

Original article

Transfer Learning from ImageNet-Pretrained Vision Models for Arabic Text Classification Using Glyph-Based Image Representations

Hosam Alzawam^{1*} , Tarik Idbeaa² 

¹Department of Computer Science, Faculty of Education, Gharyan University, Gharyan, Libya

²Department of Computer Science, Faculty of Science, University of Gharyan, Gharyan, Libya

Email: hosamzwam1@gmail.com

Abstract

Arabic text classification is challenging because of rich morphology, orthographic variation, and complex tokenization. This study investigates an alternative approach that represents Arabic documents as rendered images and classifies them using ImageNet-pretrained vision models. We developed a rendering pipeline to preserve contextual letter shaping and right-to-left layout; we also divided long documents into image tiles and classified them by averaging tile-level probabilities. EfficientNet-B3 and Xception were fine-tuned using the same two-stage training protocol without automated hyperparameter optimization. Experiments were conducted on 6,000 balanced Modern Standard Arabic news articles across six categories. EfficientNet-B3 achieved 98.77% test accuracy, compared with 98.34% for Xception, while requiring approximately half the parameters and one-fifth of the floating-point operations. The approach eliminates tokenization, vocabulary construction, and manual feature engineering while achieving competitive classification performance. These results demonstrate the potential of rendered Arabic text as a compact and effective representation for visual transfer learning in Arabic text classification.

Keywords: Arabic Text Classification, Transfer Learning, Glyph-Based Representation, EfficientNet-B3, Xception.

Introduction

Text classification involves assigning documents to predefined categories and has been widely applied in applications such as news organization, content moderation, information retrieval, and topic analysis. Arabic text classification presents particular challenges due to the language's rich morphological structure and orthographic variability. Arabic words are generated from roots through productive morphological patterns and may contain attached prefixes, suffixes, and clitics. In addition, diacritics are commonly omitted, while certain characters, including alef and taa marbuta, exhibit orthographic variation. The limited availability of large annotated Arabic corpora further constrains model development. Collectively, these characteristics increase lexical variability, distribute semantic information across multiple surface forms, and complicate the training of robust classification models, particularly when training data are limited.[1]

Deep learning has become a dominant approach to Arabic text classification. Transformer-based encoders pretrained on Arabic corpora, particularly AraBERT, have demonstrated substantial improvements across a broad range of Arabic natural language understanding tasks [2]. Hybrid architectures integrating convolutional and recurrent layers have also been successfully applied to large-scale Arabic news classification [3], while BERT-based transfer learning has been extended to informal and long-form Arabic texts [4]. Despite their architectural differences, these approaches predominantly represent text as sequences of discrete linguistic units. Consequently, they depend on tokenization and vocabulary-based representations, which may be affected by orthographic variation and previously unseen word forms. Transformer-based architectures may also impose considerable computational and memory requirements during both training and inference.

An alternative research direction represents textual content in the visual domain rather than as a sequence of discrete tokens. Convolutional neural networks pretrained on ImageNet can learn hierarchical visual representations involving edges, strokes, curves, and spatial configurations [5]. Features learned in the earlier layers of such networks are generally transferable across visual tasks, providing a basis for adaptation through transfer learning [6]. Building on this principle, the Super Characters approach represented text as images and employed two-dimensional convolutional networks for classification, demonstrating promising performance across sentiment and topic classification tasks in Chinese, Japanese, Korean, and English [7]. Subsequent approaches extended this concept by training language models directly on rendered text at the pixel level, demonstrating improved performance in some settings involving previously unseen scripts and greater robustness to orthographic perturbations.[8]

Despite these developments, the application of visual representations to Arabic document classification remains insufficiently investigated. Arabic introduces distinctive rendering requirements because its script is written from right to left and uses context-dependent character forms, with individual letters potentially assuming different shapes according to their position within a word. Ligatures and optional diacritics introduce additional visual variability that

must be appropriately represented during rendering. Accurate script rendering is therefore a fundamental component of an image-based approach to Arabic text classification. This study investigates whether visual representations derived from rendered Arabic text can be effectively classified using ImageNet-pretrained convolutional networks. To this end, we develop a rendering pipeline that preserves the structural characteristics of the Arabic script and evaluate two architectures with distinct designs. The first, Xception, is based on depthwise separable convolutions [9], whereas EfficientNet-B3 employs compound scaling across network depth, width, and input resolution [10]. EfficientNet-B3 is investigated as the proposed architecture, while Xception is employed as a comparative model under the same training protocol.

This work makes three main contributions.

- Visual transfer learning for Arabic text classification: To the best of our knowledge, this is the first systematic investigation of ImageNet-pretrained vision models for Arabic document classification using rendered Arabic script as the input representation.
- Arabic-aware text rendering: We develop a reproducible rendering pipeline that preserves contextual character shaping and right-to-left layout, adapts the rendered text to the native input resolution of each network, and handles long documents through consecutive image tiles whose predictions are aggregated to obtain a document-level classification.
- Controlled architectural and computational evaluation: We systematically compare EfficientNet-B3 and Xception under an identical fine-tuning protocol, with hyperparameters specified a priori and without metaheuristic optimization. The evaluation considers both classification performance and computational efficiency and contextualizes the results against recent Arabic news-classification studies.

