The Visual Computer
https://doi.org/10.1007/s00371-024-03371-w

RESEARCH

A novel approach for improving open scene text translation with modified GAN

Yasmeen Cheema[1] · Muhammad Nadeem Cheema[2] · Anam Nazir[3] · Fahad Ahmed Khokhar[4] · Ping Li[5] · Ayaz Ahmed[3]

Accepted: 13 March 2024

Abstract

Text, as a vital tool for communication, is playing an imperative role in modern society. Precise high-level text translation systems are essential requirements in a wide range of real-world applications, such as robot navigation, industrial automation, image search, and instant translation. Regardless of improved research, a series of grand challenges may still become upon when translating text automatically in the real-world from open scene images. The difficulties mainly stem from multiplicity and inconsistency of text in open scenes, complication and obstruction of backgrounds, and deficient imaging conditions in uncontrolled circumstances for open scene images. The existing deep learning-based text translation systems do not eliminate the text for translation, and these applications just replace text on the reconstructed scene. To address the abovementioned shortcomings, this study proposed a novel approach for open scene text translation. Our system consists of five modules including scene text detection, text recognition, text elimination, text translation, and text insertion along with scene reconstruction. The novelty presented by our model lies in the idea of first eliminating the text from the open scene for accurate translation and then reconstructs the translated text on the image for its proper alignment. We specifically modified the existing generative adversarial network (GAN) architecture for improved performance of text elimination by introducing a novel strategy of text and scene concatenation to reduce the overall loss function. For this purpose, we created a synthetic dataset to train our GAN for text elimination module. Experiments on various standard text translation systems demonstrate that our integrated system is able to outperform state-of-the-art approaches in terms of result quality. We have achieved 90.87% of precision, 83.66% of recall, 87.116% of $F1$-score, and reduced both losses (l_1 and l_2) up to 50% which is remarkable upon state-of-the-art translation systems.

Keywords Text detection · Image to text · Text recoginition · Deep learning-based translation · Text elimination · Text translation · Generative adversarial network (GAN) · Convolutional neural network (CNN)

✉ Fahad Ahmed Khokhar
fahadahmed.khokhar@unifi.it

Yasmeen Cheema
ycheema.phdcse16@rcms.nust.edu.pk

Muhammad Nadeem Cheema
itscheema786@yahoo.com

Anam Nazir
itsanam786@yahoo.com

Ping Li
p.li@polyu.edu.hk

Ayaz Ahmed
ayazahmed201819@gmail.com

[1] School of Interdisciplinary Engineering & Sciences (SINES), National University of Sciences and Technology, Islamabad, Pakistan

[2] Department of Computer Science, COMSATS University Islamabad, Attock, Pakistan

[3] Department of Computer Science, COMSATS University Islamabad, Wah Cantt, Pakistan

[4] Department of Mathematics and Information, University of Florence, Florence, Italy

[5] Department of Computing and School of Design, The Hong Kong Polytechnic University, Kowloon, Hong Kong

Published online: 13 April 2024

1 Introduction

Text is the most essential medium for communicating semantic information. Such texts appear everywhere in the natural environment is known as open scene texts. Open scene text translation draws much interest in computer vision. Recently, it is used in many art and education applications such as real-time text recognition [36] for visually impaired, tourist guided applications, robot sensing, and many more. Moreover, many machine learning technologies require open scene text translation [2] such as language translation with automatic input of text written in an unknown script, indexing large image, video databases by their textual content, and automatic information entry. With the rapid development of deep learning, each composition technology of text translation has room for improvement, which means that the overall performance can get a significant boost as long as we improve the performance of each step.

An end-to-end text translation system typically consists of text detection [22, 43], text recognition and text translation components. Numerous methods that focus solely on text localization and detection in real-world images have been published [4, 15, 21, 45, 50], where in many cases the text is manually localized by a human annotator. Though several methods have been proposed for horizontal and oriented text detection and recognition yet no open scene text translation system has yet achieved sufficient accuracy for practical applications [14, 19].

Nowadays, there are some translation systems available in the industry such as Baidu Translate [3], Sogou Translate [37], Dict [8] and Google Translate [10]. Few of them are available as open source. Moreover, most results of the commodity system have visible drawbacks in the delivery of end results that either the translated text is misplaced or open scene image got noisy due to text translation. The existing text translation systems do not eliminate the text for translation first and then replace back the text on the reconstructed scene.

Distinctive from scripts in documents, text in open scene reveals much elevated variety and unpredictability [27]. Moreover, the feature ratios and layouts of open scene text may fluctuate drastically. Backgrounds of open scene images are nearly irregular and unpredictable. Similar background patterns of text can often occur in open scene images such as traffic signs, bricks, windows, and occlusions caused by unfamiliar objects, which may potentially lead to errors during translating task [27]. Hence, due to intricate open scene backgrounds and dissimilarity of font, size, color, language, illumination condition and orientation of natural environment, open scene text translation becomes a very challenging task. Moreover, open scene text translation performance was poor when hand-designed features and traditional classifiers were used for text detection [25, 41] and recognition. On the other hand, the performance has been much improved in recent years, and significantly benefitted from the development of deep learning. Meanwhile, the research focus on text detection has shifted from image inpainting [44, 49] to text elimination [34] and trickier open scene reconstruction [18].

Our proposed model aims to resolve the problems mainly related to tourism by replacing the foreign languages in the local scenes with familiar language. Our open scene text translation system consists of five modules including scene text detection, text recognition, text elimination, text translation into the specified language, and text insertion along with scene reconstruction [24, 29, 38].

One of the key insights of this paper is that we adopt the SynthText [11] method to create our own image dataset and train a generative adversarial network (GAN) model that focuses on text elimination for the image. Using this method, our system is suitable for challenging scenarios of open scene images with different environmental conditions in terms of accuracy and efficiency.

In conclusion, the key contributions of this study are summarized as follows:

- The proposed model provides a novel approach for open scene text translation by first eliminating the text from the open scene for accurate translation and identically aligning the translated text on the open scene image.
- We have modified the existing GAN architecture to improve the performance of the text elimination task by concatenating the original text image and text mask information at the input and then combining the non-text region of the original input and text region of the network output to get the final result. This strategy helps us to reduce the loss function which is only $L2$ loss of target image and ground truth image.
- We created a synthetic dataset to train of GAN used in the text elimination process with additional fonts to fulfill the real-time environment conditions in text translation.

