ORIGINAL ARTICLE

Modular Co-attention Networks in Nepali Visual Question Answering Systems

Visual question answering has been regarded as a challenging task requiring a perfect blend of computer vision and natural language processing. As no dataset was available to train such a model for the Nepali language, a new dataset was developed during the research by translating the VQAv2 dataset. Then the dataset consisting of 202,577 images and 886,560 questions was used to train an attention-based VQA model. The dataset consists of yes/no, counting, and other questions with primarily one-word answers. Modular Co-attention Network (MCAN) was applied to the visual features extracted using the Faster RCNN framework and question embeddings extracted using the Nepali GloVe model. After co-attending the visual and language features for a few cascaded MCAN layers, the features are fused to train the whole network. During evaluation, an overall accuracy of 69.87% was obtained with 81.09% accuracy in yes/no type questions. The results surpassed the performance of models developed for Hindi and Bengali languages. Overall, novel research has been done in the Nepali Language VQA domain paving the way for further advancements.

KEYWORDS

Abbreviations: VQA, Visual Question Answering; MCAN, Modular Co-Attention Network; RCNN, Region Convolutional Neural Network.

Nepali Visual Question Answering, *Nepali VQA Dataset*, Modular Co-attention

1 | INTRODUCTION

Visual Question Answering is a fruitful combination of computer vision and natural language processing. It enables a machine to concurrently interpret images and text. That will allow machines and humans to exchange information in a humane way.

Leveraging artificial intelligence, VQA has demonstrated its value to both the general public and tech-related enterprises. Applications range from e-learning tasks and visual aids for the visually handicapped [1] to chatbots and content retrieval systems [2]. Nonetheless, the primary obstacle associated with VQA systems is their linguistic specificity. This indicates that questions posed in any other language—such as Nepali—cannot be answered by the VQA model that was trained for that language, say English. The language of Nepali itself has to be trained on a different approach.

The Nepali language has not yet had a VQA system created for it. To train such a model, there was no VQA dataset available in the Nepali language. Furthermore, it is commonly recognized that deep learning models are data-hungry, meaning they need a sufficient quantity of training data in order for all of the model's parameters to learn correctly. As a result, the VQA system's possible applications for the Nepali language were not investigated. Therefore, training a Nepali VQA model requires a well-prepared dataset.

The Nepali VQA model, which was created by following the established goals, can identify and quantify the things in the image that the questions are referring to. Nevertheless, the model generated has several limitations because of certain translation mistakes and the shortcomings of the Nepali word encoding system.

The major contributions are:

- Composing a Nepali VQA dataset by translating an English VQA dataset.
- Training a VQA model on the Nepali VQA dataset prepared, and evaluating the model based on different question types.

Section 2 provides a brief review of related works. Then a description of the dataset composed is presented in Section 3. A detailed description of the proposed methodology is provided in Section 4. Section 5 presents different parameters used to train three models and the results hence obtained. Analysis of the results is done in Section 6 and Section 7 concludes the paper.

2 | RELATED WORKS

2.1 | Combining Image and Language Features

Image captioning models were the first attempt to integrate visual and linguistic elements to address image-text difficulties. Using artificial intelligence, Oriol Vinyals et al. [3] created a model to characterize a picture. Using a knowledge base, a system created by Somak Aditya, et al. [4] derived scene description graphs from images.

2.2 | Beginning of Visual Question Answering

Question-answering systems that could respond to queries concerning a paragraph of text began to take shape. To train such models, high-quality datasets such as SQuAD [5] were used. But the issues were exclusive to one mode, i.e., text alone.

By including a visual channel, the question-answering jobs' scope was significantly expanded. The initial VQA model, referred to as the Vanilla VQA, was proposed by Aishwarya Agrawal and colleagues [6]. By presenting a system that integrated verbal and visual modes to respond to inquiries about a picture, this model established a significant milestone in the field of artificial intelligence.

2.3 | Attention and VQA

The transformer encoder-decoder model was presented by Ashish Vaswani and colleagues [7], with the aim of optimizing the attention mechanism. For the majority of the neural network models created since then, it has served as the foundation. To create visual question-answering models, Qi Wu, et al. [8] employed a few attention mechanisms, including Hierarchical Co-Attention (HieCoATT) [9].

The authors Zhou Yu and co [10] proposed a deep modular co-attention network to improve the effectiveness and efficiency of the VQA system. By cascading such co-attention networks, the researchers claimed state-of-the-art performance on the VQAv2 dataset. After employing self-attention for language and picture aspects independently, a directed attention mechanism was used to correlate the keywords with the important areas.

Every deep learning domain has been positively influenced by the employment of transformer models. In several computer vision tasks, Vision Transformers (ViTs) have outperformed CNNs[11]. Furthermore, by employing the concept of transformers as its foundation, Bidirectional Encoder Representations from Transformers (BERTs) [12] have shown to be a more effective and reliable language model.

With their Pathways Language and Image model (PaLI), Xi Chen, et al. [13] achieve the state-of-the-art performance in the VQAv2 dataset as of right now. With 17 billion parameters, the pre-trained encoder-decoder based language models and vision transformer are used. In a similar vein, Wenhui Wang et al. [14] and Hangbo Bao et al. [15] created the models BEiT and VLMO, respectively, employing deep layers of transformers to improve performance on the VQAv2 dataset. However, the lack of funding has prevented this study from constructing extremely complex networks with billions of parameters.

2.4 | Hindi and Bengali VQA

To assess performance, a Nepali VQA model may be compared against models in Bengali and Hindi. To create a VQA system for the Hindi language, Deepak Gupta and colleagues [16] created a mix of multilingual and code-mixed VQA assignments. A suitable combination of characteristics at the object and picture levels produced good results. By Google translating the VQAv1 dataset, they were able to collect the dataset needed to train their model.

SM Shahriar Islam, et al. [17] created and employed two Bengali datasets, the Bengai CLEVR dataset and the Bengali VQA dataset, to create a VQA model for the Bengali language. To get these, they used Google to translate the VQAv1 dataset and the CLEVR dataset [18]. However, the dataset's inadequate size put a barrier to the task.

Using a human-annotated dataset, Mahamudul Hasan Raf, et al. [19] attempted a top-down attention-based strategy in the Bengali VQA domain. To train a true VQA model, however, the dataset was insufficient because it only included 'yes/no' type questions taken from the VQAv2 dataset.

It was noted that there were more English-language VQA models created to date during the examination of relevant studies in the VQA area. Additionally, similar VQA models have been implemented for a few additional non-English languages. Nevertheless, there is currently no equivalent system in place for the Nepali language. A Nepali VQA model could not be trained using any dataset. A VQA model was created in Nepali, and a dataset was generated in the same language in an attempt to close the research gap.

3 | DATASET USED

An open dataset called VQAv2 [20] includes open-ended questions and responses pertaining to pictures. To create the second version, extra questions, and answers have been added to the previous version, known as VQAv1 [6]. It is made up of 204, 721 actual photos that were taken from CoCo [21]. In addition, those photographs are the subject of 1.1 million inquiries and 11 million responses.

