Pipeline Enabling Zero-shot Classification for Bangla Handwritten Grapheme

Linsheng Guo
Preferred Networks .inc
linsho@preferred.jp

Md Habibur Rahman Sifat
The Hong Kong Polytechnic University
habib.sifat@connect.polyu.hk

Tashin Ahmed
AriSaf Tech Japan K.K.
tashin@arisaftech.co.jp

Abstract

This research investigates Zero-Shot Learning (ZSL), and proposes CycleGAN-based image synthesis and accurate label mapping to build a strong association between labels and graphemes. The objective is to enhance model accuracy in detecting unseen classes by employing advanced font image categorization and a CycleGAN-based generator. The resulting representations of abstract character structures demonstrate a significant improvement in recognition, accommodating both seen and unseen classes. This investigation addresses the complex issue of Optical Character Recognition (OCR) in the specific context of the Bangla language. Bangla script is renowned for its intricate nature, consisting of a total of 49 letters, which include 11 vowels, 38 consonants, and 18 diacritics. The combination of letters in this complex arrangement provides the opportunity to create almost 13,000 unique variations of graphemes, which exceeds the number of graphemic units found in the English language. Our investigation presents a new strategy for ZSL in the context of Bangla OCR. This approach combines generative models with careful labeling techniques to enhance the progress of Bangla OCR, specifically focusing on grapheme categorization. Our goal is to make a substantial impact on the digitalization of educational resources in the Indian subcontinent.

1 Introduction

OCR, a significant technological innovation, has revolutionized the processing and examination of textual content in the contemporary digital age. OCR technology, specifically designed for the purpose of identifying and converting printed or handwritten text into text that can be processed by machines, has facilitated the retrieval, searchability, and manipulation of vast quantities of information across various languages, including Bangla.

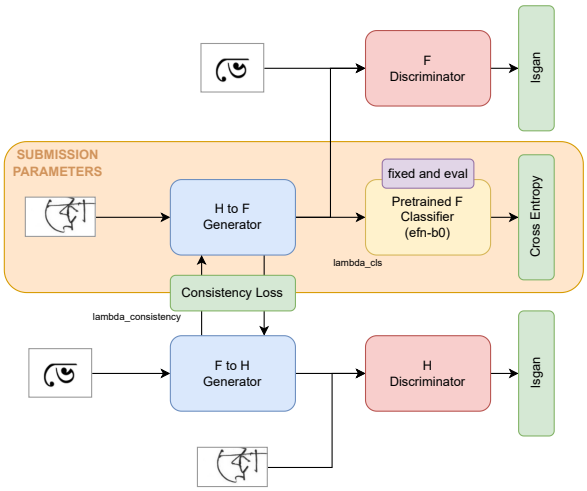

Figure 1: CycleGAN training module. The pre-trained font image classifier keeps the parameters fixed and only conveys gradients to the handwritten (H) to font (F) image generator. Additionally, the H to F image generator incorporates the CycleGAN architecture, enabling more natural generations from handwritten to fonts. *lambda_consistency* is a weight parameter that determines the amount of emphasis placed on loss of the classifier in addition to the loss of CycleGAN while performing zero-shot learning.

Bangla/Bengali has a rich and complicated writing system that makes it hard for OCR systems to read because of its complex ligatures, unique letters, and complicated calligraphy. OCR for Bangla characters aims to bridge the disparity between physical documents and digital databases by offering solutions for activities such as document digitization, language translation, and text analysis. This study is performed on a global AI competition, *Bengali.AI Handwritten Grapheme Classification* hosted by Kaggle and Bengali.AI where our study topped the final leaderboard. The main objective of this research was not exclusively to categorize handwritten characters into predetermined classes, but rather to construct a model with the ability to identify and classify classes that were not explic-

Proceedings of the First Workshop on Bangla Language Processing (BLP-2023), pages 26–33

