Bangla Character Similarity Detection

Nabeel Mohammed[1,†] Imranul Ashrafi[2,†]

[†]North South University

Abstract

Bangla is undoubtedly the most widely used language in the Indian Subcontinent. It is one of the rich languages in the world. Researches towards this language have been advancing in recent years. To contribute to the large community of Bangla Natural Language Processing, we propose several methods to detect shape similarity of Bangla characters. We investigate conventional metric based similarity calculation, namely Cosine Similarity and Normalized Cross Correlation. We also explore feature-based analysis using AlexNet and Histogram of Oriented Gradients. The comparison and results of our study yield that using pretrained neural networks perform better, and pretraining a character recognition network in Bangla is essential to further progress in this field of study. These studies should assist in different tasks regarding character and sentence recognition from images and structural analysis of Bangla characters.

1. Introduction

Research in Natural Language Processing regarding Bangla language has not faced improvements until recent years. Being a morphologically rich and complex language, finding patterns and architectures to recognize the unique visual elements of Bangla characters is quite challenging. The closest task which considers the features of characters is Optical Character Recognition (OCR). However, most of the time in OCR, the focus is on the specific task, such as license plate detection[10]. Often, the analysis of morphological features of Bangla characters remain overlooked.

If we observe the example of Bangla characters in figure 1, we can see that there are many similarities between them. When image features are used in a specific task, most of the time, the structural information is dominant. Therefore, using the similarity information and features can be useful in those cases. Also, Bangla characters are complex, and they occupy different regions and shapes[10]. A better understanding of the differences and overlapping characteristics should be advantageous.

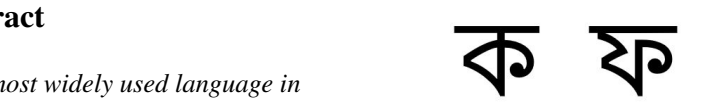

Figure 1: Bangla character examples and similarity

There have been a few previous works to address the issues as mentioned earlier. Hasan *et al.*[3] proposed Convolutional Neural Network with Bidirectional Long Short Term Memory (CNN-BiLSTM) to understand the compound structure of Bangla characters. Paramit *et al.*[7] proposed a shape decomposition-based segmentation technique to reduce the classifier's number of classes. Afroge *et al.* proposed "Discrete Frechet Distance" and "Dynamic Time wrapping" to extract relevant features from images of characters. However, to examine the relationship between characters, we investigate different methods of calculating character similarity to find similar characters.

The rest of this paper is organized in the following fashion. Section 2 describes the methodologies and details about each component. Section 3 illustrates the results and comparative analysis between architectures. And finally, Section 4 and Section 5 describes the recent relevant works and conclusion.

2. Methodology

2.1. Overview

Figure 2 shows the overall structure of our method to calculate similarity. We phrase the problem of finding similar characters to an image similarity problem. Given n number of images $im1, im2, ..., im_n$, we find the similarity score of i^{th} image im_i by comparing it with each of the other $n-1$ images. As shown in the figure, we first extract the features of both images. Then we use the features to calculate similarity score by a metric. In the following subsections, we describe the feature extractors and metrics in detail.

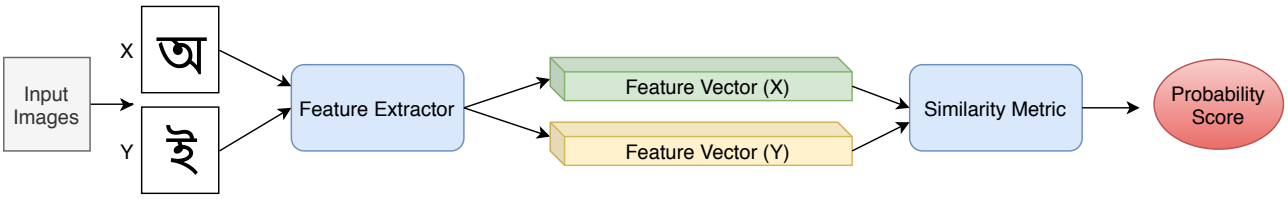

Figure 2: Our approach for detecting character similarity

2.2. AlexNet

AlexNet is a Convolutional Neural Network which was introduced by Krizhevsky *et al.*[4]. It was the winner of the ImageNet Large Scale Visual Recognition Challenge.[9]. AlexNet contains eight layers, five convolutional layers, and three fully connected layers. ReLU activation function was introduced in this architecture and used instead of Tanh function. It also uses overlapping pooling, which reduces the error rate of classification. For our task of image similarity, we take the AlexNet model which is pretrained on ImageNet. According to AlexNet, we resize our image shape to (224,224,3) and extract features from the last fully connected layer of the architecture, which is a vector of shape (1,4096).

