Hybrid deep neural network for Bangla automated image descriptor

Md. Asifuzzaman Jishan [a,1,*], Khan Raqib Mahmud [b,2], Abul Kalam Al Azad [b,3], Md. Shahabub Alam [a,4], Anif Minhaz Khan [a,5]

[a] Department of Statistics, Technische Universität Dortmund, August-Schmidt-Straße 1, 44227 Dortmund, Germany
[b] Department of Computer Science and Engineering, University of Liberal Arts Bangladesh, Dhaka-1209, Bangladesh
[1] md-asifuzzaman.jishan@tu-dortmund.de; [2] raqib.mahmud@ulab.edu.bd; [3] abul.azad@ulab.edu.bd; [4] md-shahabub.alam@tu-dortmund.de; [5] anif.khan@tu-dortmund.de
* corresponding author

ARTICLE INFO

Article history
Received April 29, 2020
Revised July 4, 2020
Accepted July 7, 2020
Available online July 12, 2020

Keywords
Convolutional neural network
Hybrid recurrent neural network
Long short-term memory
Bi-directional RNN
Natural language descriptors

ABSTRACT

Automated image to text generation is a computationally challenging computer vision task which requires sufficient comprehension of both syntactic and semantic meaning of an image to generate a meaningful description. Until recent times, it has been studied to a limited scope due to the lack of visual-descriptor dataset and functional models to capture intrinsic complexities involving features of an image. In this study, a novel dataset was constructed by generating Bangla textual descriptor from visual input, called Bangla Natural Language Image to Text (BNLIT), incorporating 100 classes with annotation. A deep neural network-based image captioning model was proposed to generate image description. The model employs Convolutional Neural Network (CNN) to classify the whole dataset, while Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) capture the sequential semantic representation of text-based sentences and generate pertinent description based on the modular complexities of an image. When tested on the new dataset, the model accomplishes significant enhancement of centrality execution for image semantic recovery assignment. For the experiment of that task, we implemented a hybrid image captioning model, which achieved a remarkable result for a new self-made dataset, and that task was new for the Bangladesh perspective. In brief, the model provided benchmark precision in the characteristic Bangla syntax reconstruction and comprehensive numerical analysis of the model execution results on the dataset.

1. Introduction

Humans can always recognize information when something is shown visually. They possess the power to understand visual information. Therefore, getting to experience new sights for the first time typically needs a quick response [1] [2]. The capacity to grasp the scene's definition is limited not just to perceiving images but also to the syntactic and semantic meaning of an image. Among the substances of the images can be found some connectivity. Once it comes to a language-based textual representation of a picture, it is generally an important area of study in computer vision, image processing, and natural language processing. As there is a rising range of practical applications focused on image captioning, research on these areas has been increasing. Many researchers worldwide have begun focusing on, such as image classification, text-based image analysis, image to object detection, allowing people with visual disabilities to grasp the digital world, and recognizing the image in social media [3]–[7].

This study aims to generate Bangla textual captions of contextual images to serve the Bangla speaking community in light of the recent advances in natural language processing. Most of the image captioning researches are conducted in English. However, this study is based on the Bengali language, commonly used in different geographical countries, and is widely used by people in many parts of the world. The Bengali language is local to the Bengal district, which incorporates Indian conditions of West Bengal and the present-day Bangladesh republic. It is used by more than 210 million people as a first or second language, with around 100 million Bengali speakers in Myanmar, about 85 million in India, and large networks of immigrants in United States, United Kingdom, and the Middle East globally.

The image captioning model is a crucial topic that generates a simple language text from the given image. The primary goal is to classify the whole dataset and then implement a hybrid model for generating text using a specific optimization technique [8]–[11]. We are attempting to construct a new dataset for that statement whose name is Bangla Natural Language Image to Text (BNLIT) [12], and that dataset is generated for a different target language. This dataset comprises 8,743 images along with an individual annotation and extracts certain annotations from the cover to the specialists. As our best idea, there is still no dataset of the image to generate Bangla language text for researching and improving the accuracy and loss score [13].

From the logical and working perspective, text generation from the given input image is an interesting sector of the machine learning, image processing, and deep learning sector. Moreover, the image to the Bangla text generation technique is a unique work from Bangladesh and Bangla language society's perspective. Also, the growing image and video datasets pose a remarkable challenge to computational natural language-based processing due to limited linguistic and semantic templates and closed vocabulary.

The image narrator's visual meaning needs to be enhanced and promoted to create a model for generating image captions. For example, how the models intercept the context, detect a region of the image, and then construct the image caption that is consistent with the content of the image. Improving accuracy is required for this role, but the challenging task is to generate Bangla text from the given image. Meanwhile, it often includes how specific textual embedding in an image may be adjusted to various contexts. To carry out this challenging task, we proposed a hybrid neural network model. The most significant and challenging element in the design component of the encoder-decoder models is to create and build a model that incorporates Convolutional Neural Network (CNN) [14], Recurrent Neural Network (RNN) [15], and Long Short-Term Memory (LSTM) models [16]. Therefore, another important aspect of image generation to text is to precisely follow this hybrid model and train the existing structure appropriately.

In the portion of image processing and the pattern recognition section, the first task is to classify the dataset, and then it is required to take an attempted image to text generation, object detection, and so on. On the other hand, there are different types of datasets existing, which are very much popular for image processing, and they are Flickr8K, Flickr30K, MS COCO, CIFAR-10, and CIFAR-100. In the classification image section, the main task is to identify objects from the images of the dataset. The accuracy improvement is the main challenge, along with the exact model development and Inception-v3 is the best method for accuracy improvement and efficiency [17]. Moreover, some researchers showed that they adopt a CNN model to classify the dataset and show the dataset's smoothness prior to the labels. This paper's methodology is related to our research because we also followed CNN for the image classification, and we also use different types of class for labels. To extend, they also implemented and showed that the accuracy improvement along with how to change the CNN parameters using the most popular stochastic gradient descent optimization technique and α-expansion min-cut-based algorithm [17][18]. Hyperspectral imaging (HSI) technique is one of the superior techniques for the image processing portion. It proposed a CNN architecture based on spectral-spatial capsule networks to achieve better accuracy and be used for classification accuracy and computational time both [19]. To do this, a generative adversarial network (GAN) is as well as the other superior model for the classification and it is the challenging task of the HSI portion [20]. In addition, HSI has a high image classification technique using a generative adversarial network (GAN)-based methods and highly configured graphics processing unit for the analysis of complex and non-linear data [21][22].