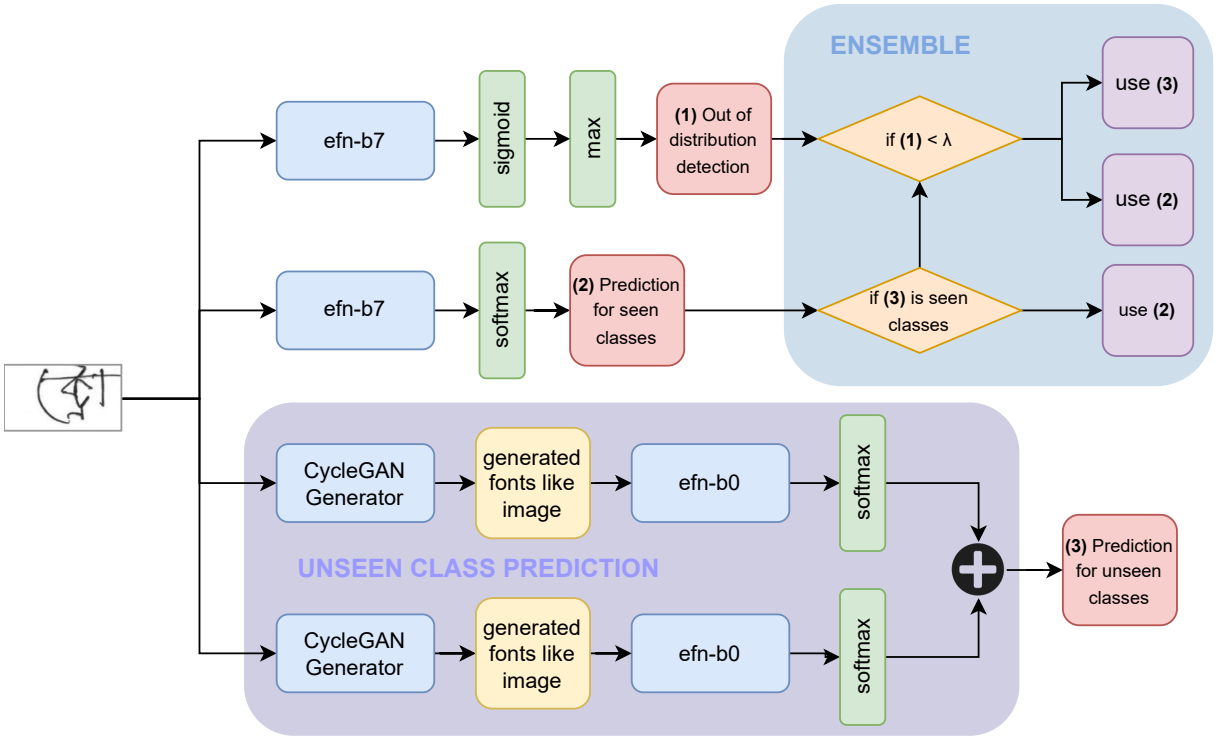

Figure 2: End to end visualization of the proposed architecture which is the top performing solution in the competition. It's created on 3 different models; (1) Out of distribution detection, (2) Seen class model and (3) Unseen class model. EfficientNet-B7 (efn-b7) is utilized as the backbone for Model 1 and 2. Innovative approach to predict unseen class is based on CycleGAN where the backbone is EfficientNet-B0 (efn-b0)

itly provided. Although the first categorization into three categories of components provided a useful foundation, we acknowledged the need for a more efficient method that involved extracting the underlying structures that could potentially arise within a character. In order to accomplish this, we utilized an innovative approach that involved the utilization of a generative model, more specifically a font image generation model based on CycleGAN. This model was employed to convert handwritten characters into images resembling fonts. When incorporated into a larger set of models leading to a handwriting classification system, this generative model produced font images that can be interpreted as intermediate features. The pixel-level representations successfully captured intricate details pertaining to the structure of the character, so effectively abstracting the fundamental qualities associated with the character. The development of this integrated system represents a significant milestone in our research, providing novel insights and enhanced functionalities in the fields of character recognition and classification.

