text objects in videos. A subset of the videos comprise triplets of ROIs with aligned source text, target text, and plain background. The datasets are available at `https://striveiccv2021.github.io/STRIVE-ICCV2021/`.

2. Related Work

2.1. Style Transfer

Our approach has its roots in deep neural style transfer [5], [12]. Specifically pix2pix [13] forms the backbone for image-to-image transfer tasks and will be used as a baseline in our experiments. Extensions to style transfer for videos have been proposed by several researchers, whereby temporal consistency constraints are added to the style losses applied on individual frames [11], [8]. We disentangle style transfer from geometric and photometric variations encountered in video, and apply temporal consistency constraints on the latter.

2.2. Scene Text Detection, Recognition

Text replacement relies upon successful text detection and recognition. As expected, deep learning methods define the state of art for both image and video input [36],[24],[3]. We use the video text detection module within the Amazon *ReKognition* toolbox to extract text ROIs for our pipeline. Related efforts to synthesize realistic scene text for training models include [33], [16], [34].

2.3. Scene Text Replacement

Recently, deep neural techniques have been proposed for editing scene text in still images. These fall into two categories: techniques that replace individual characters ([22], [1], [31], [30], [32]), and those that replace entire words ([26], [29]). While the approaches vary in the types of effects that can be modeled, the main steps are: i) inpainting to erase existing text [23]; ii) transfer of input text style to new characters or words; iii) a fusion step to combine foreground and background regions for realistic outputs. Since it is difficult to acquire paired text data in real world scenes, existing methods are trained on synthetic datasets. The closest application we are aware of is the camera mode in the Google Translate app, which overlays language translations on scene text in the camera preview mode. Since the primary purpose is translation, there is no attempt to strictly match the appearance of the original text.

2.4. Learning Image Transformations

Our text propagation module (TPM) learns photometric transformations between a reference and a non-reference frame. Several works have tackled a similar problem in the context of image enhancement. Gharbi et al. [6] learn to automatically enhance images by training a deep bilateral transform network from paired (pre- and post-enhancement) training data. Related works [7], [2] learn to predict a parameterized transformation from a pair of low-resolution images and apply this to the high resolution version. The transform models complex photo enhancement operators with a ”recipe” of local affine transforms. We draw inspiration from these techniques to learn a parametric transform between reference and non-reference frames in the original video, with several key differences. First, in previous works, the transformation learned from image I_1 to its enhanced version I_2 is then applied on a high-resolution version of I_1. Each transform is thus intimately tied to a specific image. In our case the transform learned

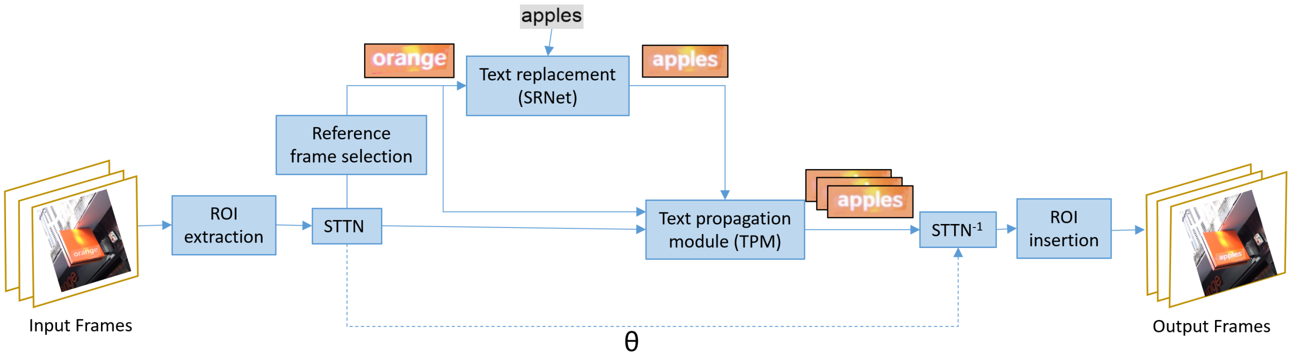

Figure 2: Overview of STRIVE. Text region of interest (ROI) is extracted from each frame and frontalized using a spatiotemporal transformation network (STTN) with parameters θ. Next a reference frame is selected and text is replaced using SRNet. The new text is transferred to other frames via a novel text propagation module (TPM), and reinserted into the frame after reverting to the original pose via $STTN^{-1}$.

from an image pair with source scene text is applied to a *different* image with target text, and hence its structure must be abstracted from the original image content. Secondly, while previous efforts are aimed at learning image enhancement operations, our purpose is to model changes in lighting and camera motion encountered across video frames, which often include distortions such as image blurring. Finally our transformations must exhibit temporal consistency, which is not applicable for still-image transforms.