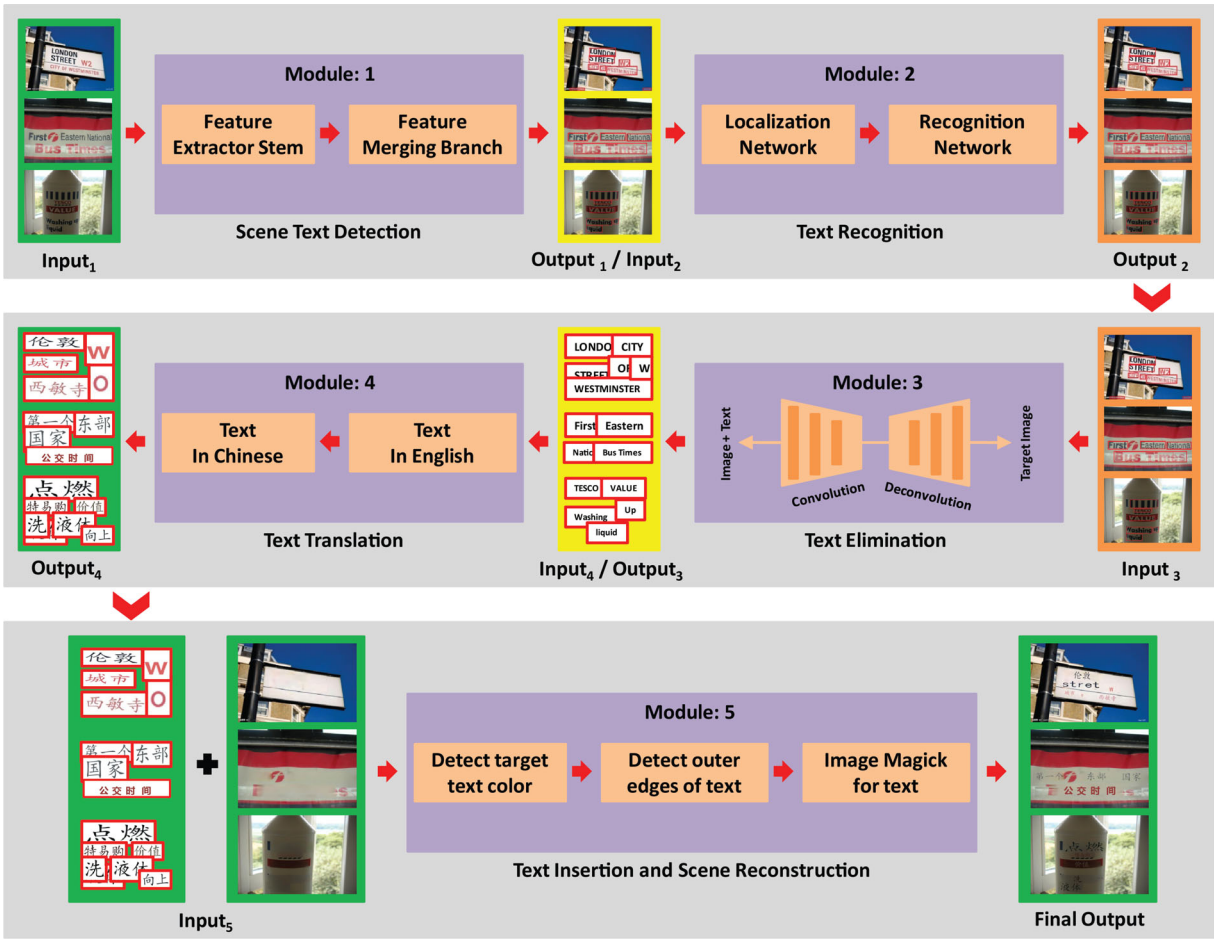

Fig. 1 The general architecture of the proposed open scene text translation system. The model consists of five modules including scene text detection, text recognition, text elimination, text translation into the specified language and scene reconstruction

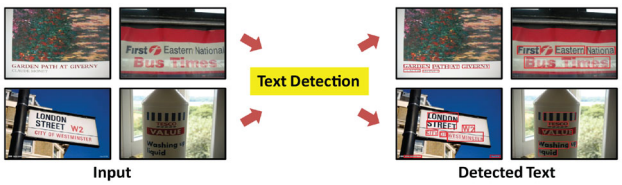

Fig. 2 The workflow of text detection module. A well-known scene text detector named EAST is employed to implement the text detection algorithm which takes input in the form of open scenes and generates outputs with marked region boxes (red rectangles) on the detected text

Our synthetic dataset comprises of almost 8K images for training and 300 images for testing.

- We have compared the proposed model with publicly available text translation systems like Google, BaiDu, Youdao and SoGou. The quantitative and qualitative results prove the superiority of our model in terms of providing detailed translated text having exact alignment on the open scenes.

2 Related Work

2.1 Text Translation System

Currently, there are few applications on translation such as Baidu Translate [3], Sogou Translate [37], Dict [8], Youdao [48] and Google Translate [10]. Among these softwares, Google Translate is the only system that can operate at real time. The results of these applications vary a lot, especially the results of text detection and text elimination, among them Google Translate works best. In the results and discussion section, we have compared our results with these publicly available translators.

2.2 Image Inpainting

Inpainting is an important part of our system, which largely determines the authenticity of text elimination. There exist

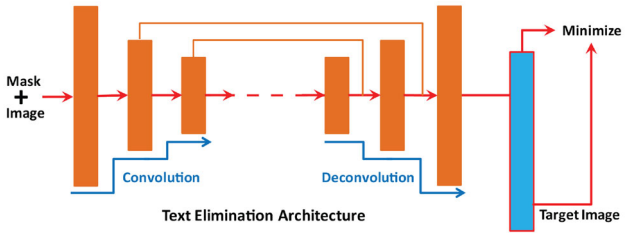

Fig. 3 General architecture of the text elimination module. The image and mask information first go through a series of convolution layers and then concatenate with previous layers with a series of deconvolution layers. The result is the image without text, the loss of our architecture is the *L*2 loss of target image and ground truth image. The novelty in our modified network is that by giving more penalties to the text area, we explicitly guide the network to learn the text part

Fig. 4 Workflow of text elimination module having modified GAN to eliminate the texts from scene. The input images having text detected in the form of marked red rectangles are inserted in this module and the output is generated with the images having the text areas eliminated from them

many open-source algorithms based on classical methods. [5] proposed a new algorithm based on picture inpainting that needs to recreate the basic methods employed by experienced restorations. The main idea is to easily convey information from nearby places in the isophote's direction. The user needs to specify the area that needs to be painted; the algorithm will do the rest in just a few minutes. The inpainted pictures are clear, with no color artifacts. The samples provided indicate a wide range of uses, including the restoration of ancient images and damaged film, the removal of overlaid writing, and the removal of objects. One of the drawbacks with our method is the replication of vast textured regions. [31] present a novel iterative regularization approach for inverse issues using Bregman distances, having a specific focus on image processing challenges. This approach is inspired by the challenge of recovering noisy and fuzzy photos with variational approaches that employ total variation regularization. Accurate convergence outcomes and effective stopping conditions are achieved in this approach. The numerical findings for denoising look to be significantly better than standard models, and the initial findings used for deblurring and denoising are quite encouraging. Some models are also based on deep learning methods.

In [34] introduced an unsupervised learning algorithm based on visual features that is dependent on context pixel prediction. A context encoder is developed utilizing a CNN model that is trained to create the material of an arbitrary image area based on its surroundings. To complete this task successfully, context encoders must first grasp the entire image's content and then provide a credible suggestion for the incomplete part(s). During context encoder training, both a regular pixel-wise reconstruction and reconstruction loss are tested including an adversarial loss. Similarly [44] presented a multi-scale neural patch synthesis approach that optimizes image content as well as texture limitations to maintain contextual structures while producing high-frequency details. Patches are matched and adapted to the most identical mid-layer that includes correlation features for a deep classification network. This approach is evaluated on the Paris Streetview dataset and ImageNet dataset that obtained high inpainting accuracy. This methodology achieves accurate and more consistent results, particularly for images with a high resolution. In Yu et al. [49] present a generative deep model-based technique that generates a new image structure and uses surrounding information as references while network training to improve predictions. The feedforward CNN is employed that analyze pictures with many holes at varying positions and sizes during testing. Analyses on several datasets, for faces CelebA and CelebA-HQ, are analyzed; for textures, DTD is utilized and ImageNet and Places2 are analyzed for natural images, and this suggested model produces more effective inpainting results than previous methods.

