Image Translation Network

Puneet Jain and **Orhan Firat** and **Qi Ge**
Google Research
1600 Amphitheater Pkwy
Mountain View, CA 94043, USA
jpuneet, orhanf, qge@google.com

Sihang Liang*
Department of Physics
Princeton University
Princeton, NJ 08544, USA
sihangl@princeton.edu

Abstract

We present an end-to-end neural network to translate images containing text from one language to another. Traditionally, a cascaded approach of optical character recognition (OCR) followed by neural machine translation (NMT) is used to solve this problem. However, the cascaded approach compounds OCR and NMT errors, and incurs longer latency, performs poorly in multiline cases. Our simplified approach combines OCR and NMT into one end-to-end model. Our neural architecture follows the encoder-decoder paradigm, with a convolutional encoder and an autoregressive Transformer decoder. Trained end-to-end, our proposed model yields significant improvements on multiple dimensions, (i) achieves higher translation accuracy due to better error propagation, (ii) incurs lower inference latency due to smaller network size, and (iii) translates multiline paragraphs and understands reading order of the lines, (iv) eliminates source side vocabulary. We train several variations of encoders and decoders on a synthetic corpus of 120M+ English-French images and show that our approach outperforms the cascaded approach with a large margin in both the automatic metrics and the detailed side-by-side human evaluation.

1 Introduction

Instant image translation refers to the problem of taking an image containing text in a source language, translating the text to a target language, and replacing the image with that text in real-time. The commercial implementations of such a feature are Google Translate's Instant Camera mode (Goo, b,c) and Google Lens (Goo, a), both of which seamlessly replace text in the source language with a translation in the target language. The camera translation feature shown in Figure 1 uses a cascade of a text detector network, a text recognizer network, and a NMT network. The text detector network is a region proposal network (RPN) which uses variants of Faster-RCNN (Ren et al., 2015) and SSD

*The work was performed during author's internship at Google.

Figure 1: Image Translation Overview

(Liu et al., 2016) to find character, word, and line bounding boxes. The text recognizer network is a convolution neural network (CNN) with an additional quantized long short-term memory (LSTM) network trained with CTC loss (Graves et al., 2006) to identify text inside the bounding box. Finally, NMT is a sequence-to-sequence network which uses variants of LSTM and Transformers (Vaswani et al., 2017; Goo, d) to translate the identified text. Figure 2 demonstrates the entire flow.

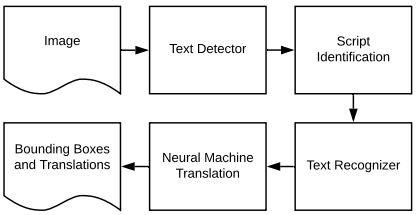

Figure 2: Image Translation Cascade Approach

The cascaded approach works fairly well on short sentences but it suffers in multiline cases. The primary reason for poor accuracy being the lack of understanding of document structure. While in a well formatted document such as newspapers or books, the document structure is relatively easy to identify, but it is much harder in real world scenes. The document structure requires learning the order in which disparate OCR lines or words should be concatenated to form a logical sentence or paragraph. For instance, imagine a case of a restaurant menu where prices of the dishes are listed next to multiline dish names and ingredients. In this case, multiline ingredients should be translated together isolated from the price. The current generation OCR engines lack this understanding, therefore translation is either performed at a line level or heuristics are used to form sentences – often resulting in a bad translation

result. Another issue with the cascaded approach is error propagation. A common error in OCR engines such as “I" or “L" being interpreted as “1" could completely change the meaning of a translation. While some of this noise can be smoothed by NMT using OCR text as the source during training, exhaustive error correction is still hard.

In this paper, we propose to fuse several of the cascaded models into a single image translation network, referred as “ItNet". ItNet shown in Figure 3 is a sequence-to-sequence model which follows the encoder-decoder paradigm of NMT (Sutskever et al., 2014). But contrary to NMT, which operates on the encoding of source sentence, ItNet operates on the output feature map generated by the CNN on input pixels. ItNet overcomes the limitations of a cascaded approach and provides additional benefits on top. Working directly with pixels also overcomes issues related to vocabularies, segmentation, and tokenization on the source side. It achieves much higher accuracy with a smaller network size, thereby providing speedup at inference time, and it takes less compute. Finally, ItNet could act as a stepping stone to build an image-to-image translation model in the future—directly emitting the translated image as the output, avoiding the complex rendering required to overlay translation on top of the source image.