2.5. Image blur estimation and correction

Techniques have been proposed for using deep CNNs to estimate blur kernels from images and videos with the goal of blind image deblurring [28], [27], [35]. Our blur estimation in TPM adopts a similar approach, however with a different purpose of estimating a *differential* blur transform between a pair of images, rather than estimating blur in an absolute sense from a single image.

3. Methodology

Referring to Fig. 2, we first extract tight ROIs (i.e. bounding boxes) for the source text from the input video using the Amazon *Rekognition* API. All operations described next are carried out on the ROIs.

3.1. Reference Frame Selection

We select a single reference frame for text replacement. To ensure successful text style transfer, we desire a frame wherein the source text clearly legible, of high contrast, and maximally frontal in pose. We compute four metrics on the text ROI. 1) Only samples with optical character recognition (OCR) confidence greater than 0.99 (as reported by *ReKognition*) are considered. This eliminates excessively blurry, distorted or occluded text objects. 2) Image sharpness is measured as the variance of the Laplacian of the luminance image [21], and the top 10 frames with the highest sharpness scores are selected. 3) The image is binarized using Otsu's algorithm [18] and the normalized interclass variance s_1 between foreground and background regions is obtained as a measure of text contrast. 4) The ratio s_2 of the area of the tight ROI bounding box to the area of the subsuming axis-aligned rectangle is calculated as an estimate of frontal pose. A composite text quality score is computed as $\alpha_1 s_1 + \alpha_2 s_2$ and the frame with the highest score is selected as the reference frame. Both α_1 and α_2 are heuristically chosen based on visual evaluation of replaced text on random videos.

3.2. Pose Normalization

Text objects in videos can undergo diverse geometric distortions due to varying object pose relative to the camera. To minimize the effect of this distortion on text style transfer, we normalize and align the source text in all ROIs to a canonical frontal pose before the replacement operation. This enables both robust text style transfer on the reference frame, and the propagation of the replaced text onto the remaining frames via learned image transforms. We make a simplifying assumption that the scene text is on a planar surface. This covers many common cases such as street and store signage, banners, etc. Under the planar assumption, pose alignment is accomplished via a perspective transform. We adopt and extend the Spatial Transformer Network [14] (STN) as a learned approach to perspective correction that is computationally efficient at runtime. STN predicts the parameters θ of a geometric correction transform via a localization network, and applies the transform to the image via grid generation and resampling operators. The original STN is trained as part of a supervised classification task. We adopt the same network architecture, but instead explicitly train it to produce temporally consistent frontal ROIs. We obtain binary masks of text ROIs with Mask R-CNN [9]. The training samples comprise a stack of distorted input ROI masks and a frontal ROI mask of synthetic text serving as the target label. The net is trained on the following loss:

$$L_{STTN} = L_\theta + \lambda_1 L_{pix} + \lambda_2 L_t \tag{1}$$

where the first two terms are from the original model: L_θ is the mean-squared error (MSE) between the true and predicted homography parameter vector θ, and L_{pix} is the pixelwise MSE between predicted and true text ROIs. We use L_2 norm for the above losses following the practice of Nguyen *et al*. [17] and DeTone *et al*. [4]; this choice produced good results in our experiments. Additionally we introduce temporal consistency loss L_t, defined as the MSE between θ for adjacent video frames: $L_t = \sum_j |\theta_i - \theta_j|^2$, where index i denotes current frame and the summation is over a number of neighboring frames $j \neq i$. This term ensures that perspective correction parameters vary smoothly over adjacent frames. Fig. 2 shows how the spatiotemporal transform network ($STTN$) is incorporated into the overall framework. After text replacement, the ROI is sent through the inverse perspective transform ($STTN^{-1}$) to restore it to the original scene geometry and inserted into the original frames to produce the output video.

3.3. Text Replacement in Reference Frame

We select SRNet [26] to replace text in the reference frame. In principle, any state-of-art still-image text replacement technique can be used. SRNet takes the input ROI and the target text in the form of a mask, and executes subnetworks for background generation, foreground creation via transfer of style from source to target text, and blending of background and foreground to produce the target text ROI. We train SRNet on video frames from our datasets. During training we introduce additional augmentations for perspective distortion, and motion and out-of-focus blur encountered in videos.

3.4. Text Propagation

The main novel element of our work is text propagation from reference frame to the rest of the video via TPM. Our key insight is to avoid repeatedly performing text replacement on every frame, and instead to solve the simpler problem of learning the changes in text appearance over the video. We posit that the image transformation between two video frames within a localized text ROI can be adequately modeled in terms of simple parametric changes in lighting and image sharpness due to changes in camera/object/lighting conditions over time. An advantage of our approach is that we are able to use self-supervision to learn the parameters of the model without relying on a large set of paired videos and labels. In detail, let I_R and I_i be respectively the ROIs from the reference and i^{th} frame containing the source text in the input video. Similarly let I'_R and I'_i be corresponding ROIs of the target text in the output video. All ROI's are pose-corrected and aligned by STTN, and are scaled to a fixed canonical image dimension prior to TPM processing. A parametric transform is learned between I_R and I_i, and is then applied to I'_R to predict I'_i. This transform comprises two stages, a lighting correction module (LCM) and differential blur prediction network (BPN), as shown in Fig. 3.

