Seeing Words Differently: Visual Embeddings for Robust English-Arabic Machine Translation

Mahdi Alshaikh Saleh[1] **Irfan Ahmad**[1,2]
[1]ICS Department, KFUPM, Dhahran 31261, Saudi Arabia
[2]SDAIA–KFUPM Joint Research Center for Artificial Intelligence
KFUPM, Dhahran 31261, Saudi Arabia
g201471540@kfupm.edu.sa, irfan.ahmad@kfupm.edu.sa

Abstract

Context: Natural Language Processing (NLP) has become an essential field with widespread applications across domains such as Large Language Models (LLMs). One of the core applications of NLP is machine translation (MT). A major challenge in MT is handling out-of-vocabulary (OOV) words and spelling mistakes, which can lead to poor translation quality. Objective: This study compares traditional text-based embeddings with visual embeddings for English-to-Arabic translation. It investigates the effectiveness of each approach, especially in handling noisy inputs or OOV terms. Method: Using the IWSLT 2017 English-Arabic dataset, we trained a baseline transformer encoder-decoder model using standard text embeddings and compared it with models using several visual embeddings strategies, including vowel-removal preprocessing and trigram-based image rendering. The translated outputs were evaluated using BLEU scores. Results: show that although traditional BPE-based models achieve higher BLEU on clean data, visual embedding models are substantially more robust to spelling noise, retaining up to 2.4× higher BLEU scores at 50% character corruption.

1 Introduction

Natural Language Processing (NLP) focuses on interactions between computers and human languages and has progressed significantly due to advances in deep learning techniques (Vaswani et al., 2017). Machine translation (MT), a central NLP task, transforms source language text into grammatically and semantically correct target language equivalents. Modern MT systems typically employ transformer-based (Vaswani et al., 2017) encoder-decoder architectures, relying heavily on dense word embeddings such as Word2Vec (Mikolov et al., 2013) or subword approaches like Byte-Pair Encoding (BPE) (Sennrich et al., 2016). Despite their effectiveness, traditional embeddings struggle with out-of-vocabulary (OOV) words and noisy text inputs, common in practical applications (Belinkov and Bisk, 2018).

Recent research proposes visual embeddings to address these limitations by converting text into image representations, leveraging visual similarity to mitigate OOV problems (Salesky et al., 2021). Instead of relying purely on textual form, these embeddings treat words or phrases as images, capturing their visual structure. This approach is particularly promising when models encounter new words or noisy inputs, as the shape of the word may still resemble known patterns. Arabic, due to its morphological richness, presents challenges in MT, motivating the exploration of machine translation to Arabic using alternative embedding methods. Rather than optimizing solely for peak performance on clean benchmark data, this work focuses on translation robustness under realistic noisy conditions, where spelling errors and out-of-vocabulary words frequently degrade subword-based models.

This paper investigates and compares the effectiveness of visual versus traditional embeddings in translating English into Arabic under clean and noisy conditions. Specific techniques explored include vowel removal preprocessing (Al-Shaibani and Ahmad, 2023) and character-level trigram tokenization rendered visually. Character trigram representations have been widely used in English NLP to improve robustness to spelling variation and to mitigate vocabulary expansion (Huang et al., 2013; Nigam et al., 2019). More recently, character trigrams have also been shown to provide strong robustness to misspelled words in Arabic text classification (Alomari and Ahmad, 2024), motivating further investigation of trigram-based representations in cross-lingual and multimodal settings. The main contributions of this paper are as follows:

- We demonstrate that visual embeddings improve robustness to spelling noise in English-

Proceedings of the 2nd Workshop on NLP for Languages Using Arabic Script (AbjadNLP 2026), pages 66–74

Arabic machine translation, maintaining substantially higher BLEU scores than BPE-based models under noise levels from 10% to 50%, despite lower performance on clean data.

- We propose a character-trigram visual tokenization strategy that reduces input sequence length by more than 50% (from ~74 to ~32 visual tokens per sentence), leading to significantly faster training while preserving robustness.

- We provide a controlled robustness evaluation across multiple noise levels, showing that visual models degrade more gracefully than text-based subword models as input corruption increases.