2 Related Works

(Fuad Rezaur Rahman, 1994) introduced a groundbreaking approach that established the basis for Bangla OCR. This approach utilized pattern recognition techniques to accurately recognize handwritten Bangla characters. In a study, (Rahman et al., 2002) introduced a multi-stage recognition system for the identification of handwritten Bangla characters. In this study, the researchers form a cohort of characters and initially identify high-level attributes to classify the characters into groups. Subsequently, they proceed to identify low-level traits in order to accurately recognize the individual characters. (Chowdhury et al., 2002) introduced an initial approach utilizing neural networks for character recognition in printed text data, which was accompanied by some limitations. (Basu et al., 2009) introduces a novel hierarchical methodology for OCR specifically designed for handwritten Bangla words. The proposed approach effectively integrates segmentation and recognition techniques, thereby addressing the inherent difficulties associated with the presence of overlapping characters in the Bangla script. The study utilizes advanced methodologies such as the two-pass approach for

certain sections and MLP-based pattern classifiers, thereby enhancing the precision and comprehensiveness of OCR systems for handwritten Bangla text. A deep neural network (DNN) approach for Bangla OCR in the context of License Plate Recognition (LPR) was proposed by (Onim et al., 2022). In a recent publication by (Emon et al., 2022), a comprehensive analysis of thirteen papers on OCR for the Bangla language was published. The authors reported that the Bidirectional Long Short-Term Memory (BLSTM) model, as proposed by (Paul and Chaudhuri, 2019), demonstrated higher accuracy among the investigated approaches.

3 Dataset

In the realm of modern Bangla literature, a distinct collection of graphemes is frequently utilized, with their recognition being established by transcriptions derived from the Google Bangla ASR dataset as the primary point of reference (Alam et al., 2021). The dataset utilized for this objective is extensive, comprising 127,565 spoken utterances that were transcribed, resulting in a cumulative count of 609,510 words and 2,111,256 graphemes.

The dataset consists of 1,295 frequently used Bangla graphemes based on specific criteria, including occurrence in words and frequency. These graphemes comprise three main components: vowel diacritics, consonant diacritics, and grapheme roots. Vowel diacritics, represented by 11 classes, are typically found at the end of Unicode strings, with a null diacritic for cases of absent vowels. Consonant diacritics, forming diverse combinations, resulted in 8 classes. The remaining grapheme roots, including vowels, consonants, and conjuncts, are limited to 168 classes based on their prevalence in everyday language.

The painstaking compilation of metadata obtained from several sources has been a great resource for conducting comprehensive investigations into the relationship between handwriting and various categories of metadata. It is noteworthy to mention that the metadata pertaining to the training set has been made publicly accessible; however, access to the metadata of the test set can be acquired through a formal request to the authority (Alam et al., 2021).

The dataset has been made available to the public domain as a fundamental element of the *Bengali.AI Handwritten Grapheme Classification* Kaggle competition[1]. In this dataset, a meticulous distribution was implemented, whereby 200,840 samples were assigned to the training set, 98,661 samples were allocated to the public test set, and 112,381 samples were selected for the private test set. Significantly, a rigorous standard was implemented to guarantee the absence of any duplication in contributions within these sets. It is worth mentioning that the graphemes that occur less frequently were predominantly allocated to the private test set, and none of them were incorporated into the training subset. During the duration of the competition, players strive to improve their performance by analyzing the results obtained from the public test set. On the other hand, the outcomes obtained from the private test set are kept undisclosed for every submission and are solely disclosed once the competition is concluded. Significantly, a deliberate decision was made to include 88.4% of the out-of-dictionary (OOD) graphemes in the private test set. This strategic choice was intended to discourage the development of models that rely entirely on public standings from overfitting. The aforementioned strategy functioned as a source of motivation for the participants to devise techniques that are capable of categorizing out-of-distribution graphemes by autonomously identifying the desired variables.