Image to text generation is the most crucial task, and in that task, the main motive is to generate image caption using a complex neural computing model. To add to do this, using Attention Generative Adversarial Networks is the best technique and achieves better accuracy [1]. On the other hand, one researcher showed that creating and developing a new model, which is a combination of CNN, RNN, and LSTM models, is also working fine and gets better accuracy for the text generation [23][24]. Like them, we also implemented a hybrid neural image captioning model with the best combination of CNN, RNN, and LSTM methods and achieved a benchmark result for BNLIT. Feature-Guiding Generative Adversarial Networks (FGGAN) is another hot research to solve the image captioning technique. It has a good efficiency, which can generate text from blur or poor quality image. Furthermore, they also showed that the text generation technique and their performance also depend on data efficiency, data resize, data re-shape, and dataset size [25][26]. Furthermore, BLEU and METEOR metric evaluation is another crucial topic for the judgment of the models, and in that paper, researchers are also highlighting how those metrics are so much important for understanding how efficient that proposed model [26]. They also achieved a large scale evaluation score, which is 63.5% and 30.6% for the human performance using the benchmark dataset, e.g., Flickr8K, Flickr30K, MS COCO, and papers as mentioned above are the state-of-the-art of our research [27].

Meanwhile, the primary objective and contribution of this research are, first, to invent a new target language dataset. Second, developing a hybrid image captioning model which is capable of generating Bangla caption from the given any image. Third, classifying the dataset with relevant classes. Fourth, improving the accuracy for that target dataset, and the sixth, successfully testing the proposed model in semantics recovery tasks of images. Our self-made BNLIT dataset is already published in the machine learning and image processing repository and available for every researcher [12].

2. Method

2.1. Hybrid deep neural network

In the domain of Computer Vision, a neural network system is a set of algorithms that enables the computational system to find patterns by matching complex input data relationships like human brains.

Convolutional Neural Network (CNN) is a deep learning algorithm that selects characteristics in the taken image and differentiates from others. In previous years, filters like blurring, sharpening, and detecting edge was needed to be hand-engineered and included enough training before CNN comes into play. The broad implementation of this algorithm, such as, has perfected facial recognition, image detection, recommendation system, and natural language processing.

We used four main layers of the architecture of CNN: Convolutional Layer, Pooling Layer, Rectified non-linear unit, and Fully-Connected Layer. Convolution layer is present at the center of the network and performs convolutions that involve linear operation utilizing multiplication, a set of weights with the array of input data called filter or kernel. The main purpose of convolution is to fetch high-quality features like edge detection and some low-quality features like color, gradient orientation, etc. To be specific, the filter usually applies following a process to the parts which overlap each other left to right and top to bottom. Using the same filter to detect a particular object in the image has been recognized powerful as it will sort out systematically all over the image where the object is present [28].

The next comes the pooling layer. The main objective is to continuously decrease the spatial size of the representation to decrease the number of parameters and computation in the network as well as controlling overfitting [29]. With the help of MAX operation, it works independently to every depth slice of the input and resizes it spatially. Fully-connected layer: Neurons in an F-C layer have full connections to all activations in the previous layer, their activations can hence be computed with a matrix multiplication followed by a bias offset. It is possible to convert FC layers to CONV layers as there is a very small difference. We showed the architecture of CNN in Fig. 1, and the proposed model in Fig. 2.

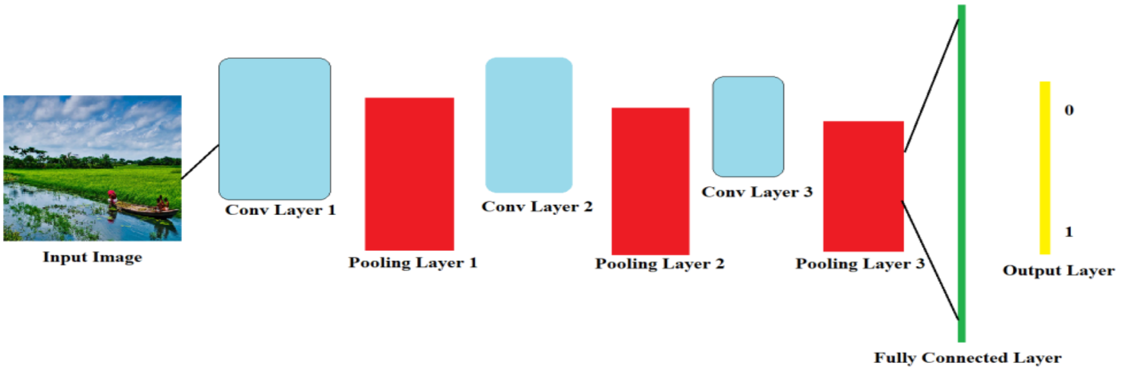


Fig. 1. Architecture of CNN.

Recurrent Neural Network (RNN) is a neural network where the output of a previous computation is implemented as the input of the current one. Usually, the inputs and the outputs are independent of each other, but whenever the system needs to predict output in a sequence, it needs to remember the previous input. RNN keeps such calculations in a memory located in a “Hidden State”. That’s why it became applicable to tasks like unsegmented, connected handwriting recognition, and speech recognition.

In speech recognition and handwriting recognition, Bidirectional RNN is implemented as sometimes there might be ambiguity in the provided input so that we need to know the next possible outputs to sequence past to present outputs, for example, words in a sentence. For translation services in NLP, the Encoder-Decoder or Sequence to Sequence RNNs is used. Here, the encoder RNN keeps updating the “Hidden State” for continuous output as ‘context’. Then the produced outputs are then fed into the decoder RNN as input to produce ‘context’ translations sequence by sequence [30].