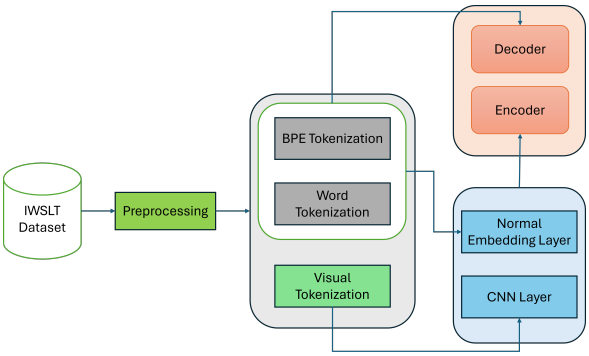

Figure 1: Overview of the end-to-end methodology. After preprocessing, source text is tokenized using either text-based or visual approaches. Visual tokens are rendered as images and encoded via a CNN, while text tokens use standard embeddings, before being passed to a transformer encoder-decoder model.

2 Related Work

Recent studies have explored replacing traditional text-based embeddings with visual text representations for various NLP tasks, particularly to improve robustness to noise and out-of-vocabulary (OOV) words.

Salesky et al. (Salesky et al., 2021) proposed representing text as images using sliding windows and applying this approach to machine translation. Their method demonstrated strong robustness to noisy inputs across multiple datasets and language pairs, showing that visual text representations can match or outperform traditional subword-based models. Building on this work, Salesky et al. (Salesky et al., 2023) extended visual embeddings to a multilingual setting, demonstrating improved performance over subword embeddings across diverse scripts and writing systems.

More generally, the PIXEL model (Rust et al., 2023) introduced a framework that converts text into images and learns representations through masked image reconstruction. PIXEL showed strong cross-script generalization, particularly for non-Latin scripts, highlighting the potential of visual representations beyond token-based text models. Subsequent work by Lotz et al. (Lotz et al., 2023) explored alternative rendering strategies for PIXEL, showing that character n-gram–based rendering can improve efficiency and sentence-level performance while reducing model size.

Character-level representations have also been widely studied as a means to improve robustness and control vocabulary growth. Prior work has shown that character n-grams are effective for handling spelling variation and OOV words in English NLP tasks. More recently, Alomari and Ahmad (Alomari and Ahmad, 2024) demonstrated that character trigrams provide strong robustness to misspelled text in Arabic classification tasks, motivating further investigation of trigram-based representations in noisy and multilingual settings.

Finally, Al-Shaibani and Ahmad (Al-Shaibani and Ahmad, 2023) proposed removing vowels from English text as a compact representation for traditional embeddings. Their approach achieved comparable performance to standard text representations across several NLP tasks, including machine translation, while significantly reducing computational requirements. This finding motivates the exploration of non-vowel preprocessing in conjunction with visual embedding methods.

3 Methodology

This section describes the techniques used to preprocess text data, tokenize input and output, generate visual embeddings, and build and train translation models. We explore both traditional and visual approaches. Figure 1 shows the methodology diagram.

3.1 Text Preprocessing

Text preprocessing is a crucial step in NLP to reduce variation and standardize the dataset. In our study, we applied language-specific preprocessing techniques to both English (source language) and Arabic (target language).

For Arabic, a preprocessing step was performed by replacing different forms of the letter “Alef”,

specifically, (”i̇“, ”i̧“ ,“ĩ”) with the standard “i”. This helped reduce redundancy in vocabulary caused by orthographic variations. In addition, the harakat (diacritics) were removed to minimize the variation in word forms.

For English, we converted all characters to lowercase to minimize case-based token duplication.

Additionally, for both Arabic and English, punctuation marks were surrounded by spaces so they would be treated as individual tokens during word-level tokenization.

One special consideration in English preprocessing was the apostrophe symbol (’), which appears frequently in contractions like “I’m” and “it’s.” Unlike other punctuation marks, apostrophes were retained to preserve the semantic meaning of such expressions in English.

We also prepared a variation of the dataset where vowels were removed from English sentences, aiming to simulate the non-vowels approach used by (Al-Shaibani and Ahmad, 2023) for visual embeddings and see its effect on the performance of machine translation.

3.2 Tokenization

Tokenization refers to dividing text into discrete units, or tokens, that can be processed by a model (Hassler and Fliedl, 2006). In this study, different tokenization strategies were employed depending on whether traditional text-based or visual embeddings were used.