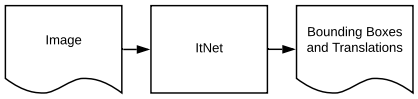

Figure 3: Image Translation Fused Approach

2 Image Translation Model

2.1 Neural Machine Translation Background

We briefly explain the encoder-decoder paradigm for NMT. Encoder-decoder paradigm consists of a sequence encoder f_{enc} that takes an input sequence of tokens $\mathbf{x} = (x_1, \ldots, x_n)$ of length n and produces a sequence of hidden states $\mathbf{h} = (h_1, \ldots, h_n)$, formally $\mathbf{h} = f_{enc}(\mathbf{x})$. Once the sequence of hidden states are generated, the decoder f_{dec} generates the output sequence $\mathbf{y} = (y_1, \ldots, y_m)$ of length m, one token at a time in an autoregressive fashion given the source sequence hidden states $\mathbf{h}$ and previously generated (outputs) prefix $y_{<j}$[1], formally $f_{dec}(y_{<j}|\mathbf{h})$. Putting it all together, the encoder-decoder model generates an output sequence $\mathbf{y}$ by modelling the conditional distribution $P(\mathbf{y}|\mathbf{x})$ given an input sequence $\mathbf{x}$ with the following factorization $P(\mathbf{y}|\mathbf{x}) = \prod_{j=1}^{m} P(y_j|y_{<y}, \mathbf{h})$.

Once an output sequence is generated, which is determined by emitting of a special end-of-sequence token, the generated hypothesis output sequence is $\mathbf{y}$ is compared against a reference output sequence in order to assess the translation quality. A commonly used automatic metric, called BLEU (Papineni et al., 2002), measures the n-gram precision of the hypothesis weighted by the brevity penalty in order to penalize short translations.[2]

2.2 Network Design

Vanilla encoder-decoder architectures for translation utilize sequence encoders/decoders in order to parameterize f_{enc} and f_{dec}. The most commonly used parametrization of these functions follow a either Transformer (Vaswani et al., 2017), or an LSTM network (Sutskever et al., 2014). In our proposed ItNet , we simply replace the f_{enc} function to be a convolutional neural network in order to input an image, rather than a sequence of tokens, and keep the decoder function f_{dec} intact, namely keep using a Transformer decoder (self-attention, cross-attention cascade).

2.3 Dataset

To the best of our knowledge, there is no public dataset available for image translation research. Therefore, image synthesis remains the lowest cost option at this time. We train ItNet on a corpus of 119.5M synthetic images. These images are constructed from an English-French corpora of parallel sentences used to train NMT models. The training set contains about 112.4M long sentences ranging from 10 to 50 words (130 tokens), 7M short sentences ranging from 3 to 10 words (10.3 tokens), and 115K single word sentences (1.4 tokens). We use an image synthesis tool which generates one image per sentence pair at 640x480 resolution. The synthesis tool generates images based on a configuration. A different configuration is generated for each sentence pair during the synthesis process by selecting a random font from a list of pre-installed fonts, a random font size from a range of 20 pixels to 50 pixel, a random font style, and various other parameters such as blur, random noise etc. The final result of the synthesis process is fairly random, given the multitude of parameters. Figure 4 shows one such example from our tool. As evident, images only render the source sentence and target sentences are used as ground truth. To convert target sentences into embeddings, a target side vocabulary is constructed from the data set. The ItNet decoder is initialized from a respective text-to-text model decoder after training the NMT model on the same parallel corpora of English-French sentences. Finally, a similar process is followed to generate ItNet test sets from corresponding NMT test sets. For ease of understanding, this paper reports the result on the images generated from the publicly available WMT 2013 test set only.

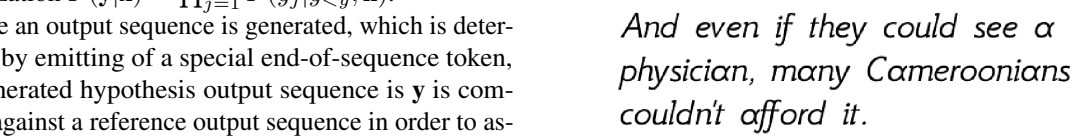

Figure 4: Example image from WMT testset

[1]Here $y_{<j}$ indicates the tokens generated previous to the target sequence position at j, where j is in range 1 to m.