To acquire the Nepali VQA dataset, the English dataset was translated using Google. Following translation, the number of questions, responses, and pictures in the final dataset was decreased by filtering the Nepali questions and answers according to whether or not each word had Nepali GloVe embeddings.

1. 202,577 images
2. 886,560 questions
3. 478,108 answers
4. 4.37 questions per image on average

3.0.1 | Composition of Nepali VQA Dataset

Different types of questions and their composition in the dataset are as follows:

1. Yes/No Questions: 45.83%
2. Other: 44%
3. Counting: 10.17%

The images, questions, and answers or annotations are stored in separate locations and are linked with their respective IDs. Questions and answers are kept in two JSON files.

3.0.2 | Question Length

- Most frequent Length: 4 words
- Average Length 4.19 words
- Maximum Length: 13 words
- Minimum Length: 1 word

3.0.3 | Answer Length

- Most frequent Length: 1 word

- Average Length 1.067 words
- Maximum Length: 7 words (1 answer)
- Minimum Length: 0 words

Figures 1 and 2 display the distribution of question and answer lengths in the final Nepali dataset.

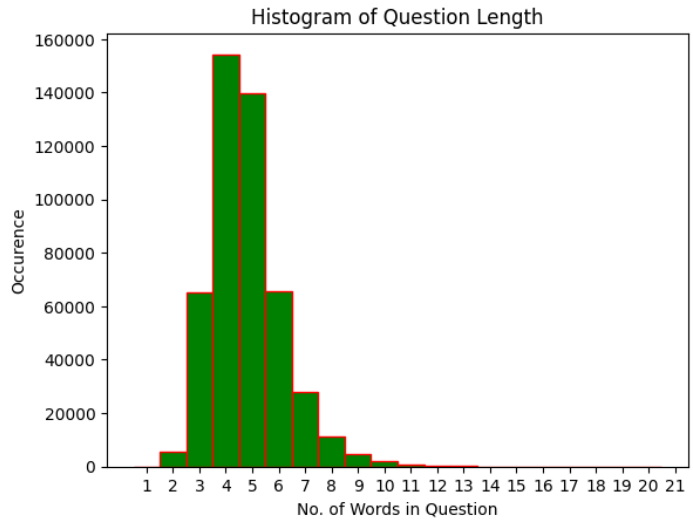

FIGURE 1 Question Length Distribution

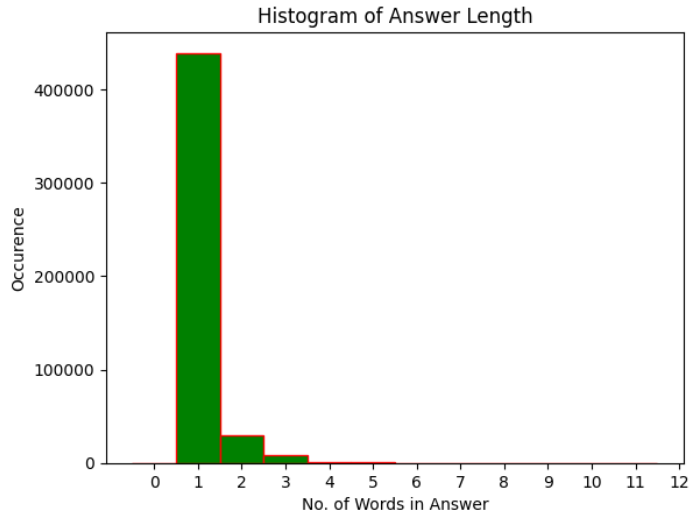

FIGURE 2 Answer Length Distribution

Following is the format for the question file:

```
{   "info" :    info,
    "task_type" :    str,
    "data_type" :    str,
    "data_subtype" :    str,
    "questions" :    [question],
    "license" :    license    }
```

Where, task_type means type of annotations i.e. OpenEnded, data_type refers to the source of image data i.e. MsCOCO or Abstract images, and data_subtype indicates whether it is train, val, or test data.
The questions are listed under the 'questions' section in the following format:

```
question{"question_id" :    int,
         "image_id" :    int,
         "question" :    str    }
```

Following is the format for the answer file:

```
{   "info" :    info,
    "data_type" :    str,
    "data_subtype" :    str,
    "annotations" :    [annotation],
    "license" :    license    }
```

```
annotation{"question_id" :    int,
           "image_id" :    int,
           "question_type"" :    str,
           "answer_type" :    str,
           "answers" :    [answer],
           "multiple_choice_answer" :    str    }
```

```
answer{"answer_id" :    int,
       "answer" :    str,
       "answer_confidence" :    str    }
```

A preview of the Nepali VQA dataset created and utilized in this study is provided by Figures 3, 4, and 5. Those figures demonstrate a sample of questions and answers in the dataset in the format as described. Similarly, a few images present in the dataset are also demonstrated in Figure 5.

```
{"info": {"description": "This is v2.0 of the VQA dataset.", "url": "http://visualqa.org", "versio
"questions": [{"image_id": 458752, "question": "यो फोटो के हेरेर खिचिएको हो ?", "question_id": 458752000},
{"image_id": 458752, "question": "यो मान्छे कुन पोजिसन खेल्दै छ?", "question_id": 458752001},
{"image_id": 458752, "question": "खेलाडीको जर्सी कस्तो रंगको हुन्छ ?", "question_id": 458752002},
{"image_id": 458752, "question": "के यो मानिस पेशेवर बेसबल खेलाडी हो?", "question_id": 458752003},
{"image_id": 262146, "question": "हिउँ कस्तो रङ हो?", "question_id": 262146000},
{"image_id": 262146, "question": "मान्छे के गर्दै छ?", "question_id": 262146001},
{"image_id": 262146, "question": "व्यक्तिको हेडवेयर कस्तो रंगको हुन्छ?", "question_id": 262146002},
{"image_id": 524291, "question": "व्यक्तिको हातमा के छ?", "question_id": 524291000}, {"image_id": 524291, "qu
```


FIGURE 3 Questions of Dataset


```
{"info": {"description": "This is v2.0 of the VQA dataset.", "url": "http://visualqa.org", "version": "2.0", "y
"annotations": [{"question_type": "what is this", "multiple_choice_answer": "नेट",
"answers": [{"answer": "नेट", "answer_confidence": "maybe", "answer_id": 1},
{"answer": "नेट", "answer_confidence": "yes", "answer_id": 2},
{"answer": "नेट", "answer_confidence": "yes", "answer_id": 3},
{"answer": "जाल", "answer_confidence": "yes", "answer_id": 4},
{"answer": "नेट", "answer_confidence": "yes", "answer_id": 5},
{"answer": "नेट", "answer_confidence": "yes", "answer_id": 6},
{"answer": "जाल", "answer_confidence": "maybe", "answer_id": 7}, {"answer": "नेट", "answer_confidence": "yes", "ar
"answer_type": "other", "question_id": 458752001}, {"answer_type": "other", "multiple_choice_answer": "सुन्तला",
"answers": [{"answer": "सुन्तला", "answer_confidence": "yes", "answer_id": 1}, {"answer": "सुन्तला", "answer_confidenc
```