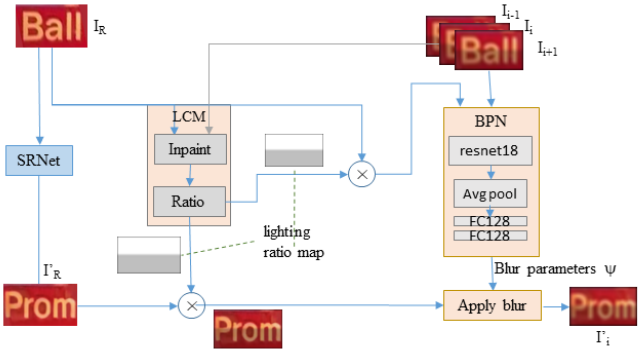

Figure 3: Text Propagation Module (TPM) derives a local image transform between reference and non-reference ROIs in the original video (”Ball”) and applies the transform to the reference ROI (”Prom”) in the altered video. The transform comprises a Lighting Correction Module (LCM) followed by a Blur Prediction Network (BPN).

The LCM captures appearance differences between reference and current ROI due to changes in illumination, including shadows and shading. Since the color of an object is a product of its reflectance and the illumination, we surmise that to first order, changes in light reflected off a fixed text object can be modeled by independent channel-wise scaling of R, G, and B channels in a spatially varying manner. Namely, changes in lighting between two aligned ROIs can be obtained from their ratio I_i/I_R, which is then multiplied by I'_R to obtain the lighting-corrected output for I'_i. In practice despite the fact that I_R and I_i are aligned via STTN, even minor imperfections in alignment can result in gross errors in the ratio map particularly around text edges. Such errors become even more noticeable when applying ratio correction to *new* text in I'_R. To address this issue, we assume that scene text is commonly placed on a smooth background, and apply inpainting to obtain estimates of the plain background, denoted I_{Rp} and I_{ip}. We use the deep inpainting module within SRNet for this purpose. The ratio of the inpainted versions defines a multiplicative correction to I'_R (see Fig. 3). In practice we compute the ratio $(I_{ip} + \epsilon)/(I_{Rp} + \epsilon)$, where a small ϵ avoids singularities near zero. The implicit assumption with the ratio model is that the reflective properties of the foreground text and background are similar. Further, to ensure temporal robustness we compute a weighted average of inpainted ROIs over N neighboring frames before computing the ratio. The latter is multiplied by both the original and altered reference frames I_R and I'_R to produce lighting-corrected versions that are then passed to the blur prediction network.

BPN is a novel CNN-based method for predicting a transformation between a pair of images that can result from spatial aberrations, including motion blur, out-of-focus blur, and resolution differences due to varying distances between camera and text object. We model possible frame-to-frame distortions within a local text ROI using the following transformation:

$$I_i(x, y) = (1 + w)I_R(x, y) - wI_R(x, y) * G_{\sigma,\rho}(x, y) \quad (2)$$

where $w \in [-1, 1]$ and $G_{\sigma,\rho}$ is an oriented 2D Gaussian filter rotated by angle ρ:

$$G_{\sigma,\rho}(x, y) = Ke^{-(\frac{x'^2}{\sigma_x^2} + \frac{y'^2}{\sigma_y^2})} \quad (3)$$

where K is a normalizing constant, $x' = x \cos \rho + y \sin \rho$ and $y' = -x \sin \rho + y \cos \rho$. The case of $\sigma_x \approx \sigma_y$ yields an isotropic point spread function (PSF) modeling out-of-focus blur and resolution differences, while a significant difference between these two parameters models an anisotropic blur in direction ρ, encountered with typical camera or object motion. As w varies from -1 to 0 to 1, the transformation varies from image sharpening to identity to image blurring. (Image sharpening with $w < 0$ is occasionally required if the current frame is sharper than the reference frame.) BPN takes one reference frame I_R and a sliding window of N frames I_i around the current (i-th) time instance, and predicts N sets of parameters $\psi = [\sigma_x, \sigma_y, \rho, w]$ that best transforms I_R to the N output frames (with respect to spatial frequency characteristics) via the blur model of Eqn. (2). The network thus takes in N+1 image inputs and predicts 4N parameters. Predicting transforms on frame stacks ensures temporal consistency. The network architecture comprises a ResNet18 backbone [10], an average pooling layer and two fully connected layers, as shown in Fig. 3. The training loss is defined as:

$$L_{BPN} = \lambda_\psi L_\psi + \lambda_R L_R + \lambda_T L_T \quad (4)$$

where L_ψ is the squared error loss between the true and predicted parameter vectors ψ; L_R is the mean-squared image reconstruction error between predicted and true ROIs I_i; and L_T is a temporal consistency loss that discourages large fluctuations in ψ over time: $L_T = \sum_j |\psi_i - \psi_j|^2$, where index i denotes current frame and the summation is over N-1 neighboring frames $j \neq i$. The predicted I_i is obtained by applying the blur model in Eqn (2) and (3) with parameters ψ to I_R. Note that the blur model is differentiable w.r.t. ψ and thus can be applied within the training loop.