[2]Briefly, BLEU= $bp \cdot \exp(1/4 \sum_{n=1}^{4} \log p_n)$, where bp is the brevity penalty, and p_n are the n-gram precision.

2.4 Training

We use Lingvo (Shen et al., 2019) to implement the base text-to-text NMT model and ItNet. We train ItNet in two steps. We first train a text-to-text transformer model (referred to as the base model). In our base model both encoder and decoder contain 6 layers each. The number of heads in multi-headed attention is set to 16 and model dimension is set to 1024. Our vocabulary contains 16000 word pieces. The hidden state dimensions are set to 2048. We use an Adam schedule with a learning rate of 1.0, a dropout probability of 0.1, and warm up steps of 50000. We then train our base model for 1M steps. The base model training takes about 3 days to complete on our platform.

In the second step, we pick the latest checkpoint from the base model and initialize ItNet training with it. We use an identical decoder as the base model and therefore only the ItNet decoder gets initialized this way. Our encoder is a ResNet-101 model and we use Xavier initialization for it. We use transformer learning rate schedule with a learning rate of 0.1 and 80000 warm up steps throughout this work. We use a batch size of 8 for training and 64 for evaluation. It is important to note that the output of the encoder in the format *Batch* $\times$ *Height* $\times$ *Width* $\times$ *Num Filters* needs to be reshaped to match the input of an NMT decoder in *Time* $\times$ *Batch* $\times$ *Model Dimension* format. This is to make sure that the number of filters in the encoder maps to the model dimension of the decoder and the 2D activation's of the encoder maps to the time dimension of the decoder. Therefore, we apply a reshape operation to each 2D feature map to convert them to a 1D vector of size ($\prod$ *Height.Width*). This creates the *Time* dimension for the decoder. We subsequently apply a transpose to switch *Batch* and *Vector* dimensions to match the ordering expected by the decoder. We train ItNet for 1M steps.

3 Experiments and Analysis

We present the experiment results ItNet on WMT 2013 test set. Methodology to render WMT test set on images is described in Section 2.3. Due to randomness during image synthesis, not all sentences in the test set can be rendered into an image. We fix font size to be between 20 and 30. Our final synthesized test set contains 1414 images.

3.1 Side-by-Side Human Evaluation

To perform side by side evaluation, we create a template containing three images and seven possible ratings with zero being a "Non-Sense" and six being "Perfect". We provide them with additional details on what does each rating imply, such as a "Non-Sense" imply "nearly all information is lost between the translated image and original image" while "Perfect" implies "the meaning of the translation is completely consistent with the original image and the grammar is correct".

The images in the template are ordered in a grid as follows: source language image, translation from the first system followed translation from the second system. We provide no system names or implementation details in the template to the raters. Raters at a time see only one task and we ask their rating on two systems (named "sys0" and "sys1") based on the displayed output. We rate each image thrice to reduce rater biases. A total of 1414 images in our test set results into 4242 tasks for a pool of 120 raters. A rater never rates the same image twice. Additional constraints such as time limit to complete a task and a maximum number of tasks per rater are imposed. All raters are professional linguists with expertise in English as well French language.

We average the three ratings to construct final rating of the image. We further divide results into cases where ItNet performs better than the baseline, performs equal to the baseline, and performs worse than the baseline. If a system performs better than or equal to the baseline in majority of cases, it is considered to be significant from product launch standpoint. To be precise, if the average rating of the new system exceeds the baseline by 0.1 rating point, the new system is typically considered to be ready to replace the baseline in production system.

3.1.1 Baseline System

We compare ItNet with a cascaded system consisting of client side OCR and server side NMT. Client side OCR model at a high-level consists of convolution encoder, LSTM decoder, and CTC loss. The server side NMT consists of transformer/LSTM encoder and decoder with additional pre and post processing steps. Both of these models are meticulously trained on a plethora of data and highly optimized to achieve best accuracy in their class. The client side OCR model performs ondevice recognition and sends recognized text lines to a server running NMT inference. Note that these text lines often contain only a part of the incomplete sentence. Therefore, NMT model on the server side translates only part-sentences. These returned translations are in-place rendered back on the image. These rendered images are then shown to the raters for side-by-side ratings against ItNet .

In the aforementioned S$\times$S evaluation, ItNet outperforms the baseline in 46.4% cases while gets an equal rating in 37.6% cases. In minority 16% cases, ItNet performs inferior to the baseline.

3.1.2 Multiline Variations

One of the key contribution of ItNet is it's ability to perform multiline translations. Figure 5 plots rating difference between ItNet and the baseline. Each bar represents average of the rating difference grouped by the number of lines in the images.

