Exploring In-Image Machine Translation with Real-World Background

Yanzhi Tian[1] **Zeming Liu**[2] **Zhengyang Liu**[1] **Yuhang Guo**[1]*
[1]School of Computer Science and Technology, Beijing Institute of Technology
[2]School of Computer Science and Engineering, Beihang University
tianyanzhi@bit.edu.cn zmliu@buaa.edu.cn
zhengyang@bit.edu.cn guoyuhang@bit.edu.cn

Abstract

In-Image Machine Translation (IIMT) aims to translate texts within images from one language to another. Previous research on IIMT was primarily conducted on simplified scenarios such as images of one-line text with black font in white backgrounds, which is far from reality and impractical for applications in the real world. To make IIMT research practically valuable, it is essential to consider a complex scenario where the text backgrounds are derived from real-world images. To facilitate research of complex scenarios IIMT, we design an IIMT dataset that includes subtitle text with a real-world background. However, previous IIMT models perform inadequately in complex scenarios. To address the issue, we propose the DebackX model, which separates the background and text-image from the source image, performs translation on the text-image directly, and fuses the translated text-image with the background to generate the target image. Experimental results show that our model achieves improvements in both translation quality and visual effect.[1]

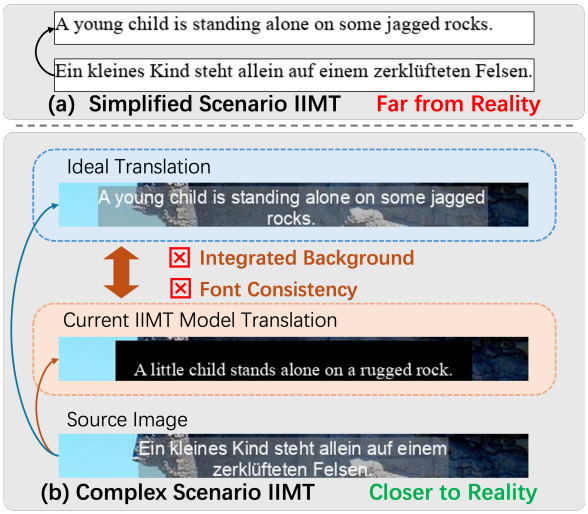


Figure 1: Illustration of simplified and complex scenario IIMT. Previous research mainly focuses on a simplified scenario IIMT, which is far from reality. Exploring complex IIMT scenarios that are much closer to reality is necessary. We find that the current IIMT model cannot fully handle the complex scenario, since the translation lacks an integrated background and fails to maintain font consistency, leading to a poor visual effect.

1 Introduction

In-Image Machine Translation (IIMT) aims to translate texts within images from one language to another, and related technologies are widely used in translation applications (Mansimov et al., 2020; Tian et al., 2023; Lan et al., 2024). IIMT performs translation on visual modality, helping people understand context in images directly, which is different from text-to-text machine translation.

IIMT is a challenging machine translation task, and previous research mainly focuses on simplified scenarios as shown in Figure 1(a). Tian et al. (2023) translate images containing texts with black font and white background. Lan et al. (2024) construct images with a single color background. However, the aforementioned research overly simplifies the IIMT task, ignoring the fact that most of the text in the real world appears with a complex background, such as video subtitles, as shown in Figure 1(b). The limitation makes previous approaches impractical for applications in the real world, highlighting the need for a scenario that is closer to reality.

To overcome the limitation and make IIMT research practical, we focus on a more realistic IIMT task, namely complex scenario IIMT, building a new IIMT dataset where the images contain text integrated with a real-world background. Compared to the simplified IIMT task that hardly aligns with reality, our designed task better reflects the real world.

On the complex scenario dataset, we experiment

*Corresponding author
[1]Code and dataset: https://github.com/BITHLP/DebackX

Findings of the Association for Computational Linguistics: ACL 2025, pages 124–137
July 27 - August 1, 2025

with a feasible OCR-NMT-Render model, which primarily consists of four steps: (1) Recognizing the source language texts with the Optical Character Recognition (OCR) model; (2) Translating the source texts recognized by the OCR model with the Neural Machine Translation (NMT) model; (3) Removing the text areas in the source image to erase the texts; (4) Rendering the translation results into the corresponding position of the image.

The above framework has the following drawbacks. Firstly, the cascading OCR and NMT models have the risk of error propagation, leading to a decrease in translation quality, which is demonstrated in the previous works (Tian et al., 2023; Lan et al., 2024). Secondly, removing text areas in step (3) affects the integrity of the source images, decreasing the visual effect of rendering results. Thirdly, OCR models generally only give recognized text results, leading to the loss of font information during the cascading process, and it is hard to maintain consistency for font styles of texts between the source images and target images. The comparison between the ideal translation result and the translation produced by the OCR-NMT-Render model is shown in Figure 1(b).

To address the drawbacks, we design the DebackX model for a complex scenario IIMT, consisting of three components: (i) A Text-Image Background Separation model is used to generate the background and text-image from the source image. Processing the background and text-image separately, our model ensures the integrity of the background. (ii) An Image Translation model is used to model the transformation from the text-image of the source language to the text-image of the target language. The direct translation mitigates the decreasing translation quality caused by error propagation between OCR and NMT and helps maintain font consistency between the target images and source images. (iii) A Text-Image Background Fusion model is used to fuse the background and target text-image to generate the final output.

The main contributions of this paper are as follows:

- We propose a complex scenario IIMT, which requires translating text within real-world backgrounds to align closely with reality.
- We build a new dataset, IIMT30k, containing backgrounds derived from real-world images for complex scenario IIMT. Existing IIMT models, including GPT-4o, fail to achieve both high-quality translation and visual effects.
- We propose the DebackX model for complex scenario IIMT, processing the background and text of the image separately, achieving both better translation quality and visual effect compared to current IIMT models.

2 Related Work

Text-Image Translation. Conventional research on Neural Machine Translation usually focuses on text and speech modality (Vaswani et al., 2023; Inaguma et al., 2023). There is also research focusing on translating the text in the image, which is an image-to-text multi-modality translation task, referred to Text-Image Translation (TIT). Ma et al. (2022) construct a synthetic dataset and employ multi-task learning in the training of end-to-end TIT model. Lan et al. (2023) annotate the OCRMT30K dataset, and propose an OCR-NMT cascade TIT model, introducing visual information with a multimodal codebook into the NMT model. Ma et al. (2023a) propose a cross-model cross-lingual interactive model for TIT with weighted interactive attention and hierarchical interactive attention. Zhu et al. (2023) design a shared encoder-decoder backbone with a vision-text representation aligner and a cross-model regularizer.