The remainder of this paper is organized as follows. Section 2 reviews the related literature and establishes the research gap. Section 3 presents the proposed methodology. Section 4 describes the experimental setup and evaluation metrics. Section 5 presents and discusses the experimental results and their comparison with previous studies. Section 6 concludes the paper and outlines directions for future research.

Literature Review

Recent advances in Arabic natural language processing (NLP) have substantially improved the performance of automated text classification systems. Nevertheless, most existing approaches rely on symbolic or token-based representations of language. In parallel, a growing body of research has investigated visual representations of text, in which linguistic content is encoded as images and processed using computer-vision architectures. This section reviews the literature relevant to the proposed approach from three complementary perspectives: deep and transfer learning for Arabic text classification, visual representations of textual information, and the use of ImageNet-pretrained convolutional architectures for Arabic-related tasks. The section concludes by identifying the specific methodological gap addressed by the present study.

Deep and transfer learning for Arabic text classification

Pretrained language models have become a major component of contemporary Arabic NLP. Among these models, AraBERT adapted the BERT architecture to the linguistic characteristics of Arabic through large-scale pretraining on Arabic corpora and language-specific segmentation strategies [2]. Its development established an important foundation for subsequent transfer-learning approaches to Arabic classification.

Several studies have subsequently investigated alternative deep architectures and transfer-learning strategies. Alruily et al. developed ARABERT4TWC for the classification of long Arabic tweets and evaluated the model on five publicly available datasets [4]. Alnagi et al. proposed a hybrid architecture combining a one-dimensional convolutional module inspired by the Inception architecture with LSTM layers. The model achieved reported accuracies of 96% on AlKhaleej, 92% on Akhbarona and the complete SANAD collection, and 89% on NADIA.[3]

Other studies have explored multibranch architectures to capture complementary linguistic characteristics. Jamaled dyn et al. introduced CLGNet, a multichannel architecture integrating convolutional, LSTM, and GRU components for modeling local and sequential information in Arabic news. The model was evaluated using the CNN, BBC, and OSAC corpora [11]. In a comparatively lightweight approach, Alayba and Altamimi combined word embeddings with a support vector machine (SVM) classifier and reported an accuracy of 90.77% on the ANACD corpus, which contains 90,000 articles distributed across nine categories [12]. Zamzami et al. investigated machine-learning and deep-learning approaches for classifying Arabic news according to country of origin and reported 94.4% accuracy and an F1-score of 0.94 for a CNN-LSTM architecture [13]. More recently, Asiri and Shoaib [14] investigated Arabic fake-news detection using BERT-based representations combined with a Bi-LSTM architecture for modeling sequential linguistic

information. Their experimental results demonstrated the effectiveness of pretrained contextual representations for Arabic misinformation classification, achieving an accuracy of 87% and an F1-score of 88%. However, the approach remains dependent on token-level textual representations and therefore differs fundamentally from the rendered-script visual representation investigated in the present study.

A number of these investigations employ established Arabic news benchmarks. For example, SANAD contains 194,797 single-label articles collected from the AlKhaleej, AlArabiya, and Akhbarona sources [15]. Despite differences in architecture and training strategy, the aforementioned approaches share a fundamental characteristic: the textual input is represented through discrete linguistic units, such as words, subwords, or embeddings derived from them. Consequently, their performance is influenced by tokenization, vocabulary construction, lexical variation, and, in the case of pretrained language models, the availability of large language-specific pretraining corpora.

These observations motivate the investigation of an alternative representation in which the textual signal is preserved in its visual form rather than decomposed exclusively into symbolic units. Such a representation may provide a complementary means of modeling orthographic and morphological variation and, importantly, may allow the use of pretrained visual models without requiring language-specific pretraining.

Visual representations of text

An alternative perspective on text classification treats written language as a visual signal. Instead of converting a document into a sequence of tokens, the textual content is rendered as an image and subsequently processed using image-classification architectures. This paradigm establishes a direct connection between natural language processing and computer vision and enables the transfer of visual representations learned from large-scale image datasets.

Sun et al. introduced the Super Characters approach, in which textual content is rendered onto an image canvas and subsequently classified using a two-dimensional convolutional neural network. The method demonstrated competitive performance on sentiment and topic classification across Chinese, Japanese, Korean, and English datasets [7]. This work established the feasibility of treating textual appearance as a discriminative representation rather than relying exclusively on symbolic linguistic features.

Subsequent studies extended the visual representation of language in several directions. The same research line introduced squared word glyph representations for English and investigated their deployment on low-power convolutional hardware for on-device language processing [16]. Glyce incorporated glyph-level visual information into Chinese character representations by extracting features from character images using convolutional networks [17]. These approaches demonstrate that visual properties of written characters can provide information complementary to conventional token representations.

A more direct pixel-level formulation was introduced by PIXEL, which pretrained a language model on rendered textual sequences using masked-patch reconstruction [8]. The resulting representations demonstrated competitive performance on linguistic tasks and increased robustness to certain orthographic perturbations. Subsequent work extended pixel-based representations to multilingual machine translation, including Arabic, and reported improvements of up to 9 BLEU over subword-based baselines in the evaluated settings.[18]

Collectively, these studies establish rendered text as a viable computational representation. However, their objectives, languages, architectures, and training requirements differ substantially from those considered in the present work. Some focus primarily on Latin or East Asian scripts, others address language understanding or generation rather than document-level topic classification, and pixel-based language models generally require dedicated pretraining. Consequently, the question of whether an off-the-shelf ImageNet-pretrained visual backbone can be transferred directly to rendered Arabic documents remains insufficiently explored.