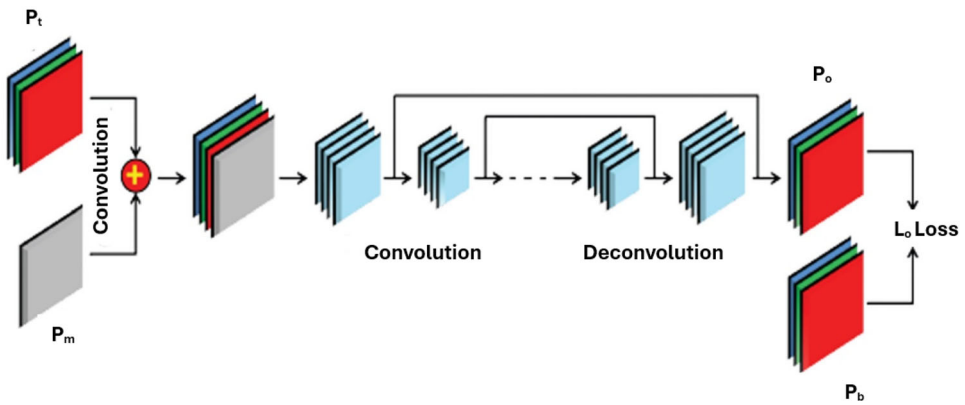

Fig. 5 Deep inside view of modified GAN . To fulfill the requirement of eliminating the text regions from open scene images, we have combined the non-text region of original input and text region of network output to get the final result. At the input, we have accelerated performance by concatenating the original text image and text mask information. By utilizing smaller lambda we have assigned the penalty only to the text area, which helps us to explicitly guide the network to learn the text part

However, the classical methods are not suitable for our system since they cannot tolerate a large inpainting area. Also, directly applying convolution network-based methods are not satisfactory due to the divergence of the dataset. As, these models are trained on scenery images, and they can hardly provide satisfactory performance in text image inpainting.

2.3 Text Detection

Scene text detection and recognition are quite popular nowadays, as there are several efficient approaches. Lukáš Neumann et al. introduced real-time end-to-end text detection and customization approach in [30]. It achieves real-time performance by presenting the character identification and segmentation issue as a fast sequential selection between different extremal regions. An algorithm based on clustering is used to efficiently organize text items into lines. It then labels character areas with an (optical character recognition) OCR classifier that has been trained on synthetic fonts. The method outperforms previous algorithms on a variety of datasets, producing cutting-edge text localization results. It performs exceptionally well on an extremely difficult dataset, exhibiting its capacity to incorporate more information about identified text. The method's efficacy is further demonstrated in a reading competition, where it outperforms the best current approaches. This ER detector is resistant to distortion, diminished contrast, and light as well as color and pattern variation.

In [39] designed a neural network design that can combine feature extraction with sequence modeling, and transform it into a cohesive framework. This approach performs well across lexicon-based tasks and non-lexicon tasks resulting in a smaller but more realistic model for real-time applications. Tests performed on benchmarks, datasets such as Street View Text, IIIT-5K, and ICDAR, show that this approach outperforms the prior methods. Furthermore, results from experiments show that it is particularly good for image-based recognition. [46] proposed a novel approach that automatically learned from labels of bounding boxes to capture the underlying structural components of text characters at various granularities. Stroke-lets can reliably detect characters and create histograms to characterize them in real settings. The scene text recognition system that depends on stroke-lets is both reliable and efficient. Numerous tests on typical benchmarks confirm the benefits of stroke-lets and show that the proposed model is more effective than previous approaches. Similarly, [47] suggest an algorithm that uses complete pictures to generate global and pixel-wise projection maps, followed by detections. To optimize text processing, a Fully Convolutional neural network Network (FCN) model is utilized that estimates three categories of data: text region, individual letters, and their relationships. This suggested approach can handle curved text, multi-oriented, and horizontal features in natural images by predicting their features. Experiments on established benchmarks such as ICDAR 2013, MSRA-TD500, and ICDAR 2015 show that the new technique outperforms earlier approaches. In this study, the first baseline results are also presented on the COCO-Text dataset.

In [50], the author provide state of- the-art effective pipeline for rapid and reliable text identification in natural settings. The pipeline uses a neural network for predicting words, quadrilateral and text lines, forms in the entire image, avoiding the need for intermediary processes like alternative collection and word division. This pipeline is simple and focuses on developing loss functions and the architecture of neural networks. Experiments on typical datasets such as ICDAR 2015 and MSRA-TD500 COCO-Text show that the proposed technique exceeds existing methods in terms of accuracy and efficiency. The proposed method achieves a 0.7820 F-score at 720p resolution and 13.2fps on the ICDAR 2015 dataset.

However, few of them can detect oblique texts and their accuracies are not always satisfactory for real time scenarios like open scene text detection and translation. We carefully compare their results.

3 Proposed Methodology

The proposed model aims to resolve the problems mainly related to tourism by replacing the foreign languages in the local scenes with the familiar language. Our open scene text translation system consists of five modules including scene text detection, text recognition, text elimination, text translation into the specified language, and text insertion along with scene reconstruction for improved performance. Figure 1 shows the general architecture of the proposed open scene text translation system. The model consists of five modules including scene text detection, text recognition, text elimination, text translation into the specified language, and text insertion along with scene reconstruction. In first module, a well-known scene text detector is employed, which generates outputs in the form of region boxes of the detected text on the scene. Second module implemented a renowned text recognition network having localization and recognition networks to recognize the text inside the detected text boxes. The third module modified the existing popular GAN to eliminate the texts from scene automatically. For text translation, the presented fourth module utilized OpenNMT. Finally, in the last module to insert text into the image, first, we have obtained the color of the original text for the insertion of eliminated text using gradients of image. Second, we convert the translated text into images and resize the image into the specified size. Third, we combine the text image and the background

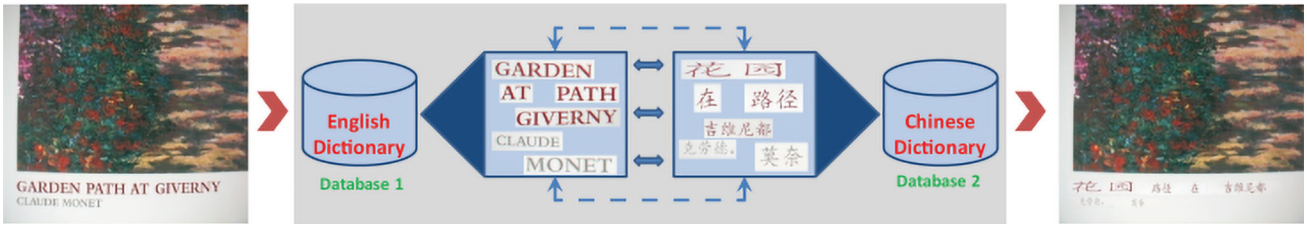

Fig. 6 Text translation module's workflow. The input is in form of eliminated text and open scene image while output is in the form of translated text and the open scene image. By implementing open-source neural machine translation (OpenNMT), we have utilized our own semantic English-Chinese dictionary for text translation task. Left is the input image, centre is showing translation procedure, and right is the translated open scene image

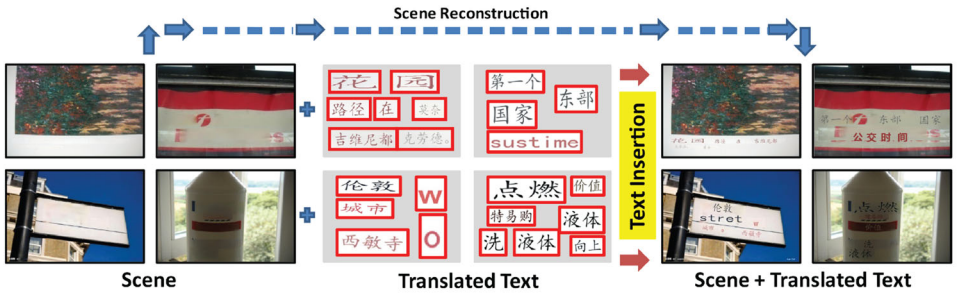

Fig. 7 Workflow of text insertion module for open scene reconstruction. The input is in the form of an open scene with translated text and output is in the form of a complete image with translated text reconstructed on its proper place having accurate alignments

Fig. 8 **a** Original image for color detection, **b** gradients of original image, **c** after flood fill of the gradients, and **d** after background color elimination

image together and flood fill algorithm is employed to fill the area of outer image edges.