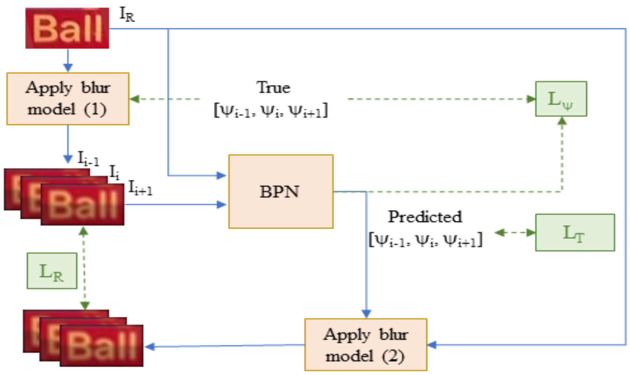

Figure 4: Training of BPN. In Stage 1, a blur model (1) with known parameters is applied to reference I_R. The input and outputs are passed to BPN which learns to regress the parameters on losses L_ψ, L_R, L_T. In Stage 2, frame tuples from real videos are passed to BPN which regresses parameters via self-supervision on L_R and L_T applying blur model (2). N=3 in this illustration.

BPN is trained in two stages, as illustrated in Fig. 4. In the first stage, Eqn (2) with known parameters ψ is applied to reference ROIs I_R from synthetic videos to obtain training pairs (I_R, I_i). In this phase, all three loss terms in Eqn (4) are minimized. As part of augmentation, inputs I_i are warped with translational jitter in x and y directions to immunize the network to minor mis-alignments between I_i and I_R encountered in real video frames. In the second phase, BPN is fine-tuned via self-supervision with (I_R, I_i) tuples extracted from real-world videos. Here only L_R and L_T are minimized since the true ψ are unknown. During inference, the ROI pair (I_R, I_i) from the original video is passed through BPN, and the predicted parameters are used to apply Eqn (2) to the altered ROI I'_R to produce the ROI I'_R for the current (i-th) frame, as shown in Fig. 3.

4. Experimental Results

To our knowledge there are no existing datasets or benchmarks for the problem of video text replacement. We thus evaluate our technique against still image replacement baselines trained and applied on video frames. Note that advances in still-image replacement are an enabler rather than competitor to STRIVE, since we rely on the still transfer for the reference frame.

4.1. Datasets and Experimental Setup

1. *Synthtext*: We have developed a dataset of 120 synthetic videos using the *Unity* simulator. Indoor and outdoor scenes are modeled with diverse text styles against varied backgrounds, captured with different lighting, camera and object motion characteristics. A given scene is generated with multiple text strings, providing many source-target pairs for training and testing our models. Unlike

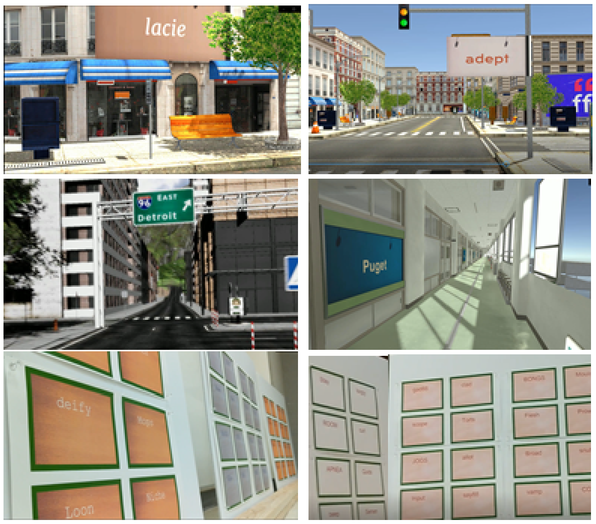

Figure 5: Example video frames from $Synthtext$ (top 2 rows) and $Robotext$ (bottom row).

existing synthetic datasets comprising clean frontal ROIs, our video simulations produce ROIs with realistic geometric and photometric distortions, including motion blur and shadows. Fig. 5 shows examples.

2. $Robotext$: We acquire first-of-a-kind videos captured by a Google Pixel 2 XL smartphone mounted on a $Create2$ robotic platform. The robot is programmed to travel in random trajectories around a gallery of text posters mounted in a large indoor hall. Trajectories include linear and curved paths at varying travel speeds. The posters are designed such that different words of the same style and background are adjacent to each other, to train text replacement models, as shown in Fig. 5. This dataset comprises approximately 5000 short video clips.

