International Journal of Science, Strategic Management and Technology
Volume **02** Issue **02** February-2026 | ISSN: **3108-1762 (Online)** | Impact Factor: 3.8
An International, Peer-Reviewed, Open Access Scholarly Journal Indexed in recognized academic databases

Enhancing Image Captioning Through Augmented Visual Comprehension with CNN

R.L.Pavan Kumar 1 , T.V.D.S.Sreyanth 2 , P.Nithin Sai 3 and G .Surendra 5

B.tech Student1, Koneru Lakshmaiah Education Foundation, Vaddeswaram, A.P., 522302, India.
1*2100080168@kluniversity.in
B.tech Student2, Koneru Lakshmaiah Education Foundation, Vaddeswaram, A.P., 522302, India.
1*2100080197@kluniversity.in
B.tech Student3, Koneru Lakshmaiah Education Foundation, Vaddeswaram, A.P., 522302, India.
1*2100080203@kluniversity.in

Assisstant Professor2, Department of AI & DS , Koneru Lakshmaiah Education Foundation, Vaddeswaram, A.P., 522302, India.
2guntisurendra@kluniversity.in

https://doi.org/10.55041/ijsmt.v2i2.044

Cite this Article: .Surendra, R. K. ,. P. S. ,. (2026). Enhancing Image Captioning Through Augmented Visual Comprehension with CNN. International Journal of Science, Strategic Management and Technology, *Volume 10*(01). https://doi.org/10.55041/ijsmt.v2i2.044

License: This article is published under the Creative Commons Attribution 4.0 International License (CC BY 4.0), permitting use, distribution, and reproduction in any medium, provided the original author(s) and source are properly credited.

ABSTRACT:

Deep Learning and Computer Vision technologies are expanding quickly, and the challenge of automatically generating informative photo captions has received considerable attention. As discoveries continue to reshape the artificial intelligence landscape, the demand for intelligent systems capable of contextualizing visual content with descriptive captions is growing. Image Captioning is a fascinating area of research that intersects computer vision and deep learning techniques. This research paper explores the application of deep learning to the task of generating descriptive captions for images. The proposed model is extended to integrate YOLO-based object detection which is incorporated into the feature extraction process, thus increasing the robustness of the image representation. The architecture includes the integration of Convolutional Neural Networks (LSTM) for feature extraction from images and RNNs for language modeling. The CNN extracts meaningful visual features from images. Attention methods are used to address the issue of matching linguistic and visual information. This enables the model to concentrate on distinct areas of the image while generating captions.

Keywords- CNN, LSTM, YOLO, BLEU

INTRODUCTION:

In the realm of advancing visual comprehension, this article investigates the incorporation of Convolutional Neural Networks and the YOLO model as a paradigm change for picture caption synthesis in the context of improving visual comprehension. The need to enhance the traditional methods of visual understanding is developing as the combination of computer vision and deep learning technologies pushes the boundaries of artificial intelligence further. Accurate captioning of videos for long-term uses like security systems is another application.[1]

© Author(s). This work is peer-reviewed, openly published, and permanently archived
This article is openly accessible and reusable with proper attribution.

A significant shift from conventional techniques is the use of CNNs, which provide a solid framework for spatial and hierarchical feature extraction from pictures. Our goal is to increase the level of context awareness and semantic understanding while simultaneously improving the generated caption accuracy by incorporating these neural networks into the picture captioning pipeline in a seamless manner. For photo captioning, numerous methods have been proposed, including the object detection model, the visual attention-based image captioning model, and the deep learning model.[10]

To address this, our approach combines the strengths of CNN and YOLO(you only look once) models, offering a synergistic solution for more contextually rich and precise image captions. Hierarchical representations of the image content are captured by the CNN, which functions as a powerful feature extractor. Our goal is to use a CNN model that has already been trained to recognize significant features and intricate patterns. Concurrently, the YOLO model which is well-known for its ability to recognize things in real time-contributes by detecting objects in the picture and delivering precise bounding box coordinates.

This article details the architecture and implementation of our combination model, which combines these two different but complementary methods. We examine how the model is trained, how to fine-tune it, and what measures are used to assess its performance. The result presented here show how well our model performs in generating comprehensive, accurate, and contextually relevant captions, which is a significant development in the domains of computer vision and natural language understanding.