FIGURE 4 Annotations of Dataset

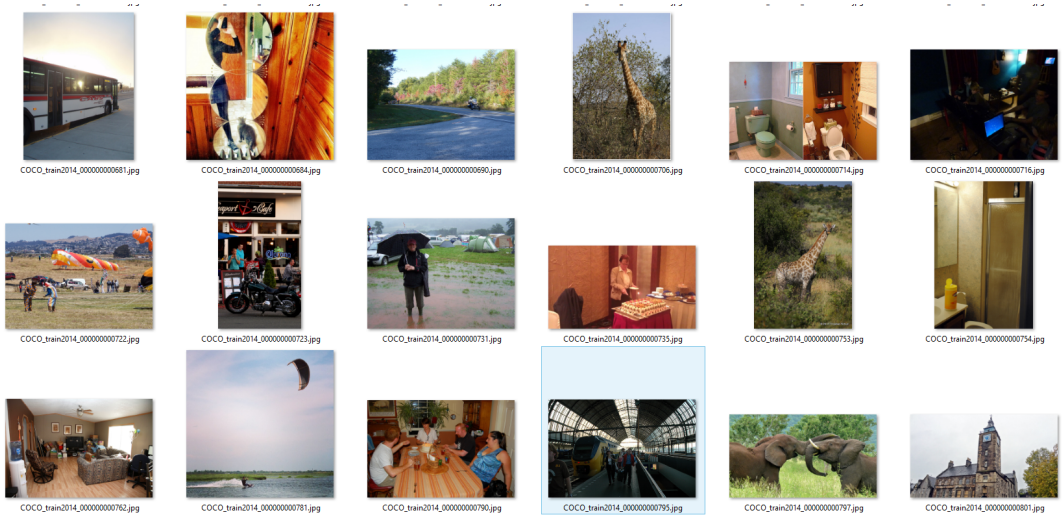

FIGURE 5 Images of Dataset

4 | METHODOLOGY

A VQA system receives an image and a question as input, uses the picture to determine the applicable response, and outputs the answer. A deep learning architecture's several steps come together to provide an output that precisely

responds to the query posed in relation to the provided picture. The purpose of this thesis is to use modular co-attention networks (MCAN) to construct a Nepali language VQA system.

4.1 | Description of Algorithms

Various algorithms process the input from earlier stages of the pipeline to produce outputs that are inputs for later stages. Among the algorithms employed in the creation of the Nepali VQA model are:

Faster RCNN:

For object detection in a picture, the deep-learning architecture called Faster Region Convolutional Network[22] is used. It puts into practice two networks: the Object Detection Network and the Region Proposal Network (RPN). RPN uses the feature map that was acquired with backbone CNN to produce region suggestions. Region proposals show which regions may hold things. After making corrections to the bounding box, the object detection network assigns one of the object classes to the region.

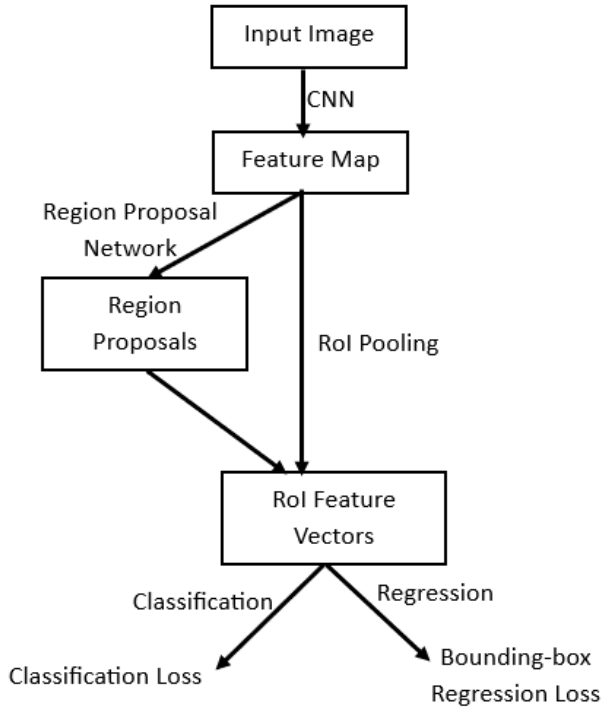

FIGURE 6 Faster RCNN Flow diagram

The faster R-CNN model incorporates two networks:

1. Region Proposal Network (RPN)
2. Object Detection Network

Faster RCNN improves the speed by training the backbone CNN and RPN at once.

1. Region Proposal Network
   The image feature map produced by the backbone CNN is fed to RPN. It generates region proposals from the feature map. Region proposals are the anchors that indicate the regions that possibly contain objects.
   For training RPNs, there are two sorts of losses, classification loss, and regression loss. Classification loss is for classification tasks where the proposed region is classified among the possible object classes. Besides, regression loss is for regressing the numerals of the bounding box that encloses the object.
2. Object Detection Network
   The object detection network basically corrects the possible object boundary detected by RPNs. The use of the Region of Interest pooling method projects the proposed regions onto the features. In this network too, classification loss and bounding box regression loss are used for the same purpose as for RPNs.
   Actually, the classification part of the Faster RCNN network is not required during feature extraction. It is required just for training the Faster RCNN model. During feature extraction from an image, the features from each detected region of the image are obtained and stacked in the form of a matrix.

Nepali GloVe Model:

For every Nepali word in questions and answers, a vector representation is provided by the GloVe model NepVec1 [23], which was trained on a Nepali corpus. The GloVe mechanism equilibrates the benefits of local and global word embedding generation techniques [24]. The matrix factorization technique is employed by GloVe to produce word embeddings. Initially, a co-occurrence matrix is made, in which the rows indicate the frequency of occurrences of each word and the columns indicate the frequency of occurrences of the word in a context, or around other words. After that, the matrix is factorized into a dense, lower-dimensional matrix where each row denotes the corresponding word's embedding.

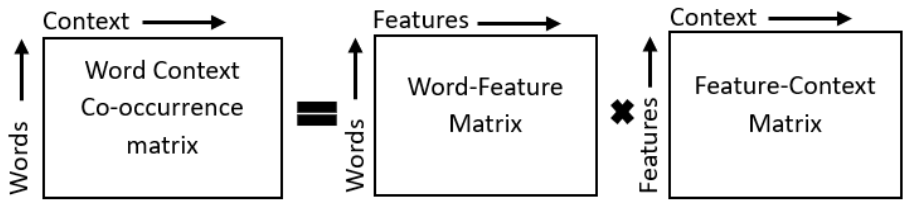

FIGURE 7 GloVe Matrix Decomposition

Modular Co-Attention:

It describes the fusion of directed attention and self-attention, two different types of attention units. MCA is necessary in order to connect the question's keywords to the image's important areas. While the main areas of an image are gained by applying self-attention to it, the keywords of a question are obtained by applying self-attention to the question. Lastly, a directed attention method is used to link the images, with the question features directing the focus over the picture characteristics.