4 Method

Our model classifies against 14784 ($168 \times 11 \times 8$) class which are all the possible combinations. Therefore, we needed to know the combinations of labels made up of grapheme. Prediction of the relationship between the combination of labels and grapheme are done from the label of the given train data. The generation of synthetic data and the conversion of the prediction results into three components are based on this correspondence.

Data splitting process was conducted randomly. As a result, an unintended grapheme root class was generated during the evaluation stage, making it impractical to conduct a thorough examination. Data splitting method presented a significant difficulty due to the considerable computational resources and time investment it demanded. Consequently, the local cross-validation (CV) procedures were implemented utilizing the data in its present condition, notwithstanding the aforementioned constraints. Split counts for train and validation for seen and unseen classes are presented in Figure 3.

[1]https://www.kaggle.com/competitions/bengaliai-cv19/

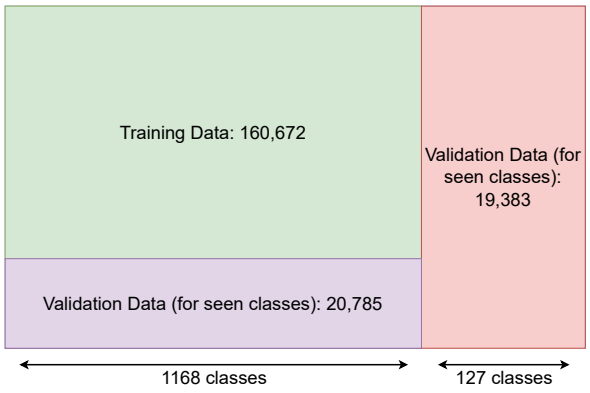

Figure 3: Data split for train data: 160,672. Validation data for seen classes: 20,785 and unseen classes: 19,383.

4.1 Out of Distribution Detection Model

The purpose of the Out of Distribution Detection Model is to categorize input images into either seen or unseen groups. The aforementioned model generates individual confidence scores for each of the 1295 classes in order to make its predictions. In situations where there is a lack of confidence, the image is classified as an unseen class. Conversely, the presence of at least one confidence score signifies that it falls within a seen class. It is worth mentioning that this particular model functions without the need for resizing or cropping input images.

4.2 Seen Class Model

The Seen Class Model has been specifically developed to classify a total of 1295 classes that are included within the training data. The utilized model in this study does not involve any resizing or cropping of input images. Instead, it leverages the AutoAugment Policy specifically designed for the preprocessing of the Street View House Numbers (SVHN) dataset.

4.3 Unseen Class Model

The Unseen Class Model comprises a learning method that consists of two stages. During the initial phase, a classifier is trained to identify images that are produced from TrueType Font (ttf) files. In the subsequent phase, the training process is centered on a generator that is responsible for transforming handwritten characters into synthetic data-like images. In order to facilitate the various stages of learning, the initial step entails the selection of a TrueType font (TTF) and the subsequent generation of a synthetic dataset. The synthetic images are generated to have dimensions that correspond to the training data, particularly 236×137 pixels. The dataset consists of 59,136 samples, each including images created in four distinct sizes: 84×84, 96×96, 108×108, and 120×120 pixels.

4.3.1 Font Classifier Pre-training

Regarding the Font Classifier During the pre-training phase, the images are subjected to several preprocessing operations, such as random affine transformations, random rotation, random cropping, and cutoff, in addition to being cropped and shrunk to dimensions of 224×224 pixels. The work at hand utilizes the EfficientNet-b0 architecture from CNN, while the AdamW optimizer is implemented with default parameter values. The learning rate scheduler utilized in this study is LinearDecay. The output layer is comprised of a Layer Normalization followed by a fully connected layer with dimensions ranging from 2560 to 14784. This is then followed by the application of a Softmax Cross-Entropy activation function. The training process consists of 60 epochs, each utilizing a batch size of 32.