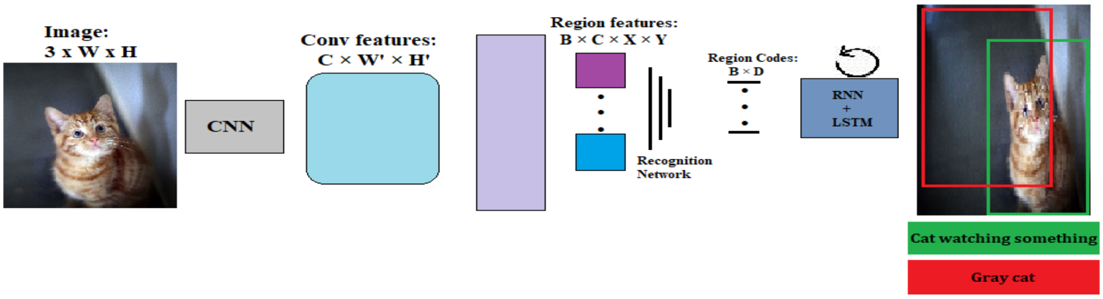


Fig. 2. Architecture and specifications of the proposed model

While remembering long sequences, RNN couldn’t process when ‘*tanh*’ or ‘*relu’* is present as an activation function. In addition to that, disadvantages like gradient vanishing and exploding problems are being fixed by the modified version of RNN which is called Long Short-Term Memory (LSTM). It performs well to classify, process, and predict time series given provided the time duration is unknown. Back-propagation is used to train the model.

LSTM has three gates- 1) Input gate, 2) Forget gate and 3) Output gate:

1. The input gate determines which value should modify the memory by the decision of “sigmoid function” through 0, 1 and “tanh function” determines the weight to the values within the range of priority of -1 to 1 to pass through.
2. Forget gate determines the details that need to be discarded from the block. It is additionally chosen by the "Sigmoid capacity". It observes the previous state and the content input and outputs a number between 0 and 1 for each number in the cell state.
3. In general, the input values and the memory will decide the output. Like previously, the "sigmoid function" will decide the values that will pass through 0, 1 and 'tanh' function determines the weight of the values within the range of priorities of -1 to 1 to pass through multiplied with the output of sigmoid [7]–[10][31].

2.2. Dataset

Data is everywhere. It requires the methodological procedure, statistical analysis, and categorical demonstration to convert it into usable information. Moreover, such information, when ordered, organized, and represented by variables using values, forms a dataset. A dataset is used to build model, recognize patterns, and generate meaningful insights widely in the field of image processing, machine learning, and deep learning sector. Real-world datasets include musical note datasets, voice clip datasets, image properties datasets, character matching datasets, recursive datasets, etc. Likewise, the image accumulated together has been established as the most dynamic and accurate in the research and development of complex data-driven application systems in recent years. For serving the purpose, we introduced a new dataset titled BNLIT [12] that is comprised of a gallery of 8,743 photos representing the life, heritage, ethnicity & culture of our country Bangladesh where every image speaks with its language. Instead of portraying western socioeconomic image collection based datasets like Flickr8K, Flickr30K, and MS COCO, we choose to reflect our country's lifestyle, culture, and beauty where the authors have been raised and brought up. The dataset is exclusively constructed by aggregating numerous sources to ensure variety, depth, and authenticity of the images. We collected images for the dataset from both urban and rural settings; images were captured on natural sceneries, shopping centers, local grocery shops, public transportation, and marriage ceremonies and so on. We used mobile phone camera, DSLR, and action camera to take the images. We also took images of ethnic and religious festivities. We also collected some photographs from various web sources that are not under any copyright obligation.

For easy referencing, recognizing and labeling the analysis of information, image annotation is a good practice that is being implemented distinctly for every picture, and the language is selected in our mother tongue "Bangla". In machine learning and deep learning, computers rely mostly on the training data that is being fed in the algorithm, and the performance also depends on absolute precision. In addition to that, image annotation is a highly qualified technique for computer vision to detect an object from training images predetermined by the scientists. However, our dataset is viewed as a kind of multi-classification image characterization with a terrible measurement of classes with the vocabulary estimate.

While training a machine to learn, the larger the dataset, the better the precision to produce results. To handle the large dataset BNLIT of 8,743 pictures, we characterized it into 100 classes. For the explanation of images, we set up a sentence for each and resized them equally as they were of varied sizes in pixels and resolutions. Here, file formats include JPG, JPEG, and PNG. We represent in the table below how many images stay in different formats e.g., JPG, PNG, JPEG, and their resolutions, image dimension, and bit depth.

In our self-made BNLIT dataset, 58 images are staying in the format of JPEG. Also, in the PNG and JPG formation, there are a number of 1,237 and 7,448 images, respectively. Before the CNN implementation, we resized the full dataset into dimension of 224 × 224. On the other hand, we represented the technical characteristics of the BNLIT dataset in Table 1. Moreover, we showed the dataset grouping in Table 2. A specialized program in Python language is developed to resize all the pictures into the same pixel and split them into different categories. Later, the pictures are prepared for CNN and RNN.

Table 1. Technical characteristics of BNLIT dataset

No of images	Resizing and Formation				
	File Format	*Image dimension (Height and Width)*	*Horizontal Resolution*	*Vertical Resolution*	*Bit depth*
58	JPEG	224 × 224	96 dpi	96 dpi	24
1237	PNG				
7448	JPG				

We divided the full BNLIT dataset into three portions: training, testing and validation. Training data is the most important because neural networks learn image captioning during the full dataset's training period, and it fits the proposed model. Our dataset size is containing 8,743 images, so that reason we choose 7,243 images for training reasons. On the other hand, testing data is important for that type of data, which is provided for an unbiased evaluation of a final model whose on the training dataset. Furthermore, validation data is used to evaluate a given model, and it is crucial for evaluating the models. We selected 1,000 and 500 images for the testing and validation period, respectively.