ItNet consistently outperforms the baseline, generally showing higher gains with the increasing number of lines. Importantly, ItNet outperforms the baseline in case of single line cases as well. This is interesting since it is indicating that not all gains in ItNet can be attributed to multiline handling. This implies that ItNet is alleviating some of the information loss happening due to cascaded approach and it is building an ability to correct OCR errors during translation. Note that there

Label	Model Dims	Layers	Heads
Shallow-Thin	512	2	8
Deep-Thin	512	6	8
Shallow-Wide	1024	2	16
Deep-Wide	1024	6	16

Table 1: Decoder labels to parameter mapping

were only handful of images with number of lines > 13 in the dataset, attributing to spikes in the chart.

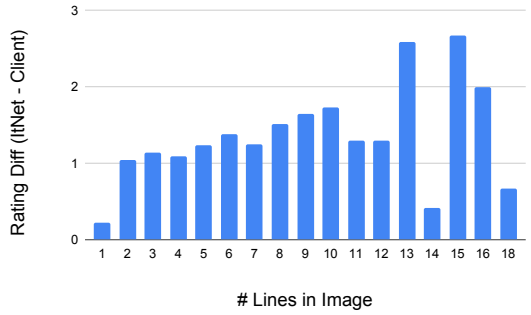

Figure 5: Avg rating difference between ItNet and the baseline, grouped by the number of lines in the image

3.1.3 Encoder and Decoder Size Variations

We study how variations in encoder and decoder size affects model accuracy. As mentioned before, accuracy of a NMT model is measured by bleu score. We report results for five different encoders (ResNet and DarkNet) and four different decoders (variants of transformer). Encoders used here have been well established in the computer vision literature. Table 1 maps transformer decoder labels to their parameters. All other parameters are kept the same across different models.

Figure 6 reports the total number of network parameters in each combination of encoder-decoder. ResNet-101 and DarkNet-53 are fairly similar models, therefore so are the total number of parameters in them. ResNet-101 with transformer deep-wide contains about 151M parameters. Figure 7 shows bleu scores achieved by different models on WMT 2013 test set. The missing bars in ResNet-18 are due to lack of convergence of those combinations on our data. We could not train any stable model on them due to their smaller capacity. As expected, larger capacity lends to higher bleu score at a high level. But for a given encoder, Deep-Thin decoders outperform Shallow-Wide which is smaller in its size than their counterpart. We notice that increasing encoder size for ResNet-18 to ResNet-50 provides large jumps in gains but the same subsides between ResNet-50 and ResNet-101. ResNet-101 with Deep-Wide decoder outperforms all other scheme, however it is also the largest model we trained. Importantly, DarkNet-53 which is of similar size as that of ResNet-101 performs inferior to it. It performs more closely to ResNet-50 which is a much smaller model. All S×S scores reported above were on DarkNet-53 and Deep-Wide combination – a model we training during initial investigations. We could not redo S×S on a better combination of ResNet-101 and Deep-Wide due to budget issues.

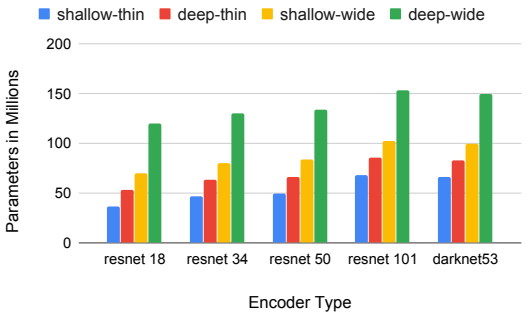

Figure 6: Parameters (millions) in ItNet variants

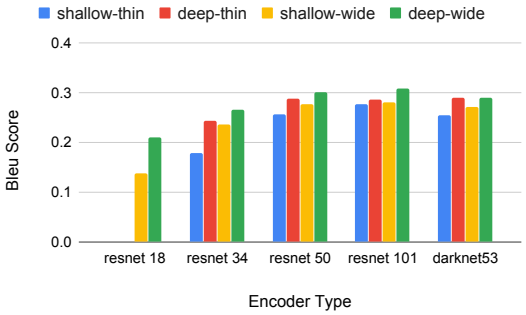

Figure 7: Bleu score of ItNet variants

4 Related Work

Text preprocessing and vocabulary construction has been an active research area leading to work on NMT systems operating on subword units (Sennrich et al., 2015), characters (Lee et al., 2017) and bytes (Wang et al., 2020) and has been highlighted to be one of the major challenges when dealing with many languages simultaneously in multilingual NMT (Arivazhagan et al., 2019), and cross-lingual natural language understanding (Conneau et al., 2019).