ImageNet-pretrained convolutional networks and Arabic

The intersection between computer vision and Arabic NLP has received comparatively limited attention. Existing studies nevertheless provide evidence that visual architectures can process representations derived from Arabic textual information. Aboulola and Umer investigated several image-classification architectures for Arabic fake-news detection by representing Arabic news using ELMo-derived embedding tensors. Their experiments reported accuracies of 96.24% for Xception, 95.89% for ResNet, 89.68% for Inception, and 97.15% for EfficientNet-B4, while a CNN-LSTM voting ensemble achieved 98.42% [19]. Although this work demonstrates the applicability of visual architectures to Arabic-derived representations, its input is a numerical embedding matrix rather than an image of the original Arabic script. Furthermore, the target task is fake-news or credibility detection rather than document-level topic classification.

Another relevant line of research concerns the recognition of Arabic characters from visual input. Al-Maamari et al. combined VGG16 and ResNet50 with a transformer encoder for Arabic printed and handwritten character recognition

and reported accuracies of 99.51% and 98.19%, respectively [20]. These findings demonstrate the capacity of vision architectures to learn discriminative representations from Arabic glyphs. However, character recognition differs fundamentally from document classification because the former operates at the character level, whereas the present study seeks to infer the semantic topic of an entire document from its rendered script. Taken together, the existing literature provides evidence for two complementary propositions. First, pretrained visual architectures can learn useful representations from Arabic-derived inputs. Second, visual representations of text can preserve information that is useful for downstream language tasks. What remains insufficiently investigated is whether these two observations can be combined by directly rendering Arabic documents and transferring ImageNet-pretrained vision models to document-level Arabic topic classification.

Research gap

Table 1 summarizes the studies most closely related to the present work. The literature reveals three specific gaps. First, existing studies have not systematically evaluated ImageNet-pretrained vision architectures for Arabic document classification when the input consists directly of rendered Arabic script. Previous Arabic studies using vision architectures have generally operated on numerical embeddings or focused on character recognition rather than document-level topic classification.

Second, the rendering requirements associated with Arabic script have received limited attention in visual text classification. Arabic requires contextual character shaping, right-to-left layout, and appropriate handling of ligatures and other script-specific characteristics. In addition, long documents must be represented within the fixed spatial dimensions expected by image-classification networks.

Third, there is limited evidence from controlled comparisons of modern ImageNet-pretrained architectures on rendered Arabic text in which both predictive performance and computational requirements are evaluated under a common training protocol.

The present study addresses these gaps by developing an Arabic-aware rendering pipeline and evaluating two ImageNet-pretrained architectures, EfficientNet-B3 and Xception, under identical fine-tuning conditions. The resulting framework enables the investigation of rendered Arabic text as a visual representation for document-level topic classification while simultaneously examining the associated computational trade-offs.

Table 1: Summary of related studies and their limitations with respect to the present work

Study	Input representation	Model	Script / task	Limitation for this study
Alnagi et al. [3]	Token embeddings	Inception-1D + LSTM	Arabic / topic	Symbolic input; no visual transfer
Jamaled dyn et al. [11]	Token embeddings	CNN + LSTM + GRU (multichannel)	Arabic / news topic	Symbolic input; no visual transfer
Asiri & Shoaib [14]	BERT-based token representations	AraBERT + Bi-LSTM	Arabic fake-news detection / X (Twitter)	Token-based representation; focused on fake-news detection rather than document-topic classification
Sun et al. [7]	Rendered text	2D CNN	CJK, English / sentiment, topic	No Arabic; no cursive shaping
Rust et al. [8]	Rendered text	Pixel transformer (from scratch)	Multilingual / NLU	Costly pretraining; no ImageNet reuse
Salesky et al. [18]	Rendered text	Pixel encoder-decoder	Incl. Arabic / translation	Generation, not classification
Aboulola & Umer [19]	Embedding matrix	Xception, EfficientNet-B4, etc.	Arabic / fake news	Not rendered script
Al-Maamari et al. [20]	Glyph images	VGG16, ResNet50 + ViT	Arabic / character recognition	Not document classification

This study	Rendered Arabic glyphs	EfficientNet-B3, Xception (ImageNet)	Arabic / topic classification	—
------------	------------------------	--------------------------------------	-------------------------------	---

Methodology

This study proposes a glyph-based transfer-learning framework for Arabic document classification. The central premise is to represent textual content as rendered images while exploiting visual representations learned from large-scale natural-image datasets. Figure 1 illustrates the complete processing pipeline. Each document undergoes lightweight preprocessing and normalization before being rendered as one or more images at the spatial resolution required by the selected backbone. The resulting images are processed by an ImageNet-pretrained classification network whose original classification head is replaced with a task-specific classifier. Model adaptation is performed through a two-stage fine-tuning strategy. For documents represented by multiple tiles, tile-level class probabilities are subsequently aggregated to obtain a single document-level prediction.

