North East University Bangladesh
Department of Computer Science and Engineering

Bangla Isolated Handwritten Characters Recognition Using DenseNet

By
Prince Chakrabarty
Reg. No:
200103020033
BSc(Engg.) in CSE
4th year 2nd semester

Supervised By
Ayon Dey
Lecturer
Department of Computer Science and Engineering

21th January, 2024

Bangla Isolated Handwritten Characters Recognition Using DenseNet

A Project submitted to the Department of Computer Science and Engineering,
North East University Bangladesh, in partial fulfillment of the requirements
for the degree of Bachelor of Science in Computer Science and Engineering

By

Prince Chakrabarty

Reg. No:
200103020033

BSc(Engg.) in CSE

4th year 2nd semester

Supervised By

Ayon Dey

Lecturer

Department of Computer Science and Engineering

21th January, 2024

Recommendation Letter from Project Supervisor

These Students, *Prince Chakrabarty*, whose project entitled *"**Bangla Isolated Handwritten Characters Recognition Using DenseNet**"*, is under my supervision and agrees to submit for examination.

Signature of the Supervisor :

Ayon Dey
Lecturer
Department of Computer Science and Engineering
North East University Bangladesh

Qualification Form of BSc(Engg.) Degree

Student Name : Prince Chakrabarty
Project Title : Bangla Isolated Handwritten Characters Recognition Using DenseNet

This is to certify that the project is submitted by the student named above in January 21, 2024. It is qualified and approved by the following persons and committee.

Head of the Dept.	**Supervisor**
Rathindra Chandra Gope	Ayon Dey
Associate Professor	Lecturer
Department of CSE	Department of CSE
North East University Bangladesh	North East University Bangladesh

Abstract

The necessity of recognizing Bangla handwritten characters is increasing day by day because of its various applications. The main objective of this report is to provide a sophisticated, effective and efficient way to classify Bangla handwritten characters such as number and vowel. We tried different network architecture like CNN, ResNet and DenseNet. Out of all this different model we choose Dense Convolution Network (DenseNet) because previously some research has shown that convolutional network can be significantly deeper, efficient to train, and more accurate if they contain shorter connections between layers close to the input and output. DenseNet connects each layer to every other layer in a feed-forward way. It has several advantages: reducing vanishing-gradient problem and number of parameters, feature re-using technique, and achieving higher accuracy. We evaluate our proposed architecture on a dataset called BanglaLekha-Isolated where we tested on bangla number(০-৯) and bangla vowel(অ–ঔ). In every class we got higher accuracy with lower error rate.

Keywords: CNN, ResNet, DenseNet, layers, input, output, feed-forward, vanishing-gradient, parameters, feature, BanglaLekha-Isolated, bangla number, bangla vowel, error rate.

Table of Contents

Abstract i

Table of Contents 1

List of Figures 2

List of tables 3

Introduction 4

Background Study 5

- Literature review 5
- Previous Work 7

Model 10

Experiment 12

- DataSet 12
- Training 13
- Classification Result 16
- User Interface 17

Discussion 18

Conclusion 18

References 19

List of Figures

Figure 1: CNN Architecture 7

Figure 2: Residual learning: a building block 8

Figure 3: A ResNet architecture 8

Figure 4: DenseNet with 3 dense blocks. The layers between 2 adjacent blocks are referred to as transition layers and change feature-map size via convolution and pooling 9

Figure 5: A 5-layer dense block with a growth rate 4. Each layer takes all preceding feature-maps as input 10

Figure 6: Dataset sample 12

Figure 7: Image preprocessing technique 13

Figure 8: Training parameters of DenseNet 13

Figure 9: Training and Testing accuracy 14

Figure 10: Training, Validation accuracy and loss graph 15

Figure 11: Classification Result 16

Figure 12: Mobile interface, (a) home page, (b) testing data, (c) and (d) classification result 17

List of Tables

Table 1: DenseNet-121 architecture 11

Introduction

Bangla is the 2nd most popular language in Indian sub-continent and being the language of around 300 million all over the world. It is also the first language of the people of Bangladesh which is also part of the greater Indo-European language family, originating its primary roots from Sanskrit and kept evolving by the absorption of foreign words over thousands of years since its existence. Bangla language holds 50 basic alphabets where it contains 11 vowel, 39 consonants, many compound characters and punctuation marks, and 10 number.

Optical character recognition or (OCR) is a technique to recognize different character from both printed and handwritten format to convert them into machine readable text format. Handwritten character identification is a complex task due to its different size and shape which vary from person to person, compare to printed character. Many research has done for English and other languages, however few research has taken place for Bangla handwriting recognition because of its complex size and shape.

