2022 25th International Conference on Computer and Information Technology (ICCIT)
17-19 December, Cox's Bazar, Bangladesh

ComNet: A Deep Convolutional Neural Network Capable of Classifying Compound Bengali Handwritten Characters High Number of Classes in a Data-Scarce State

Md. Mehedi Hasan*, Md. Ali Hossain†, Azmain Yakin Srizon‡ and Abu Sayeed§
Department of Computer Science & Engineering
Rajshahi University of Engineering & Technology, Rajshahi, Bangladesh
Emails: *mmehedihasann@gmail.com, †ali.ruet@gmail.com, ‡azmainsrizon@gmail.com, §abusayeed.cse@gmail.com

***Abstract*—Handwritten character recognition has been a well-studied problem for decades among researchers. Even though there exist character recognition systems for some non-cursive languages like English, this hasn't been as much successful for cursive languages yet. Bengali, being a cursive language, has another complication up its sleeve which is numerous number of compound characters. These facts make it very difficult to classify accurately even for a modern Deep Convolutional Neural Network (CNN). Deep CNNs thrive when plenty of data are available to optimize their parameters and don't perform so well in a data-scarce situation. In this paper, we studied both the aspects, of data scarcity and the cursive compound nature of the Bengali language. We created a Deep CNN called ComNet with EfficientNetB7 and an additional regularization layer, which achieved an accuracy of 92.42% and outperformed baseline DenseNet201 architecture which managed only 79.42% accuracy on our in-house 267 class augmented compound handwritten character dataset called Juktakkhor. Without augmentation, deep learning algorithms suffer. ComNet also suffers since the dataset has a large number of classes with quite a low number of data per class. ComNet achieved 41.16% accuracy on the non-augmented Juktakkhor dataset, while baseline DenseNet201 achieved 39.89%.**

***Index Terms*—Handwritten compound Bengali character, Deep CNN, Data Scarcity**

I. Introduction

Handwritten character recognition is a mature problem in the field of computer science because of its use in Optical Character Recognition (OCR) problems [1]. Even being studied for many decades, this problem isn't yet solved for various difficulties associated with it. One of them is the cursive nature of some languages like Bengali, which creates a whole new problem on its own. On top of it, some characters have more than one form which is another challenge.

With the emergence of deep learning and computational hardware availability for example graphics processing units and powerful processors, these types of image processing problems have been tackled via Deep Learning nowadays. As a result, we can see some tremendous improvement in computer vision tasks like classification, object detection, segmentation etc., and sometimes with higher accuracy than human beings which is particularly astounding [2].

For a few decades, handwritten character recognition received a lot of attention, and it still does. The most difficult aspect of the Bengali handwriting optical character recognition system is the recognition of consonant conjuncts or compound characters, which account for 85% of all the characters used in Bengali literature. Contributions in support of Bengali compound characters have not before garnered particularly noteworthy contributions. To identify the compound characters, considerable research has been done using the CMATERdb dataset.

In order to report overall accuracy in 2014, a convex hull and quadtree-based feature set were used [3]. Low performance was seen in [3] due to improper feature extraction, though. The Support Vector Machine (SVM) classifier was introduced in 2016, which improved performance but was unable to reach near-accurate recognition due to the vast volume of multiclass data [4]. Deep convolutional neural networks (CNNs) were used in research that was undertaken in 2017 [5], [6]. However, the improper implementation of convolution in [5] and [6] prevented the predicted performance.

Convolutional and hybrid-hog neural networks both performed similarly in a research that was proposed in 2018 [7]. Although the suggested strategy in [7] was quicker, the performance was not better. A paper in 2019 used transfer learning, such as DenseNets, however the lack of hyperparameter adjustment led to an even lower recognition rate [8]. The two most recent experiments [9], [10] found that deep CNNs performed admirably. However, the performance was still lacking because neither [9] nor [10] addressed all of the literary complex characters.

Recently researchers used Squeeze and excitation-based ResNext model to classify handwritten Bengali compound character dataset and achieved 99.82% accuracy on Mendeley compound character dataset [11]. Though the performance is praiseworthy, the model is used on a very small dataset with relatively clear. Sayeed et. al used a low-cost deep cnn to recognize handwritten Bangla characters from the different publicly available datasets and achieved 96-99% accuracy [12]. But they didn't use a data-scarce setting.