3. $Realworld$: We have collected challenging real-world videos in indoor and outdoor environments including camera motion related to walking and driving. We drew from two sources. The first is the ”Text in Videos” dataset from the ICDAR 2015 Robust Reading Competition [15] comprising 25 videos of scene text captured in the wild. From these, we curated a subset of 15 videos of sufficient quality for our task. Additionally we collected our own dataset of 22 videos of similar scene content and diversity as the ICDAR set.

All datasets are annotated with bounding boxes around text objects in each frame using AWS *ReKognition* software.

Pix2pix and SRNet are trained on frames from 1000 video clips from $Synthtext$ and $Robotext$ following the protocols of the original implementations [13], [26], with additional augmentations for pose variations. STTN is trained on frames from 100 video clips from $Synthtext$ and $Robotext$ with frontal stylized text masks serving as training targets. For reference frame selection, parameters α_1 and α_2 (see Section 3.1) were selected as 0.7 and 0.3 based on cross-validation experiments. Stage 1 of BPN is trained on 100 videos from $Synthtext$, with 10% of the training data used for validation. Stage 2 is trained on 900 videos from $Robotext$ and a subset of 25 videos from $Synthtext$. Both stages are trained with 100 epochs. We use a neighborhood of N=3 consecutive frames for imposing temporal consistency. The ADAM optimizer is used with a learning rate of 0.0005 for stage 1 and 0.0003 for stage 2. All implementations are in the Pytorch framework [19] with GPU acceleration, and all tests reported below are performed on independent datasets.

4.2. Evaluation Metrics

1. We compute MSE, peak signal-to-noise ratio (PSNR), and average structural similarity (SSIM) scores [25] between estimated and ground-truth ROIs for frames from synthetic test videos.
2. We evaluate OCR accuracy of *ReKognition* on video text in real scenes. We measure the number of word-level errors on the target text as a fraction of the number of frames in which the source text was correctly recognized.
3. We evaluate high-frequency jitter of the target text by analyzing bounding box coordinates returned by *ReKognition* from altered videos. Let a box vertex have time-varying coordinates [$x(t)$,$y(t)$]. We extract a high pass signal [$\tilde{x}(t)$,$\tilde{y}(t)$] by subtracting a lowpass-filtered version, and computing the root mean of $\tilde{x}^2(t) + \tilde{y}^2(t)$ as a measure of temporal jitter.
4. We perform timing analysis, simulating the personalized marketing scenario where a single input video is used to generate K altered output copies, each with a different target text string. We measure average frame rate over 20 randomly selected input videos on a Linux machine running Ubuntu 16.04 with a single GeForce GTX GPU.

4.3. Quantitative Analysis

Reconstruction performance on synthetic videos is shown in the first 3 columns of Table 1. Our first result is an ablation study on the effect of reference frame selection (disabled by always selecting the first frame as the reference). We see that this step has a noticeable effect in reconstruction accuracy (“w/o ref frame” in Table 1). As seen in Fig. 9, SRNet struggles to correctly replace small or strongly distorted text, and our reference frame selection criteria avoid selecting such frames for text replacement. Next we examine the impact of BPN (disabled by setting $w = 0$ in Eq. (2). As seen in Table 1, BPN also plays a crucial role (“w/o BPN” in Table 1). The most pronounced benefit is seen for videos that contain significant motion or focus blur, as shown in Fig. 6 where BPN accurately models focus blur.

Next, we compare STRIVE with two competing alternatives. The first is a pix2pix baseline [13] trained on in-

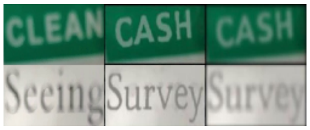

Figure 6: BPN ablation: input ROI (left), output ROI without BPN (middle), output with BPN (right).

Method	MSE ↓	PSNR ↑	SSIM ↑	OCR ↑	Jitter ↓
Pix2pix [13]	0.0593	12.27	0.531	-	-
SRNet [26]	0.0227	16.44	0.598	0.771	5.10
w/o ref frame	0.0203	16.93	0.596	-	-
w/o BPN	0.0203	16.93	0.594	-	-
STRIVE	**0.0186**	**17.31**	**0.604**	**0.957**	**1.48**

Table 1: Quantitative results on $Synthtext$ (first 3 columns) and $Robotext$ and $Realworld$ videos (last 2 columns). For MSE and jitter, smaller is better; for PSNR, SSIM, OCR greater is better.

dividual video frames. To give pix2pix a fair chance at transferring text on videos, during training we supply the network with 3 consecutive frames and incorporate a temporal coherence constraint that penalizes MSE loss between the current output and its two neighboring frames. The second (and stronger) baseline is SRNet trained and applied on individual video frames. As seen in Table 1, STRIVE outperforms both methods in terms of MSE, PSNR, SSIM.