4.3.2 CycleGAN Training

The training method of CycleGAN (Zhu et al., 2017) from scratch[2] comprises the application of a model architecture known as CycleGAN for the purpose of image translation jobs. The input images are subjected to cropping and resizing, resulting in dimensions of 224×224 pixels. These images then undergo preprocessing, which involves random affine transformations, random rotation, and random cropping. However, the dimensions of the random cropping in this phase are reduced compared to the pre-training phase. It is important to note that the cutoff operation is excluded from this preprocessing step. In addition, a pre-trained Font Classifier was incorporated into the model. The parameters of the Font Classifier were kept fixed, and it was operated in evaluation mode.

5 Experiments

Out of distribution detection model utilizes the AutoAugment Policy for preparing the SVHN dataset. The model employed in the study is based on the EfficientNet-b7 (Tan and Le, 2019) architecture, which has been pretrained on the ImageNet dataset. The optimization process leverages the AdamW

[2]https://www.kaggle.com/code/linshokaku/cyclegan-training

optimizer with default parameters. The management of the learning rate scheduling is handled by the WarmUpAndLinearDecay module. The output layer is composed of LayerNorm-FC with dimensions 2560 to 1295, and it utilizes the BCEWithLogitsLoss function. The model undergoes training for a total of 200 epochs, with a batch size of 32. The dataset is divided in a 1:0 ratio, indicating the adoption of a single-fold methodology.

Seen class model employed in this study is based on the EfficientNet-b7 architecture, which has been pretrained on the ImageNet dataset. The optimization process utilizes the AdamW optimizer with the default configuration. The management of the learning rate scheduling is handled by the WarmUpAndLinearDecay module. The output layer is comprised of a Layer Normalization followed by a fully connected layer with dimensions ranging from 2560 to 14784. This is then followed by the application of a Softmax Cross-Entropy activation function. The model is trained for a total of 200 epochs using a batch size of 32. The dataset is divided randomly into a 9:1 ratio, which follows a single-fold methodology for both training and evaluation purposes.

The optimization procedure in CycleGAN training utilizes the Adam optimizer, using a learning rate of 0.0002 and beta values of (0.5, 0.999). The implementation of learning rate scheduling involves the utilization of the LinearDecay method. The training process consists of 40 epochs, where each epoch involves a batch size of 32. The training is performed on a machine configuration consisting of 4 Tesla V100 GPUs, and the entire training process takes approximately 2.5 days to complete. The key hyperparameters of the model consist of *lambda_consistency*, which is set to 10, and *lambda_cls*, which ranges from 1.0 to 5.0. These hyperparameters play a crucial role in determining the performance of the model. The training process plays a crucial role in attaining uniformity and efficacy in tasks related to image translation. The proposed CycleGAN training module is presented in Figure 1.

The leaderboard scores and submissions are assessed using a hierarchical macro-averaged recall (HMAR).

$$HMAR = [(2 * recall_{grapheme\_root}) + recall_{vowel\_diacritic} + recall_{consonant\_diacritic}]/4 \quad (1)$$

Model Architecture	HMAR
SE-ResNeXt50 + Head	0.9584
InceptionResNetv2, SE-ResNeXt101	0.9620
pc-softmax	0.9645
SE-ResNeXt 50 & 101	0.9689
efn-b7, CycleGAN + efn-b0	**0.9762**

Table 1: Outcomes of top 5 submissions in private LB of the competition. Our approach with EfficientNet-B7 (efn-b7), CycleGAN + EfficientNet-B0 (efn-b0) [detailed visualization in Figure 2] scored the highest HMAR. Approaches of LB position 2nd to 5th are mentioned in the Appendix.

For each component (grapheme root, vowel diacritic, or consonant diacritic), a standard macro-averaged recall (MAR) is first calculated. The grapheme root receives double the weight in the final result, which is calculated as the weighted average of those three scores.

6 Results and Discussion

In the domain of handwritten character recognition, ZSL has been a prominent research focus, particularly in the context of Chinese and Japanese character recognition. Many studies have explored the sub-categorization and identification of characters through the manipulation of constituent components.