2.3. Histogram of Oriented Gradients

The histogram of Oriented Gradients (HOG) is a feature descriptor that is used in computer vision for different image processing tasks[2]. HOG architecture uses a block that slides through the image and calculates the horizontal and vertical gradients of each pixel within the block. After that, it computes the gradient magnitude and gradient angle for each pixel. Finally, it compresses the vectors to a fixed number by constructing a histogram. The HOG descriptor finds structural information of an image by losing all insignificant features. For our task, we take the original image and calculate the HOG features directly.

2.4. Cosine Similarity

Cosine Similarity measures the similarity between two vectors of an inner product space. It measures the cosine of the angle between the input vectors projected in a multidimensional space. It is generally effective since it considers the orientation of the two vectors where Euclidean Distance does not. Given two vectors A and B, the cosine similarity, $cos(\theta)$, is calculated using a dot product and magnitude as follows-

$$CS(A,B) = cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} \tag{1}$$

For our similarity calculation, we either take the one-dimensional vector from the feature vectors or flatten the input image and convert it into a one-dimensional vector.

2.5. Normalized Cross Correlation

Normalized Cross Correlation is used in a broad range of computer vision tasks. It is a simple and effective method to calculate the similarity between two images. However, it is sensitive to rotation and scale changes. Since we are using images of the same scale and rotation, this does not originate any problem. Given two vectors A and B, the Normalized Cross Correlation is calculated as follows-

$$NCC(A,B) = \frac{\sum(A-\overline{A})\sum(B-\overline{B})}{\sqrt{\sum(A-\overline{A})^2}\sqrt{\sum(B-\overline{B})^2}} \tag{2}$$

where $\overline{A}$ and $\overline{B}$ is the mean of two vectors, respectively.

3. Experiments and Analysis

Our task of image similarity for characters is unsupervised since we do not have any ground truth values for the similarities. Therefore, we experiment with different combinations based on feature extractors and metrics. Table 1 describes the experiments we carried out. In the following subsections, we describe each experiment by visualizing the features and results.

Index	Experiment
1	AlexNet + Cosine Similarity
2	AlexNet + Normalized Cross Correlation
3	HOG + Cosine Similarity
4	HOG + Normalized Cross Correlation
5	Cosine Similarity
6	Normalized Cross Correlation

Table 1: Name of experiments carried out

We take two random character images from our 50 characters set to compare among the architectures. We first examine the features obtained from the AlexNet. Since AlexNet is pretrained on ImageNet, it has a certain number of parameters to detect text from images. However, it was not trained on any Bangla text or character dataset. Figure 3 shows the activation feature heatmap of our chosen two characters. Table 2 shows the similarity score between two characters when using AlexNet.

Figure 3: AlexNet feature visualization

Experiment	Score
AlexNet + Cosine Similarity	0.75
AlexNet + Normalized Cross Correlation	0.71

Table 2: Similarity score when using AlexNet

From the figure, we can see that the network was able to capture the structural information decently. The highlighted portions of the images determine the regions upon which the network makes a prediction. When using Cosine Similarity as metric the score yields 0.75, which is similar to the score when using Normalized Cross Correlation, 0.71. However, even though these characters are relatively similar to the human eye, there are portions of the image which do not coincide with each other. Since we are taking the features from the last convolution layer, the small details are missing since the network is not trained on our dataset. Therefore, the similarity scores are much higher.

HOG descriptors are very useful when detecting objects within images. This is because these descriptors are mostly the dominant gradients of each pixel. This property justifies our use case because the images of characters are printed, and there are significant gradient changes around the outline of the characters. Figure 4 shows the visualization of the gradients attained from the HOG descriptors for our chosen two characters. Table 3 shows the similarity score between two characters when using HOG.