Text-based tokenization. For traditional models, Byte-Pair Encoding (BPE) was applied jointly to the English and Arabic training corpora using 10,000 merge operations. This resulted in a shared subword vocabulary that helps mitigate vocabulary sparsity and improves the handling of rare and unseen words during translation.

Visual tokenization. For visual models, tokenization was performed by first rendering each English source sentence as a grayscale image and then segmenting the image into smaller visual units, each of which was treated as a token.

In the two base visual models, referred to as BASE1 and BASE2, the rendered sentence image was segmented using a sliding window with a width of 15 pixels and a stride of 10 pixels, producing overlapping image slices. Each slice represents a local visual context within the sentence and serves as an individual visual token. In the NON-VOWELS model, the same sliding-window segmentation strategy was applied after removing vowel characters from the English source text, resulting in shorter rendered sequences and fewer visual tokens per sentence.

In the trigram visual model, referred to as TRIGRAM, tokenization was performed at the character level. Each sequence of three consecutive characters from the English source sentence was rendered as a separate, non-overlapping image slice, producing fixed-width visual tokens. Character trigram representations have been shown to be effective in English NLP tasks for reducing vocabulary size and improving robustness to spelling errors and out-of-vocabulary words (Huang et al., 2013; Eyecioglu and Keller, 2016). More recent work has demonstrated similar robustness properties for character trigrams in Arabic text classification under noisy conditions (Alomari and Ahmad, 2024), suggesting that trigram-based representations capture language-agnostic structural patterns that are beneficial under noisy input conditions.

Target-side tokenization. The target language (Arabic) was tokenized using BPE in all models to ensure consistency during training and evaluation. An exception was made for BASE1, where word-level tokenization using whitespace as a delimiter was employed on the target side to enable a direct comparison with BASE2, which uses BPE-based decoding.

3.3 Embedding

We consider two embedding paradigms for representing source text: (i) conventional text-based embeddings and (ii) visual embeddings derived from image representations of text.

For text-based models, source tokens are represented using standard word embeddings augmented with sinusoidal positional encodings to preserve relative token order information.

For visual models, tokens are represented as image-based units rather than discrete symbols. Each visual token is mapped to a fixed-dimensional continuous representation through a learnable convolutional projection, producing a 512-dimensional embedding per token. These visual embeddings replace conventional word embeddings and are directly consumed by the encoder–decoder architecture, allowing the model to operate on visual structure while remaining architecturally compatible with text-based transformer models.

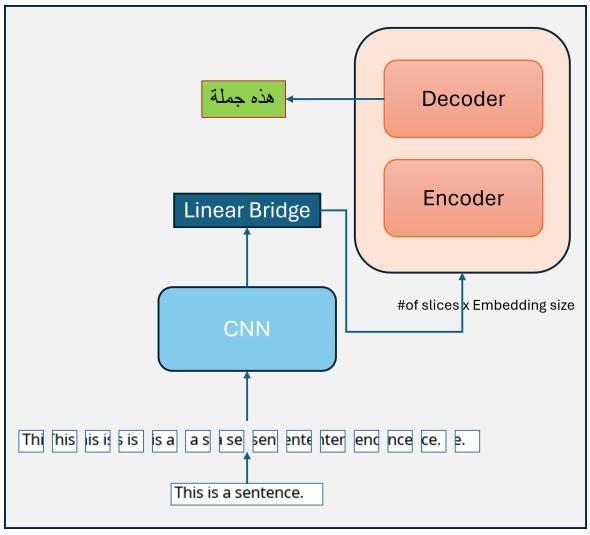

Figure 2: Example of the visual embedding pipeline. An English source sentence is rendered as image slices, which are encoded by a CNN and projected through a linear bridge to obtain fixed-dimensional visual embeddings. These embeddings are then provided as input to the transformer encoder-decoder model to generate the target Arabic sentence.

Figure 2 illustrates an example of the visual embedding process. An English source sentence is first tokenized and rendered into image slices, which are encoded using a CNN. The resulting visual features are projected through a linear layer to produce fixed-dimensional embeddings that are compatible with the transformer encoder-decoder architecture.