In-Image Machine Translation. All of the above works only generate the translated texts, but do not generate the images with translation. Mansimov et al. (2020) explore the In-Image Machine Translation (IIMT) task, aiming to transform images containing texts from one language to another, which is closer to real-world applications. Both the input and output of IIMT are entirely based on the image modality, significantly different from other machine translation tasks. Tian et al. (2023) construct a dataset containing images of one-line texts with black font and white background, improving the translation quality of IIMT with a segmented pixel transformer. Lan et al. (2024) adopt the ViT-VQGAN to IIMT with OCR and TIT multi-task learning, constructing a dataset with a single color as a background with multi-line texts. Qian et al. (2024) utilize Qwen and AnyText models to translate images with short text on a real dataset, but do not release their dataset publicly.

Text-within-Image Generation The research of text-within-image generation aims to generate clear

and readable text in the images, which is generally necessary to incorporate textual priors, such as text position and image with rendered text (Ma et al., 2023b; Tuo et al., 2024; Zhang et al., 2023). Rodríguez et al. (2023) propose OCR-VQGAN, which uses OCR pre-trained features to calculate the text perceptual loss, to mitigate the phenomenon of the generated blurring text with VQGAN (Esser et al., 2021). And it is generally necessary to incorporate textual priors into the model, such as text position and image with rendered texts (Ma et al., 2023b; Tuo et al., 2024; Zhang et al., 2023).

3 Data Construction

In previous works of IIMT, Mansimov et al. (2020); Tian et al. (2023) construct images with black font and white background of one-line texts. Lan et al. (2024) render multi-line texts into single-color backgrounds. The backgrounds of the above datasets are simple, and there is a gap between the constructed datasets and reality. Therefore, we build a dataset containing real-world images as backgrounds, which is lacking in IIMT research. The images in our dataset are more complex, which are shown in Figure 1(b).

We build the dataset with images and texts from Multi30k[2] (Elliott et al., 2016). The Multi30k dataset is designed for Multimodal Machine Translation, and there are captions in several languages for each image. Firstly, we resize the images in Multi30k to 512×512. Secondly, the German and English text captions are rendered into the image by Pillow[3] library. Finally, each image is cropped to a size of 48×512, thereby fully preserving the text area and part of the background.

Our constructed IIMT30k dataset comprises three subsets, each characterized by a specific font: Times New Roman (TNR), Arial, and Calibri. The texts in each subset of images are exclusively presented in one of these three fonts, namely IIMT30k-TNR, IIMT30k-Arial, and IIMT30k-Calibri. Each subset is further divided into training, validation, and test sets. The purpose of using multiple fonts is to investigate the font adaption capability of the model. We perform inspections on our dataset, ensuring the integrity of texts in the images. Although the dataset is synthetic, it is comparable to real-world video subtitle images, and samples from the dataset are shown in Appendix C.

[2] https://github.com/multi30k/dataset
[3] https://pypi.org/project/pillow/

4 Method

We design the DebackX model, which is shown in Figure 2, and our model contains three components, the Text-Image Background Separation, the Image Translation, and the Text-Image Background Fusion. The Text-Image Background Separation model first decomposes the source image into a background image and a source text-image. Then the Image Translation model transforms the source text-image into the target text-image. Finally, the Text-Image Background Fusion model fuses the background image and target text-image, generating the target image.

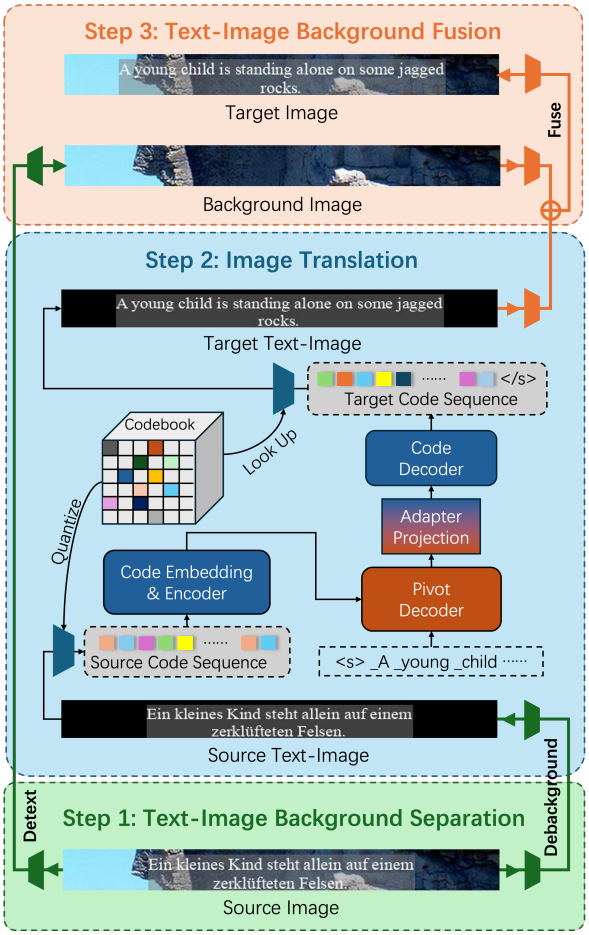


Figure 2: Architecture of our proposed DebackX.

4.1 Text-Image Background Separation

The Text-Image Background Separation model decomposes the source image into a background image and a source text-image. The input source image x is encoded with two ViT (Dosovitskiy et al., 2021) encoders, E_{deback} and E_{detext}, obtaining corresponding features separately. Then two ViT decoders G_{back} and G_{text} take the features as

inputs, outputting the source text-image and the background image. Formally as:

$$\text{Background Image} : G_{\text{back}}(E_{\text{deback}}(x)),$$
$$\text{Source Text-Image} : G_{\text{text}}(E_{\text{detext}}(x)).$$

4.2 Image Translation

The Image Translation model is used to transform source text-images into target text-images. The source text-image x is encoded to feature by a ViT encoder E, and tokenized with a codebook q into $z = (z_1, z_2, \ldots, z_N)$. The codebook q contains V learnable vectors $\{e_1, e_2, \ldots, e_V\}$, and each encoded feature x_i is quantized by the nearest vector in q, obtaining z_i. Formally as:

$$z_i = q(E(x_i)) = \arg\min_{e_k \in q} ||E(x_i) - e_k||_2. \quad (1)$$

Therefore, the images can be represented using the indices of the vectors in the codebook. By tokenizing the source and target text-images into code sequences, the continuous representation of the image is transformed into a discrete sequence. Consequently, the image-to-image translation task shifts to a transformation from source code sequence to target code sequence. Then, the source code sequence is passed through a Code Embedding layer and a Code Encoder in turn, generating the representation H^E_{code}.