We tested different bangla handwriting dataset on different network architecture such as CNN, ResNet and DenseNet. Among them we choose DenseNet because of its different structural design and many advantages. It combine features by concatenating them means it adds all previous feature-maps to the next layer and every layer is connected to each other. Along with better parameter efficiency, one big advantage of DenseNets is their improved flow of information throughout the network, which makes them easy to train. We also observe that dense connections have a regularizing effect, which reduces over-fitting on task with smaller training set sizes. We used gray-scaled bangla character images for our model and done per-processing techniques for better accuracy.

Background Study

Literature Review

Here we conclude some of the research paper we followed which helped us to complete our project.

The paper [1] discusses a Dense Convolutional Networks (DenseNet), emphasizing their effectiveness in training deep and accurate convolutional networks. DenseNet introduces dense connections, where each layer is connected to every other layer in a feed-forward manner, creating L(L+1)/2 direct connections in a network with L layers. This architecture addresses issues like the vanishing-gradient problem, enhances feature propagation, encourages feature reuse, and reduces the number of parameters. The DenseNet model is evaluated on various object recognition benchmarks, showing significant improvements over state-of-the-art methods on tasks like CIFAR-10, CIFAR-100, SVHN, and ImageNet, all while less computation for high performance.

In this paper [2], the author propose a convolutional deep model to recognize Bengali handwritten characters. They first learnt a useful set of features by using kernels and local receptive fields, and then they have employed densely connected layers for the discrimination task. Their system has been tested on BanglaLekha-Isolated dataset. It achieves 98.66% accuracy on numerals (10 character classes), 94.99% accuracy on vowels (11 character classes), 91.60% accuracy on compound letters (20 character classes), 91.23% accuracy on alphabets (50 character classes), and 89.93% accuracy on almost all Bengali characters (80 character classes). Most of the errors incurred by their model in recognition task are due to extreme proximity in shapes among characters. A significant number of errors was caused by the mislabeled, irrecoverably distorted, and illegal data examples.

Deeper neural networks are more difficult to train. The author [3] present a residual learning framework to ease the training of networks that are substantially deeper than those used previously. Author explicitly reformulate the layers as learning residual functions with reference to the layer inputs, instead of learning unreferenced functions. Author provide comprehensive empirical evidence showing that these residual networks are easier to optimize, and can gain accuracy from considerably increased depth. An ensemble of these residual nets achieves 3.57% error on the ImageNet test set. This result won the 1st place on the ILSVRC 2015 classification task. Author also present analysis on CIFAR-10 with 100 and 1000 layers. The depth of representations is of central importance for many
visual recognition tasks. Solely due to the extremely deep representations, author obtain a 28% relative improvement on the COCO object detection dataset. Deep residual nets are foundations of submissions to ILSVRC & COCO 2015 competitions1, where author also won the 1st places on the tasks of ImageNet detection, ImageNet localization, COCO detection, and COCO segmentation.

This paper [4] proposes a process of Handwritten Character Recognition to recognize and convert images of individual Bangla handwritten characters into electronically editable format, which will create opportunities for further research and can also have various practical applications. The dataset used in this experiment is the BanglaLekha-Isolated dataset. Using Convolutional Neural Network, this model achieves 91.81% accuracy on the alphabets (50 character classes) on the base dataset, and after expanding the number of images to 200,000 using data augmentation, the accuracy achieved on the test set is 95.25%. The model was hosted on a web server for the ease of testing and interaction with the model. Furthermore, a comparison with other machine learning approaches is presented.

The objective of this paper [5] is to provide a sophisticated, effective and efficient way to recognize and classify Bangla handwritten characters. Here an extended convolutional neural network (CNN) model has been proposed to recognize Bangla handwritten characters. their CNN model is tested on “BanglalLekha-Isolated” dataset where there are 10 classes for digits, 11 classes for vowels and 39 classes for consonants. Their model shows accuracy of recognition as: 99.50% for Bangla digits, 93.18% for vowels, 90.00% for consonants and 92.25% for combined classes.

A simple, lightweight CNN model has been proposed in this paper [6] for classifying Bangla Handwriting Character, which contains 50 basic Bangla characters (11 vowels and 39 consonants). Experiments have been made on three datasets along with the BanglaLekha-Isolated [1] CMATERdb and the ISI dataset. For character recognition, the proposed BornoNet model gets 98%, 96.81%, 95.71%, and 96.40% validation accuracy respectively for CMATERdb, ISI, BanglaLekha-Isolated dataset and mixed dataset. Also proposed model was trained with one dataset and cross-validated with other two datasets. Proposed model achieved the best accuracy rate so far for BanglaLekha-Isolated, CMATERdb and ISI datasets.