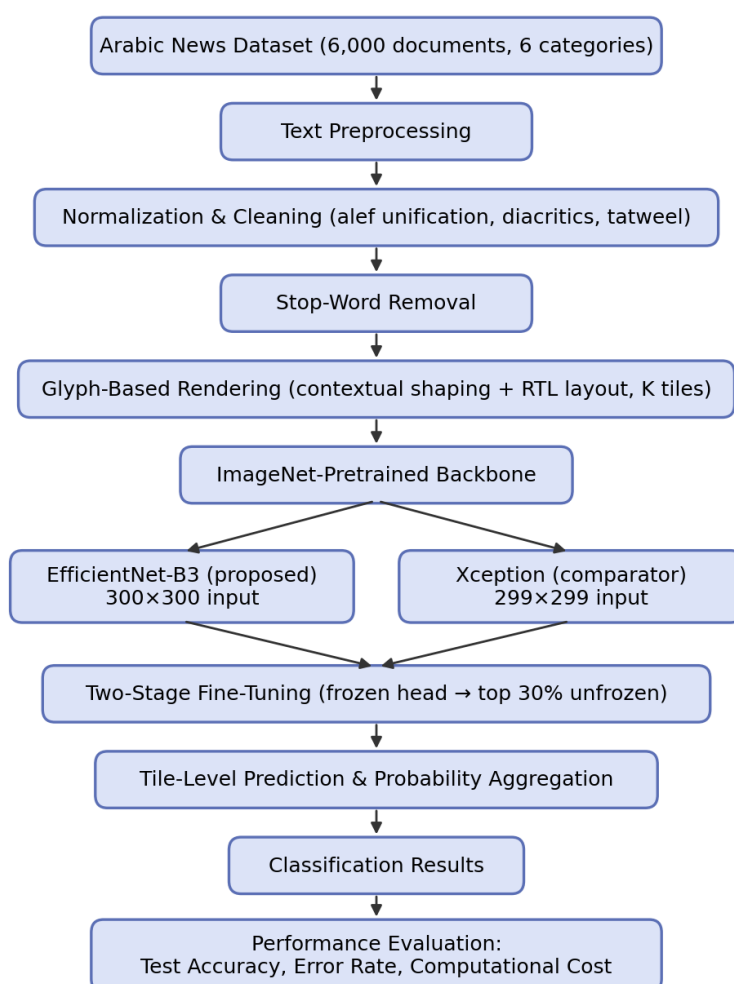


Figure 1: The proposed glyph-based transfer-learning methodology

Dataset Description

The experimental corpus comprises 6,000 Modern Standard Arabic news articles distributed uniformly across six topical categories: Sports, Politics, Economy, Health, Technology, and Culture. Each category contains exactly 1,000 documents, resulting in a balanced classification setting and avoiding class-frequency disparities during evaluation.

The articles were collected between January 2023 and December 2024 from three publicly accessible Arabic news sources: Al-Jazeera, Al-Arabiya, and BBC Arabic. The category assigned to each article was manually verified, and near-duplicate articles were removed to reduce redundancy and minimize the possibility of overlapping content across categories. Given the formal nature of the selected news sources, the corpus is predominantly composed of Modern Standard Arabic, with only a limited presence of dialectal material. The resulting class distribution is presented in Table 2.

Table 2: Number of documents in each category

Category	Number of Articles
Sports	1000
Politics	1000
Economy	1000
Health	1000
Technology	1000
Culture	1000
Total	6000

Data Preprocessing

Preprocessing was intentionally restricted to operations that reduce irrelevant variation while preserving the natural visual structure of Arabic words. Three stages were applied. First, non-linguistic content was removed. This included URLs, email addresses, Latin characters, numerical digits, and non-linguistic symbols, while consecutive whitespace characters were collapsed into a single space. Second, orthographic normalization was applied. The alternative forms of alef (ا, إ and آ) were mapped to the bare form (ا) while Arabic diacritics and the elongation character (tatweel) were removed. Third, common stop words, including frequent prepositions and conjunctions, were removed to increase the relative proportion of topical lexical content represented within each image canvas. No stemming procedure was applied. Unlike conventional token-based classifiers, the proposed approach relies on the visual form of the written text; consequently, aggressive morphological reduction could alter the natural appearance of Arabic words and potentially remove visual information relevant to the proposed representation.

Glyph-Based Rendering of Arabic Text

Let a preprocessed document be represented by the token sequence $d = (w_1, w_2, \dots, w_n)$. A rendering function R maps a sequence of tokens to a square image canvas of side length S . The text is rendered in black on a white background, with S determined by the native spatial input requirement of the selected backbone: 300 pixels for EfficientNet-B3 and 299 pixels for Xception. Rendering Arabic text requires more than conventional character drawing because the visual form of a character depends on its contextual position and because the writing direction is right-to-left. Accordingly, two script-specific operations are incorporated into the rendering pipeline. The first is contextual shaping, which determines the appropriate isolated, initial, medial, or final form of each character and resolves required ligatures. The second is bidirectional text layout, which ensures correct right-to-left ordering.

These operations are implemented using the HarfBuzz text-shaping engine through the Raqm layout backend integrated with the Pillow imaging library. Words are positioned on right-aligned lines according to their measured widths. A new line is initiated when adding the next word would exceed the available horizontal space. The rendering configuration uses a font size of 18 pixels, line spacing of 6 pixels, and a margin of 16 pixels. Under these settings, each canvas accommodates approximately 50–60 words. Representative rendered documents are presented in Figure 2.