To incorporate the semantic information required for the translation task into the code sequence, a Pivot Decoder is trained with TIT auxiliary task. The target language text shifted to the right and prefixed with a Beginning of Sentence (BOS) token, obtaining word embedding E_{TIT} through a Text Embedding layer. The word embedding E_{TIT} and the Code Encoder representation H^E_{code} are jointly used as inputs to the Pivot Decoder, formally as $\text{PivotDecoder}(E_{\text{TIT}}, H^E_{\text{code}}, H^E_{\text{code}})$[4].

The output of the Pivot Decoder H^D_{pivot} serves two purposes. First, it is projected to the vocabulary size through a linear layer for training the auxiliary TIT task. Second, it is used as input for the subsequent Code Decoder that generates the target code sequence.

For the Code Decoder, the target code sequence is first prefixed with a BOS token and then passed through the Code Embedding layer to obtain the vector representation E_{code}. The representation from the Pivot Decoder H^D_{pivot} is passed through a Linear Adapter to produce H^A_{pivot}. Both E_{code} and H^A_{pivot} are used as inputs for the Code Decoder, formally as $\text{CodeDecoder}(E_{\text{code}}, H^A_{\text{pivot}}, H^A_{\text{pivot}})$.

The output code sequence of the Code Decoder is transformed into vector representations by looking up in the codebook and then decoded into the target text-image using a ViT Decoder.

During the inference stage, the target text is decoded autoregressively with the source code sequence. After the target text is completely decoded, the complete representation from the Pivot Decoder H^D_{pivot} can be obtained, which is used for decoding the target code sequence autoregressively.

4.3 Text-Image Background Fusion

The Text-Image Background Fusion model fuses the background image and the target text-image, which are generated by the Text-Image background Separation model and Image Translation model to the target image. Two ViT encoders E_{back} and E_{text} are used to encode the background image i_b and the target text-image i_t, obtaining the corresponding features. The sum of the features is passed to a ViT decoder G_{fuse} to get the final output of the model, formally as:

$$\text{Target Image} : G_{\text{fuse}}(E_{\text{back}}(i_b) + E_{\text{text}}(i_t)). \quad (2)$$

5 Training

5.1 Text-Image Background Separation

We adopt the training method of the image generation task to train the Text-Image Background Separation model. The reconstruction loss and perceptual loss[5] (Zhang et al., 2018) are two commonly used criteria for image generation, and we use the combination of them as described in Equation 3.

$$\mathcal{L}_{\text{img}}(y, \hat{y}) = ||y - \hat{y}||^2 + \lambda_{\text{p}} \mathcal{L}_{\text{Perceptual}}(y, \hat{y}), \quad (3)$$

where y is the generation result of the model and $\hat{y}$ is the ground truth image. λ_{p} is loss weight, and it is set to 0.1 in our experiments.

Both the background image i_b and text-image i_t generation use the loss function. The total loss function at this stage is the sum of those mentioned above, formally as $\mathcal{L}_{\text{sep}} = \mathcal{L}_{\text{img}}(i_b, \hat{i_b}) + \mathcal{L}_{\text{img}}(i_t, \hat{i_t})$.

[4]The input of the Transformer Decoder can generally be represented as $\text{Decoder}(Q, K, V)$.

[5]https://github.com/richzhang/PerceptualSimilarity

5.2 Image Translation

The training of our Image Translation model consists of two stages.

Stage 1: Vector Quantization. The codebook and ViT Encoder-Decoder, which are used to tokenize and detokenize the images, are trained with image reconstruction tasks. We use the commitment loss for the training of the codebook, and the total loss function for Stage 1 is described as Equation 4. Besides, Exponential Moving Average (EMA) is also used for updating the codebook, which is a more robust training method (Łańcucki et al., 2020).

$$\mathcal{L}_{\text{VQ}}(y, \hat{y}) = ||y - \hat{y}||^2 + \lambda_{\text{p}}\mathcal{L}_{\text{Perceptual}}(y, \hat{y}) + ||\text{sg}[z_q] - E(x)||_2^2, \quad (4)$$

where y is the generated image of the model and $\hat{y}$ is the ground truth image. $||\text{sg}[z_q] - E(x)||_2^2$ is the commitment loss with stop-gradient operation $\text{sg}[\cdot]$, and z_q is the vector obtained by quantization using the codebook, while x is the source image and $E(x)$ is the output feature of the ViT encoder which also serves as the input to the quantization layer. λ_{p} is loss weight, and it is set to 0.1 in our experiments.

Stage 2: Translation. We train the translation model to transform the codes of source text-images to the codes of target text-images with the TIT auxiliary task. Both codes and auxiliary texts utilize cross-entropy with a label-smoothing factor of 0.1 as the loss function, denoted as $\mathcal{L}_{\text{code}}$ and $\mathcal{L}_{\text{TIT}}$. The total loss function of this stage is the sum of two components, formally as $\mathcal{L}_{\text{trans}} = \mathcal{L}_{\text{code}} + \mathcal{L}_{\text{TIT}}$.

In this stage, the Embedding Layers, the Code Encoder, the Pivot Decoder, the Code Decoder, the Adapter Projection, and the Output Projections are trained together. Besides, it should be noted that we construct additional text-images only for training at this stage. The additional text-images are used to pre-train the model, while the IIMT30k training set is for fine-tuning.

5.3 Text-Image Background Fusion

The Text-Image Background Fusion model takes the separated background image and the translated text-image as input, the target image as output, which is trained with the image generation loss function described as Equation 3.