In this paper [7] convolutional neural network (CNN) based model has been developed to recognize handwritten isolated Bangla compound characters. The performance of the proposed model has been analyzed by training it over CMATERdb 3.1.3.3 and comparing the result over test dataset with other existing methods for handwritten Bangla compound character recognition. The result of the network showed 95.5% accuracy on the test dataset which is better than some current approaches.

Previous Work

CNN

In the beginning we considered a simple CNN or Convolutional Neural Network architecture for bangla handwriting recognition. This network has:

- Image input layer
- 3 convolutional layer
- 3 max pooling layer
- 3 fully connected layer
- Classification layer

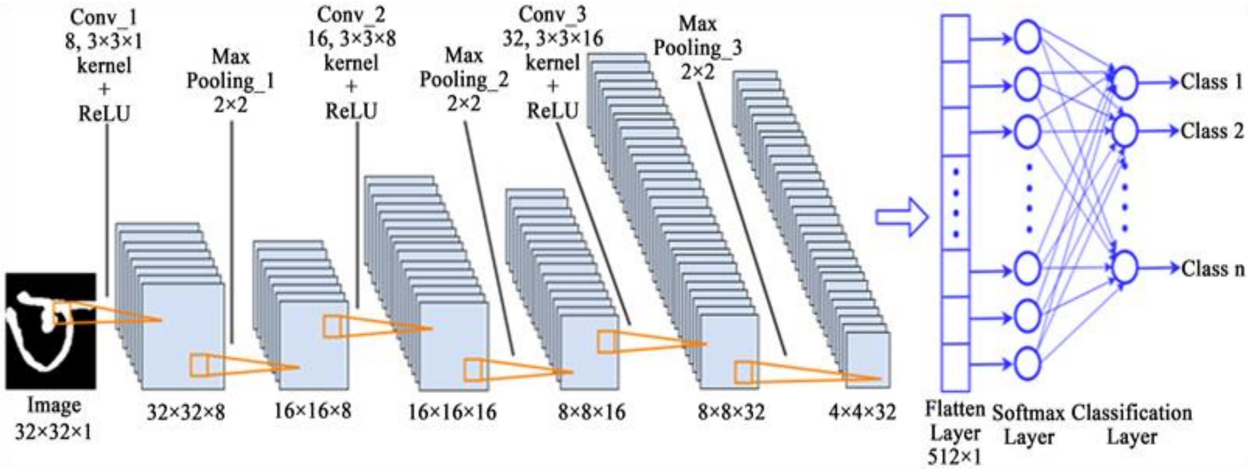

Figure 1: CNN Architecture

This network works well for simple (number) and clean image classification with higher accuracy however, if the characters gets complex then the network can not classify properly because of its limited layers. For this reason we move to our next network architecture is called ResNet.

ResNet

Residual Network (ResNet) is a deep learning model used for computer vision applications. It is a Convolutional Neural Network (CNN) architecture designed to support hundreds or thousands of convolutional layers. Previous CNN architectures were not able to scale to a large number of layers, which resulted in limited performance. However, when adding more layers, researchers faced the “vanishing gradient” problem.

Neural networks are trained through a back-propagation process that relies on gradient descent, shifting down the loss function and finding the weights that minimize it. If

there are too many layers, repeated multiplications will eventually reduce the gradient until it "disappears", and performance saturates or deteriorates with each layer added.
ResNet provides an innovative solution to the vanishing gradient problem, known as "skip connections". Skipping speeds up initial training by compressing the network into fewer layers.
Then, when the network is retrained, all layers are expanded and the remaining parts of the network—known as the residual parts—are allowed to explore more of the feature space of the input image.

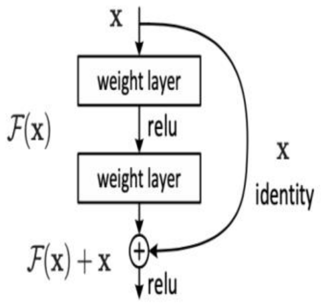

Figure 2. Residual learning: a building block.

The figure above shows a typical residual block. This can be expressed in Python code using the expression *output* = *F(x)* + *x* where x is an input to the residual block and output from the previous layer, and F(x) is part of a CNN consisting of several convolutional blocks.

The simplified version of ResNet is shown below.

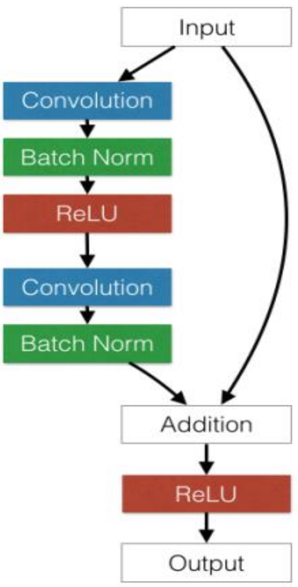

Figure 3: A ResNet architecture

ResNet works well at image classification task such as bangla handwritten character recognition due to its Feature Propagation such as skip connection and other techniques also. However, DenseNet is more efficient than ResNet in many cases such as Parameter Efficiency, Feature Propagation, Model Compactness and Performance.