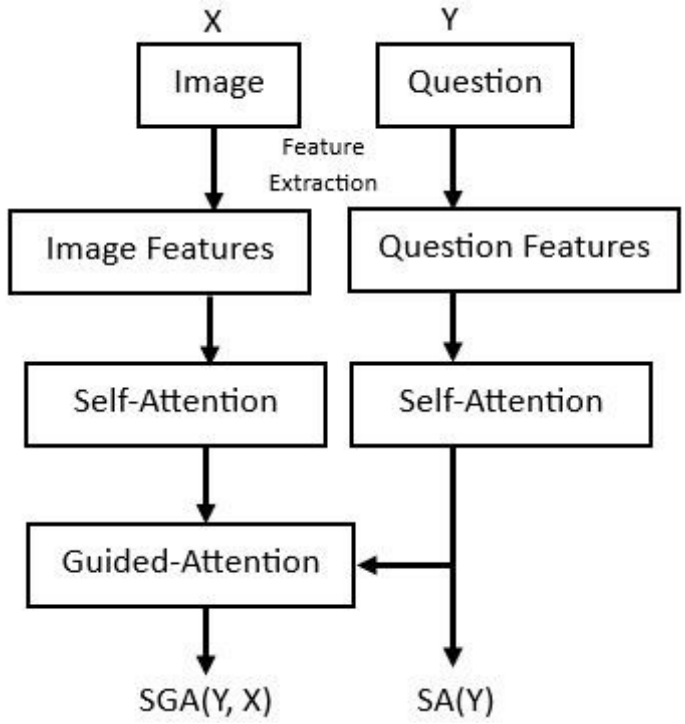

FIGURE 8 MCA Flow Diagram

The query (Q), key (K), and value (V) vectors for the inputs are created using the corresponding weight matrices for an input matrix X provided as $X = x_1, x_2, ..x_n$.

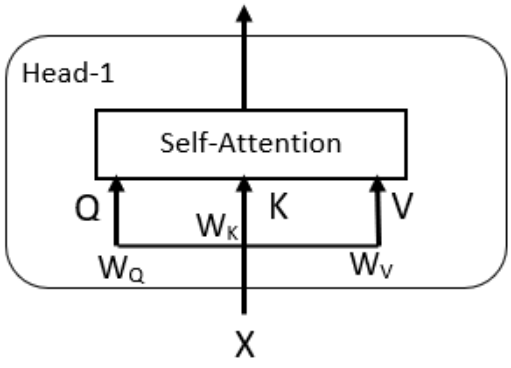

FIGURE 9 Self-Attention

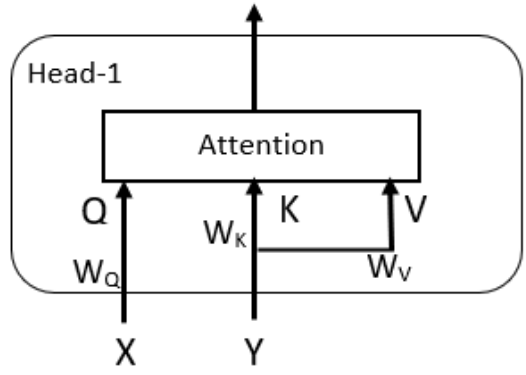

FIGURE 10 Guided-Attention

It means,

$$Q = XW_Q, K = XW_K, V = XW_V \tag{1}$$

The attention Score is provided as:

$$AttentionScore(A)' = Softmax\left(\frac{QK^T + Mask}{\sqrt{QuerySize}}\right) \times V \tag{2}$$

Multi-Head Attention (MHA) is the result of calculating and concatenating the attention ratings for each of the 'h' heads.

$$MHA = Concatenate(A_1, A_2, ..., A_h)W \tag{3}$$

Multimodal Fusion:

In this instance, the algorithm is just necessary for combining the characteristics from several modes or channels, pictures, and language. The last layer of the co-attention network's output visual and question characteristics are sent into the fusion algorithm. Each of the picture areas and question sentences has a different distribution of characteristics. As a result, for each channel, the features of many words and picture regions are first condensed into a single attended feature. Next, each channel's attended feature is sent to a fully linked layer. Fused feature representation can be obtained by combining the outputs of the two FC layers. Prior to merging the features, attended features for channels Y and X, respectively, are collected.

Mathematical formulation to obtain attended feature is given as,

$$\alpha = softmax(MLP(X_L)) \tag{4}$$

In the Equation 4, $\alpha = [\alpha_1, \alpha_2, ...\alpha_m]$ are the weights of learned attention.

$$x' = \sum_{i=1}^{m} \alpha_i x_{iL} \tag{5}$$

Similar is the case for finding the attended feature y'.

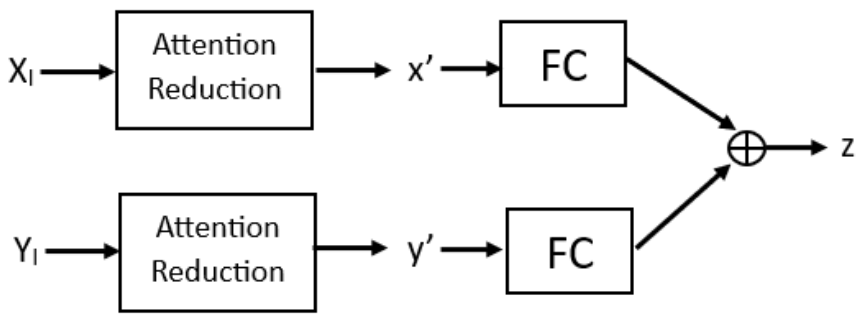

FIGURE 11 Features Fusion

The two attended features, y' and x', are now sent into the fusion function shown in Equation 6, which fuses them into a single fused feature, z.

$$z = LayerNorm(W_x^T x' + W_y^T y') \tag{6}$$

4.1.1 | Working Principle

The system's working mechanism consists of a pipeline and a series of steps that enable it to learn the textual and visual elements and model their relationships using the labeled dataset that is provided to it. After that, the generated system may be used to forecast future occurrences.

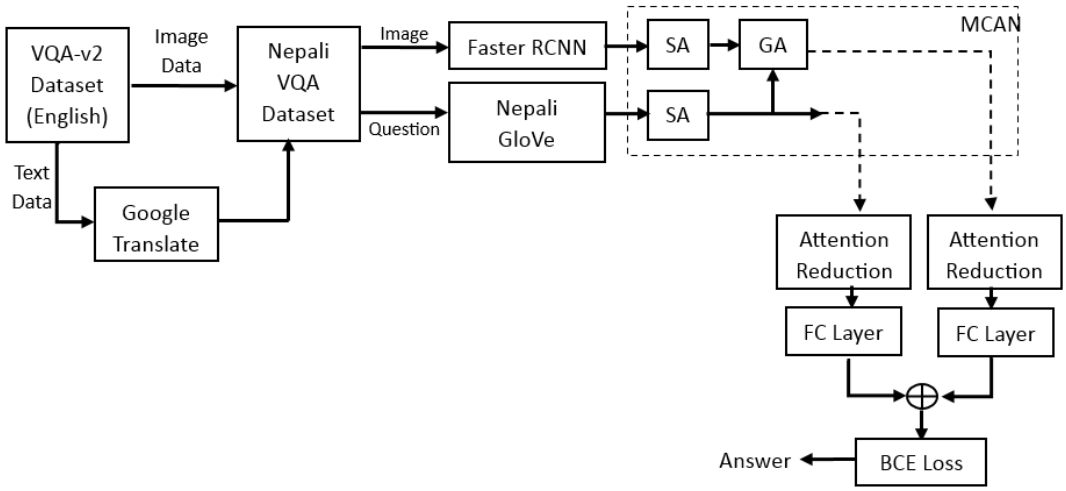

FIGURE 12 Overall Working Principle

The overall working procedure is divided into three sections:

Text and Image Encoding by their features. For text, 300-dimensional GloVe embeddings are employed. Using a pre-trained Faster RCNN model with a bottom-up attention mechanism [25] and a ResNet-101 backbone, 2048-dimensional features were obtained for each area of an image.