For Chinese characters (Zhang et al., 2018), which pose a considerable challenge due to their complex grapheme structure, approaches involving the classification of approximately 500 components using recurrent neural networks (RNN) series have been adopted. In contrast, for Bangla characters, an attempt was made to categorize them into three component-based groups, simplifying the multi-class classification process compared to RNN-based methods. However, it became evident that this approach led to significant overfitting in zero-shot recognition, as evidenced by both private validation experiments and competition outcomes. The issue of overfitting in multi-class classification arises from the model's reliance on the entire character for predicting each class, necessitating intricate engineering to dissect the relevant features effectively.

Motivated by the need to address these challenges, an unconventional approach was pursued, where each character was treated as an individual class, even when data for certain classes were scarce. A fundamental assumption underlying this

approach was that if there is a software running on a computer that can recognize a certain character, then that computer is capable of handling the character code and can output it as an image. This assumption was deemed valid within the context of the competition. Several attempts[3] were made to construct models that output characters, and the most successful method among these is presented in this paper. The subsequent sections outline the proposed method and discuss the results obtained.

The findings suggest that the model trained on font images acquired statistically informed, fine-grained character components to efficiently discriminate among font images. Furthermore, the H to F image conversion model appeared to perform the desired transformation, emphasizing recognizable components. This transformation operated on local features of handwriting, with the range of local features being statistically inferred and generalized from the font image-trained model through back-propagation. This generalization facilitated the recognition of zero-shot classes, rendering it feasible. It is worth noting that this technique's strengths extend beyond zero-shot class generalization, encompassing its universal applicability to languages and its straightforward implementation.

Using a dataset composed of 1168 seen classes and 127 unseen classes, the out of distribution detection model results in 0.9967 local CV area under the receiver operating characteristic (AUROC). The CV score achieved for the seen class model is 0.9985, while the Leaderboard score stands at 0.9874. To determine the threshold for this model, a local CV was created weighted to replicate the predicted ratio of seen/unseen classes in the leaderboard. The model's threshold was adopted when this local CV was maximized.

At the initial stage, the loss calculation of the discriminator incorporated a supervised loss obtained from a font classifier that had been pretrained. The configuration that yielded the maximum performance, as indicated by a *lambda_cls* value of 4.0, produced the subsequent CV scores: a local CV score of 0.8804 for previously unobserved classes, and a CV score of 0.9377 for previously observed classes. It is worth noting that the calculation of the MAR involved the exclusion of non-existent classes, hence preventing the assignment of recall values of either 0.0 or 1.0 to these classes.

[3]https://www.kaggle.com/competitions/bengaliai-cv19/discussion/135984

Following that, the hyperparameter modification was conducted, and two further models were trained using different font data, without assessing their local cross-validation performance. These models exhibited a noteworthy achievement on the leaderboard (LB) when compared to the original model, indicating the potential for improved ability to generalize to unfamiliar classes in private data.

In contrast to early hypotheses, the development of synthetic data that closely resembles real samples did not yield the anticipated enhancement in consistency and discriminator losses. Furthermore, this unforeseen inclination did not yield a higher level of generalization towards classes that were not before encountered. During the process of fine-tuning hyperparameters, an observation was made that the model's overall performance exhibited improvement in terms of generalization. However, this improvement was accompanied by a deterioration in the visual quality of the generated images.

The aforementioned observations suggest the potential existence of complex interconnections among hyperparameters, model architecture, and the ability to generalize to novel classes. Consequently, it is imperative to do additional research in order to explore this matter in greater depth.

HMAR scores on the final LB for the top 5 performing architectures are mentioned in Table 1.

7 Conclusion

This work introduces an innovative architecture for Zero-Shot Learning (ZSL) in the context of Optical Character Recognition (OCR), specifically for the complex Bangla script. Our objective was not solely to assign characters to seen classes, but also to enhance our model's ability to identify and classify classes that were unseen. By utilizing the CycleGAN for image synthesis and implementing accurate label mapping techniques, a robust correlation between labels and graphemes has been developed.