RELATED WORK:

At the beginning of 2018, Shuang Liu and others proposed an analysis of RNN, CNN, and Reinforcement Learning's training settings and training times and also concluded that we could train CNN faster. Compared to RNN and Reinforcement framework, CNN has a faster pre- parameter. Two popular techniques that are based on machine-translation are BLEU and METEOR[6]. While certain indicators based on image captioning are used to construct CIDEr and SPICE. ROGUE is derived from text abstraction. For those five criteria, they have displayed the best outcomes from the three methodologies. CNN, CNN-RNN, and CNN- CNN methods from which the CNN-CNN-based framework has given weaker performance.

Grishma Sharma and others stated that they employed the Visual Geometry Group (VGG16) for object detection and the LSTM and GRU architectures for sentence framing from the provided input photos[1]. They compared the performances of LSTM and GRU using the BLEU(Bilingual Evaluation Understudy) score.They evaluated different photos on both VGG+LSTM and VGG+GRU. The loss calculated for LSTM was less than that of GRU, and it took about 13 minutes for LSTM and 10 minutes for GRU per epoch. However, their findings indicate that because of its complexity, the LSTM model performs marginally better than GRU in most cases.

According to Simao Herdade and others research, they used a "Object Relation Transformer" that explicitly incorporates spatial relationship information between input-detected items through geometric attention. They have used encoder-decoder architecture for picture captioning using the MS-COCO dataset. In their initial method, they extracted appearance and geometry information using an object detector, then generated caption text using ORT. Using CIDEr-D, SPICE, BLEU and ROGUE, they
assessed the models. Based on the CIDEs-D scores, they found that adding comparable ordering, such as object-size, left-to-right, or top- to-bottom, reduces performance. It is concluded that the model's interpretability has improved with the addition of geometric attention[9].

Megha J. Panicker and colleagues' work, "Image Caption Generator", set out to create a system that could take an image as input in the form of a dimensional array and generate a sentence summarizing the image. They used an 8K Flickr dataset, which is preprocessed by feeding its 8000 phots through Keras' Xception program. Their model is a CNN+LSTM, a combination of an RNN and CNN architecture, that receives an image as input and generates a caption. The method uses a "decoder" RNN and an "encoder" RNN to map the source sentence and convert it into a fixed-length vector. In the end, RNN predicts the last meaningful sentence, which is [5]. To extract features from it, transfer learning will also be used. Since the BLEU score can be used to compare different models and assess translated text, it may be used to determine which model offers the highest level of accuracy.

© Author(s). This work is peer-reviewed, openly published, and permanently archived
This article is openly accessible and reusable with proper attribution.

Aishwarya Maroju and other co-researchers have developed a deep-learning model that is appropriate for captioning military images. The CNN-RNN framework is primarily used.[10] To reduce the gradient descent issue, they employed

LSTM networks in conjunction with the Inception model for image encoding. They concentrated on how the CNN-RNN model's vanishing gradient issue prevents the RNN from learning and from receiving effective training. They have used the FLICKR 8K dataset and given inputs to ResNet to get preprocessed. They proposed a model called ResNet-LSTM, in which ResNet is used as an encoder to extract features from images, converts those features into single-layered vectors, and then feeds the gathered features into an LSTM, a decoder that generates each caption word sequentially using input from vocabulary dictionary and picture features. They concluded that the ResNet-LSTM model is more accurate than the VGG and CNN- RNN models.

Yajush Pratap Singh and others published a journal in 2021 titled "Image Captioning using Artificial Intelligence". Their suggested approach primarily focuses on attention mechanisms, which are widely employed in picture caption generation tasks and have a significant impact on computer vision. The problem they are working on is generating language descriptions of an image that can be used for understanding and supporting visually impaired people and considering the large number of images that are shared online, it might be challenging to create inscriptions[12] that accurately represent the object and its surroundings. They have created a photo captioning tool that uses CNN and AI to decipher and computerize text age. They have developed a combined model of two architectures which are CNN(element extractor) and RNN(LSTM). To generate consecutive information or word clusters that ultimately result in a depiction, CNN is used to synthesize and include vectors from spatial information in images. These are handled by fully associated straight layers into RNN and have used Flickr 8K dataset, also included steps like working of tokenization, embedding layer, words to vectors. They have achieved 80 to 90 % accuracy on the model they have developed.