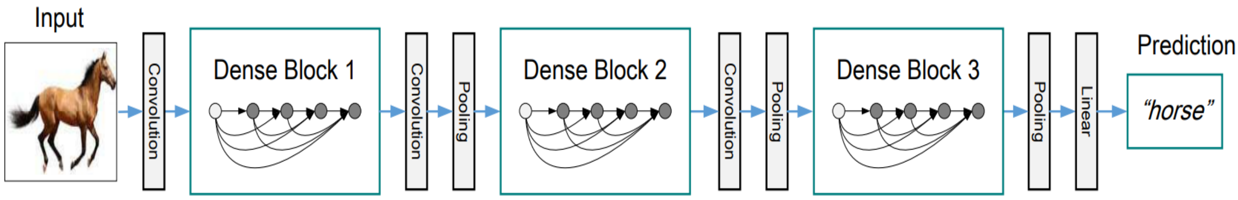

Figure 4: DenseNet with 3 dense blocks. The layers between 2 adjacent blocks are referred to as transition layers and change feature-map size via convolution and pooling.

Model

DenseNets

DenseNet, short for Dense Convolutional Network, is a deep learning architecture designed for image classification tasks. It was introduced by Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger in the paper [1] titled "Densely Connected Convolutional Networks," presented at CVPR 2017. Motivated by their works we also used their densenet-121 network architecture into our project. DenseNet has some unique features which makes the image classification easy.

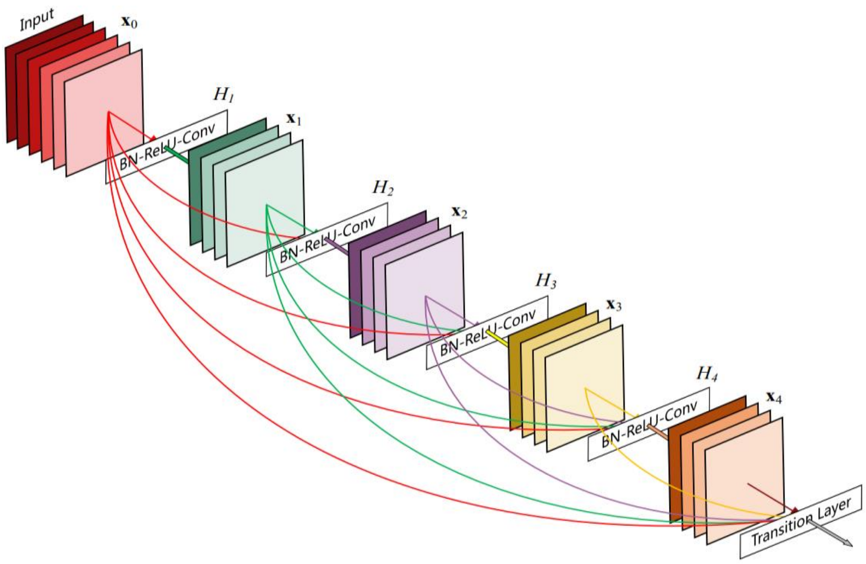

Figure 5: A 5-layer dense block with a growth rate of k = 4. Each layer takes all preceding feature-maps as input.

Dense connectivity. To further improve the information flow between layers author propose a different connectivity pattern: to introduce direct connections from any layer to all subsequent layers. Figure 5 illustrates the layout of the resulting DenseNet schematically. Consequently, the ℓ^{th} layer receives the feature-maps of all preceding layers, x0, . . . , $x_{\ell-1}$, as input:

$$x_\ell = H_\ell([x_0, x_1, \ldots, x_{\ell-1}]), \quad (1)$$

where $[x_0, x_1, \ldots, x_{\ell-1}]$ refers to the concatenation of the feature-maps produced in layers 0, . . . , $\ell-1$. Because of its dense connectivity author refer to this network architecture as Dense Convolutional Network (DenseNet). For ease of implementation, author concatenate the multiple inputs of $H_\ell(\cdot)$ in eq. (1) into a single tensor.

Composite function. The author define $H_{\ell}(.)$ as a composite function of three operation: batch normalization, ReLU and 3x3 convolution.

Pooling layers. Concatenation is a great thing which add the feature-maps to the next layer but when the size of feature-maps changes then the eq. (1) is not viable. To overcome this situation we divide the network into multiple dense blocks which illustrates in figure 4. The layers between 2 dense blocks is called transition layer which consist of 1x1 convolution layer followed by a 2x2 average pooling layer.