**Figure 2: Glyph-based representations of Arabic news text from three categories**

Because a complete article may exceed the capacity of a single canvas, the document is represented using consecutive tiles. If each canvas accommodates C tokens, the k -th tile corresponds to the token interval

$$t_k = R(w_{(k-1)C+1}, \dots, w_{kC}), \quad k = 1, \dots, K \quad (1)$$

In the present configuration, a maximum of $K = 3$ tiles is used. This configuration is intended to capture the lead and early body of each article, where substantial topical information is expected to occur. Each grayscale image is replicated across the three input channels required by the pretrained backbones and subsequently normalized using the preprocessing convention associated with the corresponding architecture. Data augmentation is restricted to transformations that preserve the linguistic structure and reading direction of Arabic text. During training, random translations of up to 4%, random zoom of up to 5%, and rendering-time font-size jitter of ± 2 pixels are applied. Horizontal flipping and image rotation are excluded because they would either reverse the writing direction or alter the baseline and geometric structure of the script.

ImageNet-Pretrained Backbones

- *Xception*: Xception [9] replaces the Inception module with depthwise separable convolutions, which factorize a standard convolution into a per-channel spatial convolution followed by a 1×1 pointwise convolution. For a kernel of size $D_k \times D_k$ and N output channels, the cost relative to a standard convolution is reduced by the ratio given in Eq. (2). The network stacks 36 convolutional layers in 14 modules with residual connections and contains about 23 million parameters.

$$\text{cost ratio} = \frac{1}{N} + \frac{1}{D_k^2} \quad (2)$$

- *EfficientNet-B3 (proposed)*: EfficientNet [10] is built from mobile inverted bottleneck (MBConv) blocks with squeeze-and-excitation channel attention and scales a baseline network by a compound coefficient φ that jointly controls depth d , width w , and input resolution r , as expressed in Eq. (3). EfficientNet-B3 contains about 12 million parameters, requires 1.8 billion floating-point operations (FLOPs), and reaches 81.6% top-1 accuracy on ImageNet, compared with 79.0% for Xception at 8.4 billion FLOPs [10].

$$d = \alpha^\varphi, w = \beta^\varphi, r = \gamma^\varphi, \alpha \cdot \beta^2 \cdot \gamma^2 \approx 2 \quad (3)$$

EfficientNet-B3 is designated as the proposed architecture because it provides a favorable combination of predictive capability and computational efficiency and because the EfficientNet family has previously demonstrated strong performance on Arabic text-derived inputs [19]. Its squeeze-and-excitation mechanism also provides an architectural mechanism for adaptive channel weighting, which may be useful when discriminative visual patterns are distributed unevenly across feature channels. For both architectures, the original classification layer is replaced with a task-specific classification head consisting of global average pooling, dropout with a rate of 0.3, and a fully connected softmax layer containing one output unit for each of the six categories.

Two-Stage Fine-Tuning and Document-Level Aggregation

Fine-tuning proceeds in two stages. In the first stage the backbone is frozen and only the new head is trained for five epochs with a learning rate of 10^{-3} . In the second stage the top 30% of backbone layers are unfrozen, batch-normalization layers are kept in inference mode, and the network is trained for up to 25 epochs with a learning rate of 10^{-5} under cosine decay and early stopping. Both stages use the AdamW optimizer with a weight decay of 10^{-4} and a cross-entropy loss with label smoothing:

$$L = -\sum_{c=1}^M q_c \log p_c, \quad q_c = (1 - \varepsilon) y_c + \frac{\varepsilon}{M} \quad (4)$$

where y_c is the one-hot label, p_c the predicted probability of class c , M the number of classes, and $\varepsilon = 0.1$. All hyperparameters are fixed a priori and are identical for both backbones; no metaheuristic or automated search is performed, so that performance differences can be attributed to the architectures themselves. At inference time, the class probabilities of the K' non-empty tiles of a document are averaged:

$$p(y | d) = \frac{1}{K'} \sum_{k=1}^{K'} \text{softmax}(f_\theta(t_k)), \quad \hat{y} = \text{argmax}_y p(y | d) \quad (5)$$

Experimental Setup

This section describes the experimental configuration used to evaluate the proposed framework. It specifies the computational environment, data partitioning strategy, common training configuration, and evaluation measures used for both backbones.

Implementation and Data Partitioning

The experiments were implemented in Python using TensorFlow/Keras and executed on the Kaggle cloud computing platform with two NVIDIA Tesla T4 GPUs, each providing 16 GB of memory. Data-parallel training was employed across the two GPUs. Arabic text rendering was performed using Pillow together with the Raqm layout engine. The complete corpus was randomly divided into training, validation, and independent test subsets using stratified partitioning to preserve the class distribution across the three subsets. The validation set was used exclusively for

monitoring training and implementing early stopping. The held-out test set was not used during model training or model-selection decisions, and all reported performance measures were calculated on this independent subset. The principal training configuration shared by both architectures is summarized in Table 3.