6 Experiments

6.1 Metrics

It is hard to evaluate the translation quality of IIMT directly, so we first recognized the texts in the generated images with the OCR model and calculated the BLEU (Papineni et al., 2002) and COMET (Rei et al., 2020) scores between the OCR results and the reference texts, which is a commonly used method. We use EasyOCR[6], a widely used OCR toolkit that supports both German and English to recognize texts in the output images. BLEU and COMET scores are calaulted by Sacrebleu[7] and Unbabel-COMET[8] respectively.

To evaluate the visual effect of the output images, we use a widely used metric, Fréchet Inception Distance (FID) (Heusel et al., 2018), which is a measure of similarity between two images, calculated by computing the Fréchet distance between two Gaussians fitted to feature representations of the Inception network. FID is shown to correlate well with the human judgment of visual quality, and it is most often used to evaluate the quality of generated images (Lucic et al., 2018). In our experiments, we employ pytorch-fid[9] to calculate FID scores.

6.2 Experimental Settings

From the perspective of input and output modalities, IIMT is an image-to-image generation task. Therefore, we adopt several image-to-image generation models (Esser et al., 2021; Tian et al., 2024; Chang et al., 2022) for the IIMT task. Besides, we build models consisting TIT models (Zhu et al., 2023; Lan et al., 2023) and Rendering. We also investigate the performance of the latest IIMT model, Translatotron-V (Lan et al., 2024). The experiments are conducted on the IIMT30k-TNR dataset. The training set includes $25,205$ image pairs containing German and English parallel texts, and the valid set includes 864 image pairs, while the test set includes $2,740$ image pairs. The following is a brief introduction to each system.

VQGAN (Esser et al., 2021). A system that tokenizes images using a codebook and generates images using an autoregressive Transformer Decoder. In our experiments, we train the codebook using all the German and English images and train the

[6] https://github.com/JaidedAI/EasyOCR
[7] https://github.com/mjpost/sacrebleu
[8] https://github.com/Unbabel/COMET
[9] https://github.com/mseitzer/pytorch-fid

Systems	Translation Quality (BLEU ↑ / COMET ↑)				Visual Effect (FID ↓)			
	De-En		En-De		De-En		En-De	
	Valid	Test	Valid	Test	Valid	Test	Valid	Test
VQGAN	0.7 / 24.2	0.6 / 24.4	0.6 / 17.9	0.8 / 20.2	25.5	21.3	27.2	20.7
VAR	0.1 / 22.5	0.1 / 22.2	0.3 / 18.3	0.2 / 17.7	21.2	15.7	20.1	14.1
MaskGIT	0.2 / 23.7	0.2 / 24.0	0.1 / 18.9	0.2 / 18.5	21.5	20.9	22.2	18.8
TIT-Render	13.8 / 53.1	12.1 / 49.6	14.0 / **46.3**	10.2 / **43.9**	137.4	133.2	121.7	119.0
PEIT-Render	10.5 / 50.4	8.6 / 47.0	12.3 / 41.1	7.9 / 35.6	149.9	140.1	125.3	119.7
McTIT-Render	14.2 / **53.5**	11.7 / 47.3	13.5 / 43.3	10.5 / 42.8	141.2	137.5	124.9	117.5
Translatotron-V	2.7 / 30.1	1.6 / 24.8	2.1 / 26.5	1.9 / 24.6	22.4	10.1	23.2	17.5
DebackX (ours)	**14.9** / 51.2	**12.8 / 50.0**	**14.6** / 42.2	**11.1** / 40.0	**20.4**	**9.0**	**19.5**	**8.7**

Table 1: Experimental results of different systems. Metrics include translation quality (BLEU, COMET) and visual effect (FID). ↑ or ↓ indicates higher or lower values are better.

Transformer Decoder models separately for each translation direction.

VAR (Tian et al., 2024). An improved system based on VQGAN, utilizing a more efficient and effective autoregressive generation paradigm, namely "next-resolution prediction".

MaskGIT (Chang et al., 2022). An image-to-image generation system that uses a bidirectional transformer decoder for synthesizing high-fidelity and high-resolution images. It predicts masked tokens in all directions during training and refines images iteratively during inference.

TIT-Render. A pipeline system contains TIT model and text rendering. In our experiments, we use Transformer based image encoder and text decoder to complete the TIT task, obtaining the translated texts in the source images.

PEIT-Render. An advanced TIT model PEIT (Zhu et al., 2023) and text rendering pipeline.

McTIT-Render. An OCR-NMT TIT model McTIT (Lan et al., 2023) and text rendering pipeline.

Translatotron-V (Lan et al., 2024). An IIMT model adopts the architecture of ViT-VQGAN, with OCR and TIT multi-task training.

DebackX. Our IIMT model introduced in Section 4. The implementation details are introduced in Appendix A and Appendix B.

6.3 Main Results

The experimental results of different systems are shown in Table 1. The autoregressive-based image-to-image generation model (VQGAN) can produce relatively clear characters while maintaining a sharp background. However, it fails to generate complete words and semantically meaningful phrases in the image, resulting in poor translation quality. The image-to-image generation models that do not adopt conventional autoregressive (VAR and MaskGIT) produce relatively clear backgrounds but fail to generate clear characters, resulting in even worse translation quality.

The methods consisting of TIT models and Rendering (TIT-Render, PEIT-Render, and McTIT-Render) can obtain meaningful translation results, but the rendering procedure removes the areas containing texts, significantly affecting the visual effect. The Translatotron-V is designed for IIMT with a simple background, but cannot adapt to IIMT with complex scenarios, resulting in poor translation quality.

Compared with other IIMT methods, our system specifically designs an Image Translation model, achieving a better translation quality. Furthermore, the Text-Image Background Separation and Fusion models lead to the improvement of visual effects.

7 Analysis

IIMT produces visual output, requiring evaluation of both translation quality and visual effect. In this section, we conduct the following analytical experiments focusing on the above evaluation aspects:

Translation quality. (1) We analyze the effectiveness of the pre-training followed by the fine-tuning strategy adopted in the Image Translation model and demonstrate how pre-training contributes to improvements in translation quality. (2) We conduct an ablation study to investigate the contribution of each component in our model to the translation quality.

Visual effect. (3) Maintaining font consistency is crucial for the visual effect of translated outputs, yet this aspect is not captured by the used FID metric directly. To address the lack of font adaptation in current IIMT research, we use a dataset containing multiple fonts to evaluate whether our DebackX model can handle this challenge. (4) We conduct a case study to illustrate how the output of our model differs from other IIMT models, including TIT-Render and GPT-4o. Visually demonstrate the advantages of our DebackX model.