Table 2. Class frequencies of BNLIT dataset

Class Name	Classification		
	Number of Object	*Class Name*	*Number of Object*
Market	680	Book	64
CNG	568	Washing Machine	25
Gun	352	Dish Washer	42
Train	241	Plane	77
Scanner	85	Soil	98
Printer Machine	546	Snake	565
Stapler	47	Shopping Complex	54
Moon	85	Movie Theater	78
Sun	98	Hippopotamus	25
Star	56	Monkey	45
Wood	54	Lion	86
Cloudy Sky	75	Horse	546
Sky	47	Tiger	85
Photo Frame	58	Dog	47
Switch Board	45	Cat	98
Circle	51	Coconut	56
Rectangle	585	Cake	54
Triangle	522	Birthday	75
Box	98	Marriage Ceremony	47
Watch	24	Sea Beach	58
Kite	8	Sunglass	45
Knife	58	Band	51
Ding Dong Bell	25	Water	585
Hawker	39	River	524
Rickshaw Puller	258	Shoe	98
Driver	59	Light	24
Nurse	27	Fan	8
Bride	26	Clock	58
Doctor	87	Power Bank	680
Hospital	54	Desktop	568
Police	48	Laptop	241
Traffic Light	98	TV	85
Station	258	Charger	85
Bus	647	Speaker	47
University	542	Harmonium	85
College	54	Guitar	98
School	68	Tree	98
Building	25	Fish	258
Banana	65	Aquarium	25
Cherry	28	Keyboard	65
Strawberry	65	Mouse	28
Jackfruit	25	Jeans	65
Mango	24	Shirt	25
Village	83	Door	56
Town	25	Bat	54
Scenery	357	Calculator	75
Eraser	55	Headphone	47
Pen	98	Book Self	58
Pencil	75	Table	45
Notebook	25	Mobile	51

2.3. Simulation

2.3.1. Image and annotation processing using hybrid deep model

We resize the images of all our datasets to affirm higher all-inclusive statements and to maintain a strategic distance from any numerical irregularity all through training and testing stages. We tend to utilize crude image documents of dataset nearby CNN and VGG16 highlights. We set pixels to measure 224 x 224 x 3. The images of the dataset are doubtlessly concealing images with pixel regards running from 0 to 255 with a part of 224 x 224, so before feeding the data into the model, it is indispensable to pre-process it. First, adhere each 224 x 224 image of the dataset into a framework of size 224 x 224 x 3, which we would then be able to bolster into the CNN arrangement. On the other hand, we classified a full dataset utilizing CNN and VGG16 highlights. Utilized 100 types of classes with the batch size of 16, we completed full dataset image classifications.

Furthermore, we implemented Conv2D features with the Maxpooling 2D and ReLU activation function. To extend, we conducted a dropout layer on CNN and a dropout layer, which value 0.5 because mainly the dropout layer regularized the neural network, and it can reduce the overfitting tendency. We implemented categorical cross-entropy for the measure losses of CNN and selected Stochastic Gradient Descent *(SGD)* for the CNN optimization period. The defined learning rate was 0.01, decay rate = 1e-6, momentum = 0.9 and neserove = True. We ran 20 epochs for CNN. After completing classification training, pick the best weight during the preparation period, and create an hdf5 record with misfortune and exactness versus ages chart and store it in an index.

At that point, we were concerned about the RNN and RNN, for the most part, used to produce content from the given information images. We picked one Bangla annotation for each picture. We centered on Keras, LSTM model close by with NumPy, Matplotlib library, and trained up our dataset's comment record. We defined 256 filters in the LSTM and set dropout value 0.2. Finally, we conducted a fully connected layer with the NAdam optimization technique, chosen batch size 128, and measured loss for used categorical cross-entropy. After training up, make an index and pick the best weight from the preparation time frame and make an hdf5 model which creates misfortune, exactness versus ages chart and run 50 epochs for RNN.

At long last, we joined both the model of CNN and RNN highlights of our dataset and trained up 30 epochs again and produced an accuracy and loss diagram. In the wake of finishing the dataset train up, all prepared models were put away in our CV organizer. At that point, we took the endeavor to assess our prepared model for these datasets to show signs of improvement.

2.3.2. Implementation

Image recognition is known for being one of the essential aspects of image processing. Typically it is easy to get big loads of ideas when we inspect a few late works [32]. Searching for those sentence portrayals assemble visit references is required for the things and their qualities. In this scenario, owing to their good precision, we used CNN for image classification. We have used CNN on ImageNet to prearrange, and we have good results [33]. We have identified 100 ImageNet Recognition Task classifications [17] that are then optimized using CNN. The plan [5] is to use the support of Region Convolutional Neural Network to scan for each item from each image. Having followed the paper [34], we continue to use the initially known space of nineteen irrespective of the pixel all-out pictures by using the jump box, as stated in the equation:

$$v = W_m[CNN\theta_c(I_b)] + b_m \tag{1}$$

There is a fully connected layer that is placed before the classifier, usually in a split second. Within the bounding box (I_b), the pixels are transformed into dimensional 4096. So $CNN\,(I_b)$ executes this cycle, which has a θ_c parameter comprising about 60 million parameters. We used the matrix W_m with a measurement of h × 4096 for multimodal inserting volume. Here, h is the scale of the position for multimodal insertion.

Bidirectional Recurrent Neural Network (BRNN) [21][35] is used to evaluate the representation of the title. Several pieces of BRNN are contained in the RNN field. Thus sentence composition was a critical part of our plan. Alternatively, BRNN is often used to predict a specific structure. Any component of the BRNN model category depends upon a component's past and future context. BRNN executes this process, where the close yield of two RNNs and one strategic planning of the sequence is conducted from left to right. Because of this activity, we obtained the subsequent yields that forecast the target signals given. There is an arrangement of N terms, as per our model. To transform each element into an h-dimensional matrix, the BRNN selects such N terms. We also used the overview t = *1.... N* to represent the situation of a term in a text. The precise BRNN condition is conducted as per the following:

$$\begin{aligned} x_t &= W_w I_t \\ e_t &= f(W_e x_t + b_e) \\ h_t^f &= f\left(e_t + W_f h_{t-1}^f + b_f\right) \\ h_t^b &= f(e_t + W_b h_{t+1}^b + b_b) \\ s_t &= f(W_d\left(h_t^f + h_t^b\right) + b_d) \end{aligned} \tag{2}$$

Thus, the weights W_w specify a word that is incorporated into the network such that 300-dimensional word2vec [33] weights will be used to inject it. It also stays set, as overfitting occurs. We also used a pointer column vector (I_t) in a word vocabulary that has a single component of the t-th word. There are two different guiding sources located at BRNN. The first passes from left to right (h_t^f) and the latter from right to left (h_t^b). We implemented the activation function f for rectification of the linear unit (ReLU).