Growth rate. It determine the number of feature-maps generated by each layer within a dense block. Growth rate referred by a hyper-parameter k. We can denote the feature-maps as the global state of the network where each layer adds k features-maps of its own to this state. The global state once written, can be accessed from everywhere within the network.

Bottleneck layers. This layers are a key component designed to reduce the number of input feature-maps before passing them through more computationally expensive operations. The 1x1 convolution use as bottleneck layer before each 3x3 convolution to reduce the number of input features and improve computational efficiency.

Layers	Output Size	DenseNet-121 (k = 32)
Convolution	16 x 16	7 x 7 conv, stride 2
Pooling	8 x 8	3 x 3 max pool, stride 2
Dense Block (1)	8 x 8	[1 x 1 conv 3 x 3 conv] x 6
Transition Layer (1)	8 x 8	1 x 1 conv
	4 x 4	2 x 2 average pool, stride 2
Dense Block (2)	4 x 4	[1 x 1 conv 3 x 3 conv] x 12
Transition Layer (2)	4 x 4	1 x 1 conv
	2 x 2	2 x 2 average pool, stride 2
Dense Block (3)	2 x 2	[1 x 1 conv 3 x 3 conv] x 24
Transition Layer (3)	2 x 2	1 x 1 conv
	1 x 1	2 x 2 average pool, stride 2
Dense Block (4)	1 x 1	[1 x 1 conv 3 x 3 conv] x 16
Classification Layer	1 x 1	7 x 7 global average pool
		fully-connected, softmax

Table 1: DenseNet-121 architecture.

Experiment

We discuss the effectiveness and advantages of DenseNet compare with some state-of-the art network architecture like ResNet and CNN. We also tested the quality and effectiveness on BanglaLekha-Isolated benchmark dataset.

Dataset

BanglaLekha-Isolated dataset is being used for our DenseNet model. This dataset contain different classes however we especially work with bangla number and vowel in total 41,531 gray-scaled character images in 21 class.

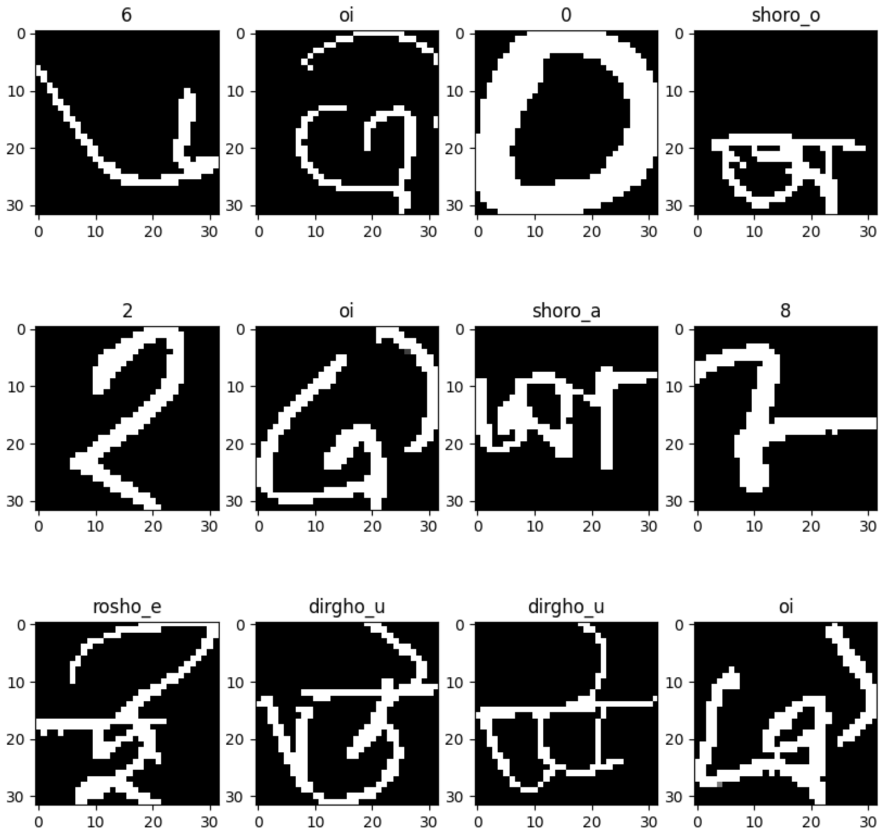

Figure 6: Dataset sample

We used different data pre-processing technique on out dataset so that the model can predict the desired output properly with minimal error rate(%).