Parth Kotak and Prem Kotak aim to put in place an image caption generator that reacts to the user to obtain the captions for a picture that they have supplied. An image caption generator's main goal is to improve user experience by producing automatic captions. They used a high-level, general-purpose programming language which was Python. Their project's overall use is to Read caption files, clean up data, load training and testing data, preprocess data using generator function, word embedding, model architecture, train our model, make predictions, and lastly apply the iterative development process model[13] for analysis, which builds little versions of each feature across all components to design a system. They have given a detailed explanation of the cost analysis. CNN, an image-based model, pulls features from an image, whereas LSTM, a language-based model, converts the data and objects recovered by CNN into a phrase that makes sense.

Muhammad Abdelhadie Al-Malla [3]and others have proposed a methodology on the "Image captioning model that imitates human attention and object features. They have used MSCOCO and Flickr30k dataset, the model they have developed contains an attention-based encoder- decoder architecture made up of the YOLO model's object feature detection and the CNN model's pre-trained Xception model. Using their method the feature extraction schemes raise the CIDEr score by 15.04%. They have used BLEU, METEOR, and ROGUE evaluation metrics which have shown poor correlations with human quality testing; stronger correlations with SPICE and CIDEr, but more challenging optimization. Language generation is done by using the attention module GRU and two connected layers. Here Xception CNN pre-trained on ImageNet is used to extract spatial features.

Janv Jambhale and co-researchers have proposed a model called Resnet 50 (residual neural network) based on CNN-LSTM neural networks. They are using CNN to detect objects in images and RNN-based LSTM for captioning the images. They have used the Python programming framework is called Flask Rest API. The method they have proposed contains 5 stages such as dataset collection, image preprocessing, defining the model, fitting, and testing. Resnet50 is a powerful convolutional Neural Network architecture that stands out with its 50 deep layers. It surpasses the performance of traditional CNNs and VGG networks. While traditional CNNs typically consist of a pooling layer, a

© Author(s). This work is peer-reviewed, openly published, and permanently archived
This article is openly accessible and reusable with proper attribution.

rectified linear unit(ReLU) layer, and a convolutional layer, Resnet50 takes it a step further. It incorporates a range of layers, including convolutional layers, ReLU activation, batch normalization, pooling, and fattening. This comprehensive combination of layers contributes to ResNet50's exceptional performance.

Akash Verma and other researchers have proposed [4] an encoder-decoder-based model They have used LSTM and the decoder are placed at 1365 positions in the VGG16 hybrid. These models are trained on the labeled datasets of Flick8k and MSCOCO captions. Evaluation is done through BLEU, METEOR, GLEU, and ROGUE_L and obtained BLEU-1=0.6666, METEOR=0.5060, GLEU-0.2469 on Flickr8k BLEU-1=0.7350, METEOR=0.4768, GLEU=0.2798 on MSCOCO captions. This model performs better than any state-of-the-art methodology. CNN models used for feature extraction-VGG16, Inception V3, and ResNet50. For Sentence generation is used.

Rajendra Prasad Mahapatra and Tejal Tiwary have proposed an AIC(Automatic Image Captioning) model that contains data gathering, the selection of uncaptioned images, appearance and texture feature extraction, and the creation of automatically generated image captions. Data has been gathered from two sources. The Selection of pictures without captions are form of ARO(adaptive Rain Optimization) which combines fuzzy C Means (FCM) clustering in a hybrid way.. They have used two types of approaches for the feature extraction which are Multi-scale, Weighted Patch Local Binary Pattern with Spatial Derivative [14]Lastly, an extended convolutional atom neural network, or ECANN, is used to automatically create the captions., further combining CNN and LSTM architectures along with usage of an Optimization algorithm named Adaptive atom search. The dataset they used was the grocery dataset(5125 images having 80 classes) and Freiburg 25 image cat categories out of 4947 images and there is a 70:30 ratio between testing and training. When tested on the grocery shop dataset and the Freiburg groceries dataset, their suggested model produced results of 99.46% and 99.32%, respectively, accuracy.