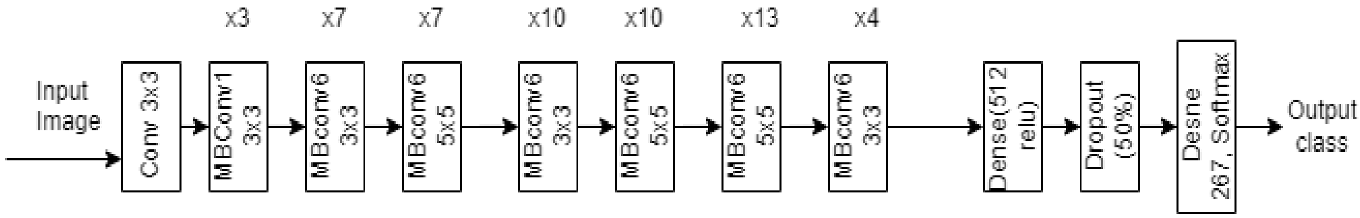

Fig. 1: ComNet Architecture

TABLE I: Image augmentation parameters

Parameter Name	Value
Random distortion	Probability = 1.0, grid height = 3, grid width = 3, magnitude = 5
Rotate	Probability = 0.7, max left rotation=3, max right rotation=3
Zoom	Probability = 0.5, min factor=1.1, max factor=1.5
Resize	Probability = 1.0, height = 224, width = 224

TABLE II: ComNet hyperparameters

Hyperparameter Name	Value
Batch size	Train = 6, Validation = 3, Test = 9
Image size	224x224x3
Optimizer	Adam
Learning Rate	0.00001
Loss	Categorical Crossentropy
Epoch	50

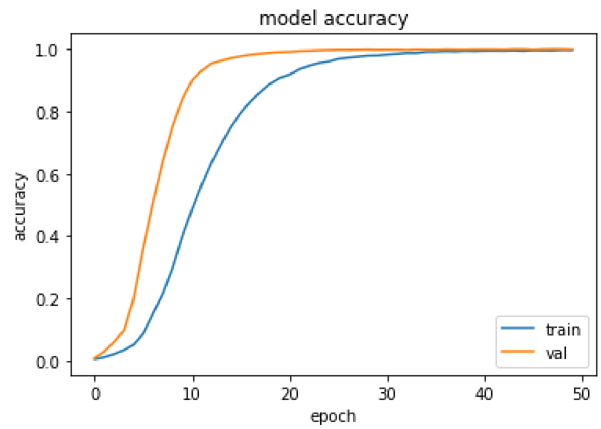

Fig. 2: Accuracy curve of train vs validation

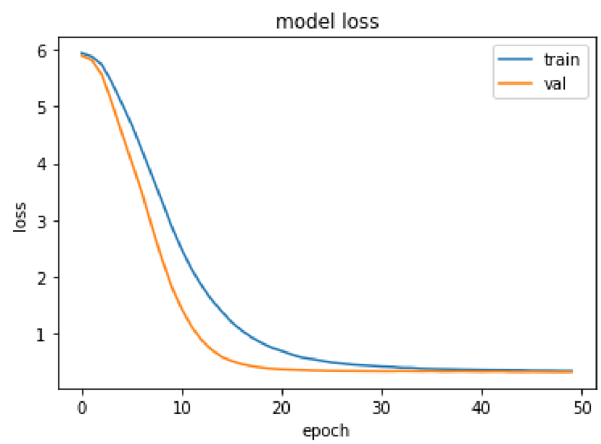

Fig. 3: Loss curve of train vs validation

II. Methodology

A. Dataset Collection

We have collected one of the biggest datasets regarding the number of classes for Bengali handwritten compound characters. It contains 267 classes, having 14 images per class on average, totaling 3846 images. The dataset was scanned from paper form and preprocessed to reduce unnecessary blank spaces. Then the dataset was partitioned into test and train data. 20% of all data were selected as test data and the rest of 80% of data remains as train data. Since the training data is quite low per class, the images were augmented and 50 images per class were generated. The parameters used for augmentation are listed in Table I. Among those 50 images, 40 were used for training purposes and the rest 10 images were used as validation data. The dataset can be found at https://bit.ly/3VvyUBS

B. Transfer Learning

Transfer learning is a machine learning approach in which a model generated for one job is utilized as the foundation for a model on a different task. Given the extensive computing and time resources needed to design neural network models on these concerns and the huge jumps in the skill that they provide on related problems, it is a popular approach in deep learning where pre-trained models are used as the starting point on computer vision and natural language processing tasks.