7.1 Pre-training Study

Due to the size of the dataset, which consists of images and their captions in multiple languages is small, it is hard to build a large amount of images with complex backgrounds to train IIMT models directly. However, text-images used to train the Image Translation model are easy to construct witha parallel corpus. Other IIMT systems cannot utilize such data, which is an advantage of our model. We construct 100K German-English text-image pairs with the IWSLT dataset used to pre-train the Image Translation model ("+IWSLT PT"). It should be noted that we use this system in the main experiments (Table 1) since its parameter is comparable to other systems. Moreover, we construct 1M text-image pairs with WMT14 datasets for pre-training ("+WMT14 PT"). To match the parameter of the model with the 1M size of the dataset, we use a larger size of the model.

Training Sets	De-En		En-De	
	Valid	Test	Valid	Test
IIMT30k-TNR	7.4	5.9	8.1	5.4
+IWSLT PT	14.9	12.8	14.6	11.1
+WMT14 PT	20.6	17.5	18.9	14.4

Table 2: BLEU scores on different training sets, "IIMT30k-TNR" refers to only training with the IIMT30k-TNR training set, while "+IWSLT PT" and "+WMT14 PT" refer to pre-training with the corresponding dataset first and then fine-tuning on the IIMT30k-TNR training set.

The experimental results are shown in Table 2, and results demonstrate the effectiveness of pre-training for the Image Translation Model. By increasing the size of the dataset, the translation quality is improved.

7.2 Ablation Study

We conduct an ablation study to further investigate the contribution of each part in our model. Firstly, we remove the Pivot Decoder, which also means there is no TIT auxiliary task (#2), and the Image Translation model learns to transform the code sequence directly. Secondly, we remove both the Text-Image Background Separation and Fusion models (#3), training a new image-to-image translation model that takes the source image as input and produces the target image as output directly, which is similar to the two-pass model (Inaguma et al., 2023), and there is no procedure of deback-ground in the model. Finally, we remove both parts mentioned above (#4).

Since the experiment #3 and #4 cannot utilize the large number of text-images for pre-training, to ensure the comparability of the experimental results, we do not use any pre-training data in the ablation study.

#	Pivot	Deback	De-En		En-De	
			Valid	Test	Valid	Test
1	✓	✓	7.4	5.9	8.1	5.4
2	✗	✓	1.5	1.4	1.6	1.4
3	✓	✗	1.2	1.1	1.3	1.3
4	✗	✗	0.6	0.6	0.7	0.9

Table 3: BLEU scores of ablation study, "Pivot" refers to the TIT auxiliary task with Pivot Decoder, and "Deback" refers to the Text-Image Background Separation and Fusion models.

The experimental results are shown in Table 3, which demonstrates that the Pivot Decoder, the Text-Image Background Separation, and Fusion models contribute to the translation quality. Moreover, the results also indicate that using only a two-pass model leads to lower translation quality in complex scenarios IIMT.

7.3 Multiple Fonts Adaption Study

To investigate the adaptation capability of multiple fonts, we mix the three subsets, IIMT30k-TNR, IIMT30k-Arial, and IIMT30k-Calibri. All the training sets, valid sets, and test sets consist of multiple font types.

The experimental results are shown in Table 5, and the results illustrate that the datasets constructed with multiple fonts do not affect the translation quality significantly. The reason why the BLEU scores for "+Arial" and "+Arial Calibri" are

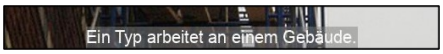
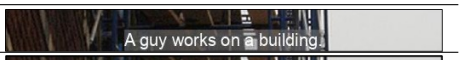
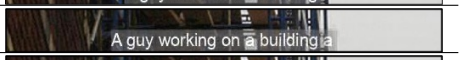
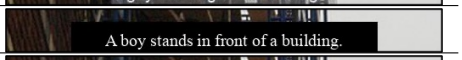
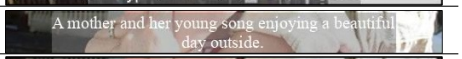
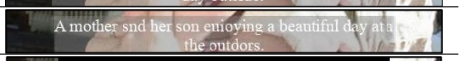
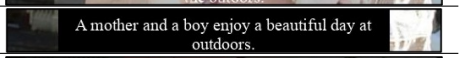
#	Source Image (Input)	Systems	Target Image (Output)
1		Ground Truth	
		DebackX (Ours)	
		TIT-Render	
		GPT-4o	
2		Ground Truth	
		DebackX (Ours)	
		TIT-Render	
		GPT-4o	

Table 4: Comparison of the outputs of different systems, including our DebackX model, TIT-Render, and GPT-4o.

higher than that for "TNR" is the OCR model used for evaluation has a lower error rate in recognizing Arial and Calibri fonts in images compared to Times New Roman.

Training Sets	De-En		En-De	
	Valid	Test	Valid	Test
TNR	7.4	5.9	8.1	5.4
+Arial	10.4	8.5	9.2	6.5
+Arial Calibri	10.8	8.6	9.5	6.9

Table 5: BLEU scores on training sets with multiple fonts. "TNR" refers to the dataset that is only constructed with Times New Roman. "+Arial" refers to the dataset that was constructed with 2 fonts, Times New Roman and Arial. "+Arial Calibri" refers to the dataset that was constructed with 3 fonts, Times New Roman, Arial, and Calibri.

We conduct a manual font style recognition for the output images. If the font of characters in the output images differs from that in the input image, they are marked as inconsistent. After the manual evaluation, the font style consistency between the output images and the input images for "+Arial" is 97.3%, and for "+Arial Calibri" is 96.5%. Experimental results show that our model has the capability to adaption to multiple fonts.

7.4 Case Study

We illustrate the differences between the outputs of our DebackX model and TIT-Render with cases in Table 4. Moreover, we conduct preliminary experiments with GPT-4o, which is one of the best Multimodal Large Language Models, to test whether it can perform IIMT.

The input of Case #1 contains German texts with Arial font, and Case #2 contains German texts with Times New Roman font. For both Case #1 and Case #2, the output images of our DebackX can keep the same backgrounds as the source images, and the fonts of the text in the output images match the fonts in the input images.

The TIT-Render removes the text area in the source image, leading to the degradation of the visual effect severely. Since the Render procedure typically uses a fixed font for text rendering, it cannot ensure font consistency in Case #1.