The scientists[7] Antonio M. Rinaldi and colleagues demonstrate how combining the current ethodologies with a combined strategy might increase performance efficiency. The image utilizes a hierarchical object detection system with two distinct levels. Three CNNs are combined to provide an encoder-based framework for object recognition, and in the decoder stage, a feature extraction-based combing mechanism is developed to carry out natural language processing. The approach is divided into three parts which are detection, combination, and captioning modules. The model made use of the COCO dataset, wherein each model was repeated for both class and superclass detection, and the detection module employed Mask R-CNN, YOLOv3, and RetinaNet. The combination module includes all with a combiner lastly captioning module combines CNN and RNN to examine the image's visual characteristics.

Taraneh and others suggested a thorough taxonomy of all deep-learning models utilized for captioning images.. They have used dense captioning methods which are used to give captions for each entity and also use whole captioning methods that give captions for whole input images [8]. They have developed models based on deep learning methods such as attention-based, generating Multi-Style Captions, Transformer-based+Graphs, VLP, Graph-based, Graph-based+Attention, Convolution Network- based, Unsupervised Approaches and reinforcement Learning, Vision language Pre- Training Methods, Generating Multi-style captions. They have attempted to address the issues raised by their research, which include contextual understanding, illumination circumstances, object hallucinations, exposure bias, loss-evaluation mismatch, disappearing gradient, and exploding gradient problems, and have trained the models using Flickr30K, Flickr30K Entites, Visual Genome, TextCaps, Vizwix-captions and evaluated on basis of BLEU, ROUGE, METEOR, CIDEr, SPICE.

METHODOLOGY:

The methodology we have proposed includes two models like a multi-modal fusion. The models which are being used are YOLOv4, VGG16, Encoder, and Decoder model. Figure 1 is the flowchart represented on the methodology we have proposed.The problem definition is taken to develop a image caption generator. Data is collected from Flick8K dataset that contains

8000 images and the augmenting technique used is reshaping the images to have same image size. We

© Author(s). This work is peer-reviewed, openly published, and permanently archived
This article is openly accessible and reusable with proper attribution.

have used libraries like numpy, tensorflow, PIL, tqdm.notebook, yolov4.tf for developing the model.

FIGURE 1

BRIEF OVERVIEW OF STEPS:

Problem Definition

Researchers in computer vision and natural language processing are interested in picture captioning, or the automatic generation of natural language descriptions for images. In this work, we provide an innovative approach to picture captioning that combines the capabilities of cutting-edge object recognition models like YOLO (You Only Look Once) with advanced language creation approaches. Suggested technique uses the recognized items and their connections as structured input to language model, which creates natural language captions that appropriately represent the visual content.Data Collection and Preprocessing

For our picture caption generator, we use the Flickr8k dataset, which contains 8000 photographs from the popular image-sharing site Flickr. Each image is accompanied by five human-annotated caption, which provides a wide range of natural language explanations of the visual information. To preprocess the data, we first divided it into training, validation, and test sets, as in the usual procedure. The photos are then downsized to a consistent resolution and normalized so that the YOLO object detector can handle them more efficiently**.**

Feature Extraction

The proposed image caption generator uses two basic sources of characteristics. Those are visual features derived from pictures using YOLO and textual features collected from the captions. The retrieved visual and textual data are then integrated and supplied into the caption generation model, which is trained to map the visual representations to their natural language descriptions.

Model Architecture

The model architecture is a combination of the YOLO object detection model and an encoder- decoder network for caption generation. The YOLO model will recognize and localize items inside the input photos, while the encode- decoder network will use these object detections, coupled with image attributes, to create natural language captions. To get the best performance on the Flickr8k dataset, the architecture must be built and fine-tuned.

Word-Embeddings

Word-embeddings will be used to represent the words in captions. Word embeddings are dense vector representations that capture the semantic and syntactic links between words. These embeddings can be taught from scratch or derived from pre-trained models such as Word2Vec or GloVe. Using word embeddings enables the model to comprehend the context and meaning of words, which is critical for creating cohesive and relevant captions.

Multi-model Fusion

YOLO object detection model and the caption generating model must be combined or fused so that the two components interact and align properly. One approach might be to use attention processes to help the caption generation model focus on the most relevant recognized elements and their spatial relationships while constructing captions. Another option is to use a visual co- attention mechanism, which may focus on both picture attributes and partially constructed captions, allowing the model to

© Author(s). This work is peer-reviewed, openly published, and permanently archived
This article is openly accessible and reusable with proper attribution.

generate more meaningful and contextual descriptions.