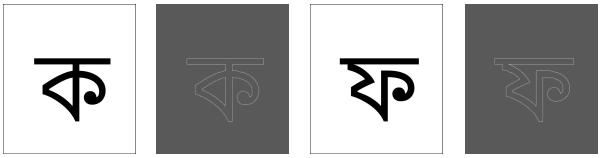

Figure 4: HOG feature descriptor visualization

Experiment	Score
HOG + Cosine Similarity	0.31
HOG + Normalized Cross Correlation	0.30

Table 3: Similarity score when using HOG descriptors

We can observe that the HOG descriptors captured the outline of the characters quite nicely if we look carefully at the figure. Nevertheless, if we look at the scores, we can see that they are reduced significantly to 0.31 when using Cosine Similarity and 0.30 when using Normalized Cross Correlation. This is because the HOG descriptors are sensitive to edges in images. Therefore, the scores calculated here are vaguely dependent on the edges' gradient values rather than the actual shape information. This is almost the opposite scenario of the AlexNet results. For our similarity task, we need both structural and semantic information of the characters.

We also experimented without using any feature vectors to understand the impact of the metrics further. Figure 5 shows the correlation map of the two characters when using the raw images. Table 4 shows the similarity scores when calculating without any feature vector.

Figure 5: Normalized Cross Correlation map

Experiment	Score
Cosine Similarity	0.98
Normalized Cross Correlation	0.76

Table 4: Similarity score without using feature vectors

Observing the correlation map, we can see that it extracts the two images' overlapping portions and excludes the non-matching areas. If we look closely, we can find that some non-overlapping regions are not easily visible to the human eye. When we used AlexNet features, we saw that they were dominant towards the overall structural information of the characters. But the cross correlation map indicates that it successfully detects the non-overlapping regions. The score we obtained when only using cross correlation maps is 0.76, which is very similar to the ones when using AlexNet (0.71). Therefore, it is evident that both AlexNet and the correlation maps extract useful information for our similarity task. However, when using Cosine Similarity without any feature, the score yields 0.98, which is not accurate. We hypothesize that Cosine Similarity is compelling using it together with a feature vector. This is because when using raw images, the similarity metric takes in irrelevant pixel information, which substantially does not contribute to similarity detection.

Figure 6 represents a side by side comparison among the experiments and their predicted three most similar characters.

Index	Experiment	Input	Most Similar
1	AlexNet + Cosine Similarity	ঊ	ঊ: 0.83, ঐ: 0.77, ঙ: 0.73
2	AlexNet + Normalized Cross Correlation	ঊ	ঊ: 0.81, ড: 0.79, ঠ: 0.75
3	HOG + Cosine Similarity	ঊ	ভ: 0.49, ড: 0.34, ড়: 0.34
4	HOG + Normalized Cross Correlation	ঊ	ভ: 0.48, ড: 0.33, ড়: 0.32
5	Cosine Similarity	ঊ	ড: 0.98, ভ: 0.98, ড়: 0.98
6	Normalized Cross Correlation	ঊ	ড: 0.78, ড়: 0.76, ভ: 0.75

Figure 6: Comparison among different models and their predicted most similar characters

From the above comparison, we can understand that most of the models predicted almost the same characters regarding the closest matches. This indicates that our approaches have the potential to detect similar characters. However, in the case of AlexNet based experiments (1 and 2), we can see that they predicted different characters than the rest of the models. Moreover, it also matches the human perspective. It is distinct that deep convolutional features extract different information than others. Since we would be using this similarity information in different predicting tasks, it is vital to understand how different models perceive similarities rather than human beings.

To use the similarity information effectively for any Bangla character related task, similarity scores must be accurate and precise. According to previous researches, it is useful to train a network solely on Bangla characters and alphabets to extract novel features. We believe training a network for Optical Character Recognition in Bangla using a character level language model will bring about improved results.

4. Previous Works

Our fundamental task is to find structural and semantic information regarding Bangla characters. We experimented by extracting similarity information. There have been several tasks regarding Bangla NLP, which focuses on extracting character level information. In the following subsections, we will review some of the notable approaches.