3.1 Open scene Text Detection

We use the efficient and accurate scene text detector named EAST [50] to implement the text detection algorithm, see Fig. 2. The output of the EAST is the region boxes of the scene text detected. Each region box have eight parameters denoting the coordinate shift from four corner vertices $\{p_i = (\Delta x_i, \Delta y_j)|i\epsilon\{1, 2, 3, 4\}\}$ of the rectangle to the pixel location. Moreover, the two parameters of each distance offset are the vertical and horizontal coordinates value. In order to make the part of text insertion easier, we transform the parameters of the region box and the final data format is represented as $(x_b, y_b, w_b; h_b, \theta_b)$, where x_b, y_b are the coordinates of central pixel of the detected text region, w_b, h_b the width and height, and θ_b the rotation angle.

As the region box is a bit compact for the part of text elimination, we appropriately increase the length and the width of the region box. $w'_b = k_1 w_b$, $h'_b = k_2 h_b$.

3.2 Text Recognition

After getting the text regions with detected text, we need to recognize the text inside them. Following work of [4], we use the same recognition network for this subtask. The extracted text regions first go through a deep residual convolutional neural network [13], then a bidirectional long short-term memory (BiLSTM) network [12] is implemented with fully connected layers and softmax layers to recognize the characters inside input regions [20].

3.3 Text Elimination

To improve text elimination results of our proposed model we have modified the existing popular GAN [9] to eliminate the texts from scene automatically as illustrated in Fig. 3 and Fig. 4. we have converted the text elimination task as a variation of image inpainting task. The difference between two tasks is that the pictures are not fully corrupted in open scene text, which means that the information in corrupted box is still useful. Following the method in [34], we modify its generator and use it to solve the text elimination task. Because of the unavailability of text elimination datasets, we

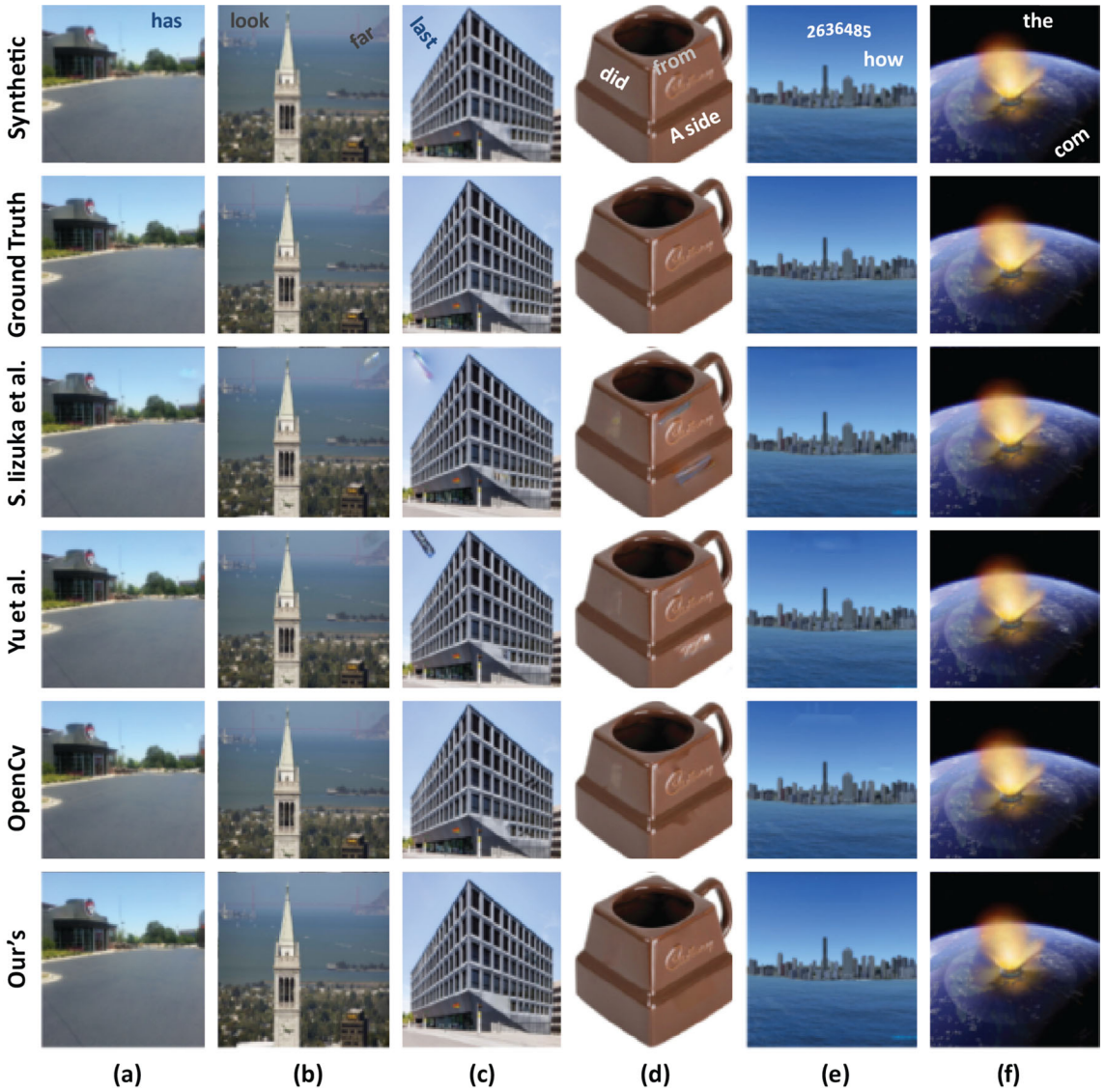

Fig. 9 Qualitative results to illustrate comparison of various text elimination methods with the proposed open scene text elimination module. First two rows are presenting synthetic dataset and ground truth images respectively, while third to fifth rows are the results obtained from three state-of-the-art methods (Lizuka et al. [17], Yu et al. [49] and OpenCV [32]) compared with the proposed model which are presented in the last row. Note that, as the number of lost pixels in our work is not the same among different pictures

have generated our own synthetic dataset for training and testing of text elimination module.