Table 3: Fixed training configuration shared by both backbones

Setting	EfficientNet-B3	Xception
ImageNet weights	Yes	Yes
Input size (pixels)	300 × 300	299 × 299
Tiles per document (K)	3	3
Classification head	GAP → Dropout (0.3) → Softmax	GAP → Dropout (0.3) → Softmax
Stage 1 (frozen backbone)	5 epochs, lr = 10 ⁻³	5 epochs, lr = 10 ⁻³
Stage 2 (top 30% unfrozen)	≤ 25 epochs, lr = 10 ⁻⁵ , cosine	≤ 25 epochs, lr = 10 ⁻⁵ , cosine
Optimizer / weight decay	AdamW / 10 ⁻⁴	AdamW / 10 ⁻⁴
Loss	Cross-entropy, label smoothing 0.1	Cross-entropy, label smoothing 0.1
Batch size	32	32
Hyperparameter search	None (fixed a priori)	None (fixed a priori)

Performance Metrics

Because the corpus is balanced across the six target categories, test accuracy provides an appropriate primary measure of classification performance without being dominated by a majority class. Accuracy is defined as:

$$Accuracy = \frac{N_{correct}}{N_{test}}, \quad Error\ rate = 1 - Accuracy \quad (6)$$

where $N_{correct}$ denotes the number of correctly classified documents and N_{test} represents the total number of documents in the test set. The corresponding error rate is calculated as

$$Error\ Rate = 100 - Accuracy. \quad (7)$$

In addition to predictive performance, computational efficiency is evaluated using the number of trainable parameters and the estimated floating-point operations (FLOPs) associated with each backbone.

Results and Discussion

Both ImageNet-pretrained architectures achieved high classification accuracy when applied to rendered Arabic documents, indicating that visual representations learned from natural-image data can be effectively adapted to the proposed task. The test-set results are summarized in Table 4.

Table 4: Test accuracy of the two backbones on the six-domain corpus

Model	Input size	Test accuracy (%)	Error rate (%)	Relative error reduction
Xception	299 × 299	98.34	1.66	—
EfficientNet-B3 (Proposed)	300 × 300	98.77	1.23	25.9%

As reported in Table 4, EfficientNet-B3 achieves the higher test accuracy, 98.77%, compared with 98.34% for Xception. Although the absolute gain is 0.43 percentage points, it corresponds to a relative reduction of about 26% in the error rate, from 1.66% to 1.23%; in other words, EfficientNet-B3 makes about 26% fewer classification errors than Xception on the same test documents. Because both networks were trained with the same rendered inputs, the same tiling, the same classification head, and the same fixed hyperparameters, this difference is most likely attributable to the backbones themselves, although its statistical significance was not assessed in this study. The advantage of EfficientNet-B3 is consistent with its higher ImageNet accuracy at a lower computational cost. A plausible explanation is that its squeeze-and-excitation blocks can emphasize channels that respond to informative glyph patterns, a form of channel re-weighting that is not part of the depthwise separable design of Xception; confirming this mechanism would require a layer-level analysis.

That both backbones exceed 98% accuracy is itself noteworthy. Neither network was pretrained on Arabic text, and no tokenizer, vocabulary, stemmer, or term-weighting scheme was involved. The only language-specific components of the pipeline are the light normalization and stop-word filtering steps and the shaping-aware renderer. This suggests that general visual features learned from natural images can, once fine-tuned, discriminate the recurring Arabic word shapes associated with different topics.

Beyond accuracy, the two backbones differ substantially in cost. As Figure 3 shows, EfficientNet-B3 obtains its higher accuracy with roughly half the parameters of Xception (12 M versus 23 M) and less than a quarter of its FLOPs (1.8 B versus 8.4 B) [10], which makes it the more attractive option for deployment on resource-constrained hardware.

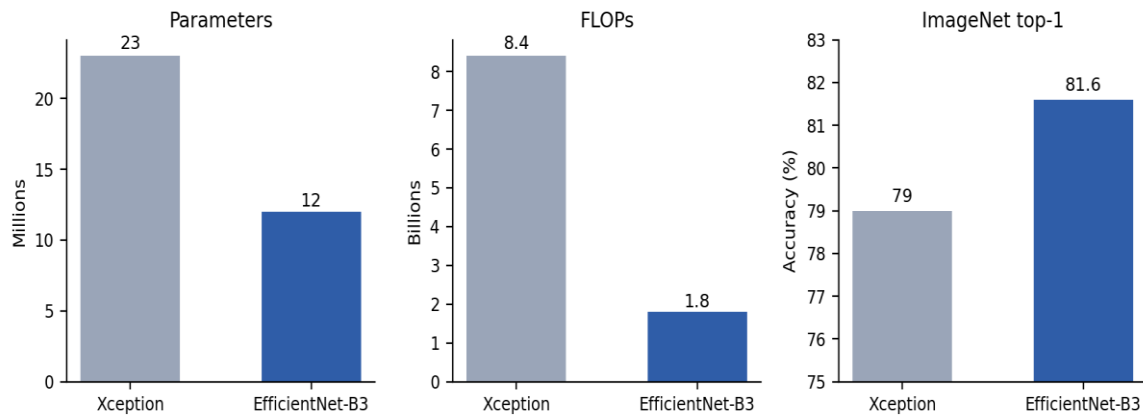


Figure 3: Parameters, FLOPs, and ImageNet top-1 accuracy of the two backbones (values from [10])