Multimodal MT is an application of NMT which helps computers to understand visual objects and their relations with natural languages. Image GPT (Chen et al., 2020; Ramesh et al., 2021) fuses boundaries between the two areas further by shows that a transformer model trained on pixel sequences can generate coherent image completions and samples similar to a transformer model trained on text. Some of the problems in this space are translating source sentences that describe an image into target language or directly describing an image in target language other than English (Elliott et al., 2016; Elliott et al., 2017). (Liu et al., 2017) shows translation results on various multimodal tasks such as street scene image translation, animal image translation, and face image translation. (Mansimov et al., 2020) attempt to render translations back to the source image - an extension of this work which could enable true end-to-end image translation. (Caglayan et al., 2016; Huang et al., 2016; Su et al., 2019) show that providing visual cues to encoder can improve text only translation accuracy. Finally, image transformer (Parmar et al., 2018) generalizes architecture to image generation problem.

References

a. *Giving Lens New Reading Capabilities in Google Go*. https://ai.googleblog.com/2019/09/giving-lens-new-reading-capabilities-in.html.

b. *Google Translate's instant camera translation gets an upgrade*. https://blog.google/products/translate/google-translates-instant-camera-translation-gets-upgrade/.

c. *How Google Translate squeezes deep learning onto a phone*. https://ai.googleblog.com/2015/07/how-google-translate-squeezes-deep.html.

d. *Recent Advances in Google Translate*. https://ai.googleblog.com/2020/06/recent-advances-in-google-translate.html.

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George Foster, Colin Cherry, et al. 2019. Massively multilingual neural machine translation in the wild: Findings and challenges. *arXiv preprint arXiv:1907.05019*.

Ozan Caglayan, Walid Aransa, Yaxing Wang, Marc Masana, Mercedes García-Martínez, Fethi Bougares, Loïc Barrault, and Joost Van de Weijer. 2016. Does multimodality help human and machine for translation and image captioning? In *ACL*.

Mark Chen, Alec Radford, Rewon Child, Jeff Wu, Heewoo Jun, Prafulla Dhariwal, David Luan, and Ilya Sutskever. 2020. Generative pretraining from pixels.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

D. Elliott, S. Frank, K. Sima'an, and L. Specia. 2016. Multi30k: Multilingual english-german image descriptions. In *Proceedings of the 5th Workshop on Vision and Language*.

Desmond Elliott, Stella Frank, Loïc Barrault, Fethi Bougares, and Lucia Specia. 2017. Findings of the Second Shared Task on Multimodal Machine Translation and Multilingual Image Description. In *SIGMT*.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *ICML*.

Po-Yao Huang, Frederick Liu, Sz-Rung Shiang, Jean Oh, and Chris Dyer. 2016. Attention-based multimodal neural machine translation. In *SIGMT*.

Jason Lee, Kyunghyun Cho, and Thomas Hofmann. 2017. Fully character-level neural machine translation without explicit segmentation. *TACL*, 5:365–378.

Ming-Yu Liu, Thomas Breuel, and Jan Kautz. 2017. Unsupervised image-to-image translation networks. In *NeurIPS*.

Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. 2016. Ssd: Single shot multibox detector. In *ECCV*.

Elman Mansimov, Mitchell Stern, Mia Chen, Orhan Firat, Jakob Uszkoreit, and Puneet Jain. 2020. Towards end-to-end in-image neural machine translation.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*.

Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Łukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. 2018. Image transformer. In *ICML*.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. 2021. Zero-shot text-to-image generation.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Jonathan Shen, Patrick Nguyen, Yonghui Wu, Zhifeng Chen, et al. 2019. Lingvo: a modular and scalable framework for sequence-to-sequence modeling.

Yuanhang Su, Kai Fan, Nguyen Bach, C-C Jay Kuo, and Fei Huang. 2019. Unsupervised multi-modal neural machine translation. In *CVPR*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *NeurIPS*, pages 3104–3112. Curran Associates, Inc.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *NeurIPS*.

Changhan Wang, Kyunghyun Cho, and Jiatao Gu. 2020. Neural machine translation with byte-level subwords. In *AAAI*.