3.3.1 Dataset generation

In order to generate synthetic images of text, we use the synthetic engine [11] to produce realistic scene-text images and record their corresponding region boxes that contain different texts. The original processing pipeline of the dataset generation is presented as follows:

- Calculate the region segmentation of image based on local color and texture [1].
- Calculate the pixel-wise depth map by CNN network [16, 23, 26, 28, 40].
- Calculate the local normal vector for each region segmentation.
- Choose the color for text based on the region's color.
- Generate text with selected color and random font. orientation is adjusted to the region's normal vector.

- Render the text into the image using Poisson image editing [35]. However, in the above method, there remains a series of issues needed to resolve:

- First, the region box cannot be directly obtained using the available source code. We need to analyze and revise a large part of its code to produce the region boxes while synthesizing the images.
- Second, in the original pipeline, the text is rendered partially through alpha-blending to produce the blended color region of texts, but the result of this method is quite inapplicable in reality for open scene scenarios. Hence, instead, texts commonly overlay background objects. Thus, we revise the text rendering code to produce more realistic scene-text images.
- Third, the applied text fonts contain only Ubuntu-Bold, UbuntuCondensed-Regular, UbuntuMono-Regular which are uncommon in the realistic settings regarding open scene images, so we extend the text font set in order to cover several most popular fonts to fulfill real-time environmental requirements.
- Finally, we modify the occurring possibility of the shadow texts, curved texts and other variates of text to make the dataset more suitable for the reality.

In Eq. 3.1, T.L is the text location, O is original scene, R is the reconstructed scene, x and y are the 2D coordinates for both O and R. Similarly d_{12}, d_{21}, d_{23}, and d_{32} are calculated distances in both direction: left to right,right to left, bottom to top and top to bottom respectively for coordinates x and y for both O and R. Finally, we have computed text insertion (T.I) in parts $\mathrm{T.I}^{a}_{S,T}(x, y, d_{12}, d_{21}, d_{23}, d_{32})$ and $\mathrm{T.I}^{b}_{S,T}(x, y, d_{12}, d_{21}, d_{23}, d_{32})$ and integrated in $\mathrm{T.I}^{\text{integrated}}_{S,T}$ $(x, y, d_{12}, d_{21}, d_{23}, d_{32})$ as shown in equation 3.2, 3.3, 3.4 respectively. S is the background scene and T is the text which is eliminated from the image. Finally, in text insertion (T.I) based on calculated distances, we have place text on exact place as compared to input image.

3.3.2 Modified GAN for Text elimination

The general architecture of our text network is shown in Fig. 5. We take the text-corrupted picture $P_t \epsilon \mathbb{R}^{d_h \times d_w \times 3}$ and text mask $P_m \epsilon \mathbb{R}^{d_h \times d_w}$ as input, and make the network to implicitly learn to output the background truth picture $P_b \epsilon \mathbb{R}^{d_h \times d_w \times 3}$ without text.
To leverage both the text-corrupted picture P_t and text mask P_m, we use concatenation to combine these two information together. The combined input goes through a series of convolution layers and deconvolution layers with skip connection.

$$\mathrm{T.L}_{O,R}\Big(x, y, d_{12}, d_{21}, d_{23}, d_{32}\Big) = \sum_{i,j}\Bigg[O\Big((x+i)d_{12}, (x+i)d_{21}, (y+j)d_{23}, (y+j)d_{32}\Big) - R\Big((x+i+d_{12}), (x+i+d_{21}), (y+j+d_{23}), (y+j+d_{32})\Big)\Bigg]^2 \tag{3.1}$$

$$\mathrm{T.I}^{a}_{S,T}\Big(x, y, d_{12}, d_{21}, d_{23}, d_{32}\Big) = \sum_{i,j} O\Big((x+i)d_{12}, (x+i)d_{21}, (y+j)d_{23}, (y+j)d_{32}\Big) - \overline{O\Big(x, y, d_{12}, d_{21}, d_{23}, d_{32}\Big)} - \Bigg[R\Big((x+i)d_{12}, (x+i)d_{21}, (y+j)d_{23}, (y+j)d_{32}\Big) - \overline{R\Big(x+d_{12}, x+d_{21}, y+d_{23}, y+d_{32}\Big)}\Bigg]^2 \tag{3.2}$$

$$\mathrm{T.I}^{b}_{S,T}\Big(x, y, d_{12}, d_{21}, d_{23}, d_{32}\Big) = \sqrt{\sum_{i,j}\Big(O\Big((x+i)d_{12}, (x+i)d_{21}, (y+j)d_{23}, (y+j)d_{32}\Big) - \overline{R\Big(x, y, d_{12}, d_{21}, d_{23}, d_{32}\Big)}\Big)^2} * \sqrt{\sum_{i,j}\Big(R\Big((x+i+d_{12}), (x+i+d_{21}), (y+j+d_{23}), (y+j+d_{32})\Big) - \overline{R\Big(x+d_{12}, x+d_{21}, y+d_{23}, y+d_{32}\Big)}\Big)^2} \tag{3.3}$$

$$\mathrm{T.I}^{\text{integrated}}_{S,T}\Big(x, y, d_{12}, d_{21}, d_{23}, d_{32}\Big) = \frac{\mathrm{T.I}^{a}_{S,T}(x, y, d_{12}, d_{21}, d_{23}, d_{32}) * O\Big((x+i)d_{12}, (x+i)d_{21}, (y+j)d_{23}, (y+j)d_{32}\Big)}{\mathrm{T.I}^{b}_{S,T}(x, y, d_{12}, d_{21}, d_{23}, d_{32}) - R\Big((x+i+d_{12}), (x+i+d_{21}), (y+j+d_{23}), (y+j+d_{32})\Big)} \tag{3.4}$$

The final output is a picture $P_o \epsilon \mathbb{R}^{d_h \times d_w \times 3}$ mimicking the ground truth P_b. In order to force the network learn better the text part, we apply mask region-specific loss strategy. The final target loss is calculated as following:

$$\text{loss}_{l2} = \sum_{p} (p_o - p_b)^2 \times p_m + \lambda (p_o - p_b)^2 \times (1 - p_m) \tag{3.5}$$

where p_o, p_b and p_m are pixels sharing the same position in picture. λ is the parameter that represents the ratio of loss in different region in mask.
The output of this network P_o is the full image including the non-text region. To gain a higher performance, we leverage the usage of original text-corrupted image P_{t}, using P_{m} as mask, merging these two picture as our final output P_{f}, which can be shown below:

$$p_f = p_o \times p_m + p_t \times (1 - p_m) \forall p_f \epsilon P_f \tag{3.6}$$

where p_{f}, p_{o}, p_{t} and p_{m} pixels sharing the same position in picture.

3.4 Text Translation

For the text translation module, we have implemented and utilized open-source neural machine translation (OpenNMT), which is an open-source neural machine translation network. Since this part is not the core purpose of our work, so we do not dive into the further. Note that the model online does not have an English-Chinese (EN-Ch) version. Therefore, we need to train an En-Ch model using our own semantic dictionary, see Fig. 6.

3.5 Text Insertion

In order to insert text into the image, we divide the procedures into three parts. First, we need to obtain the color of the original text for the follow-up insertion of new text. Second, we convert the translated text into images and resize the image into the specified size. Third, we combine the text image and the background image after text elimination together. See Fig. 7 for illustration of above mentioned processes.
In the first step, inspired by the idea of watermark detection [7], as shown in Fig. 8 we use the gradients of image to obtain the outer edge of the text. After that, we use flood fill algorithm implemented in OpenCv [32] to fill the overall area of the outer edge. Thus, now we can obtain the rough segmentation of each character. Note that it is not enough to directly analyze the average color inside each character. After the above processes, the characters such as a, b and o, the inside closed part will be included too which will affect the accuracy of color detection. Instinctively, we find that the color of this part will resemble the background color, and thus we have eliminated this kind of intervention-based similarity.
In the second step, we use ImageMagick [18] to create and resize the image. ImageMagick is free software delivered as a ready-to-run binary distribution or as source code that can be made use of, copied, modified, and distributed in both open and proprietary applications. Using the information obtained in previous steps, the text image is created with the same features as the raw image.
In the third step, we insert the text image into the specified region of the background image after text elimination per pixel. Based on the grayscale of each pixel, we combine the two pictures together.