By actively engaging in the *Bengali.AI Handwritten Grapheme Classification* competition, we achieved the highest rank on the scoreboard, effectively demonstrating the exceptional capabilities of our novel model. The performance of our system in detecting out-of-distribution instances was exceptional. It achieved an Area Under the Receiver Operating Characteristic (AUROC) score of 0.9967 for a dataset incorporating 1168 seen classes and 127 unseen classes. Additionally, the system demon-

strated a Cross-Validation (CV) score of 0.9985 for the classes it had encountered during training, and a leaderboard score of 0.9874. And finally, it achieved a hierarchical macro-averaged recall (HMAR) score of 0.9762 and topped the leaderboard amongst other contestants. Through our investigation, we have uncovered the intricate relationships among hyperparameters, model architecture, and the ability to generalize to unfamiliar classes. This study represents a noteworthy achievement in the field of character recognition. It is posited that our proposed methodology possesses the capacity to fundamentally transform the field of Bangla OCR development, hence expediting the process of digitizing educational materials in the Indian subcontinent. Furthermore, it is suggested that the prospective uses of this strategy may expand beyond the realm of character recognition. The comprehensive examination of these complex interconnections is important in order to fully realize the potential of our pioneering approach.

Limitations

The process of splitting data for cross-validation of unseen classes is conducted randomly. During the assessment process, it was found that a grapheme root class that did not exist had been generated, which resulted in the inability to conduct a proper evaluation. The rationale for the expansion of the Unseen class using this approach cannot be substantiated. There was an anticipation that the production of images closely resembling the synthetic data would have an effect on the Consistency Loss and the Discriminator, thus leading to an enhancement in generalization for the unseen class. Nevertheless, when adjusting the hyperparameters, it was seen that the overall performance of generalization was enhanced, but at the cost of the images exhibiting an artificial aspect.

Ethics Statement

This study adheres to ethical principles, such as obtaining informed consent, ensuring confidentiality, maintaining integrity, preventing harm, complying with applicable regulations, acknowledging sources, and disclosing conflicts of interest.

Acknowledgements

This project is supported by HKSAR RGC under Grant No. PolyU 15221420. Finally, thanks to the anonymous reviewers for their valuable input.

References

Samiul Alam, Tahsin Reasat, Asif Shahriyar Sushmit, Sadi Mohammad Siddique, Fuad Rahman, Mahady Hasan, and Ahmed Imtiaz Humayun. 2021. A large multi-target dataset of common bengali handwritten graphemes. In *International Conference on Document Analysis and Recognition*, pages 383–398. Springer.

Subhadip Basu, Nibaran Das, Ram Sarkar, Mahantapas Kundu, Mita Nasipuri, and Dipak Kumar Basu. 2009. A hierarchical approach to recognition of handwritten bangla characters. *Pattern Recognition*, 42(7):1467–1484.

Ahmed Asif Chowdhury, Ejaj Ahmed, Shameem Ahmed, Shohrab Hossain, and Chowdhury Mofizur Rahman. 2002. Optical character recognition of bangla characters using neural network: A better approach. In *2nd ICEE*.

Md Imdadul Haque Emon, Khondoker Nazia Iqbal, Md Humaion Kabir Mehedi, Mohammed Julfikar Ali Mahbub, and Annajiat Alim Rasel. 2022. A review of optical character recognition (ocr) techniques on bengali scripts. In *International Conference for Emerging Technologies in Computing*, pages 85–94. Springer.

Ahmad Fuad Rezaur Rahman. 1994. Recognition of bangla hand written characters using pattern recognition techniques.

Md Saif Hassan Onim, Hussain Nyeem, Koushik Roy, Mahmudul Hasan, Abtahi Ishmam, Md Akiful Hoque Akif, and Tareque Bashar Ovi. 2022. Blpnet: A new dnn model and bengali ocr engine for automatic licence plate recognition. *Array*, 15:100244.