Comparison with Previous Studies

To position the proposed model within the field, its results were compared with recent studies on Arabic news classification, as summarized in Table 5 and Figure 4. Because these studies use different corpora, the comparison is contextual rather than a controlled benchmark. With an accuracy of 98.77%, the proposed EfficientNet-B3 is higher than the 96% reported for the hybrid Inception-1D and LSTM model on AlKhaleej [3], the 94.4% of the CNN-LSTM hybrid of Zamzami et al. [13], and the 90.77% of the embedding-based SVM of Alayba and Altamimi [12]. Results from related tasks provide further context. Recent Arabic NLP research continues to rely predominantly on token- and embedding-based representations. For example, Asiri and Shoaib [14] proposed an AraBERT-Bi-LSTM framework for Arabic fake-news detection on X (Twitter), achieving an accuracy of 87% and an F1-score of 88%. Their study demonstrates the effectiveness of combining pretrained Arabic contextual representations with recurrent sequence modeling. However, the approach remains dependent on tokenization and contextual embedding, and its objective is fake-news detection rather than multi-class Arabic news-topic classification. This distinction highlights the methodological difference between conventional text-based Arabic classification and the rendered-script visual representation investigated in the present study. Notably, the proposed approach reaches it with an off-the-shelf pretrained backbone, fixed hyperparameters, and no tokenizer, vocabulary, or hand-crafted features.

Table 5: Comparison of the proposed approach with previous studies on Arabic news classification

Study	Method	Dataset / Task	Accuracy (%)
Alayba & Altamimi [12]	Word embeddings + SVM	ANACD, 9 classes / topic	90.77
Zamzami et al. [13]	CNN-LSTM	Hajj news / country of origin	94.40
Alnagi et al. [3]	Inception-1D + LSTM	SANAD-AlKhaleej / topic	96.00
This study	Xception (glyph input)	Six-domain Arabic news / topic	98.34
This study	EfficientNet-B3 (glyph input, proposed)	Six-domain Arabic news / topic	98.77

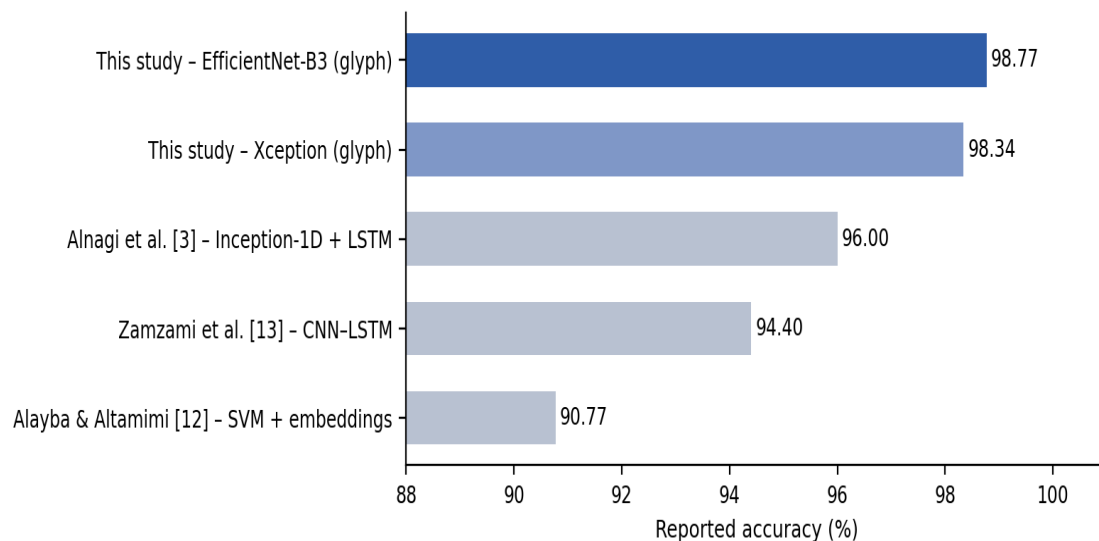


Figure 4: Accuracy of the proposed model compared with previously reported results on Arabic news classification

Limitations

The results suggest that the lower-layer filters learned from ImageNet, which are largely general and respond to edge-like patterns [6], provide a useful basis for recognizing recurring Arabic word shapes, and that fine-tuning the upper layers adapts the network to topic-discriminative patterns. Because the input is rendered script, the approach removes the need for a tokenizer, a vocabulary, or hand-engineered features, so orthographic variants and rare words are processed by the same visual mechanism as frequent ones.

Several limitations should be acknowledged. First, the fixed capacity of each canvas means that only the first $K \times C$ words of an article are observed, which may omit relevant content from long documents. Second, results may depend on rendering choices such as font family and size. Third, the evaluation is restricted to a single Modern Standard Arabic news corpus and to test accuracy, which is appropriate for a balanced corpus but could be complemented by per-class and repeated-run analyses; performance on public benchmarks, dialectal text, and social-media content also remains to be established. Finally, no text-based model, such as a fine-tuned AraBERT encoder or a TF-IDF classifier, was trained on the same data split, so the accuracy and efficiency trade-off between visual and symbolic input remains to be quantified directly.

Conclusion and Future Work

This study investigated the feasibility of transferring visual representations learned from ImageNet to Arabic text classification by representing documents as images of their rendered Arabic script. A script-aware rendering pipeline was developed to preserve contextual character shaping and right-to-left layout while representing long documents through multiple image tiles. Two ImageNet-pretrained architectures, EfficientNet-B3 and Xception, were subsequently fine-tuned under an identical two-stage training protocol with hyperparameters fixed a priori.