3.4 Encoder-Decoder Architecture

All models whether using traditional or visual embeddings shared the same transformer-based encoder-decoder architecture to ensure a fair comparison. The architecture consisted of:

- 6 transformer encoder layers and 6 transformer decoder layers
- 4 attention heads per multi-head attention module
- Two fully connected layers in each transformer block: one with size (512 $\rightarrow$ 1024), followed by another with size (1024 $\rightarrow$ 512)
- Embedding dimension: 512
- Optimizer: Adam (Kingma and Ba, 2015)

To encode token order information, sinusoidal positional encodings were added to the encoder inputs in all models. In the traditional setup, the sinusoidal positional embeddings were applied to the BPE-based text embeddings of the English source sentences. In the visual setup, the same sinusoidal positional encoding scheme was added to the sequence of 512-dimensional visual embeddings produced by the CNN over image token slices, ensuring that positional information was modeled consistently across both embedding types.

4 Experiment

4.1 Dataset

We used the IWSLT 2017 (Cettolo et al., 2017) English-Arabic parallel corpus, a widely used benchmark dataset for low-resource machine translation tasks. The dataset was accessed using the datasets library in Python, which provides the data pre-split into training, validation, and test sets enabling a consistent and reproducible evaluation. The training data was used to train all models, while the validation set was used for hyperparameter tuning and the test set for final evaluation. The training set consists of 231713 sentence pairs, while the validation and test sets include 888 and 8,583 pairs, respectively. Each pair in the dataset contains a source sentence in English and its corresponding translation in Arabic. The dataset covers a wide range of spoken language topics derived from TED talks, which makes it a useful resource for evaluating the generalization and robustness of translation systems. All sentences in the corpus were preprocessed to ensure consistency across experiments, including punctuation handling, and case standardization.

4.2 Experiment Setup

This section describes the training environment, preprocessing steps, and model-specific configurations used to conduct the experiment and do comparison between traditional and visual embedding models.

4.3 Preprocessing

All experiments used the same preprocessing pipeline described in Section 3.1 to ensure comparability across models. This included English lowercasing, Arabic normalization, punctuation separation, and diacritic removal.

For the NON-VOWEL setting, English vowels were removed prior to tokenization. No additional preprocessing steps were applied.

4.4 Visual Model Configuration

For visual models, English source sentences were rendered as grayscale images using the Pygame library with the *NotoSans-Regular* font at a size

of 10pt. Padding was applied during rendering to ensure consistent image dimensions across samples.

Tokenization was performed directly in the image space using multiple strategies:

- **BASE1, BASE2, NON-VOWELS**: sliding window slicing with a window width of 15 pixels and a stride of 10 pixels.
- **TRIGRAM**: fixed-width slicing corresponding to three consecutive characters, where the slice width was set to three times the maximum character width to ensure uniform trigram representation.

The image height was determined by the maximum height of the rendered sentence plus padding:

- BASE1, BASE2, and NON-VOWELS: 22 pixels (16 pixels text height with 3 pixels padding on both the top and bottom).
- TRIGRAM: 18 pixels (including 1 pixel padding on both the top and bottom).

To guarantee uniform input dimensions for convolutional processing, image slices whose width was smaller than the predefined window size were padded with a white background. After slicing and padding, each sentence was represented as a stack of n visual tokens, resulting in an input tensor of shape $n \times h \times w$, where n denotes the number of image slices and h and w correspond to the height and width of each slice.

Each image slice was processed by a one-layer convolutional neural network followed by batch normalization and a ReLU activation function, producing a 512-dimensional visual embedding. These embeddings were then provided as input to the transformer encoder–decoder model.

4.5 Training Procedure

All experiments were conducted on Google Colab Pro using the PyTorch framework, with an NVIDIA L4 GPU (22.5 GB memory) and 53 GB of system RAM. All models were trained under identical conditions using the Adam optimizer (Kingma and Ba, 2015). Visual models are trained with a learning rate of 5×10^{-4}, dropout of 0.3, and token-based batching with a maximum of 10k tokens per batch; all other training hyperparameters follow standard fairseq defaults. The maximum number of epochs was set to 100 for both traditional and visual models. For the NON-VOWEL model, additional training runs were conducted to examine the effect of extending training on model performance.