We compare STRIVE with frame-wise SRNet in terms of OCR accuracy on real and synthetic videos. From column 4 of Table 1, STRIVE significantly outperforms SRNet. We owe this largely to the failure of SRNet to replace text under strong geometric distortions and photometric blur. Finally we compare temporal smoothness of the replaced text in column 5 of Table 1 and observe that STRIVE produces substantially less jitter, owed to temporal consistency constraints that are absent in per-frame SRNet. The jitter is clearly evident when viewing the videos (see the supplemental section).

In comparing the three methods, pix2pix attempts holistic transfer, accounting for text style, color, background, geometry, and lighting. SRNet decouples the transfer of foreground text from background, which helps performance. STRIVE benefits from an additional level of disentanglement, namely separation of single-image transfer from modeling of geometric and appearance variations over time. Interestingly, frame-wise SRNet offers a form of ablation, as it is essentially STRIVE without the pose normalization and text propagation step.

We perform inference timing tests for different output run lengths (K=1, 50, 100) of output videos created from a given input video. The average SRNet inference speed is 1.67 frames per second (fps) for all K, while STRIVE achieves faster rates of 2.11, 7.83, and 8.01 fps for K = 1, 50, 100 respectively. This is because STRIVE performs expensive text replacement only on one reference frame.

x-y misalignment	w	σ_x	σ_y	ρ
0	0.14	1.18	1.19	44.76
+/- 5	0.15	1.18	1.21	45.3

Table 2: Mean absolute error in blur parameter predictions without and with alignment error (ρ is specified in degrees).

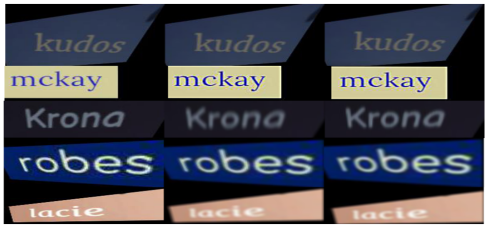

Figure 7: Results of BPN simulation. From top to bottom, transforms applied are identity, sharpening, motion blur, and two different levels of isotropic Gaussian blur. Left, middle and right columns indicate input, blur model output, and BPN output.

Furthermore, parameters for all text propagation functions (STTN, lighting, blur model in Eqn 2) are computed once offline per input video, and reused in runtime for all K output copies. In contrast, framewise SRNet must replace text in every frame in every video, and hence suffers from the same low throughput for all run lengths.

4.4. BPN Simulation

We study BPN's ability to predict the parameters of the blur model (Eqn 2) from image pairs. An independent test set of text ROIs from 995 frames covering 5 different scenes from both synthetic and real datasets are used. Model parameters ψ are chosen at random in the following ranges: $w \in (-1, 1)$, σ_x, $\sigma_y \in (0, 5)$, $\rho \in (0, 180)$, and the blur model is applied with these parameters to the ROIs to produce transformed outputs. The original and transformed images are sent through BPN and predicted parameters are compared to ground truth. Regression performance is shown in Table 2. From observing real videos, we noted translational alignment errors in the order of 0-5 pixels between reference and non-reference ROI. Hence we report BPN performance for both ideal alignment and random simulated x-y jitter between 0-5 pixels. We note that network predictions are robust to such misalignment. The high errors for ρ are presumably due to the fact that blur angle estimation becomes ill-conditioned for near-isotropic kernels. Qualitative results are shown in Fig. 7 confirming that BPN effectively models a variety of differential blurring and sharpening transformations between frames.

4.5. Qualitative Evaluation

Fig. 8 demonstrates TPM on $Robotext$ and $Realworld$ videos. Effects of varying lighting and text sharpness as the robot moves across the scene are effectively incorporated into the altered video. Additionally we estimated blur of

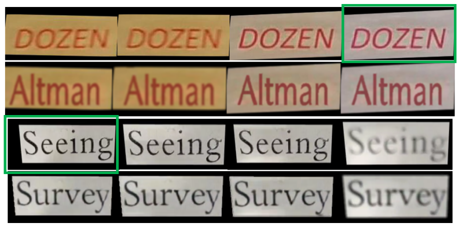

Figure 8: Results of TPM on *Robotext* (upper half) and *Realworld* (lower half). First/second row are original/altered video. Frames are in temporal sequence but not consecutive. Reference frame marked in green.

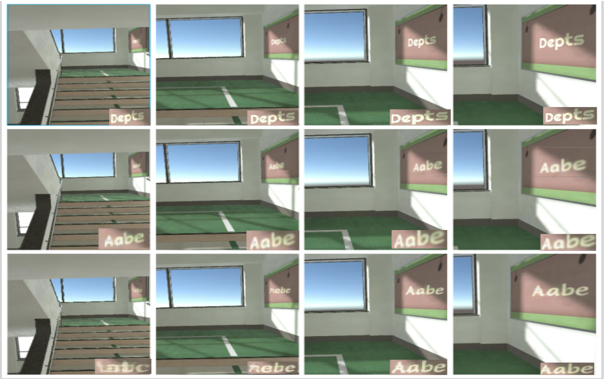

Figure 9: Comparison of original video frames (upper), STRIVE (middle) and SRNet outputs (lower) [26] for a *Synthtext* scene. Zoomed ROI insets are in lower right.