To put it another way, in transfer learning, we train a base network first on a base dataset and task, and then we transfer the acquired features to a second target network to be trained on a target dataset and task. If the traits are general—that is, applicable to both the base task and the target task—rather than task-specific, this procedure is more likely to succeed. Inductive transfer is the name of the type of transfer learning that deep learning uses. This is where employing a model fit on a distinct but related job helps to advantageously reduce the range of potential models (model bias).

C. EfficientNet

EfficientNet is a convolutional neural network design and scaling technique that uses a compound coefficient to consis-

TABLE III: Classwise accuracy

Class	Accuracy	Class	Accuracy	Class	Accuracy	Class	Accuracy
0	100.0	169	66.67	235	66.67	38	100.0
1	100.0	17	66.67	236	100.0	39	100.0
10	100.0	170	100.0	237	100.0	4	66.67
100	100.0	172	100.0	238	100.0	40	100.0
101	66.67	173	100.0	239	66.67	41	66.67
102	100.0	174	33.33	24	100.0	42	100.0
103	100.00	175	100.0	240	100.0	43	100.0
104	100.00	176	100.0	241	100.0	44	100.0
105	66.67	177	66.67	242	100.0	45	100.0
106	100.0	178	100.0	243	100.0	46	100.0
107	100.0	179	100.0	244	100.0	48	100.0
108	100.0	18	66.67	245	66.67	49	100.0
109	100.0	181	100.0	246	100.0	5	100.0
11	100.0	182	100.0	247	33.33	50	100.0
110	66.67	183	66.67	248	100.0	52	66.67
111	100.0	184	100.0	249	100.0	53	100.0
112	100.0	186	100.0	25	100.0	54	100.0
113	66.67	187	100.0	250	100.0	55	100.0
114	66.67	188	100.0	251	100.0	56	100.0
115	100.0	189	100.0	252	100.0	58	66.67
116	66.67	19	66.67	253	100.0	59	100.0
117	66.67	190	100.0	254	100.0	6	100.0
118	100.0	191	66.67	255	100.0	60	100.0
119	100.0	192	100.0	256	100.0	61	66.67
12	100.0	193	100.0	257	100.0	62	100.0
120	100.0	194	100.0	258	100.0	63	66.67
122	100.0	195	66.67	259	100.0	64	66.67
123	100.0	196	100.0	26	100.0	65	100.0
124	100.0	197	100.0	260	33.33	66	66.67
125	100.0	198	100.0	262	100.0	67	100.0
126	100.0	2	100.0	265	66.67	68	100.0
127	100.0	20	66.67	266	100.0	69	100.0
128	100.0	200	100.0	267	66.67	7	100.0
13	66.67	201	100.0	268	100.0	70	100.0
131	100.0	202	100.0	269	66.67	71	100.0
132	133.33	203	100.0	27	100.0	72	100.0
133	166.67	204	100.0	270	66.67	73	100.0
134	100.0	205	100.0	271	66.67	74	100.0
135	100.0	206	100.0	272	100.0	75	100.0
136	100.0	207	100.0	273	100.0	76	66.67
137	66.67	208	66.67	274	66.67	77	100.0
138	66.67	209	100.0	275	100.0	79	33.33
139	66.67	21	100.0	276	100.0	8	100.0
14	100.0	210	100.0	277	100.0	80	100.0
140	33.33	211	100.0	278	100.0	81	100.0
141	100.0	212	100.0	279	100.0	82	100.0
142	100.0	213	66.67	28	66.67	83	100.0
143	66.67	214	100.0	280	100.0	84	100.0
144	100.0	215	66.67	282	100.0	85	100.0
145	100.0	216	100.0	283	100.0	86	100.0
146	100.0	217	100.0	284	100.0	87	100.0
147	100.0	218	100.0	285	100.0	88	100.0
148	66.67	219	100.0	286	100.0	89	100.0
149	100.0	22	66.67	287	100.0	9	66.67
15	66.67	220	66.67	288	100.0	90	66.67
150	100.0	221	100.0	289	100.0	91	100.0
151	100.0	222	100.0	29	100.0	92	100.0
153	100.0	223	100.0	290	100.0	93	100.0
154	100.0	225	100.0	291	100.0	94	100.0
158	100.0	227	66.67	292	100.0	95	100.0
159	100.0	228	100.0	293	100.0	97	100.0
16	66.67	229	100.0	3	100.0	98	100.0
160	66.67	23	66.67	31	66.67	99	100.0
161	100.0	230	100.0	32	66.67		
162	66.67	231	100.0	33	100.0		
163	100.0	232	100.0	34	66.67		
164	100.0	233	100.0	35	33.33		
167	100.0	234	100.0	37	100.0		