2.3.3. Optimization

We used stochastic gradient descent (SGD) for the optimization section in CNN part with a mini-batch of 16 image sentence sets moreover. We implemented a large dataset, and SGD can show the best and faster performance on that. Furthermore, it can converge faster than other optimization techniques with the definition of a batch size because it can perform and update data more frequently. On the other hand, the SGD optimization technique is a simple combination of gradient descent, whereas the stochasticity comes with a mini-batch measurement technique and computes the gradient technique at each descent [10]-[12]. To extend that, it has a regularization effect, making it appropriate for the exceptionally non-raised function of losses, for example, those involved in preparing profound systems for an order. In addition, it can update weights on the fly for the raw and extraordinary data, but as the frequently update the weight loss and cost functions are heavily fluctuates. We used learning rate 0.001, decay rate 1e-6, momentum=0.9, and nesterov=True because we implemented a large dataset containing 8,743 images so that we chose the SGD optimization technique. From that point onward, for measured misfortunes utilized misfortune all out categorical cross-entropy, and for measure exactness, used precision metric. We utilized 100 types of classes for the classification of the entire dataset.

For the RNN part, we used Nesterov-accelerated Adaptive Moment Estimation (NAdam) optimizer for the image to Bangla caption generation. NAdam optimization technique is the combination of NAG and Adam techniques, and it is utilized for uproarious slopes or for inclinations with high bends [24][25]. On the other hand, the learning procedure is quickened by summarizing the exponential rot of the moving midpoints for the past and current slope. We also maintained accuracy and loss in the RNN part for measuring accuracy and loss vs. epoch.

3. Results and Discussion

We implemented a hybrid neural image captioning model which is capable to generate Bangla text based practical depiction of the given image. We prepared our model to get familiar with the connection between better bits of the image alongside the pertinent segment of the sentences. To measure exactness, we utilized grouping exactness measurements.

3.1. Encoding: CNN implementation

In this section, we gave concern and talked about the CNN feature execution of the BNLIT dataset. We picked CNN strategy for image classifications and utilized 100 classes on CNN. Furthermore, we utilized stochastic gradient descent (SGD) as an optimized technique because we implemented a large dataset, and SGD can show the best and faster performance on that. Moreover, we ran 20 epochs and with the select batch size 16, and we implemented categorical cross-entropy for the measure losses of CNN and selected Stochastic Gradient Descent (SGD) for the CNN optimization period with the defined learning rate 0.01, decay rate = 1e-6, momentum = 0.9 and neserove = True. From the first epoch of CNN training time, we got improvement results for our self-made dataset. After training and completing one epoch, model saved in a directory and finally completed all epochs, choose the best weight from them, and create a final model in the hdf5 file, which saved in model.hdf5.

After running 20 epochs, we got 0.824538 training accuracy, which is the best precision for this dataset for CNN results. We got 0.801161 validation accuracy for the BNLIT dataset. After 20 epochs, overfitting happened, and that outcome did not store in graphically. We showed that training time accuracy and validation time accuracy vs. epoch for CNN in Fig. 3. We demonstrated that outcome graphically for the entire dataset.

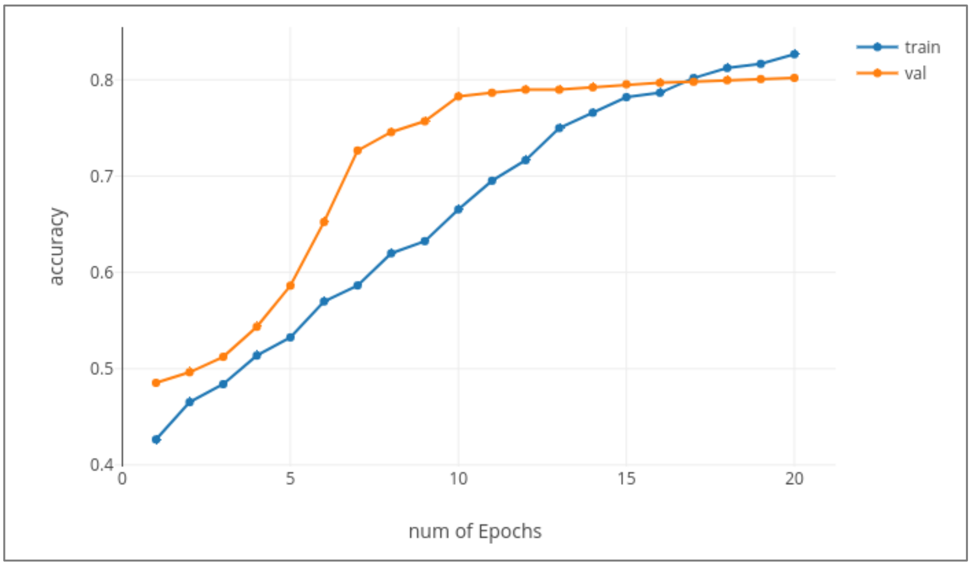
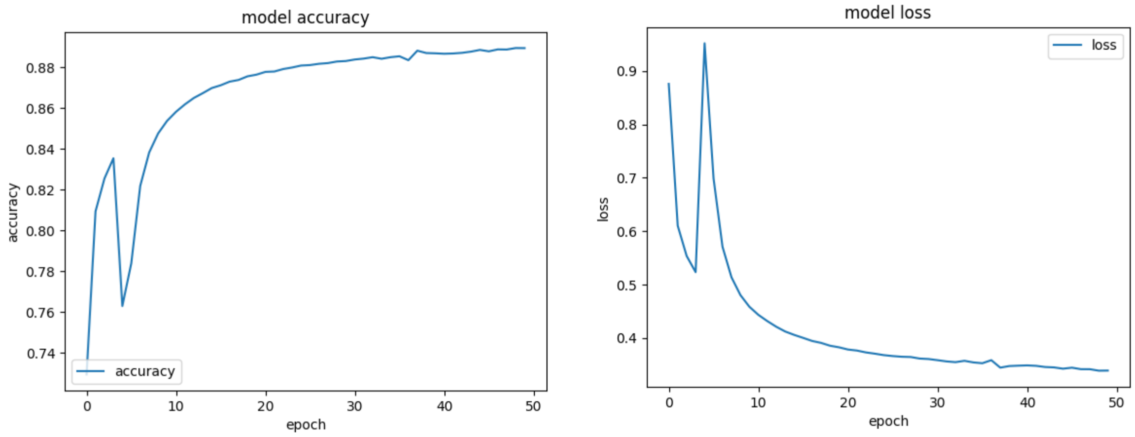


Fig. 3. Graphical representation of training time and validation time accuracy for image classification of CNN part

3.2. Decoding: RNN and LSTM implementation

After CNN, for the most part, talked about the RNN implementation portion of our dataset. For the RNN part, we utilized the NAdam optimizer technique for this thesis. We also maintain accuracy and loss in the RNN part for measuring accuracy and loss vs. epoch.