Co-attention Network for Training in which layers are used to attend the textual and visual features among themselves and between each other.

Classifier Training after Fusion of Features which takes in the attended features of visual and textual channels and combines them into a single feature representation. After feeding the combined feature to a sigmoid function, an N-dimensional vector is produced, where N is the number of most likely (or most frequent) responses in the training set. It denotes that a binary cross-entropy (BCE) loss function is used during classifier training.

4.2 | Verification and Validation Procedures

The phases of validation and verification are when the model's performance is assessed. To conduct validation, test and validation subsets of the entire dataset must be allocated.

The most appropriate assessment metric for the VQA problem is accuracy because the entire challenge has been examined as a classification assignment. The percentage of the total number of questions that the model properly predicts as responses is known as accuracy. For the VQAv2 dataset with 10 human-annotated answers for a question, accuracy is introduced as:

$$Accuracy(Ans) = min\left\{\frac{No.of\,HumansThatAnnotatedAns}{3}, 1\right\} \tag{7}$$

5 | EXPERIMENTS AND RESULTS

5.1 | Models Prepared

Using the supplied Nepali VQA dataset, three models have been created. Certain model parameters and hyperparameters are shared by all models, while others are unique.

5.1.1 | Common Hyperparameters

Visual Features Parameters

TABLE 1 Visual Features parameters

Image Feature Size	2048
Image Feature Pad Size	100
No. of Features	10-100

Image feature size actually refers to the size of the feature for each detected region in an image. The overall feature from an image is a collection of features of each region stacked in the form of a matrix. Even though 10-100 features are obtained from an image, padding is done to maintain the fixed size of 14x2048 for the feature matrix.

Textual Features Parameters

TABLE 2 Textual Features parameters

Maximum Tokens	14
Word Embedding Type	GloVe
Embedding Size	300

Even though the maximum number of tokens is set to 14, every question may not have the same number. Thus to maintain consistency, padding is done such that the textual feature matrix has a size of 14x300 in each case.

Architectural Parameters

TABLE 3 Architectural parameters

Feed Forward Size	2048
Flat Out Size	1024
Hidden Size	512
No. of Heads for MHSA	8

Optimizer Parameters

TABLE 4 Optimizer parameters

Optimizer	Adam
Learning Rate	0.0001
LR Decay Rate	0.2
Betas	0.9, 0.98

5.1.2 | Differing Parameters

Textual Parameters

TABLE 5 Differing Textual parameters

Parameter	Model 1	Model 2	Model 3
No. of Answer Classes	2505	3439	2505
Answer Frequency	≥5	≥3	≥5

Architectural Parameters

An experiment is done by increasing the depth of model 3. In such case dropout rate is increased such as to overcome the risk of overfitting due to much complex model.

TABLE 6 Differing Architectural Parameters

Parameter	Model 1	Model 2	Model 3
No. of MCAN Layers	6	6	8
Dropout Rate	0.1	0.1	0.2

5.2 | Model Training

Table 7 illustrates the composition of the dataset used to train the models:

TABLE 7 Test-Train-Val Split

	No. of Questions	No. of Images
Training Set	323,414	81,478
Validation Set	154,694	39,830
Test Set	408,452	81,269
Total	886,560	202,577

Train to Val ratio: 2.091:1

All of the models were trained using a training batch size of 256. Following the limitations placed on memory resources and training time determines the size of the training batch. The proper batch size for training and validation was established because of the previously mentioned factors.

Plotting the model's loss at each training step against the step (or epoch) number is known as a training loss curve. The step number is just the total number of iterations required to train an entire epoch. In Figures 13, 14, and 15, the training loss curves for each of the trained models are displayed.

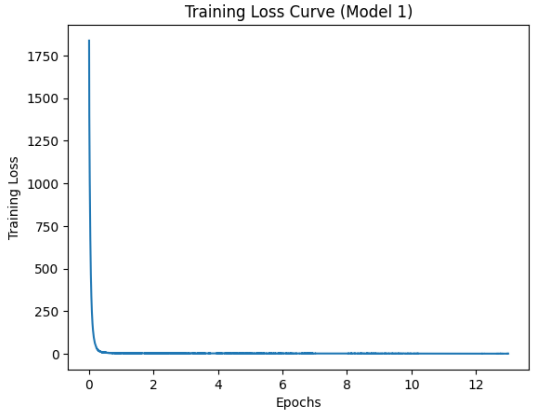

FIGURE 13 Training Loss Curve (Model 1)

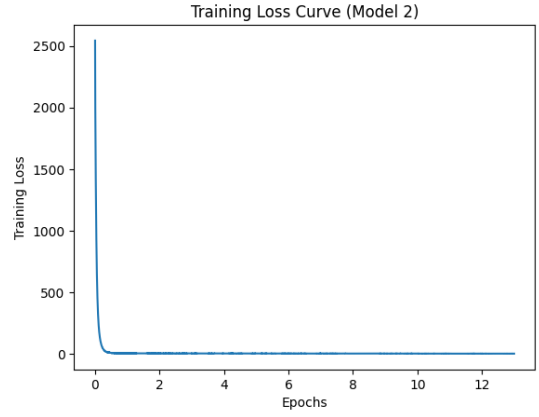

FIGURE 14 Training Loss Curve (Model 2)

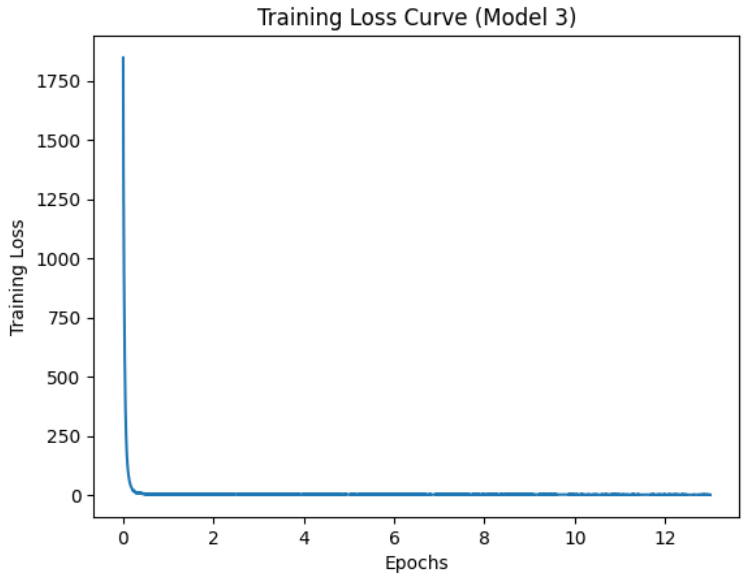

FIGURE 15 Training Loss Curve (Model 3)

Every model has a loss curve that resembles one another. Up to the second epoch, all of the models' loss values showed a clear downward trend. The loss figures appear to have stabilized after that. Examining the actual loss figures, however, revealed that the losses were actually declining up to the 13th epoch. But the decrease in loss values is too little to be seen graphically.