4 Evaluations and Discussions

we have conducted qualitative as well as quantitative evaluation with recent state-of-the-art methods for showing the improved performance of proposed model. Our main focus is on presenting results regarding text elimination module in terms of peak signal-to-noise ratio (PSNR) along with two losses. Moreover, we have provided visual performance of our system as a whole in comparison with the famous text translation systems. The settings for hyperparameters for the proposed model include a learning rate of 0.00019 for the max loss of the generator while 0.3 for minimum loss of generator. The number of epochs used was 12 with batch size 10 and sample interval of 100. The loss function for the modified GAN is l_2 loss of targeted image and ground truth image. Detailed results are described in the following subsections.

4.1 Datasets

4.1.1 Dataset for Text Translation

For text translation module, we have utilized the model similar to open-source neural machine translation (OpenNMT) [21]. However, there is no available En-Ch off-the-shelf model provided on-line. So we train an En-Ch translation model using our own semantic dictionary. The English-Chinese parallel dataset is referred from [33]. The En-Ch corpus utilized in our case is TED2013, which comprises 0.2M sentences.

4.1.2 Text Elimination Dataset

Our synthetic dataset is originally based on [11] which includes 8000 pictures without text information, each of which has size 512×512. The text font dataset is collected from Internet[4], comprises of following eight font styles i.e.

Arial, Century, BMW Helvetica, Impact, Stencil, Tahoma, Times News Roman, and Verdana. Using method described in dataset generation subsection, we output 10 synthesized text corrupted pictures for each picture of dataset, and finally generated 8000 images in total, with their text regions and ground truth. We use the synthesized dataset as our training dataset, and create another 300 images following the same process as our test dataset.

4.1.3 Results of Text Elimination

The performance of our model is compared with well-reputed state-of-the-art models [17, 49] and the methods implemented in openCv [6, 32, 42] on the same test dataset. The results are shown in Table 1. The number of lost pixels in our work is not the same among different images. So, we have calculated the three quantitative metrics as follow:

- *Mean L_1 Loss:* We apply mean function on average value l_1 loss in lost part of every picture compared to ground truth as our L_1 loss.

$$\text{loss}L_1 = \frac{1}{T}\sum_{i=1}^{T}\frac{1}{|L_T|}\sum_{p\epsilon L_T}|p_\text{f} - p_\text{b}| \tag{4.1}$$

  where T stands for number of pictures in test set, L_T means the set of lost pixels in Tth picture, and p_f and p_b is representing final output pixel and background pixel.
- *Mean L_2 Loss:* We calculate the average value of mean l_2 loss in lost part of every picture compared to ground truth as our L_2 loss.

$$\text{loss}L_2 = \frac{1}{T}\sum_{i=1}^{T}\frac{1}{|L_T|}\sum_{p\epsilon L_T}(p_\text{f} - p_\text{b})^2 \tag{4.2}$$

- *Mean PSNR:* The formation of PSNR used in this study as a benchmark measure is slightly different from the standard PSNR, which is calculated concerning whole picture. We calculate mean PSNR_m perpixel value as our benchmark.

$$\text{PSNR}_\text{m} = \frac{1}{T}\sum_{i=1}^{T}\frac{1}{|L_T|}\sum_{p\epsilon L_T}\text{PSNR}(p_\text{f}, p_\text{b}) \tag{4.3}$$

  where PSNR means standard PSNR calculation.

From the qualitative analysis of Fig. 9, we can deduce that our method presents better results than OpneCV and other two approaches for delivering the background in a more natural way. Three state-of-the art approaches used in our model are compared in Graphical form in terms of three quantitative metrics i.e. L_1, L_2 losses and PSNR ratio is described in Table 1. The bars show that our method eliminates the text from open scene images with lower L_1 and L_2 losses having values of 11.69 $\pm$ 0.05, 01.46 $\pm$ 0.01, and higher PSNR ratio up to 01.14 $\pm$ 0.02 as compared to other state-of-the-art techniques.

4.2 Result Comparison

Here we compare our result with Google Translation [10], BaiDu Translation [3], SoGou Translation [37]. Figure 10 shows the visual comparison of aforementioned approaches with our proposed model. We test these systems under different open scenes scenario. In Fig. 10, First column from (a–f) is showing the input images under different scenes environments. The second column presents results of proposed method from (a–f) compared with obtained results in third, fourth, and fifth columns for BaiDu Translation [3], SoGou Translation [37], Google Translation [10], and Youdao [48] systems respectively. From the obtained visual results, we can conclude that, regardless of translation results, our system performs better than Baidu Translation and SoGou Translation (rows a, b, and f of our system output more details than Baidu and SoGou). Our system is comparatively robust when there are highlights (row b and f of our method can still detect the text inside and finish the text elimination jobs with proper alignments) whereas other systems either do not detect texts or have poor elimination and alignment results. From the results, we can conclude that our text translation module regardless of translation results, our system performs better than other translation systems. The text elimination work of proposed method is better than Baidu Translation and SoGou Translation (in Fig. 10 (a, b, f), our system can output more details than Baidu and SoGou). Our system is more robust when there are highlights, as shown in Fig. 10 (b, f), our system can still detect the text inside and finish the text elimination jobs, whereas other systems either do not detect texts, or have poor elimination. Moreover, the proposed model takes into consideration the color of texts like Baidu [3] and Google [10].

As quantitative metrics for performance comparison of various algorithms, we have utilized and calculate their precision, recall, and $F1$-score. To calculate these performance markers, we have compared and matched the list of translated text examples from open scene images to the ground truth labels. Pr represents precision, is premeditated as the proportion of translated text instances that can be matched with original text perfectly. Recall is denoted with Re, is the proportion of ground truth labels that have correspondents in the translated text list. In the end $F1$-score is calculated using formula $F1 = \frac{2*\text{Pr}*\text{Re}}{\text{Pr}+\text{Re}}$ taking both precision and recall into relation. We have presented the obtained results in Table 2, which shows better results obtained by our translation system in

Table 1 Quantitative results of the proposed method in terms of L_1, L_2 losses, and PSNR ratio to show a comparison with recent state-of-the-art techniques

Method	L_1 loss (%)	L_1 loss (%)	PSNR (%)
S. Iizuka et al. [17]	19.00 ± 0.11	03.48 ± 0.12	01.03 ± 0.14
Yu et al. [49]	30.19 ± 0.05	11.62 ± 0.42	00.96 ± 0.03
OpenCV [32]	14.90 ± 0.13	02.75 ± 0.08	01.07 ± 0.07
Proposed	11.69 ± 0.05	01.46 ± 0.01	01.14 ± 0.02