Segmentation is one of the effective approaches to extract information about characters. Zahan *et al.*[12] proposed a segmentation-based approach which divides a character into two regions: upper and lower. Since Bangla characters are complex, this helped in distinguishing characters effectively. Rabby *et al.*[8] proposed BornoNet which is a lightweight CNN to extract features for compound characters. They reported improvements for classifying complex characters. Pervin *et al.*[6] proposed combining two feature vectors, namely Zoning and Gabor filter. The combined feature vector is then used for the classification task.

Detecting handwritten characters properly is often a difficult task since they vary from person to person. Singh *et al.*[11] proposed a skip connected multi-column convolutional network. Essentially it extracts global and local features by using different filters and combines them to produce the final feature descriptor. Sen *et al.* proposed a simple approach of splitting a character into different regions and taking the distance between them. These split distances are finally used as features for the specific task. Zunair *et al.*[13] proposed an unconventional approach of finetuning VGG16 network. They show that using different types of augmentation and parameter tuning produces higher accuracy on the OCR task.

Our task involves finding image similarity. The metrics we used also have been used for different similarity measure tasks. Buniatyan *et al.*[1] proposed template matching architecture, which uses Normalized Cross Correlation and Siamese convolutional network. Mirzaei *et al.*[5] recently proposed a novel approach to estimate tissue displacement in quasi-static elastograph using Normalized Cross Correlation.

5. Conclusion

In this work, we have approached the problem of finding Bangla character feature information by calculating the similarity among characters. We found that using deep convolutional pretrained network produces better semantic descriptors than others. We also acknowledge that training a network solely on Bangla dataset should achieve much higher accuracy when producing useful features.

References

[1] D. Buniatyan, T. Macrina, D. Ih, J. Zung, and H. S. Seung. Deep learning improves template matching by normalized cross correlation. *arXiv preprint arXiv:1705.08593*, 2017.

[2] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893 vol. 1, 2005.

[3] M. J. Hasan, M. F. Wahid, and M. S. Alom. Bangla compound character recognition by combining deep convo-

lutional neural network with bidirectional long short-term memory. In *2019 4th International Conference on Electrical Information and Communication Technology (EICT)*, pages 1–4, 2019.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, page 1097–1105, Red Hook, NY, USA, 2012. Curran Associates Inc.

[5] M. Mirzaei, A. Asif, M. Fortin, and H. Rivaz. 3d normalized cross-correlation for estimation of the displacement field in ultrasound elastography. *Ultrasonics*, 102:106053, 2020.

[6] M. T. Pervin, S. Afroge, and A. Huq. A feature fusion based optical character recognition of bangla characters using support vector machine. In *2017 3rd International Conference on Electrical Information and Communication Technology (EICT)*, pages 1–6. IEEE, 2017.

[7] R. Pramanik and S. Bag. Shape decomposition-based handwritten compound character recognition for bangla ocr. *Journal of Visual Communication and Image Representation*, 50:123–134, 2018.

[8] A. S. A. Rabby, S. Haque, S. Islam, S. Abujar, and S. A. Hossain. Bornonet: Bangla handwritten characters recognition using convolutional neural network. *Procedia computer science*, 143:528–535, 2018.

[9] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[10] N. Saif, N. Ahmmed, S. Pasha, M. S. K. Shahrin, M. M. Hasan, S. Islam, and A. S. M. M. Jameel. Automatic license plate recognition system for bangla license plates using convolutional neural network. In *TENCON 2019 - 2019 IEEE Region 10 Conference (TENCON)*, pages 925–930, 2019.

[11] A. Singh, R. Sarkhel, N. Das, M. Kundu, and M. Nasipuri. A skip-connected multi-column network for isolated handwritten bangla character and digit recognition. *arXiv*, pages arXiv–2004, 2020.

[12] T. Zahan, M. Zafar Iqbal, M. Reza Selim, and M. Shahidur Rahman. Connected component analysis based two zone approach for bangla character segmentation. In *2018 International Conference on Bangla Speech and Language Processing (ICBSLP)*, pages 1–4, 2018.

[13] H. Zunair, N. Mohammed, and S. Momen. Unconventional wisdom: A new transfer learning approach applied to bengali numeral classification. In *2018 International Conference on Bangla Speech and Language Processing (ICBSLP)*, pages 1–6, 2018.