In our preliminary experiments, GPT-4o cannot complete the IIMT task directly. As a result, we decompose the IIMT task and design the following prompt for GPT-4o. *"Translate the German text into English, erasing the text in this image and rendering the translated text into the processed image at the corresponding position."*

GPT-4o generates the images based on its "Analysis", and the detailed outputs of GPT-4o are shown in Appendix E. For the outputs of GPT-4o, the issue with Case #1 is that the translated text is not rendered in the corresponding position, occupying too much area. The problem of Case #2 is that the rendered text extends beyond the image boundary. Moreover, the fonts in the output images are not consistent with the input images. Although GPT-4o can generate meaningful translated text, it fails to comprehend the layout and font of the source image, resulting in poor visual quality in the output image.

8 Conclusion

In this paper, we propose the complex scenario IIMT task where the backgrounds are sourced from real-world images. To support the proposed task, we build a new IIMT dataset named IIMT30k, which comprises images with backgrounds derived

from real-world. We introduce the DebackX model, specifically designed to address the challenge of complex scenario IIMT. Experimental results on the IIMT30k dataset demonstrate that our model outperforms previous IIMT models in both translation quality and visual effect.

Limitations

While our DebackX model achieves a better performance on the IIMT task, this work has certain limitations.

Firstly, for the sub-modules of our model, such as the vector quantization layer, the vision transformer encoder, and the decoder, we only conduct experiments using the most basic modules, lacking the investigation into other advanced modules.

Secondly, due to the multiple components in our model, which requires multi-stage training, leading to high computational resource costs.

Acknowledgment

We thank all the anonymous reviewers for their insightful and valuable comments. This work is supported by the National Natural Science Foundation of China (Grant No. U21B2009, 62406015) and Beijing Institute of Technology Science and Technology Innovation Plan (Grant No. 23CX13027).

References

Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T. Freeman. 2022. Maskgit: Masked generative image transformer. *Preprint*, arXiv:2202.04200.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. *Preprint*, arXiv:2010.11929.

Desmond Elliott, Stella Frank, Khalil Sima'an, and Lucia Specia. 2016. Multi30k: Multilingual english-german image descriptions. In *Proceedings of the 5th Workshop on Vision and Language*, pages 70–74. Association for Computational Linguistics.

Patrick Esser, Robin Rombach, and Bjorn Ommer. 2021. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12873–12883.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. 2018. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Preprint*, arXiv:1706.08500.

Hirofumi Inaguma, Sravya Popuri, Ilia Kulikov, Peng-Jen Chen, Changhan Wang, Yu-An Chung, Yun Tang, Ann Lee, Shinji Watanabe, and Juan Pino. 2023. UnitY: Two-pass direct speech-to-speech translation with discrete units. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15655–15680, Toronto, Canada. Association for Computational Linguistics.

Zhibin Lan, Liqiang Niu, Fandong Meng, Jie Zhou, Min Zhang, and Jinsong Su. 2024. Translatotron-v(ison): An end-to-end model for in-image machine translation. *Preprint*, arXiv:2407.02894.

Zhibin Lan, Jiawei Yu, Xiang Li, Wen Zhang, Jian Luan, Bin Wang, Degen Huang, and Jinsong Su. 2023. Exploring better text image translation with multimodal codebook. *Preprint*, arXiv:2305.17415.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. *Preprint*, arXiv:1711.05101.

Mario Lucic, Karol Kurach, Marcin Michalski, Sylvain Gelly, and Olivier Bousquet. 2018. Are gans created equal? a large-scale study. *Preprint*, arXiv:1711.10337.

Cong Ma, Yaping Zhang, Mei Tu, Xu Han, Linghui Wu, Yang Zhao, and Yu Zhou. 2022. Improving end-to-end text image translation from the auxiliary text translation task. *Preprint*, arXiv:2210.03887.

Cong Ma, Yaping Zhang, Mei Tu, Yang Zhao, Yu Zhou, and Chengqing Zong. 2023a. CCIM: Cross-modal cross-lingual interactive image translation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 4959–4965, Singapore. Association for Computational Linguistics.

Jian Ma, Mingjun Zhao, Chen Chen, Ruichen Wang, Di Niu, Haonan Lu, and Xiaodong Lin. 2023b. Glyphdraw: Seamlessly rendering text with intricate spatial structures in text-to-image generation. *Preprint*, arXiv:2303.17870.

Elman Mansimov, Mitchell Stern, Mia Chen, Orhan Firat, Jakob Uszkoreit, and Puneet Jain. 2020. Towards end-to-end in-image neural machine translation. In *Proceedings of the First International Workshop on Natural Language Processing Beyond Text*, pages 70–74, Online. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Zhipeng Qian, Pei Zhang, Baosong Yang, Kai Fan, Yiwei Ma, Derek F. Wong, Xiaoshuai Sun, and Rongrong Ji. 2024. AnyTrans: Translate AnyText in the image with large scale models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 2432–2444, Miami, Florida, USA. Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Juan A. Rodríguez, David Vazquez, Issam Laradji, Marco Pedersoli, and Pau Rodriguez. 2023. Ocr-vqgan: Taming text-within-image generation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 3689–3698.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. 2024. Visual autoregressive modeling: Scalable image generation via next-scale prediction. *Preprint*, arXiv:2404.02905.

Yanzhi Tian, Xiang Li, Zeming Liu, Yuhang Guo, and Bin Wang. 2023. In-image neural machine translation with segmented pixel sequence-to-sequence model. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 15046–15057, Singapore. Association for Computational Linguistics.

Yuxiang Tuo, Wangmeng Xiang, Jun-Yan He, Yifeng Geng, and Xuansong Xie. 2024. Anytext: Multilingual visual text generation and editing. *Preprint*, arXiv:2311.03054.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. Attention is all you need. *Preprint*, arXiv:1706.03762.

Jiahui Yu, Xin Li, Jing Yu Koh, Han Zhang, Ruoming Pang, James Qin, Alexander Ku, Yuanzhong Xu, Jason Baldridge, and Yonghui Wu. 2022. Vector-quantized image modeling with improved VQGAN. In *International Conference on Learning Representations*.

Lingjun Zhang, Xinyuan Chen, Yaohui Wang, Yue Lu, and Yu Qiao. 2023. Brush your text: Synthesize any scene text on images via diffusion model. *Preprint*, arXiv:2312.12232.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. 2018. The unreasonable effectiveness of deep features as a perceptual metric. *Preprint*, arXiv:1801.03924.