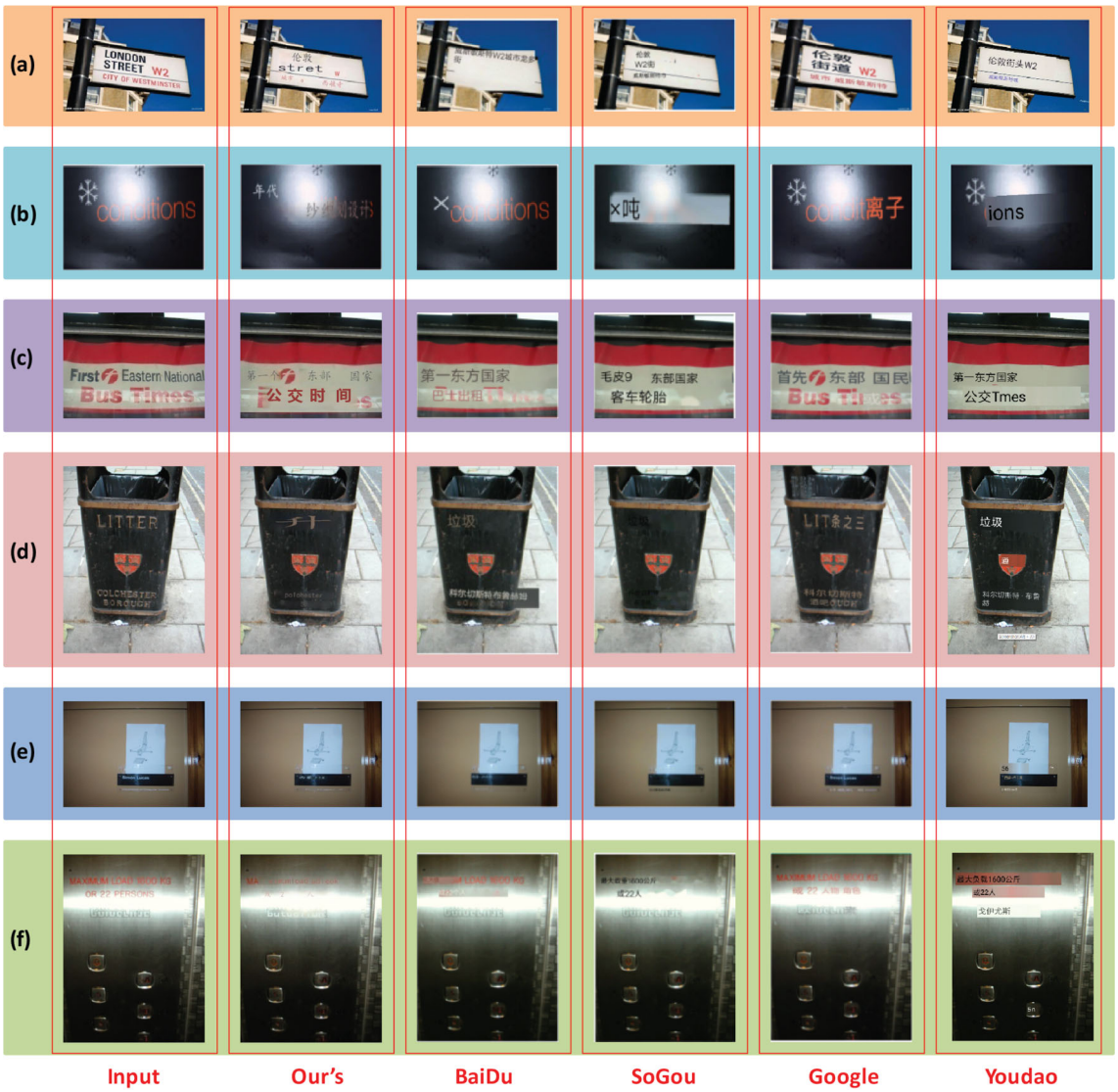

Fig. 10 Qualitative results to visualize the comparison of various text translation systems with the proposed method

Table 2 Quantitative results of the proposed method with three well-known translators in terms of precision, recall, and $F1$-score

Method	Precision	Recall	$F1$-Score
Google translation [10]	88.60 ± 1.54	82.44 ± 1.66	85.429
Baidu translation [3]	89.56 ± 2.33	81.87 ± 2.77	85.542
SoGou translation [37]	85.34 ± 2.77	80.88 ± 2.45	83.050
Youdao translation [48]	89.88 ± 1.73	82.78 ± 2.01	86.157
Proposed	90.87 ± 1.44	83.66 ± 1.67	87.116

terms of precision, recall, and $F1$ scores as compared to other stated approaches.

5 Conclusion

As our world becomes progressively more allied, language translation plays a crucial role in bridging cultural and economic gaps among people from diverse countries and ethnic backgrounds. Text, being the written form of human languages, serves as a practical means to exchange information across time and space. Consequently, automatic text translation from real-world environments has emerged as a cutting-edge research area in machine learning applications. While current text translation systems demonstrate efficiency in text detection and recognition, translating text automatically from open scene images remains a formidable challenge due to factors such as text diversity, complex backgrounds, and suboptimal imaging conditions in natural environments. Our proposed model aims to address these challenges by offering a comprehensive solution consisting of five modules: scene text detection, text recognition, text removal, text translation into the specified language, and text insertion alongside scene reconstruction. Our model surpasses popular translation services such as Google, Baidu, and Sogou. Moreover, it ensures that the translated text retains its original meaning while maintaining fluency and accuracy.

Funding Open access funding provided by Universitá degli Studi di Firenze within the CRUI-CARE Agreement.