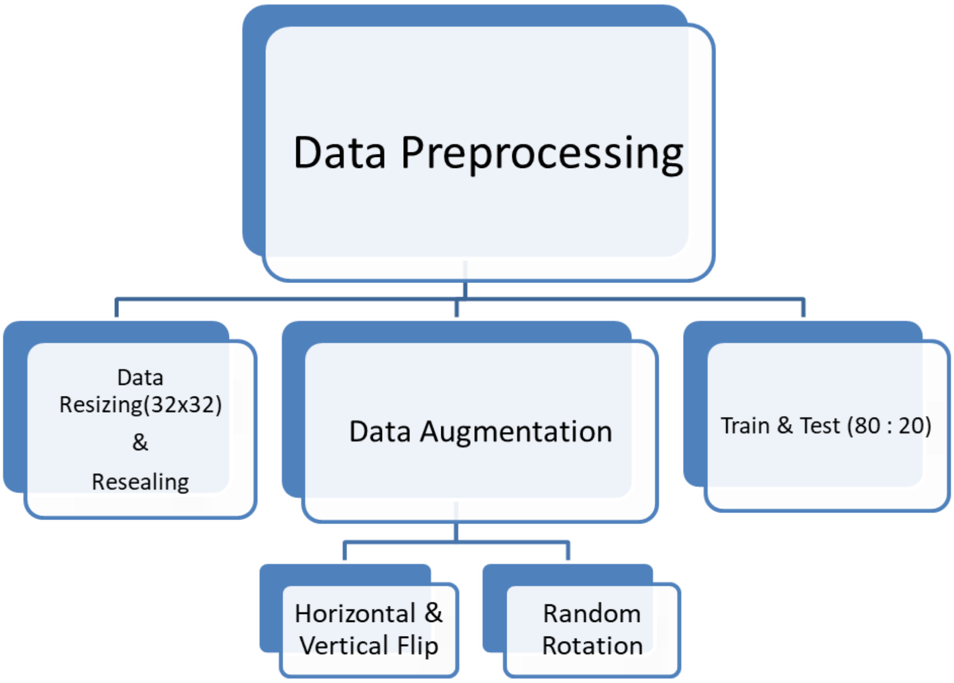

Figure 7: Image preprocessing technique

Training

To train our DenseNet model we used different parameter like training dataset, epochs, batch size, validation dataset. We use kaggle for training purposes.

```
In [5]:
IMAGE_SIZE = 32
BATCH_SIZE = 32
CHANNELS = 3
EPOCHS = 10
```

Figure 8: Training parameter of DenseNet

bangla_ocr

0 Edit

Notebook Input Output Logs Comments (0) Settings

```
Epoch 1/10
1038/1038 [==============================] - 358s 345ms/step - loss: 0.2323 - accuracy: 0.9325 - val_loss: 0.3170 - val_accuracy:
0.9162
Epoch 2/10
1038/1038 [==============================] - 364s 351ms/step - loss: 0.2053 - accuracy: 0.9431 - val_loss: 0.1632 - val_accuracy:
0.9586
Epoch 3/10
1038/1038 [==============================] - 364s 351ms/step - loss: 0.1500 - accuracy: 0.9570 - val_loss: 0.1782 - val_accuracy:
0.9520
Epoch 4/10
1038/1038 [==============================] - 361s 348ms/step - loss: 0.1412 - accuracy: 0.9601 - val_loss: 0.1810 - val_accuracy:
0.9508
Epoch 5/10
1038/1038 [==============================] - 361s 348ms/step - loss: 0.1561 - accuracy: 0.9612 - val_loss: 0.1878 - val_accuracy:
0.9545
Epoch 6/10
1038/1038 [==============================] - 374s 361ms/step - loss: 0.0935 - accuracy: 0.9723 - val_loss: 0.1340 - val_accuracy:
0.9622
Epoch 7/10
1038/1038 [==============================] - 366s 353ms/step - loss: 0.0840 - accuracy: 0.9752 - val_loss: 0.0881 - val_accuracy:
0.9797
Epoch 8/10
1038/1038 [==============================] - 368s 354ms/step - loss: 0.0814 - accuracy: 0.9768 - val_loss: 0.0820 - val_accuracy:
0.9792
Epoch 9/10
1038/1038 [==============================] - 362s 348ms/step - loss: 0.0746 - accuracy: 0.9776 - val_loss: 0.0923 - val_accuracy:
0.9777
Epoch 10/10
1038/1038 [==============================] - 363s 350ms/step - loss: 0.0607 - accuracy: 0.9819 - val_loss: 0.1263 - val_accuracy:
0.9707
```

In [49]:

```
scores = model.evaluate(test_ds)
```

```
129/129 [==============================] - 11s 82ms/step - loss: 0.1172 - accuracy: 0.9736
```

Figure 9: Training and Testing accuracy

Our model gives higher training and testing accuracy with minimal error so that over fitting and other kind of problems does not occur generally.