Given such saturation, it may appear that the model reached its peak learning between the second and third epochs, at which point it began to overfit. However, as the validation accuracy curves show, the model was still learning and generalizing effectively, thus this was not the case. Despite the saturated training loss, the accuracies as measured in the validation set were rising.

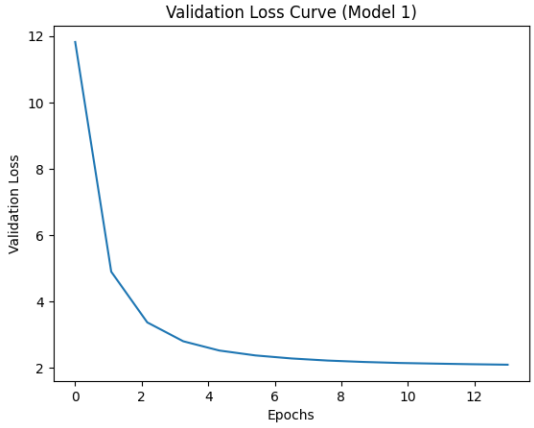

FIGURE 16 Validation Loss Curve(Model 1)

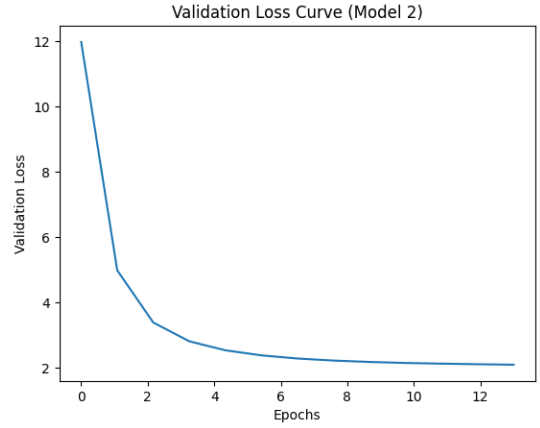

FIGURE 17 Validation Loss Curve(Model 2)

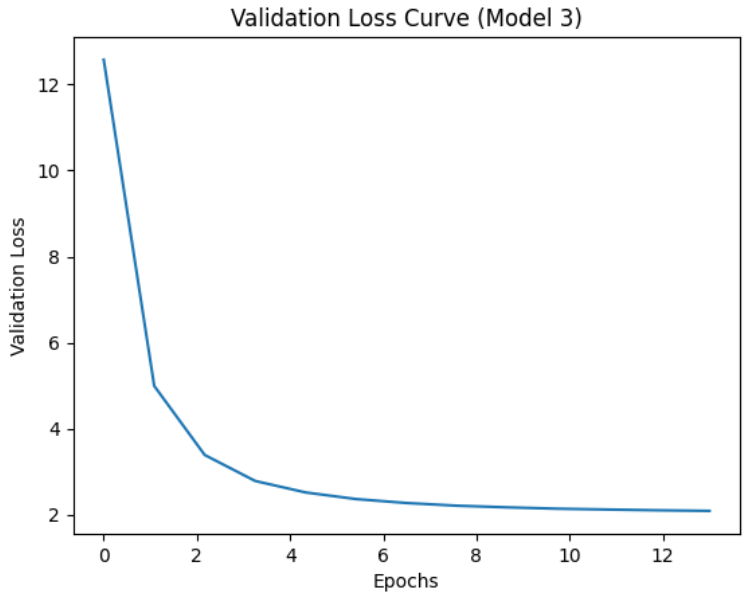

FIGURE 18 Validation Loss Curve (Model 3)

5.3 | Model Evaluation

Three kinds of questions were used to test the constructed models: yes/no, counting, and other. After the 13th period was over, the three models' combined accuracies were also determined.

Table 8 presents the evaluation and comparison of accuracy.

TABLE 8 Accuracy Table

Acc. (%) vs Qn. Type	Model 1	Model 2	Model 3
Overall	**69.87**	69.83	69.58
Yes-No	80.89	**81.09**	80.80
Other	**62.20**	62.04	61.71
Counting	**53.18**	52.64	52.88

Based on the assessed accuracy measure, Table 8 shows that the three models perform quite similarly to each other. For all question kinds, each model's accuracy is about the same.

For the three models—yes-no, other, and counting type questions—as well as the total accuracy, accuracy vs. epoch curves are presented.

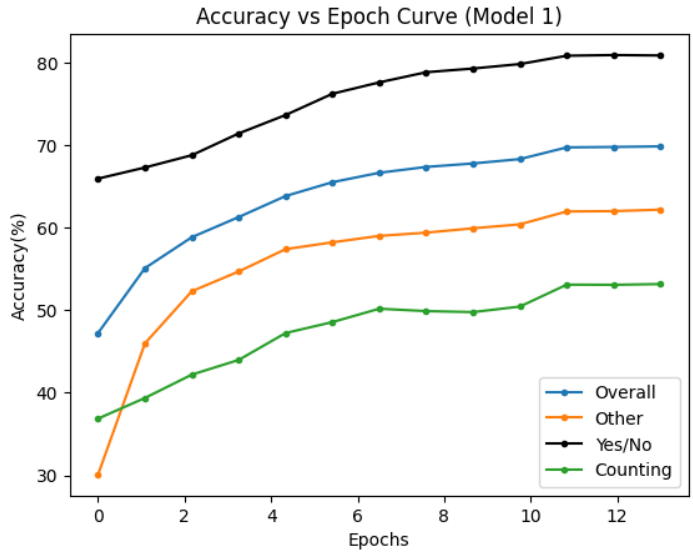

FIGURE 19 Accuracy vs Epoch Curve (Model 1)

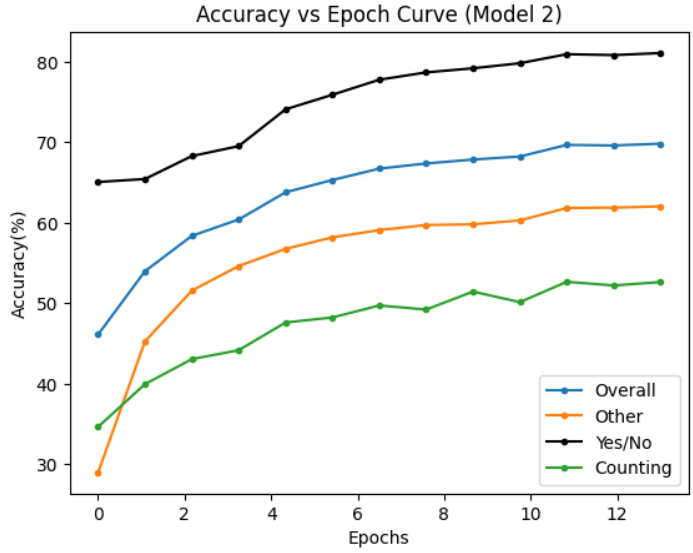

FIGURE 20 Accuracy vs Epoch Curve (Model 2)