We selected batch size 128 during RNN train up. We ran 50 epochs during RNN training time and chose a vocabulary size of 98. From the first epoch of RNN training time, we got better accuracy. We showed that epoch vs. accuracy and loss in Fig. 4. After running 50 epochs, we got 0.889419 accuracy, which is the best accuracy for this dataset for RNN results.



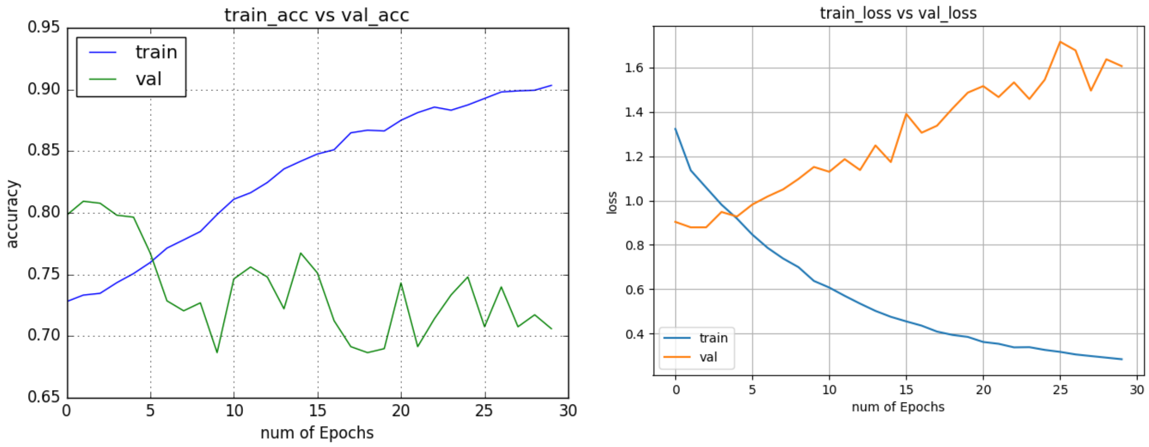
(a) Epoch vs. training time accuracy

(b) Epoch vs. training time losses

Fig. 4. Graphical representation of the RNN part result.

3.3. Image to Text Generation: Hybrid Model Implementation For Generate Caption

For Generate Text, we generated features.pkl file from the whole dataset, which contains 8,743 images. Pickle library mainly serialized objects in python. Finally, we ran 30 epochs for generating text from the given input image. Each epoch took approximately 1 hour 50 minutes, and our accuracy reached 0.917895 for training and 0.768651 for validation. We got approximately 0.181132 losses in the training period and 1.605326 losses invalidation purposes. We showed a graphical representation of the period training, and validation accuracy and loss result in Fig. 5. Therefore, the initial accuracy value was 0.688928 in the first epochs for the training period; however, from the second epoch with the accuracy value coming down to 0.7112296. The accuracy value increased, which indicates that the model is starting to pick up the new Bangla language.



(a) Accuracy of training and validation time vs. Epoch.

(b) Epoch vs. training and validation timing loss.

Fig. 5. Graphical representation of during final train up for training and validation time accuracy along with their loss.

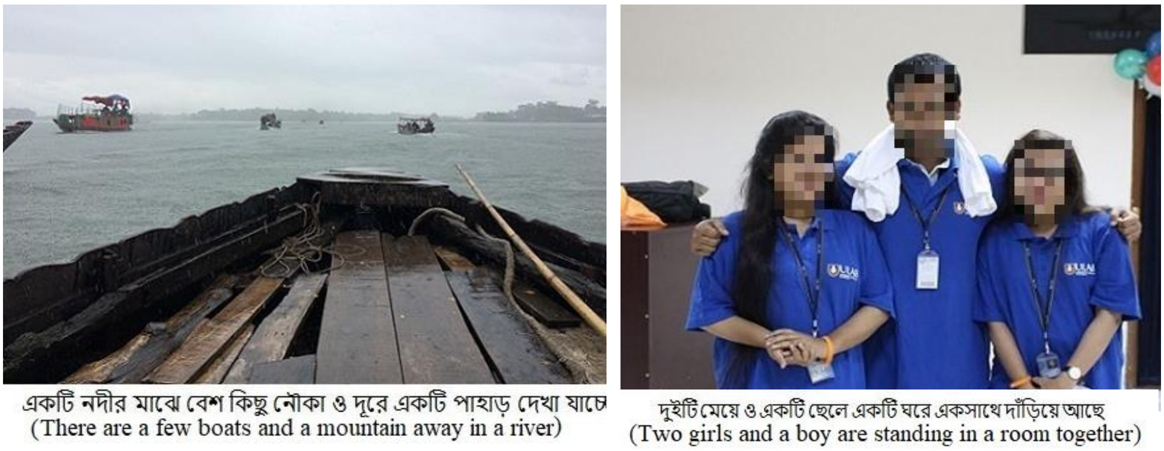
3.4. Evaluation Result

After successfully training up, we need to evaluate our model and represent that result to the readers. There are different types of evaluation in the machine learning and image processing sector, but among them, BLEU (bilingual evaluation understudy) and METEOR (Metric for Evaluation of Translation with Explicit Ordering) evaluation are so much popular for NLP. First of all, if we give concern towards the BLEU evaluating system, then we can see it is an algorithm, which is used for machines translated from one language to another. The thought behind BLEU is the closer a machine interpretation is to an expert human interpretation; the better it is. The BLEU's assessment framework requires two sources of info: 1) a numerical interpretation closeness metric, which is then appointed and estimated against, 2) a corpus of human reference interpretations. BLEU midpoints out different measurements utilizing an n-gram strategy, a probabilistic language model frequently utilized in computational linguistics. We used BLEU-1, BLEU-2, BLEU-3, and BLEU-4 for the evaluation score.