the text ROIs using variance of Laplacian [20]. The Pearson correlation coefficient between blur scores of original vs. replaced ROIs is 0.9912 indicating effective blur transfer. Figure 9 compares STRIVE and SRNet outputs for a *Synthtext* video. As seen in the zoomed insets, SRNet is unable to cope with strong perspective (although being trained with such examples), and even when replacement is successful, the geometry is incorrect. STRIVE achieves both accurate text replacement and geometry by solving each task individually. Figure 10 shows results for real-world videos. The first scene encounters out-of-focus blur. STRIVE is able to model this effect and maintain character integrity, while SRNet outputs are distorted. The next two scenes are from the ICDAR dataset, where videos are captured with human walking motion. The fourth scene is captured in a moving vehicle with atmospheric noise. In all cases, STRIVE is visibly superior to SRNet in preserving text integrity and the geometric and photometric effects of the original scene. The supplemental section contains videos of challenging realistic scenarios with moving shadows, focus blur, human and vehicle motion. As seen in the videos, SRNet exhibits considerable jitter that is avoided by STRIVE, thanks to temporal smoothness constraints.

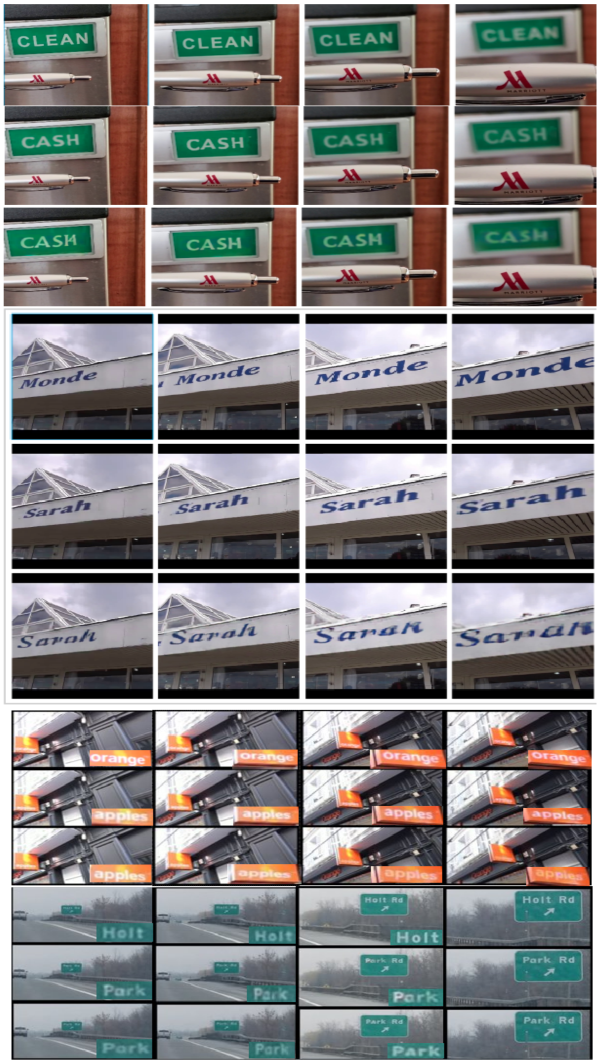

Figure 10: Comparison of original (upper), STRIVE (middle) and SRNet (lower) [26] frames for real videos. Zoomed ROI insets are shown in lower right for selected scenes.

5. Conclusions

We propose an effective and efficient method to replace text in videos by decoupling still-image text transfer from temporal changes in geometry and appearance. The latter are modeled via a novel learnable transform that captures photometric differences between pairs of video frames. This type of differential transform learning has broad applicability for image and video editing. We offer new datasets for further progress in text-related video tasks. The efficacy of STRIVE relies on certain assumptions, including planar geometry and spectral coherence between text and background. We argue that these assumptions are not excessively limiting as such cases are commonly encountered in real world scenes. Future work includes generalizing the approach to handle occlusions, non-planar surfaces, self-lit text, and to non-text objects such as graphics and 3D shapes.

References

[1] Samaneh Azadi, Matthew Fisher, Vladimir Kim, Zhaowen Wang, Eli Shechtman, and Trevor Darrell. Multi-content gan for few-shot font style transfer. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018.

[2] Jiawen Chen, Andrew Adams, Neal Wadhwa, and Samuel W. Hasinoff. Bilateral guided sampling. *ACM Trans. Graph.*, 35(6), 2016.

[3] Zhanzhang Cheng, Jing Lu, Yi Nio, Shiliang Pu, FEi Wu, and Shuigeng Zhou. You only recognize once: Towards fast video text spotting. *Arxiv 1903.03299*, 2019.

[4] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Deep image homography estimation. *CoRR*, abs/1606.03798, 2016.