Table 1: Some Statistics for Training Models

Model	#Epochs	Max tokens	Slice (h x w)	Mean of # slices (Training/Testing)
Baseline (Text)	100	4096	No images	No images
BASE1	81	20000	22 x 31	74/71.3
BASE2	100	20000	22 x 31	74/71.3
NON-VOWELS	172	20000	22 x 31	56.3/54.1
TRIGRAM	100	20000	18 x 54	32.3/31.5

Table 2: Visual Models Training Time

Model	Training Time (s)
BASE2	17047.6
NON-VOWELS	13687.7 + 9807.5
TRIGRAM	10693.5

For visual models, an early stopping criterion was applied using a patience value of 10, halting training if no improvement was observed over 10 consecutive epochs. After training, all outputs were post-processed to remove BPE artifacts (if any) before evaluating model performance using BLEU scores on the test set.

To compare visual embeddings with traditional text-based embeddings under conditions with out-of-vocabulary (OOV) words and spelling mistakes, we created five additional test sets. These were derived from the original test dataset by introducing random character swaps within words. Character swaps were chosen as they simulate common typographic errors frequently observed in user- generated text. Specifically, two consecutive characters in a word were swapped with probabilities of 10%, 20%, 30%, 40%, and 50%, respectively.

We evaluated the text-based baseline model (using traditional embeddings) and TRIGRAM (using visual embeddings) on these modified test sets to examine how spelling errors and OOV words affect each embedding type.

5 Results and Discussion

This section presents and analyzes the experimental results of both traditional text-based and visual embedding models. We discuss training dynamics, computational efficiency, translation quality on clean data, and robustness under noisy input conditions.

Table 1 summarizes key training characteristics of all models, including the number of epochs until convergence, maximum token limits, visual slice dimensions, and the average number of visual to-

Table 3: BLEU Scores for the Models on Clean Test Set

Model	BLEU Score (Clean)
Baseline (Text)	17.26
BASE1	14.09
BASE2	14.16
NON-VOWELS	14.28 (100 epochs) - 14.60 (172 epochs)
TRIGRAM	14.28

Table 4: BLEU Scores for Noisy Inputs with Different Swap Probabilities

Swap Probability	Baseline (Text)	TRIGRAM
$p = 10\%$	12.93	13.86
$p = 20\%$	9.22	12.26
$p = 30\%$	6.70	10.91
$p = 40\%$	4.80	9.45
$p = 50\%$	3.39	8.27

kens per sentence. Beyond reporting configurations, the table provides insight into how different tokenization and embedding strategies affect training behavior and computational cost.

The BASE1 model converged after only 81 epochs, substantially earlier than the other visual models. While this early convergence indicates faster saturation during training, it is also associated with weaker generalization performance, as reflected by its lower BLEU score on the clean test set (Table 3). This behavior can be attributed to the use of word-level tokenization on the target side, which restricts the model's ability to learn subword-level patterns compared to Byte-Pair Encoding (BPE).

Table 1 also highlights the effect of visual token granularity on input sequence length. Sliding-window models (BASE1, BASE2, and NON-VOWELS) generate a large number of visual tokens per sentence. For instance, BASE2 produces an average of 74 slices per training sentence, whereas the TRIGRAM model produces only 32.3 slices. This substantial reduction in sequence length directly impacts computational efficiency and explains the shorter training time of the TRIGRAM model reported in Table 2. These results suggest that non-overlapping trigram segmentation offers a more compact and efficient visual representation.

Table 2 reports the total training time for visual models. The TRIGRAM model completed training in approximately 2.97 hours (10,693.5 seconds), making it the fastest among all visual configurations. In contrast, BASE2 and NON-VOWELS required longer training times due to their higher number of visual tokens per sentence. The NON-VOWELS model, however, benefits from reduced sequence length compared to BASE2 as a result of vowel removal, leading to lower computational cost while maintaining comparable translation quality.

5.1 Clean Dataset Performance

Table 3 presents BLEU scores for all models evaluated on the clean (unmodified) test set. The traditional text-based baseline achieved the highest BLEU score of 17.26, outperforming all visual embedding models. This result confirms that subword-based text embeddings remain well-optimized for standard machine translation when input data is clean and well-formed.

Among the visual models, NON-VOWELS and TRIGRAM achieved the highest BLEU scores (14.28), while BASE1 and BASE2 performed slightly worse. Extending the training of the NON-VOWELS model resulted in a modest improvement, reaching a BLEU score of 14.60 after 172 epochs. This suggests that visual models can benefit from longer training durations, although they still do not reach the performance level of traditional text-based embeddings on clean inputs.