On the other hand, METEOR is presented as another assessment approach, in view of the incomplete request hypothesis. METEOR joins straightforward choice help and advantageous devices for information investigation with the capacity to remember partners' inclinations for the choice procedure. The essential thought is an orderly step-by-step total of pointers, including their weighting. We implemented and used different types of the hidden layer, and they are 64, 128, and 256, respectively. We illustrated our evaluation metrics score of our model and represented it in Table 3, which is the benchmark result of our dataset. Furthermore, we showed how we could generate a Bangla language caption from the given image in Fig. 6.

Table 3. BLEU scores and METEOR score for BNLIT dataset

Hidden Layer Size	Evaluation				
	BLEU-1	*BLEU-2*	*BLEU-3*	*BLEU-4*	*METEOR*
64	65.8	45.8	32.1	22.3	19.613227
128	65.3	43.1	32.4	19.8	18.625489
256	65.9	43.4	32.3	22.8	19.983625


একটি নদীর মাঝে বেশ কিছু নৌকা ও দূরে একটি পাহাড় দেখা যাচে
(There are a few boats and a mountain away in a river)
দুইটি মেয়ে ও একটি ছেলে একটি ঘরে একসাথে দাঁড়িয়ে আছে
(Two girls and a boy are standing in a room together)

(a) The best most preferable text for each test images.

(b) The absolute best test sentence for the test image.

Fig. 6. Example of the image to Bangla language caption generation using hybrid image captioning model.

3.5. Discussion

We represented a deep hybrid neural image captioning model for generating images to Bangla text. Our hybrid model is the combination of CNN, RNN, and LSTM models and is used for implementing the self-made BNLIT dataset. We achieved better accuracy, and it can generate text from the image. We implemented CNN features for the image classification using the VGG16 architecture. Moreover, we achieved great performance in the CNN portion, which is 0.824538 during the training time period and 0.801161 for validation time. We set batch size 16 and implemented the SGD optimization technique in that CNN period. After that, we gave concern about the RNN part and achieved 0.889419 accuracies during training time with the implemented NAdam optimization technique. Finally, we combined both models and trained the full system, and our accuracy hit 0.917895 for training and 0.768651 for validation. In Table 3, we represented our BNLIT dataset evaluation test score using the effects of BLEU-1, BLEU-2, BLEU-3, BLEU-4, and METEOR, respectively.

4. Conclusion

In this study, we proposed a complex neural network with sufficient deep structure that efficiently generates a Bangla natural language sentence by comprehending intricacies in the contents of an image. We implemented our proposed model using the combination of CNN, RNN, and LSTM architecture and obtained benchmark accuracy for our self-made dataset. Furthermore, we implemented and represented the classification of our full dataset along with the annotation portion implemented by RNN, which is crucial for the image to text generation. Then we combined both architecture and achieved a benchmark result for BNLIT. Our analysis with the model indicates that improved implementation through more comprehensive databases may be achieved by methods to improve model fine-tuning and engineering. Moreover, we have an intension in the future to improve accuracy by implementing an extraordinary neural image captioning model and object detection.

Acknowledgment

The authors are thankful to the Department of Computer Science and Engineering at the University of Liberal Arts Bangladesh for their research lab and other crucial technical support along with the Department of Statistics at Technische Universität Dortmund, Germany.

Declarations

Author contribution. All authors contributed equally to the main contributor to this paper. All authors read and approved the final paper.

Funding statement. None of the authors have received any funding or grants from any institution or funding body for the research.

Conflict of interest. The authors declare no conflict of interest.

Additional information. No additional information is available for this paper.

References

[1] T. Yao, Y. Pan, Y. Li, and T. Mei, "Exploring visual relationship for image captioning," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2018, doi: 10.1007/978-3-030-01264-9_42.

[2] S. Sharma, D. Suhubdy, V. Michalski, S. E. Kahou, and Y. Bengio, "ChatPainter: Improving text to image generation using dialogue," in *6th International Conference on Learning Representations, ICLR 2018 - Workshop Track Proceedings*, 2018, available at: Google Scholar.

[3] K. Fu, J. Jin, R. Cui, F. Sha, and C. Zhang, "Aligning Where to See and What to Tell: Image Captioning with Region-Based Attention and Scene-Specific Contexts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2321–2334, Dec. 2017, doi: 10.1109/TPAMI.2016.2642953.

[4] L. Chen *et al.*, "SCA-CNN: Spatial and Channel-Wise Attention in Convolutional Networks for Image Captioning," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 6298–6306, doi: 10.1109/CVPR.2017.667.

[5] H. Wang, Y. Zhang, and X. Yu, "An Overview of Image Caption Generation Methods," *Comput. Intell. Neurosci.*, vol. 2020, pp. 1–13, Jan. 2020, doi: 10.1155/2020/3062706.

[6] T.-H. Chen, Y.-H. Liao, C.-Y. Chuang, W.-T. Hsu, J. Fu, and M. Sun, "Show, Adapt and Tell: Adversarial Training of Cross-Domain Image Captioner," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 521–530, doi: 10.1109/ICCV.2017.64.

[7] J. Aneja, A. Deshpande, and A. G. Schwing, "Convolutional Image Captioning," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5561–5570, doi: 10.1109/CVPR.2018.00583.

[8] F. Fang, H. Wang, Y. Chen, and P. Tang, "Looking deeper and transferring attention for image captioning," *Multimed. Tools Appl.*, vol. 77, no. 23, pp. 31159–31175, Dec. 2018, doi: 10.1007/s11042-018-6228-6.

[9] M. A. Jishan, K. R. Mahmud, and A. K. Al Azad, "Natural language description of images using hybrid recurrent neural network," *Int. J. Electr. Comput. Eng.*, vol. 9, no. 4, p. 2932, Aug. 2019, doi: 10.11591/ijece.v9i4.pp2932-2940.

[10] Q. Wang and A. B. Chan, "Cnn+ cnn: Convolutional decoders for image captioning," *arXiv Prepr. arXiv1805.09019*, 2018, available at : Google Scholar.

[11] P. Anderson *et al.*, "Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6077–6086, doi: 10.1109/CVPR.2018.00636.