Shaolin Zhu, Shangjie Li, Yikun Lei, and Deyi Xiong. 2023. PEIT: Bridging the modality gap with pre-trained models for end-to-end image translation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13433–13447, Toronto, Canada. Association for Computational Linguistics.

Adrian Łańcucki, Jan Chorowski, Guillaume Sanchez, Ricard Marxer, Nanxin Chen, Hans J. G. A. Dolfing, Sameer Khurana, Tanel Alumäe, and Antoine Laurent. 2020. Robust training of vector quantized bottleneck models. *Preprint*, arXiv:2005.08520.

A Details of DebackX

Our implementation for all ViT encoders and decoders refers to timm[10]. We use learnable position encoding, and the patch embedding of the encoder uses a convolutional layer with a kernel size and stride equal to the patch size, while the output layer of the decoder uses a transposed convolutional layer with a kernel size and stride equal to the patch size. All ViT encoders and decoders have patch_size=16, d_model=512, l=8, head=8, and d_ff=2,048.

We implement the codebook based on vector-quantize-pytorch[11], with a size of 8,192 and a dimension of 32. Therefore, two linear layers are placed both before and after the codebook to match the dimensions of the ViT features. This approach, known as Factorized Codes (Yu et al., 2022), can improve the utilization of the codebook.

We use different model sizes for the Pivot Decoder and Code Decoder in the Image Translation model, since the auxiliary TIT task is easier than the task of code-to-code transformation. To prevent the auxiliary TIT task is overfitting while the other task is still underfitting, we use a smaller model size for Code Encoder and Pivot Decoder and a larger model size for Code Decoder. Both German and English texts for TIT auxiliary task are applied 10,000 BPE (Sennrich et al., 2016) to build a joint text vocabulary.

The detailed hyperparameters of experiments are shown in Table 6.

[10] https://github.com/huggingface/pytorch-image-models

[11] https://github.com/lucidrains/vector-quantize-pytorch

#	Systems	Code Encoder & Pivot Decoder				Code Decoder			
		d_model	l	head	d_ff	d_model	l	head	d_ff
1	DebackX (Table 1)	256	3	4	1,024	1,024	6	16	4,096
2	IIMT30k-TNR (Table 2)	256	3	4	512	512	6	8	2,048
3	+IWSLT PT (Table 2)	256	3	4	1,024	1,024	6	16	4,096
4	+WMT14 PT (Table 2)	512	6	8	2,048	1,024	6	16	4,096
5	TNR (Table 5)	256	3	4	512	512	6	8	2,048
6	+Arial (Table 5)	256	3	4	1,024	1,024	6	16	4,096
7	+Arial Calibri (Table 5)	256	3	4	1,024	1,024	6	16	4,096

Table 6: Detailed hyperparameters for Image Translation model of our experiments.

B Training Details

We use the Hugging Face Accelerate framework [12] with fp16 mixed precision to train our model. All parts of our models are trained with AdamW optimizer (Loshchilov and Hutter, 2019), inverse square root learning rate schedule, and dropout rate of 0.3.

Text-Image Background Separation. Two ViT encoders used to encode the source image and two ViT decoders used to decode the background image and text-image are trained for 25K steps.

Image Translation. In **stage 1**, a ViT encoder is used to encode the text-image before the codebook, a ViT decoder is used to decode the text-image after the codebook, and the vectors containing in the codebook are trained for 50K steps together. In **stage 2**, the additional text-images are used to pre-train the model for 100K steps, and the IIMT30k training set is used to fine-tune the model for 50K steps.

Text-Image Background Fusion. Two ViT encoders are used to encode the background image and text-image, and a ViT decoder fusing two encoder features to decode the target image is trained for 15K steps.

C Samples from IIMT30k Dataset

The sizes of subsets in our dataset are shown in Table 7. The reason for the size difference in IIMT30k-Arial is that the Arial font occupies more space in the images, and we remove the images that do not fully render the text to ensure the integrity of the text in the images.

We sample some images with multiple fonts from the IIMT30k dataset, as shown in Figure 3.

[12] https://github.com/huggingface/accelerate

Subsets	Training	Valid	Test
IIMT30k-TNR	25,205	864	2,740
IIMT30k-Arial	25,187	864	2,739
IIMT30k-Calibri	25,205	864	2,740

Table 7: Statistic of IIMT30k dataset.

D Impact of OCR Errors on Results Evaluation

To investigate the impact of OCR errors on the evaluation of the results, we calculate the BLEU score and Word Error Rate (WER)[13] for the OCR recognized results of the golden output images with English (En) and German (De) texts. If there were no errors in the OCR-recognized results, the BLEU score should be 100, and the WER should be 0.

The results are shown in Table 8, and we test on both text-images and images with background (real images). The results show that even with the golden output images, the errors in OCR-recognized results caused a certain decrease in the BLEU score.

		Text-Image		Real Image	
		En	De	En	De
BLEU	Valid	74.1	80.9	64.6	69.2
	Test	76.0	81.0	67.5	69.2
WER	Valid	0.25	0.19	0.32	0.30
	Test	0.23	0.18	0.27	0.26

Table 8: BLEU score and WER of OCR recognized results for golden output images.

The results in Table 8 also indicate that the OCR model has a lower error rate in recognizing text-images, suggesting that our Debackground method, which separates the image into text-image and background, is not only effective for IIMT but also has

[13] Calculate by the jiwer library.