5.2 Robustness to Noisy Inputs

Table 4 highlights the robustness of the models under increasing levels of noise introduced through character swaps. As the swap probability increases, the BLEU score of the baseline model degrades rapidly, dropping from 17.26 on clean input to 12.93 at 10% noise and further to 3.39 at 50% noise. This sharp decline indicates the sensitivity of text-based tokenization methods to spelling errors and out-of-vocabulary (OOV) words.

In contrast, the TRIGRAM visual model demonstrates substantially greater robustness. It achieves a BLEU score of 13.86 at 10% noise and maintains a score of 8.27 even at 50% noise. Figure 3 illustrates this trend, showing that the performance gap between the baseline and the visual model widens as noise levels increase.

These results demonstrate a key advantage of visual embeddings: their ability to retain structural information from character images even when characters are partially corrupted. Unlike traditional text-based tokenization, visual representations are less sensitive to minor spelling variations, making them particularly suitable for noisy or user-generated text scenarios.

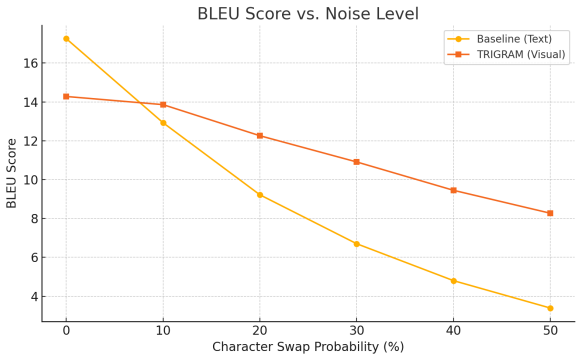

Figure 3: BLEU score vs. noise level for Baseline and TRIGRAM.

5.3 The Role of Target-Side Tokenization

An important observation from the experiments is the influence of target-side tokenization on convergence behavior and translation quality. While the primary focus of this study is on source-side embedding strategies, the choice of tokenization for the Arabic target language plays a critical role in overall model performance.

Models employing Byte-Pair Encoding (BPE) on the target side, such as NON-VOWELS, achieved better generalization and higher BLEU scores than models relying on word-level tokenization, such as BASE1. Although BPE-based models required more epochs to converge, they were able to capture finer-grained morphological patterns in Arabic, a morphologically rich language. This suggests that even when visual embeddings are used on the source side, retaining subword-level processing on the target side remains beneficial for translation quality and generalization.

6 Limitations

Despite the promising results, this study has several limitations that should be considered when interpreting the findings.

First, the experimental setup for visual embeddings was intentionally constrained to reduce variability. Only a single font (NotoSans-Regular) and font size (10pt) were used during image rendering. While this choice follows prior work (Salesky et al., 2021) and ensures comparability across models, different font styles or sizes may influence the visual appearance of characters and, consequently, the quality of the learned embeddings. Similarly, the visual tokenization process relied on fixed window and stride sizes for the sliding-window models. Alternative configurations could lead to different token structures and potentially affect model performance.

Second, several training-related factors may limit the conclusions drawn from the results. Hyperparameters such as learning rate, dropout, and optimizer settings were fixed across all experiments following recommendations from prior work. Although this ensures a fair comparison, some models—particularly the NON-VOWELS configuration—may benefit from further tuning. In addition, early stopping with a fixed patience value was applied uniformly to all models. While this promotes consistency, it may prevent certain models from fully converging, especially those that require longer training to stabilize.

Third, the scope of the experimental evaluation is limited. All experiments were conducted on a single language pair (English–Arabic) using the IWSLT 2017 dataset. While this dataset is a well-established benchmark for low-resource machine translation, the findings may not generalize directly to other language pairs, larger-scale datasets, or domain-specific translation tasks. Further experiments across diverse languages and domains are necessary to assess the broader applicability of visual embedding approaches.

Finally, evaluation relied primarily on BLEU as the main performance metric. Although BLEU is widely used and enables quantitative comparison with prior work, it does not fully capture semantic adequacy, fluency, or human-perceived translation quality. Future work could incorporate complementary evaluation methods, such as human judgments or semantic similarity metrics, to provide a more comprehensive assessment of translation performance.