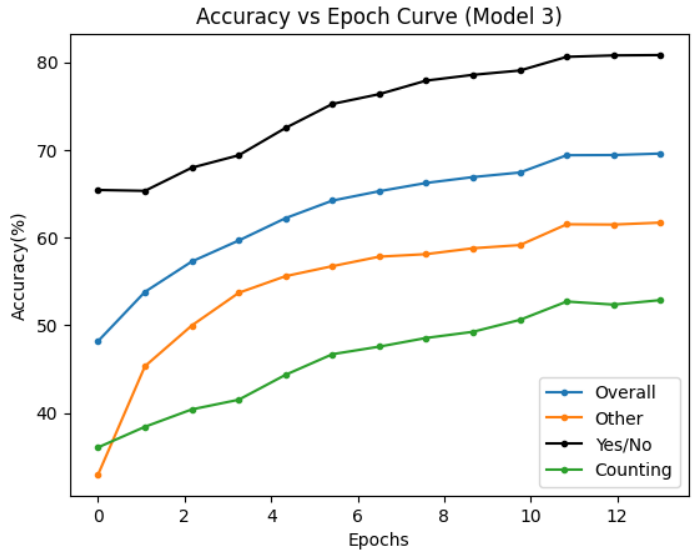

FIGURE 21 Accuracy vs Epoch Curve (Model 3)

It was found that the accuracy values for each question type for every model tend to somewhat saturate after the 11th epoch. On inspecting the saturation of training and validation loss curves at those epochs, it seemed that the models had trained enough and were not learning much after that. Thus, to prevent the model from overfitting the training of all the models was halted after the 13th epoch.
Furthermore, the accuracy plot of the 'other' type question and 'counting' type question overlap for each model. It is because the accuracy plot for 'other' begins with a very low value at the first epoch since there is a larger number of possible targets for 'other' type questions than the rest question types. As the training proceeds, the model learns well and the accuracy values for the 'other' type question increase.

5.4 | Results on Test Dataset

All three models were inferred using the test dataset. The anticipated response is shown next to the corresponding question and image to provide a clearer illustration of the prediction. Sample predictions made after inferencing Model 1 to the test dataset are displayed in Figures 24, 23, 22, and 25.

Question: यो कस्तो प्रकारको घटना हो?

Predicted Answer: खाने

FIGURE 22 'Other' Question

Question: उसको हातमा कुनै औंठी छ?

Predicted Answer: हो

FIGURE 23 'Yes' Question

Question: के मानिस बाइक चलाउँदै हुनुहुन्छ?

Predicted Answer: छैन

FIGURE 24 'No' Question

Question: कति जिराफ देखाइन्छ?

Predicted Answer: पाँच

FIGURE 25 'Count' Question

The model can accept questions of three types namely yes/no type, counting type, and other or general type questions. Besides the output of the model is a short (mostly single word) answer to the question being asked.

5.5 | Evaluation of Translated Dataset

The VQAv2 dataset was translated into Nepali using Google Translate. The dataset's questions and responses were translated into Nepali. It is evident that employing a machine translation system to translate a text from one language to another results in certain inaccuracies in the translated content. Semantic or syntactic mistakes are possible. The performance of the model may suffer as a result of these inaccuracies spreading across it.

There were two translations of each of the 100 English questions in the dataset—one made by human knowledge and the other by Google Translate—into Nepali. Table 9 displays a snapshot of the samples.

TABLE 9 Comparing Google Translate with Human Translation

English Text	Human Translated (Reference)	Google Translated (Predicted)
How many people fit in this room?	यो कोठामा कति जना अट्छन?	यो कोठामा कति जना बस्छन्?
Is the toilet lid up?	शौचालयको ढक्कन माथि छ?	शौचालय ढक्कन छ?
Is this in a motorized vehicle?	के यो मोटर चालित गाडीमा छ?	के यो मोटर चालित गाडीमा छ?
How many children?	कति वटा सन्तान ?	कति सन्तान ?
How many towel racks are in the room?	कोठामा कतिवटा तौलिया र्याकहरू छन्?	कोठामा कतिवटा तौलिया र्याकहरू छन्?
Which room is this?	यो कुन कोठा हो?	यो कुन कोठा हो?
Where is this?	यो कहाँ छ?	यो कहाँ छ?
What is hanging on the wall outside the bathroom?	बाथरूम बाहिरको भित्तामा के झुण्डिएको छ?	बाथरूम बाहिर भित्तामा के झुण्डिएको छ?
What is above the toilet on wall?	शौचालय भन्दा माथिको भित्तामा के छ?	भित्तामा शौचालय भन्दा माथि के छ?
Is there a stove in this photo?	के यो फोटोमा चुलो छ?	के यो फोटोमा चुलो छ?

The computation of Word Error Rate (WER) and BiLingual Evaluation Understudy (BLEU) values was done using these 100 samples. These measurements can offer a broad perspective on how well the Google Translation system is operating.

- WER for 100 samples: 13.69%
- Average BLEU for 100 samples (bi-gram overlap): 0.7965

6 | DISCUSSION AND ANALYSIS

6.1 | Error Analysis

Throughout the assessment and inference stages, the model's predictions for various inputs were noted. Errors occur in many circumstances, which may be categorized according to the circumstances in which they occur.

Translation Error:

This study's dataset preparation step is entirely dependent on translating the questions and responses from English to Nepali. To complete the assignment, the Google Translate library and API were utilized.

1. Ambiguity of terms.
2. Technical Terms and Named Entities
3. Out of Vocabulary Words for GloVe model
4. Improper Yes/No Questions in Nepali

Error in Exactness of Answer:

Certain predictions made by the model during the test dataset's inference are imprecise. It indicates that while the responses are not nearly as precise as the questions need, they are nonetheless pertinent to the topic and the image being discussed. Answer precision is lost when one uses a frequency count to restrict the set of viable responses. Since there are only so many alternative responses, this mistake cannot be totally removed.

Yes/No Question Label Error:

There are just two possible responses to "yes/no" questions in the English language: yes or no. Nevertheless, the model is compelled to provide "yes/no" questions with two viable responses. However, this method uses two different kinds of non-exact equivalents for yes/no responses.

6.2 | Comparison with State-of-the-art Works

Since no research has been done on the visual question-answering system in Nepali to date, the findings of this study are contrasted with comparable studies conducted in other languages.