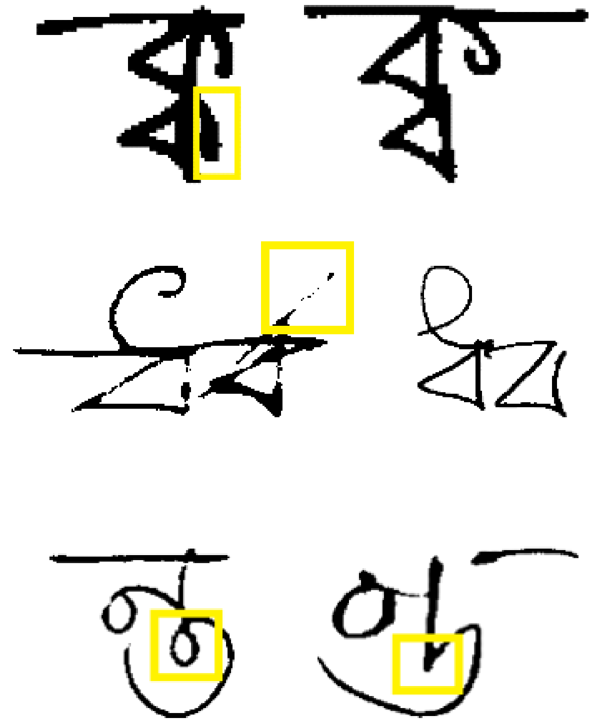

Fig. 4: Examaple of misclassification

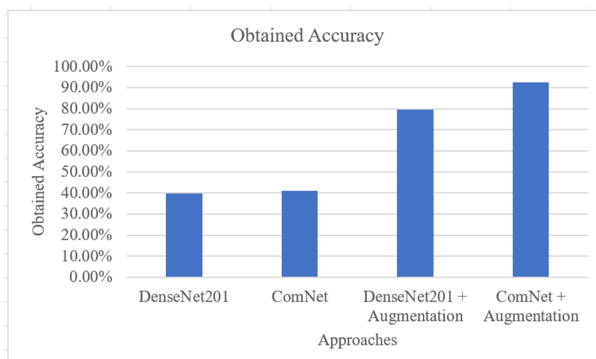

Fig. 5: Accuracy comparison among different approaches and proposed method.

tently scale all depth, breadth, and resolution parameters. The EfficientNet scaling approach evenly scales network breadth, depth, and resolution using a set of preset scaling coefficients, in contrast to standard practice that scales these variables freely. The rationale behind the compound scaling approach is that larger input images require more layers in order to expand the network's receptive area and more channels in order to catch more fine-grained patterns on the larger picture.

In addition to squeeze-and-excitation blocks, the foundational EfficientNet-B0 network is built upon the MobileNetV2 inverted bottleneck residual blocks. On the CIFAR-100 (91.7%), Flowers (98.8%), and 3 additional transfer learning datasets, EfficientNets likewise transfer well and reach state-of-the-art accuracy while using orders of magnitude fewer parameters. In this study, EfficientNetB7 has been used.

D. Experimental Setup

The experiment was conducted on a Kaggle instance with a 4 Core Intel Xeon CPU, 16 GB of RAM, 13 GB of VRAM from a Tesla P100 GPU.

E. ComNet Architecture

The proposed ComNet architecture is a combination of EfficientNetB7 and a regularization layer (using dropout of 50%) and the final softmax layer. EfficientNetB7 was used as a feature extractor from the images and the regularization layer was used to control the overfitting of the model. All the parameters of EfficientNetB7 were set to be trainable, which lets the parameters update themselves to adjust to the specific problem we were dealing with. EfficientNet is a deep learning algorithm that is designed to be efficient at scale-up with a compound coefficient value. This model achieves this via depth, width, and even resolution all at once which makes this model stand out from the rest of the models. Other models either use depth, width or resolution, or a combination of these but not all at once to scale up. But EfficientNet uses all of these which makes it very efficient and hence the name. For these properties, we have selected EfficientNet and in particular, EfficientNetB7 which is the top-of-the-line model in the family of EfficientNets regarding accuracy. Since the output of the EfficientNet is a large feature vector, we used a dropout layer as a regularization measure to prevent the model from being overfitted during the training procedure. To train the proposed ComNet model, hyperparameters from Table II have been used. We achieved an overall accuracy of 92.42%. The classwise accuracy is listed in Table III.