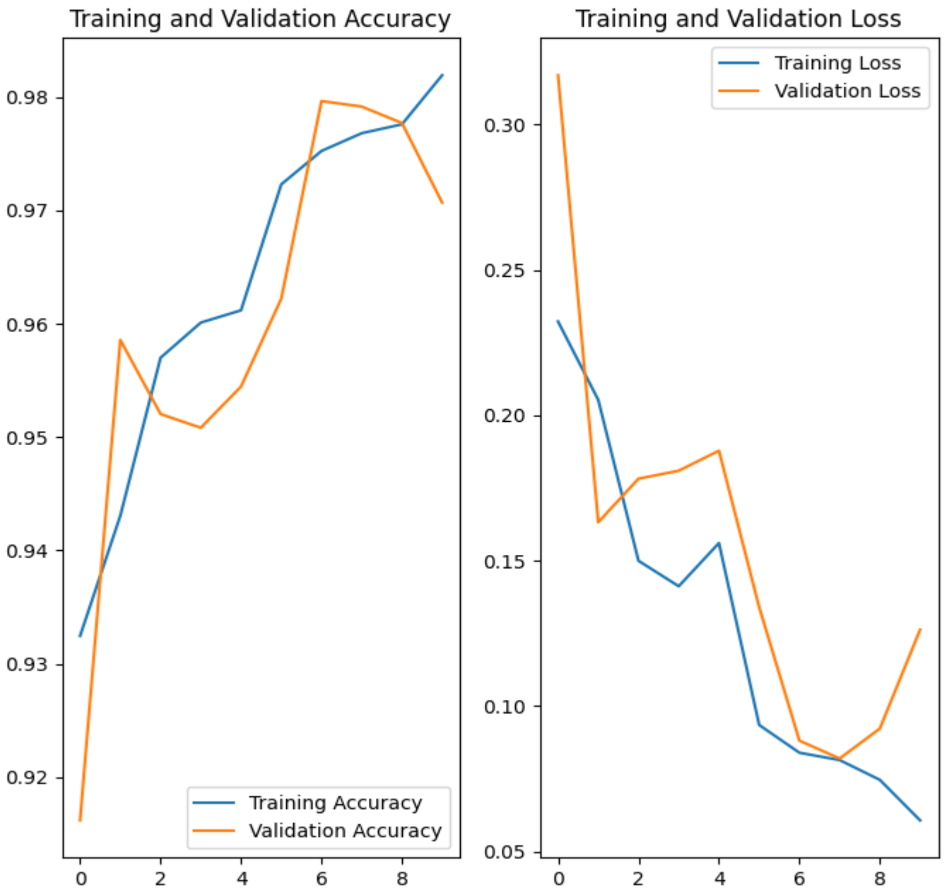

Figure 10: Training, Validation accuracy and loss graph.

Classification Result

We train DenseNet with depths, L , and growth rates = 32 and we use different technique like data pre-processing , tweaking different training agent so that we can get better accuracy and parameter efficiency. The classification result are shown here.

Figure 11: Classification Result

User Interface

Any project is incomplete without a user interface/UI because it can help to optimize the project and remove bugs. That’s why we build an mobile application with help of Tensoflow lite, we use Flutter for programming and Android Studio for code editor.