Debabrata Paul and Bidyut Baran Chaudhuri. 2019. A blstm network for printed bengali ocr system with high accuracy. *arXiv preprint arXiv:1908.08674*.

Ahmad Fuad Rezaur Rahman, R Rahman, and Michael C Fairhurst. 2002. Recognition of handwritten bengali characters: a novel multistage approach. *Pattern Recognition*, 35(5):997–1006.

Mingxing Tan and Quoc Le. 2019. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR.

Jianshu Zhang, Yixing Zhu, Jun Du, and Lirong Dai. 2018. Trajectory-based radical analysis network for online handwritten chinese character recognition. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 3681–3686. IEEE.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232.

Appendix

SE-ResNeXt50 + Head (5th place solution): This solution involved using SE-ResNeXt50 model with a customized head for improving scores on the LB. Notable improvements were made in consonant diacritic prediction. Preprocessing included image normalization. Architecture had a SE-ResNeXt50 model as a backbone with multiple heads for different tasks. Training used all available data with a cosine annealing schedule and various augmentation techniques. Optimized used: Adam. Loss functions were specified for different tasks, and the final loss combined them. Postprocessing involved a uniform threshold for consonant diacritic prediction. Cosine similarity helped match predictions with training samples. Submissions were selected based on binarization and metric learning criteria, with a focus on LB performance. Overfitting to the public LB was considered, resulting in the selection of the best and a slightly modified submission, both performing well on the private LB.

InceptionResNetv2, SE-ResNeXt101 (4th place solution): The primary objective is to divide a dataset of 1,295 graphemes into In-Dictionary (ID) and Out-of-Dictionary (OOD) categories. The strategy entails training Arcface models to calculate the centroid of each of these 1,295 graphemes' features. The test images are then classified as either ID or OOD based on the shortest distance of a feature to the grapheme centers. The threshold of 0.15 (cosine distance) was estimated locally and contributed to the fourth place submission for the competition.

pc-softmax (3rd place solution): The technique aims to categorize of both seen and unseen graphemes. The methodology involves the utilization of preprocessed images that undergo flipping operations to generate triple identities, which are subsequently standardized to dimensions of 137×236 pixels. The model's architecture consists of two encoders, namely "phalanx" and "earthian," which are subsequently followed by global average pooling, batch normalization, and fully connected layers. The model is trained using the Arcface loss function. To improve performance, a secondary encoder is incorporated, accompanied by augmentations such as cut mix and geometric alterations to promote resilience. The training process involves distinguishing between graphemes that are familiar and those that are unknown. This is achieved by first pretraining on a specific dataset, followed by fine-tuning for the familiar graphemes. Subsequently, additional training is conducted on the original dataset to address the unfamiliar graphemes. The conventional softmax function is substituted with pc-softmax, which utilizes negative log probability as the loss function. The utilization of Stochastic Gradient Descent (SGD) with CosineAnnealing is employed to optimize the model, while Stochastic Weighted Average is utilized to facilitate the training process. The process of inference involves the utilization of Arcface, a technique that computes cosine similarities. The determination of a threshold is based on the minimum similarity between the embeddings of the training and validation datasets. This technique demonstrates a high level of efficacy in addressing grapheme categorization tasks, irrespective of the level of familiarity with the task at hand.

SE-ResNeXt 50 & 101 (2nd place solution): The implemented technique utilized a sequence of strategic modifications to improve the classification of graphemes. The initial approach involved a transition from predicting grapheme components to predicting individual graphemes, hence enabling the implementation of more sophisticated enhancements such as FMix. Post-processing techniques were employed to enhance the accuracy of predictions for both familiar and unfamiliar graphemes. In order to address the issue of overfitting, distinct models were developed for the R and C components, incorporating the utilization of synthetic grapheme creation. The utilization of model blending was of utmost importance, necessitating the implementation of separate methodologies for each individual component. The study employed SE-ResNeXt50 and 101 models, employed different image sizes, and utilized optimization strategies to attain better outcomes.