III. Result and Analysis

After 50 epochs, the ComNet model has achieved 92.42% accuracy on the test set beating DenseNet201 by a large margin. The classwise accuracy of ComNet architecture is presented in Table III. We can clearly see that the architecture performed really well in most of the classes achieving 100% accuracy on the unseen data. But for some classes, the model misclassified 1 or 2 samples. The main reason behind this misclassification is the striking similarity between different classes which makes it harder to distinguish. Figure 5 showcases the comparison between state-of-the-art methods and proposed approach.

IV. Limitations and Future Works

The dataset has a significantly low number of images per class. Any deep learning model will struggle to accommodate such a large quantity of classes with a such low number of images. This makes the dataset interesting as well as challenging. Some characters are very similar which creates another challenge for any model to differentiate accurately. As a result, misclassification occurs quite frequently. These challenges needs to be addressed in the upcoming works.

V. Conclusion

Handwritten compound character recognition is quite a challenge and researchers have been working on this for quite some time now. Even then, this problem has not been solved due to the cursive nature of some languages like Bengali. And if the problem is associated with data scarcity, the combination makes it a tough one. In this paper, we have introduced a dataset for handwritten compound character classification and a deep learning model named ComNet which achieves 92.41% on this dataset beating DenseNet201 by a long margin. Since this problem is a complex one, many more aspects are yet to be explored.

References

[1] J. Memon, M. Sami, R. A. Khan, M. Uddin, ‘Handwritten optical character recognition (OCR): A comprehensive systematic literature review (SLR)’, IEEE Access, 8, 142642–142668, 2020.

[2] K. He, X. Zhang, S. Ren, J. Sun, ‘Delving deep into rectifiers: Surpassing human-level performance on imagenet classification’, In Proceedings of the IEEE international conference on computer vision, 2015, pp. 1026–1034.

[3] N. Das, K. Acharya, R. Sarkar, S. Basu, M. Kundu and M. Nasipuri, ”A benchmark image database of isolated Bangla handwritten compound characters,” International Journal on Document Analysis and Recognition (IJDAR), vol. 17, no. 4, pp. 413-431, April 2014.

[4] R. Sarkhel, N. Das, A. K. Saha and M. Nasipuri, ”A multi-objective approach towards cost effective isolated handwritten Bangla character and digit recognition,” Pattern Recognition, vol. 58, no. 4, pp. 172-189, April 2016.

[5] S. Roy, N. Das, M. Kundu and M. Nasipuri, ”Handwritten isolated Bangla compound character recognition: A new benchmark using a novel deep learning approach,” Pattern Recognition Letters, vol. 90, no. 3, pp. 15-21, March 2017.

[6] A. Ashiquzzaman, A. K. Tushar, S. Dutta and F. Mohsin, ”An efficient method for improving classification accuracy of handwritten Bangla compound characters using DCNN with dropout and ELU,” 2017 Third International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN), pp. 147-152, Kolkata, West Bengal, India, 2017.

[7] S. M. A. Sharif, N. Mohammed, S. Momen and N. Mansoor, ”Classification of bangla compound characters using a hog-cnn hybrid model,” Proceedings of the International Conference on Computing and Communication Systems, pp. 403-411, Singapore, 2018.

[8] M. M. Hasan, M. M. Abir, M. Ibrahim, M. Sayem and S. Abdullah, ”AIBangla: A Benchmark Dataset for Isolated Bangla Handwritten Basic and Compound Character Recognition,” 2019 International Conference on Bangla Speech and Language Processing (ICBSLP), pp. 1-6, Sylhet, Bangladesh, 2019.

[9] A. Fardous and S. Afroge, ”Handwritten isolated Bangla compound character recognition,” 2019 International Conference on Electrical, Computer and Communication Engineering (ECCE), pp. 1-5, Coxs Bazar, Chittagong, Bangladesh, 2019.

[10] P. Keserwani, T. Ali and P. P. Roy, ”Handwritten Bangla character and numeral recognition using convolutional neural network for low-memory GPU,” International Journal of Machine Learning and Cybernetics, vol. 10, no. 12, pp. 3485-3497, February 2019.

[11] M. M. Khan, M. S. Uddin, M. Z. Parvez, L. Nahar, ‘A squeeze and excitation ResNeXt-based deep learning model for Bangla handwritten compound character recognition’, Journal of King Saud University-Computer and Information Sciences, vol. 34, no. 6, pp. 3356–3364, 2022.

[12] A. Sayeed, J. Shin, M. A. M. Hasan, A. Y. Srizon, M. M. Hasan, ‘BengaliNet: A Low-Cost Novel Convolutional Neural Network for Bengali Handwritten Characters Recognition’, Applied Sciences, vol. 11, no. 15, 2021.