TABLE 10 Comparison with State-of-the-art Works

	Accuracy Comparison in %				Dataset Comparison			Model Used
	Overall	Other	Count	Yes/No	Dataset	#Images	#Questions	
Nepali VQA	69.87	62.20	53.18	80.89	VQAv2 (Translated)	202,577	886,560	MCAN
English MCAN	70.63	60.72	53.26	86.82	VQAv2	204,721	1.1 million	MCAN
Vanilla VQA	57.75	43.08	36.77	80.50	VQAv1	204,721	760k	VGGnet, LSTM
Hindi VQA	64.51	55.37	42.09	84.21	VQAv1 (Translated)	204,721	369k	RCNN, LSTM
Bengali VQA	63	n/a	n/a	n/a	VQAv1+CLEVR (Translated)	n/a	5k + 12k	MobileNetv2, LSTM
Attention-based Bengali VQA	n/a	n/a	n/a	63.3	VQAv2 (Human Translated)	3280	13,046	VGG19, Bi-GRU

The original Deep MCAN paper (English MCAN) by Zhou Yu, et al. achieved better performance than other VQA models in terms of overall accuracy and for 'yes/no' and 'other' type questions. The vanilla VQA proposed in 2015 by Aishwarya Agrawal, et al. introduced the combination of visual and linguistic features to accomplish one of the most sophisticated tasks related to artificial intelligence. Mahamudul Hasan Raf, et al. (Attention-based Bengali VQA) developed a Bengali VQA system based on the attention mechanism. They took a subset of data from the VQA v2 dataset. they manually translated questions and answers in the dataset into the Bengali language. They translated the VQA v1 and CLEVR datasets into Bengali language using Google Translate to develop a Bengali VQA system. They used MobileNet, a CNN algorithm, and LSTM for visual and textual features respectively. Deepak Gupta, et al. (Hindi VQA) proposed a multilingual and code-mixed visual question-answering system for Hindi, English, and the mixed language. They used Google Translate API to translate the VQA v1 dataset to Hindi.
English counterparts were trained on a larger dataset that used better word embedding models along with better text pre-processing algorithms. Consequently, they performed better than the model developed in this research.

The Nepali VQA model utilized the co-attention of features extracted from a larger dataset along with pre-trained word embeddings. As a result, a better performance than the Bengali VQA models was ensured.

7 | CONCLUSION

To create a Nepali VQA model trained on a translated English dataset, the application of layers of modular co-attention networks for attending the visual features with textual characteristics was investigated. As a result, the performance could not match that of the English VQA systems. Nonetheless, it was unquestionably superior to a few non-English VQA systems, such as Bengali and Hindi VQA. Better text pre-processing techniques and word embedding models can be used to further enhance performance.
The foremost objective of this research was to compose a Nepali VQA dataset. The goal was achieved by translating a benchmark English VQA dataset called the VQAv2 dataset which consisted of enough examples to train a VQA model. Then, a pre-trained Nepali GloVe model was used to map the words in questions and answers to embedding vectors which were fed as textual features to the VQA model. Finally, the accuracy of the model was calculated and compared against similar works, and the comparisons were analyzed. This research paved the way for future researchers to use this word as a baseline for future works in the Nepali VQA domain.

Conflict-of-Interest

The authors declare that they have no conflict of interest.

references

[1] Gurari D, Li Q, Stangl AJ, Guo A, Lin C, Grauman K, et al. Vizwiz grand challenge: Answering visual questions from blind people. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2018. p. 3608–3617.

[2] Geramifar P, Yazdani E. NuMeVChat: A Conceptual AI-Driven Visual Chatbot for Advancing Personalized Cancer Care in Nuclear Medicine. Frontiers in Biomedical Technologies 2023;.

[3] Vinyals O, Toshev A, Bengio S, Erhan D. Show and tell: A neural image caption generator. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2015. p. 3156–3164.

[4] Aditya S, Yang Y, Baral C, Fermuller C, Aloimonos Y. From images to sentences through scene description graphs using commonsense reasoning and knowledge. arXiv preprint arXiv:151103292 2015;.

[5] Rajpurkar P, Zhang J, Lopyrev K, Liang P. Squad: 100,000+ questions for machine comprehension of text. arXiv preprint arXiv:160605250 2016;.

[6] Antol S, Agrawal A, Lu J, Mitchell M, Batra D, Zitnick CL, et al. Vqa: Visual question answering. In: Proceedings of the IEEE international conference on computer vision; 2015. p. 2425–2433.

[7] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. Advances in neural information processing systems 2017;30.

[8] Wu Q, Teney D, Wang P, Shen C, Dick A, Van Den Hengel A. Visual question answering: A survey of methods and datasets. Computer Vision and Image Understanding 2017;163:21–40.

[9] Lu J, Yang J, Batra D, Parikh D. Hierarchical question-image co-attention for visual question answering. Advances in neural information processing systems 2016;29.

[10] Yu Z, Yu J, Cui Y, Tao D, Tian Q. Deep modular co-attention networks for visual question answering. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition; 2019. p. 6281–6290.

[11] Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, Unterthiner T, et al. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:201011929 2020;.

[12] Devlin J, Chang MW, Lee K, Toutanova K. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:181004805 2018;.

[13] Chen X, Wang X, Changpinyo S, Piergiovanni A, Padlewski P, Salz D, et al. Pali: A jointly-scaled multilingual language-image model. arXiv preprint arXiv:220906794 2022;.

[14] Wang W, Bao H, Dong L, Bjorck J, Peng Z, Liu Q, et al. Image as a foreign language: Beit pretraining for all vision and vision-language tasks. arXiv preprint arXiv:220810442 2022;.

[15] Bao H, Wang W, Dong L, Liu Q, Mohammed OK, Aggarwal K, et al. Vlmo: Unified vision-language pre-training with mixture-of-modality-experts. Advances in Neural Information Processing Systems 2022;35:32897–32912.

[16] Gupta D, Lenka P, Ekbal A, Bhattacharyya P. A unified framework for multilingual and code-mixed visual question answering. In: Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing; 2020. p. 900–913.

[17] Islam SS, Auntor RA, Islam M, Anik MYH, Islam AAA, Noor J. Note: Towards devising an efficient vqa in the bengali language. In: ACM SIGCAS/SIGCHI Conference on Computing and Sustainable Societies (COMPASS); 2022. p. 632–637.

[18] Johnson J, Hariharan B, Van Der Maaten L, Fei-Fei L, Lawrence Zitnick C, Girshick R. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2017. p. 2901–2910.

[19] Rafi MH, Islam S, Labib SHI, Hasan SS, Shah FM, Ahmed S. A Deep Learning-Based Bengali Visual Question Answering System. In: 2022 25th International Conference on Computer and Information Technology (ICCIT) IEEE; 2022. p. 114–119.

[20] Goyal Y, Khot T, Summers-Stay D, Batra D, Parikh D. Making the V in VQA Matter: Elevating the Role of Image Understanding in Visual Question Answering. In: Conference on Computer Vision and Pattern Recognition (CVPR); 2017. .

[21] Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, et al. Microsoft coco: Common objects in context. In: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13 Springer; 2014. p. 740–755.

[22] Girshick R. Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision; 2015. p. 1440–1448.

[23] Koirala P, Niraula NB. Npvec1: Word embeddings for Nepali-construction and evaluation. In: Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP-2021); 2021. p. 174–184.

[24] Pennington J, Socher R, Manning CD. Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP); 2014. p. 1532–1543.

[25] Anderson P, He X, Buehler C, Teney D, Johnson M, Gould S, et al. Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering. In: CVPR; 2018. .