On a balanced corpus of 6,000 Modern Standard Arabic news articles spanning six topical categories, EfficientNet-B3 achieved a test accuracy of 98.77%, compared with 98.34% for Xception. In addition to achieving the higher observed accuracy, EfficientNet-B3 required approximately half the number of parameters and less than one-quarter of the reported FLOPs of Xception. These results provide empirical support for the use of rendered Arabic text as a visual representation for document-level classification and demonstrate that ImageNet-pretrained visual features can be adapted to Arabic script without language-specific visual pretraining.

The findings should nevertheless be interpreted within the scope of the current experimental design. The evaluation is based on a single Modern Standard Arabic news corpus, a fixed rendering configuration, and a maximum of three document tiles. Future research should therefore investigate the generalizability of the approach across public Arabic benchmarks, dialectal and social-media text, and different document genres. Further experiments should also directly compare glyph-based models with strong Arabic transformer baselines and conventional statistical classifiers under identical data partitions. Additional research directions include sensitivity analysis of fonts and rendering parameters, repeated-run statistical evaluation, per-class error analysis, visual explanation techniques for identifying influential textual regions, and hybrid architectures that combine glyph-based visual representations with token-level linguistic representations.

Conflict of interest. Nil

References

1. Abdhood SF, Omar N, Tiun S. Data augmentation for Arabic text classification: a review of current methods, challenges and prospective directions. *PeerJ Comput Sci.* 2025;11:e2685. doi:10.7717/peerj-cs.2685.
2. Antoun W, Baly F, Hajj H. AraBERT: transformer-based model for Arabic language understanding. In: *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT)*; 2020; Marseille, France. p. 9-15.
3. Alnagi E, Ghnemat R, Abu Al-Haija Q. Boosting Arabic text classification using hybrid deep learning approach. *Discov Appl Sci.* 2025. doi:10.1007/s42452-025-07025-x.
4. Alruily M, Fazal AM, Mostafa AM, Ezz M. Automated Arabic long-tweet classification using transfer learning with BERT. *Appl Sci.* 2023;13(6):3482. doi:10.3390/app13063482.
5. Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L. ImageNet: a large-scale hierarchical image database. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*; 2009. p. 248-255.
6. Yosinski J, Clune J, Bengio Y, Lipson H. How transferable are features in deep neural networks? In: *Advances in Neural Information Processing Systems*. Vol. 27. 2014. p. 3320-3328.
7. Sun B, Yang L, Dong P, Zhang W, Dong J, Young C. Super Characters: a conversion from sentiment classification to image classification. In: *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis (WASSA)*; 2018. p. 309-315. doi:10.18653/v1/W18-6245.
8. Rust P, Lotz JF, Bugliarello E, Salesky E, de Lhoneux M, Elliott D. Language modelling with pixels. In: *Proceedings of the 11th International Conference on Learning Representations (ICLR)*; 2023.
9. Chollet F. Xception: deep learning with depthwise separable convolutions. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*; 2017. p. 1251-1258.
10. Tan M, Le QV. EfficientNet: rethinking model scaling for convolutional neural networks. In: *Proceedings of the 36th International Conference on Machine Learning (ICML)*. PMLR. 2019;97:6105-6114.
11. Jamaledyn I, El Ayachi R, Biniz M. Novel multichannel deep learning model for Arabic news classification. *Jordanian J Comput Inf Technol.* 2024;10(4):453-468. doi:10.5455/jcit.71-1720086134.
12. Alayba AM, Altamimi M. Optimization of Arabic text classification using SVM integrated with word embedding models on a novel dataset. *Int J Adv Appl Sci.* 2025;12(9).
13. Zamzami N, Himdi H, Sabbeh SF. Arabic news classification based on the country of origin using machine learning and deep learning techniques. *Appl Sci.* 2023;13(12):7074. doi:10.3390/app13127074.
14. Asiri HA, Shoaib M. Arabic fake news detection on X (Twitter) using Bi-LSTM algorithm and BERT embedding. *IEEE Access.* 2025;13:189372-189386. doi:10.1109/ACCESS.2025.3628530.
15. Einea O, Elnagar A, Al Debsi R. SANAD: single-label Arabic news articles dataset for automatic text categorization. *Data Brief.* 2019;25:104076. doi:10.1016/j.dib.2019.104076.
16. Sun B, Yang L, Lin M, Zhang W, Dong P, Young C, et al. System demo for transfer learning across vision and text using domain specific CNN accelerator for on-device NLP applications. *arXiv [Preprint]*. 2019;arXiv:1906.01145.
17. Meng Y, Wu W, Wang F, Li X, Nie P, Yin F, et al. Glyce: glyph-vectors for Chinese character representations. In: *Advances in Neural Information Processing Systems*. Vol. 32. 2019.
18. Salesky E, Verma N, Koehn P, Post M. Multilingual pixel representations for translation and effective cross-lingual transfer. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*; 2023.
19. Aboulola OI, Umer M. Novel approach for Arabic fake news classification using embedding from large language features with CNN-LSTM ensemble model and explainable AI. *Sci Rep.* 2024;14:30463. doi:10.1038/s41598-024-82111-5.
20. Al-Maamari MR, Ramteke R, Al-Hejri AM, Alshamrani SS, et al. Integrating CNN and transformer architectures for superior Arabic printed and handwriting characters classification. *Sci Rep.* 2025;15:29936. doi:10.1038/s41598-025-12045-z.