[5] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016.

[6] Michael Gharbi, Jiawen Chen, Jonathan T. Barron, Samuel W. Hasinoff, and Fredo Durand. Deep bilateral learning for real-time image enhancement. *ACM Trans. Graph.*, 36(4), 2017.

[7] Michael Gharbi, YiChang Shih, Gaurav Chaurasia, Jonathan Ragan-Kelly, Sylvain Paris, and Fredo Durand. Transform recipes for efficient cloud photo enhancement. *ACM Trans. Graph.*, 34(6), 2015.

[8] Agrim Gupta, Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Characterizing and improving stability in neural style transfer. In *Int. Conf. Comput. Vis.*, 2017.

[9] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016.

[11] Haozhi Huang, Hao Wang, Wenhan Luo, and Lin Ma. Real-time neural style transfer for videos. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017.

[12] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017.

[13] Phillip Isola, Jun-Yan Zhu, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017.

[14] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. In *Adv. Neural Inform. Process. Syst.*, 2015.

[15] Dimosthenis Karatzas et al. Icdar 2015 competition on robust reading. In *International Conference on Document Analysis and Recognition (ICDAR)*, 2015.

[16] Shangbang Long and Cong Yao. Unrealtext: Synthesizing realistic scene text images from the unreal world. *Arxiv 2003.10608v6*, 2020.

[17] T. Nguyen, S. W. Chen, S. S. Shivakumar, C. J. Taylor, and V. Kumar. Unsupervised deep homography: A fast and robust homography estimation model. *IEEE Robotics and Automation Letters*, pages 2346–2353, 2018.

[18] Nobuyuki Otsu. A threshold selection method from gray-level histogramss. *IEEE Trans. System, Man, Cybernetics*, 9(1):62–66, 1979.

[19] Adam Paszke et.al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. 2019.

[20] J.L. Pech-Pacheco, G. Cristobal, J. Chamorro-Martinez, and J. Fernandez-Valdivia. Diatom autofocusing in brightfield microscopy: a comparative study. In *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*, volume 3, pages 314–317 vol.3, 2000.

[21] Said Pertuz, Domenec Puig, and Miguel Angel Garcia. Analysis of focus measure operators for shape-from-focus. *Pattern Recognition*, 46(5):1415 – 1432, 2013.

[22] Prasun Roy, Saumik Bhattacharya, Subhankar Gosh, and Umapada Pal. Stefann: Scene text editor using font adaptive neural network. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020.

[23] Osman Tursun, Rui Zeng, Simon Denman, Sabesan Sivapalan, Sridha Sridharan, and Clinton Fookes. Mtrnet: A generic scene text eraser. In *International Conference on Document Analysis and Recognition (ICDAR)*, 2019.

[24] Wenhai Wang, Enze Xie, Xiang Li, Tong Lu, Gang Yu, and Shuai Shao. Shape robust text detection with progressive scale expansion network. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019.

[25] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.*, 13(4):600–612, 2004.

[26] Liang Wu, Chengquan Zhang, Jiaming Liu, Junyu Han, Jingtuo Liu, Errui Ding, and Xiang Bai. Editing text in the wild. In *ACM Int. Conf. Multimedia*, 2019.

[27] Qingbo Wu, Rui Ma, King Ngi Ngan, Hongliang Li, and Fanman Meng. Blind image sharpness assessment and enhancement via deep auxiliary learning. In *Int. Conf. Multimedia and Expo*, 2019.

[28] Ruomei Yan and Ling Shao. Blind image blur estimation via deep learning. *IEEE Trans. Image Process.*, 25(4):1910–1924, 2016.

[29] Qiangpeng Yang, Jun Huang, and Wei Lin. Swaptext: Image based texts transfer in scenes. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020.

[30] Shuai Yang, Jiaying Liu, Zhouhui Lian, and Zongming Guo. Awesome typography: Statistics-based text effects transfer. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017.

[31] SHuai Yang, Jiaying Liu, Wenjing Wang, and Zongming Guo. Tet-gan: Text effects transfer via stylization and destylization. In *AAAI*, 2019.

[32] Shuai Yang, Zhangyang Wang, Zhaowen Wang, Ning Xu, Jiaying Liu, and Zongming Guo. Controllable artistic text style transfer via shape-matching gan. In *Int. Conf. Comput. Vis.*, 2019.

[33] Fangneng Zhan, Hongyuan Zhu, and Shijian Lu. Scene text synthesis for efficient and effective deep network training. *Arxiv 1901.09193*, 2019.

[34] Fangneng Zhan, Hongyuan Zhu, and Shijian Lu. Spatial fusion gan for image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[35] Youjian Zhang, Stephen J. Maybank, and Dacheng Tao. Self-supervised exposure trajectory recovery for dynamic blur estimation. *Arxiv 2010.02484*, 2020.

[36] Xinyu Zhou, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiran He, and Jiajun Liang. East: An efficient and accurate scene text detector. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017.