[12] M. A. Jishan, K. R. Mahmud, and A. K. Al Azad, *Bangla Natural Language Image to Text (BNLIT)*, 2020, doi: 10.17632/ws3r82gnm8.4.

[13] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene Parsing through ADE20K Dataset," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 5122–5130, doi: 10.1109/CVPR.2017.544.

[14] T. Xu *et al.*, "AttnGAN: Fine-Grained Text to Image Generation with Attentional Generative Adversarial Networks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1316–1324, doi: 10.1109/CVPR.2018.00143.

[15] K. H. Lee, X. Chen, G. Hua, H. Hu, and X. He, "Stacked Cross Attention for Image-Text Matching," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, pp. 201-216, 2018, doi: 10.1007/978-3-030-01225-0_13.

[16] Y. Zhu *et al.*, "Texygen: A Benchmarking Platform for Text Generation Models," in *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, 2018, pp. 1097–1100, doi: 10.1145/3209978.3210080.

[17] M. Hussain, J. J. Bird, and D. R. Faria, "A Study on CNN Transfer Learning for Image Classification," *Advances in Computational Intelligence Systems*, vol. 840, pp. 191–202, 2019, doi: 10.1007/978-3-319-97982-3_16.

[18] L. Zhu, Y. Chen, P. Ghamisi, and J. A. Benediktsson, "Generative Adversarial Networks for Hyperspectral Image Classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 9, pp. 5046–5063, Sep. 2018, doi: 10.1109/TGRS.2018.2805286.

[19] I. Dhall, S. Vashisth, and S. Saraswat, "Text Generation Using Long Short-Term Memory Networks," *Micro-Electronics and Telecommunication Engineering*, vol. 106, pp. 649–657, 2020, doi: 10.1007/978-981-15-2329-8_66.

[20] C. Rebuffel, L. Soulier, G. Scoutheeten, and P. Gallinari, "A Hierarchical Model for Data-to-Text Generation," *Advances in Information Retrieval*, vol. 12035, pp. 65–80, 2020, doi: 10.1007/978-3-030-45439-5_5.

[21] T. Qiao, J. Zhang, D. Xu, and D. Tao, "MirrorGAN: Learning Text-To-Image Generation by Redescription," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 1505–1514, doi: 10.1109/CVPR.2019.00160.

[22] L. Ma, Q. Sun, S. Georgoulis, L. Van Gool, B. Schiele, and M. Fritz, "Disentangled Person Image Generation," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018, pp. 99-108, doi: 10.1109/CVPR.2018.00018.

[23] S. Bai and S. An, "A survey on automatic image caption generation," *Neurocomputing*, vol. 311, pp. 291–304, Oct. 2018, doi: 10.1016/j.neucom.2018.05.080.

[24] Z. Zhang, Y. Xie, and L. Yang, "Photographic Text-to-Image Synthesis with a Hierarchically-Nested Adversarial Network," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6199–6208, doi: 10.1109/CVPR.2018.00649.

[25] E. Laloy, R. Hérault, D. Jacques, and N. Linde, "Training-Image Based Geostatistical Inversion Using a Spatial Generative Adversarial Neural Network," *Water Resour. Res.*, vol. 54, no. 1, pp. 381–406, Jan. 2018, doi: 10.1002/2017WR022148.

[26] J. Chen and H. Zhuge, “Extractive summarization of documents with images based on multi-modal RNN,” *Futur. Gener. Comput. Syst.*, vol. 99, pp. 186–196, Oct. 2019, doi: 10.1016/j.future.2019.04.045.

[27] W. Xu, H. Sun, C. Deng, and Y. Tan, “TextDream: Conditional Text Generation by Searching in the Semantic Space,” in *2018 IEEE Congress on Evolutionary Computation (CEC)*, 2018, pp. 1–6, doi: 10.1109/CEC.2018.8477776.

[28] J. Xu, X. Ren, J. Lin, and X. Sun, “Diversity-Promoting GAN: A Cross-Entropy Based Generative Adversarial Network for Diversified Text Generation,” in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 3940–3949, doi: 10.18653/v1/D18-1428.

[29] H. Kwon, Y. Kim, H. Yoon, and D. Choi, “CAPTCHA Image Generation Systems Using Generative Adversarial Networks,” *IEICE Trans. Inf. Syst.*, vol. E101.D, no. 2, pp. 543–546, 2018, doi: 10.1587/transinf.2017EDL8175.

[30] T. Jiang, Z. Zhang, and Y. Yang, “Modeling coverage with semantic embedding for image caption generation,” *Vis. Comput.*, vol. 35, no. 11, pp. 1655–1665, Nov. 2019, doi: 10.1007/s00371-018-1565-z.

[31] A. Gatt and E. Krahmer, “Survey of the State of the Art in Natural Language Generation: Core tasks, applications and evaluation,” *J. Artif. Intell. Res.*, vol. 61, pp. 65–170, Jan. 2018, doi: 10.1613/jair.5477.

[32] C.-C. Wu, R. Song, T. Sakai, W.-F. Cheng, X. Xie, and S.-D. Lin, “Evaluating Image-Inspired Poetry Generation,” in *CCF International Conference on Natural Language Processing and Chinese Computing*, 2019, pp. 539–551, doi: 10.1007/978-3-030-32233-5_42.

[33] G. Ding, M. Chen, S. Zhao, H. Chen, J. Han, and Q. Liu, “Neural Image Caption Generation with Weighted Training and Reference,” *Cognit. Comput.*, vol. 11, no. 6, pp. 763–777, Dec. 2019, doi: 10.1007/s12559-018-9581-x.

[34] G. Zhang, F. Wang, and W. Duan, “Study on Star-Galaxy Image Generation Method Based on GAN,” *Xibei Gongye Daxue Xuebao/Journal Northwest. Polytech. Univ.*, vol. 37, no. 2, pp. 315–322, Apr. 2019, doi: 10.1051/jnwpu/20193720315.

[35] Y. Sagawa and M. Hagiwara, “Face image generation system using attribute information with DCGANs,” in *Proceedings of the 2nd International Conference on Machine Learning and Soft Computing - ICMLSC ’18*, 2018, pp. 109–113, doi: 10.1145/3184066.3184071.