References

1. Arbeláez, P., Maire, M., Fowlkes, C., et al.: Contour detection and hierarchical image segmentation. IEEE Trans Pattern Anal Mach Intell **33**(5), 898–916 (2011). https://doi.org/10.1109/TPAMI.2010.161
2. Bai, X., Yao, C., Liu, W.: Strokelets: a learned multi-scale mid-level representation for scene text recognition. IEEE Trans Image Process **25**(6), 2789–2802 (2016). https://doi.org/10.1109/TIP.2016.2555080
3. Baidu translate (2023) Accessed 27 Nov 2023 https://fanyi.baidu.com/translate
4. Bartz, C., Yang, H., Meinel, C.: SEE: towards semi-supervised end-to-end scene text recognition. CoRR abs/1712.05404. (2017). arXiv:1712.05404
5. Bertalmio, M., Sapiro, G., Caselles, V., et al.: Image inpainting. In: Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, SIGGRAPH '00, pp. 417–424 (2000). https://doi.org/10.1145/344779.344972
6. Bertalmio, M., Bertozzi, A.L., Sapiro, G.: Navier-stokes, fluid dynamics, and image and video inpainting. In: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, pp. I–I (2001). https://doi.org/10.1109/CVPR.2001.990497
7. Dekel, T., Rubinstein, M., Liu, C., et al.: On the effectiveness of visible watermarks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6864–6872 (2017) https://doi.org/10.1109/CVPR.2017.726
8. Eudic (european dictionary) (2023) Accessed 27 Nov 11 2023 https://eudict.com/
9. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., et al.: Generative adversarial nets. In: NIPS (2014)
10. Google translator (app) (2023). Accessed Nov 27 2023 https://translate.google.com/
11. Gupta, A., Vedaldi, A., Zisserman, A.: Synthetic data for text localisation in natural images. CoRR abs/1604.06646. (2016) arXiv:1604.06646
12. Hanson, J., Paliwal, K., Litfin, T., et al.: Accurate prediction of protein contact maps by coupling residual two-dimensional bidirectional long short-term memory with convolutional neural networks. Bioinformatics **34**(23), 4039–4045 (2018). https://doi.org/10.1093/bioinformatics/bty481
13. He, K., Zhang, X., Ren, S., et al.: Deep residual learning for image recognition. CoRR abs/1512.03385. (2015) arXiv:1512.03385
14. He, T., Huang, W., Qiao, Y., et al.: Text-attentional convolutional neural network for scene text detection. IEEE Trans Image Process **25**(6), 2529–2541 (2016). https://doi.org/10.1109/TIP.2016.2547588
15. He, W., Zhang, X., Yin, F., et al.: Multi-oriented and multi-lingual scene text detection with direct regression. IEEE Trans Image Process **27**(11), 5406–5419 (2018). https://doi.org/10.1109/TIP.2018.2855399
16. Huang, M., Liu, Y., Peng, Z., et al.: Swintextspotter: Scene text spotting via better synergy between text detection and text recognition. In: proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4593–4603 (2022)
17. Iizuka, S., Simo-Serra, E., Ishikawa, H.: Globally and locally consistent image completion. ACM Trans. Graph. (Proc of SIGGRAPH 2017) **36**(4), 107:1-107:14 (2017)
18. Imagemagick: Convert, edit, and compose images (2023) https://www.imagemagick.org/script/index.php
19. Karaoglu, S., Tao, R., van Gemert, J.C., et al.: Con-text: text detection for fine-grained object classification. IEEE Trans. Image Process. **26**(8), 3965–3980 (2017). https://doi.org/10.1109/TIP.2017.2707805
20. KhoKhar, F.A., Shah, J.H., Khan, M.A., et al.: A review on federated learning towards image processing. Comput. Electr. Eng. **99**, 107818 (2022)
21. Klein, G., Kim, Y., Deng, Y., et al.: Opennmt: open-source toolkit for neural machine translation. CoRR abs/1701.02810. (2017) arXiv:1701.02810
22. Koo, H.I.: Text-line detection in camera-captured document images using the state estimation of connected components. IEEE Trans. Image Process. **25**(11), 5358–5368 (2016). https://doi.org/10.1109/TIP.2016.2607418
23. Kumar, P., Raman, B.: A Bert based dual-channel explainable text emotion recognition system. Neural Netw. **150**, 392–407 (2022)

24. Liao, J., Buchholz, B., Thiery, J., et al.: Indoor scene reconstruction using near-light photometric stereo. IEEE Trans. Image Process. **26**(3), 1089–1101 (2017). https://doi.org/10.1109/TIP.2016.2636661
25. Liao, M., Shi, B., Bai, X.: Textboxes++: a single-shot oriented scene text detector. IEEE Trans. Image Process. **27**(8), 3676–3690 (2018). https://doi.org/10.1109/TIP.2018.2825107
26. Liu, F., Shen, C., Lin, G.: Deep convolutional neural fields for depth estimation from a single image (2014)
27. Long, S., He, X., Yao, C.: Scene text detection and recognition: the deep learning era. CoRR abs/1811.04256. (2018) arXiv:1811.04256
28. Lu, S., Ding, Y., Liu, M., et al.: Multiscale feature extraction and fusion of image and text in VQA. Int. J. Computat. Intell. Syst. **16**(1), 54 (2023)
29. Mustafa, A., Kim, H., Hilton, A.: Msfd: multi-scale segmentation-based feature detection for wide-baseline scene reconstruction. IEEE Trans. Image Process. **28**(3), 1118–1132 (2019). https://doi.org/10.1109/TIP.2018.2872906
30. Neumann, L., Matas, J.: Real-time lexicon-free scene text localization and recognition. IEEE Trans. Pattern Anal. Mach. Intell. **38**(9), 1872–1885 (2016). https://doi.org/10.1109/TPAMI.2015.2496234
31. Osher, S., Yin, W., Goldfarb, D., et al.: An iterative regularization method for total variation-based image restoration. Multiscale Model. Simul. **4**, 23 (2005). https://doi.org/10.1137/040605412
32. Opencv: (2023) Accessed 27 Nov 2023 https://opencv.org/
33. Opus dataset: Translated text. (2023). http://opus.nlpl.eu/
34. Pathak, D., Krähenbühl, P., Donahue, J., et al.: Context encoders: feature learning by inpainting. CoRR abs/1604.07379. (2016) arXiv:1604.07379
35. Pérez, P., Gangnet, M., Blake, A.: Poisson image editing. ACM Trans Graph **22**(3), 313–318 (2003). https://doi.org/10.1145/882262.882269
36. Rong, X., Yi, C., Tian, Y.: Unambiguous scene text segmentation with referring expression comprehension. IEEE Trans. Image Process. **29**, 591–601 (2023). https://doi.org/10.1109/TIP.2019.2930176
37. Sogou translate. (2023). Accessed 27 Nov 2023 http://fanyi.sogou.com/
38. Shen, S.: Accurate multiple view 3d reconstruction using patch-based stereo for large-scale scenes. IEEE Trans. Image Process. **22**(5), 1901–1914 (2013). https://doi.org/10.1109/TIP.2013.2237921
39. Shi, B., Bai, X., Yao, C.: An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. IEEE Trans. Pattern Anal. Mach. Intell. **39**(11), 2298–2304 (2017). https://doi.org/10.1109/TPAMI.2016.2646371
40. Tamilselvi, M., Ramkumar, G., Anitha, G., et al.: A novel text recognition scheme using classification assisted digital image processing strategy. In: 2022 International Conference on Advances in Computing, pp. 1–6. Communication and Applied Informatics (ACCAI), IEEE (2022)
41. Tang, Y., Wu, X.: Scene text detection and segmentation based on cascaded convolution neural networks. IEEE Trans. Image Process. **26**(3), 1509–1520 (2017). https://doi.org/10.1109/TIP.2017.2656474
42. Telea, A.: An image inpainting technique based on the fast marching method. J. Graph. Tools (2004). https://doi.org/10.1080/10867651.2004.10487596
43. Xu, Y., Wang, Y., Zhou, W., et al.: Textfield: learning a deep direction field for irregular scene text detection. IEEE Trans. Image Process. **28**(11), 5566–5579 (2019). https://doi.org/10.1109/TIP.2019.2900589
44. Yang, C., Lu, X., Lin, Z., et al.: High-resolution image inpainting using multi-scale neural patch synthesis. CoRR abs/1611.09969. (2016) arXiv:1611.09969
45. Yang, C., Yin, X., Pei, W., et al.: Tracking based multi-orientation scene text detection: a unified framework with dynamic programming. IEEE Trans. Image Process. **26**(7), 3235–3248 (2017). https://doi.org/10.1109/TIP.2017.2695104
46. Yao, C., Bai, X., Shi, B., et al.: Strokelets: a learned multi-scale representation for scene text recognition. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition, pp. 4042–4049 (2014). https://doi.org/10.1109/CVPR.2014.515
47. Yao, C., Bai, X., Sang, N., et al.: Scene text detection via holistic, multi-channel prediction. CoRR abs/1606.09002 (2016). arXiv:1606.09002
48. Youdao (app). (2023). Accessed 28 Nov 2023 https://www.youdao.com/
49. Yu, J., Lin, Z., Yang, J., et al.: Generative image inpainting with contextual attention. CoRR abs/1801.07892 (2018). arXiv:1801.07892
50. Zhou, X., Yao, C., Wen, H., et al.: EAST: an efficient and accurate scene text detector. CoRR abs/1704.03155 (2017). arXiv:1704.03155