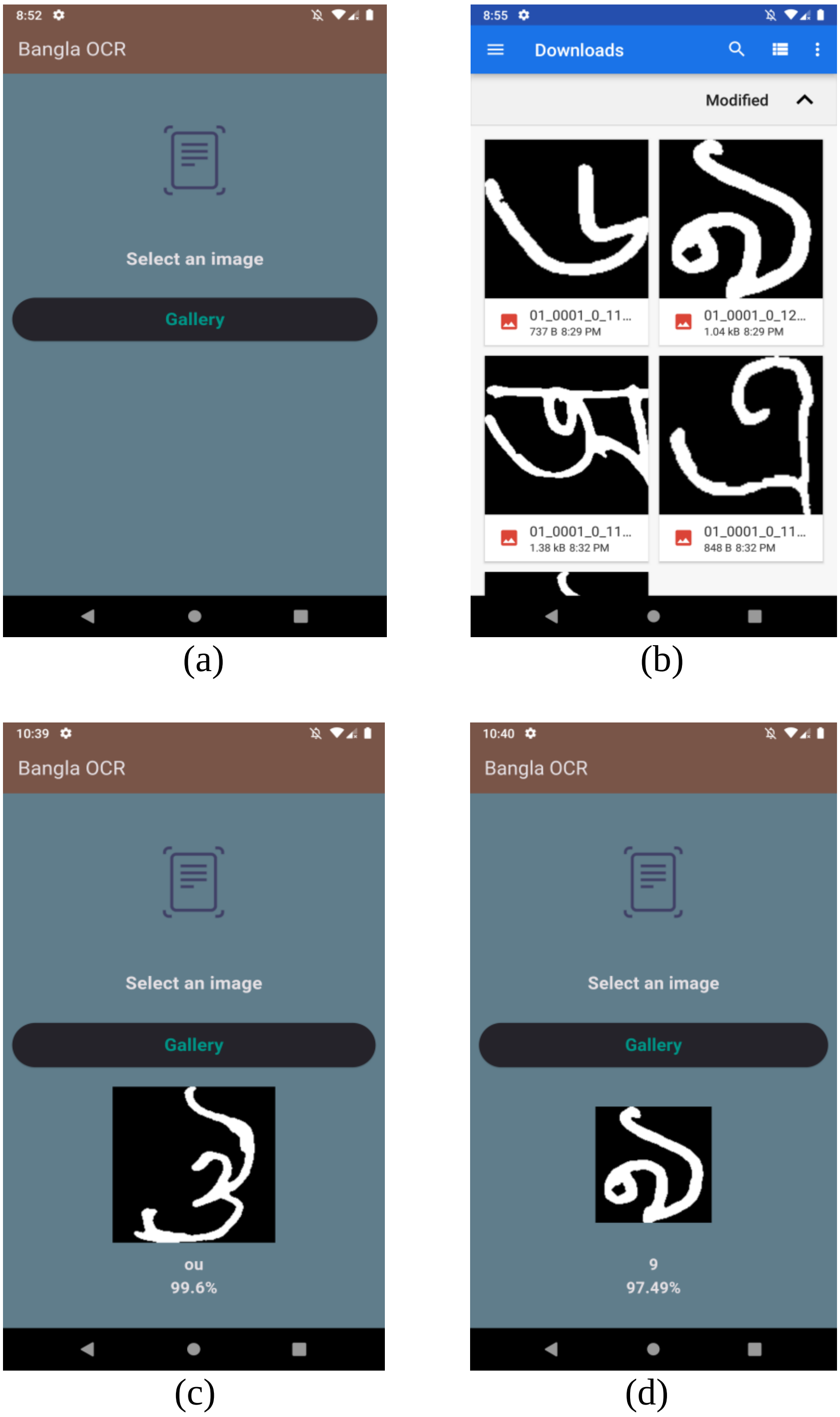

Figure 12: Mobile interface, (a) home page, (b) testing data, (c) & (d) classification result.

Discussion

From surface level DenseNet are quite similar to ResNet however the input are concatenated instead of summed. This two network are suitable for image classification but in some complex scenario like our project DenseNet work better because it has some advantages like model compactness, implicit deep supervision, stochastic vs. deterministic connection and feature reuse.

Conclusion

We present a different convolutional network architecture called Dense Convolutional Network (DenseNet). It introduces direct connections between any two layers with the same feature-map. This network gives an significant boost to our bangla isolated handwritten character recognition project with maximum accuracy and precision. We tested different model by our dataset and DenseNet gives us the better result at every scenario. And we build a mobile application so that people can use our project any time. We plan to extend our work and solve different problem related to computer vision with efficient and smart manner in near future.

References

[1] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q Weinberger. "Densely Connected Convolutional Networks," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017,

[2] Bishwajit Purkaystha, Tapos Datta, Md Saiful Islam. "Bengali handwritten character recognition using deep convolutional neural network," 20th International Conference of Computer and Information Technology (ICCIT)(2017).

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren and Jian Sun "Deep Residual Learning for Image Recognition," IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016).

[4] Rumman Rashid Chowdhury, Mohammad Shahadat Hossain, Raihan ul Islam, Karl Andersson and Sazzad Hossain. "Bangla Handwritten Character Recognition using Convolutional Neural Network with Data Augmentation," 2019 Joint 8th International Conference on Informatics, Electronics & Vision (ICIEV) and 2019 3rd International Conference on Imaging, Vision & Pattern Recognition (icIVPR).

[5] Tandra Rani Das, Sharad Hasan, Md. Rafsan Jani, Fahima Tabassum and Md. Imdadul Islam. "Bangla Handwritten Character Recognition Using Extended Convolutional Neural Network," Journal of Computer and Communications(2021).

[6] AKM Shahariar Azad Rabbya , Sadeka Haquea , Md. Sanzidul Islama , Sheikh Abujara and Syed Akhter Hossaina. "BornoNet: Bangla Handwritten Characters Recognition Using Convolutional Neural Network." 8th International Conference on Advances in Computing & Communications (ICACC-2018).

[7] Asfi Fardous, Shyla Afroge. "Handwritten Isolated Bangla Compound Character Recognition," International Conference on Electrical, Computer and Communication Engineering (ECCE)(2019).

[8] DataSet: https://data.mendeley.com/datasets/hf6sf8zrkc/2