7 Conclusion and Future Work

This study investigated the effectiveness of visual word embeddings compared to traditional text-based embeddings for English-to-Arabic machine translation under both clean and noisy input conditions. While the traditional text-based baseline achieved the highest BLEU score on clean data, visual embedding models demonstrated substantially greater robustness to spelling noise and out-of-vocabulary (OOV) tokens. Among the visual approaches, the TRIGRAM model achieved the best balance between translation quality and computational efficiency, benefiting from its non-overlapping tokenization strategy and reduced se-

quence length.

These findings highlight that visual representations provide a promising alternative to conventional text embeddings in scenarios where input text is noisy or user-generated. Importantly, this robustness comes at the cost of lower BLEU scores on clean data, indicating a deliberate trade-off between peak benchmark performance and stability under input corruption. In addition, the use of subword-level tokenization (BPE) on the target language proved crucial for effective learning, enabling better convergence and improved translation quality even when visual embeddings were used on the source side.

For future work, we plan to extend this study in several directions. First, we aim to evaluate visual embedding models on larger and more diverse datasets to further assess their generalization capabilities and robustness across domains. Second, we plan to explore additional character-level segmentation strategies for visual tokenization, including bigram and higher-order n-gram representations, to better understand the trade-offs between translation performance, robustness, and computational cost. Finally, motivated by recent findings on the effectiveness of character trigrams for Arabic text processing under noisy conditions, future work will investigate the application of visual n-gram tokenization directly to Arabic source text in Arabic-English translation settings.

Acknowledgments

The authors would like to thank Saudi Data and AI Authority (SDAIA) and King Fahd University of Petroleum & Minerals (KFUPM) for supporting this work through SDAIA-KFUPM Joint Research Center for Artificial Intelligence grant number JRC-AI-CAI02563.

References

Maged Al-Shaibani and Irfan Ahmad. 2023. Consonant is all you need: a compact representation of English text for efficient NLP. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 11578–11588. Association for Computational Linguistics.

Dorieh Alomari and Irfan Ahmad. 2024. Exploring character trigrams for robust arabic text classification: A comparative analysis in the face of vocabulary expansion and misspelled words. *IEEE Access*, 12:57103–57116.

Yonatan Belinkov and Yonatan Bisk. 2018. Synthetic and natural noise both break neural machine translation. In *Proceedings of the 6th International Conference on Learning Representations*, Vancouver, Canada.

Mauro Cettolo, Marcello Federico, Luisa Bentivogli, Jan Niehues, Sebastian Stüker, Katsuhito Sudoh, Koichiro Yoshino, and Christian Federmann. 2017. Overview of the IWSLT 2017 evaluation campaign. In *Proceedings of the 14th International Conference on Spoken Language Translation*, pages 2–14, Tokyo, Japan. International Workshop on Spoken Language Translation.

Asli Eyecioglu and Bill Keller. 2016. Character-based neural models for semantic similarity. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 248–255. Association for Computational Linguistics.

Markus Hassler and Gernot Fliedl. 2006. Text preparation through extended tokenization. https://www.witpress.com/elibrary/wit-transactions-on-information-and-communication-tec 37/16699. Accessed: 2022-08-11.

Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM International Conference on Information and Knowledge Management*, pages 2333–2338. ACM.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*.

Jonas Lotz, Elizabeth Salesky, Phillip Rust, and Desmond Elliott. 2023. Text rendering strategies for pixel language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10155–10172, Singapore. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Sarthak Nigam, Sayan Borah, and Vasudha Bhatnagar. 2019. Semantic textual similarity using character trigram embeddings. *Expert Systems with Applications*, 128:63–72.

Phillip Rust, Jonas F. Lotz, Emanuele Bugliarello, Elizabeth Salesky, Miryam de Lhoneux, and Desmond Elliott. 2023. Language modelling with pixels. In *International Conference on Learning Representations*.

Elizabeth Salesky, David Etter, and Matt Post. 2021. Robust open-vocabulary translation from visual text representations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7235–7252, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Elizabeth Salesky, Neha Verma, Philipp Koehn, and Matt Post. 2023. Multilingual pixel representations for translation and effective cross-lingual transfer. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13845–13861, Singapore. Association for Computational Linguistics.

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.

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 *Advances in Neural Information Processing Systems*, volume 30.