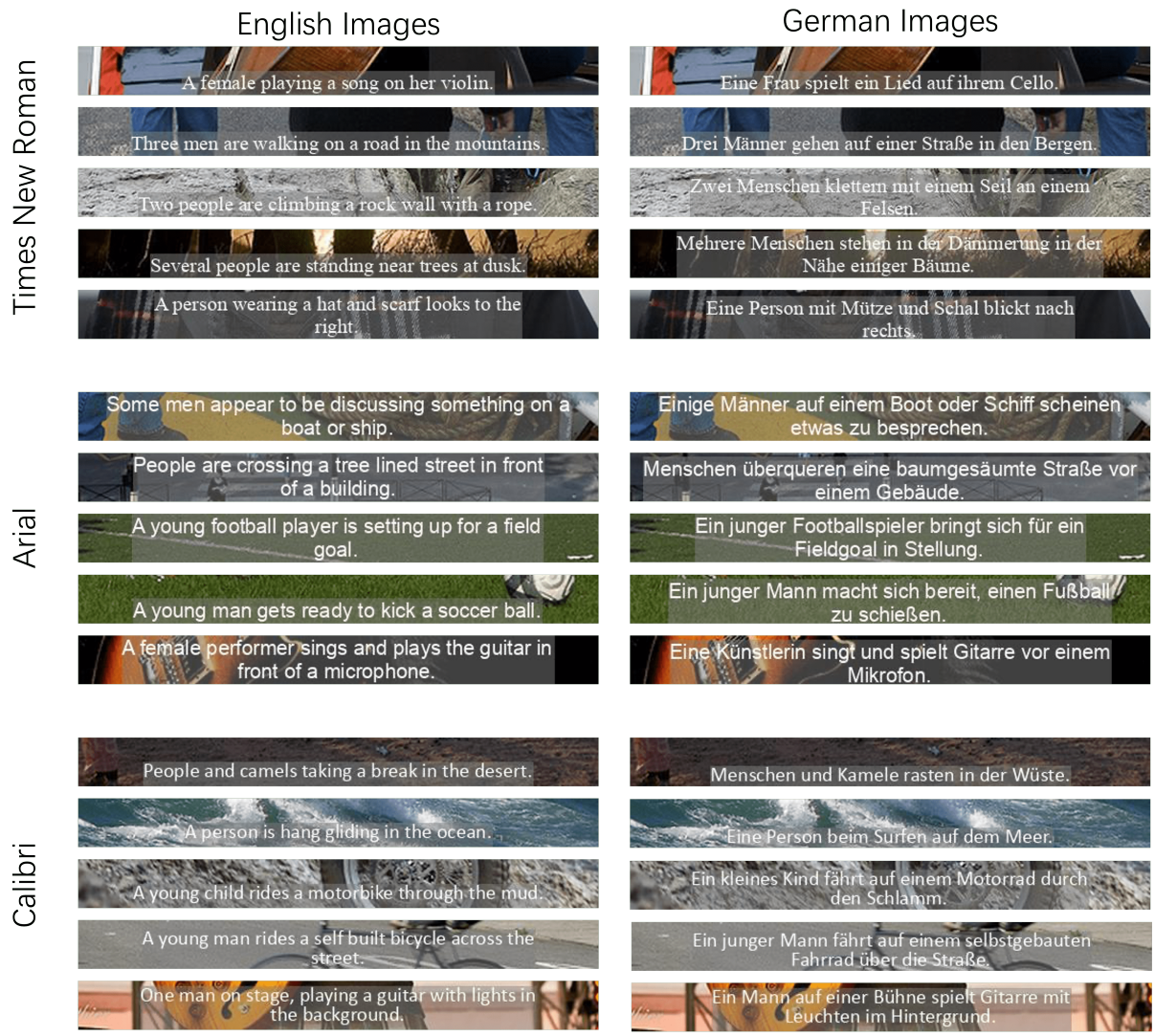


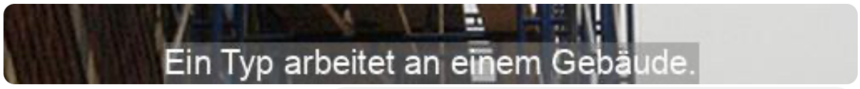
Figure 3: Samples from the IIMT30k dataset with multiple fonts (Times New Roman, Arial, and Calibri).

potential reference value for other tasks, such as OCR.

E Detailed Outputs of GPT-4o

The detailed outputs of GPT-4o that include Python code analysis of the case study are shown in Figure 4 and Figure 5. The prompt for GPT-4o is *"Translate the German text into English, erasing the text in this image, and render the translated text into the processed image at the corresponding position."*

Case #1



Translate the German text into English, erasing the text in this image and render the translated text into the processed image at corresponding position.

I have processed the image with the translated text. You can download the updated image using the link below:

Download the processed image

Analysis Python code generated by GPT-4o:

```
from PIL import Image, ImageDraw, ImageFont

# Load the image
image_path = "/mnt/data/src-1.jpg"
image = Image.open(image_path)

# Translate the German text into English
translated_text = "A guy is working on a building."

# Define font size and position (Approximated as we don't have the exact font
details)
font_size = 30
font = ImageFont.truetype("/usr/share/fonts/truetype/dejavu/DejaVuSans-
Bold.ttf", font_size)
draw = ImageDraw.Draw(image)

# Define position for text (Approximated position similar to original image)
text_position = (10, image.height - 40)

# Draw rectangle over the existing text
text_width, text_height = draw.textsize(translated_text, font=font)
draw.rectangle([text_position, (text_position[0] + text_width, text_position[1]
+ text_height)], fill="white")

# Add translated text to image
draw.text(text_position, translated_text, fill="black", font=font)

# Save the processed image
processed_image_path = "/mnt/data/processed_image.jpg"
image.save(processed_image_path)
processed_image_path
```

Output image:

Figure 4: Detailed GPT-4o output of Case #1.

Case #2

Translate the German text into English, erasing the text in this image and render the translated text into the processed image at corresponding position.

The image has been processed with the translated text. You can download it using the link below:

Translated Image

Analysis Python code generated by GPT-4o:

```python
from PIL import Image, ImageDraw, ImageFont
import matplotlib.font_manager as fm

# Load the image
image_path = "/mnt/data/src-2.jpg"
image = Image.open(image_path)

# Translation
translated_text = "A mother and her little son enjoy a beautiful day outdoors."

# Determine position to draw the text
draw = ImageDraw.Draw(image)
width, height = image.size
text_position = (10, height - 30)  # Adjust the position as needed

# Load a font
# Using a basic font as we don't have access to system fonts in this environment
font_path = fm.findSystemFonts(fontpaths=None, fontext='ttf')[0]
font = ImageFont.truetype(font_path, 20)

# Erase the original text area (assuming it is a rectangle, white background)
draw.rectangle(((0, height - 40), (width, height)), fill="white")

# Draw the translated text
draw.text(text_position, translated_text, font=font, fill="black")

# Save the processed image
processed_image_path = "/mnt/data/translated_image.jpg"
image.save(processed_image_path)

processed_image_path
```

Output image:

Figure 5: Detailed GPT-4o output of Case #2.