Model Training

Once the model architecture and components are in place, the picture caption generator must be trained using the flickr8k dataset. To do this, appropriate loss functions must be established (for example, cross-entropy loss in the caption production process), and gradient-based optimization methods like Adam and RMSProp must be used to optimize the model parameters. Using a planned sampling strategy is one way to increase the model's capacity to provide original, neat captions during training.

Evaluation

To Using a held-out test set taken from the Flickr8k dataset, the trained model's performances will be evaluated. Metrics such as METEOR (Metric for Evaluation of Translation with Explicit Ordering) and BLEU (Bilingual Evaluation Understudy) are frequently used to access the quality and accuracy of the gen.

Hyperparameter Tuning

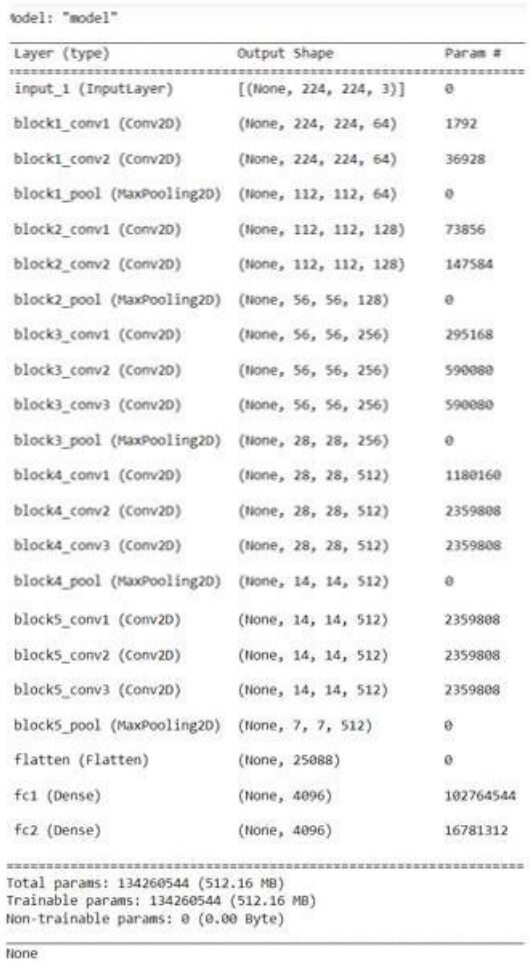
The photo caption generator's hyperparameters, which can significantly affect its performance, including the learning rate, batch size, and regularization parameters, just like all other machine learning models. To maximize the model's effectiveness, a methodical process of hyperparameter tuning will be carried out, which may include grid search, random search, or more approaches like Bayesian optimization, which can effectively examine through the hyperparameter space and find the best configurations.

Transfer Learning & Fine Tuning

Using weights from pre-trained models that have been learned on larger datasets, such ImageNet or Microsoft COCO, to initialize the training of the model from start, the model might be a helpful method of using transfer learning. This can provide the model with a solid knowledge base and feature representations, enabling it to learn from scratch

I. VGG16 Model

This model is for extracting the features from the images. Beginning with the images to get loaded with the target_size of (224,224), converting the image to the array, reshaping, and then predicting the feature. Specifically, the pre-trained VGG16 model is loaded from Tensorflow keras, the output layer is removed, and the output from the second -to-last layer is retained. This extracted feature is used as a representation of the input image and is later used in the image captioning model,



```
Model: "model"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         [(None, 224, 224, 3)]     0
block1_conv1 (Conv2D)        (None, 224, 224, 64)      1792
block1_conv2 (Conv2D)        (None, 224, 224, 64)      36928
block1_pool (MaxPooling2D)   (None, 112, 112, 64)      0
block2_conv1 (Conv2D)        (None, 112, 112, 128)     73856
block2_conv2 (Conv2D)        (None, 112, 112, 128)     147584
block2_pool (MaxPooling2D)   (None, 56, 56, 128)       0
block3_conv1 (Conv2D)        (None, 56, 56, 256)       295168
block3_conv2 (Conv2D)        (None, 56, 56, 256)       590080
block3_conv3 (Conv2D)        (None, 56, 56, 256)       590080
block3_pool (MaxPooling2D)   (None, 28, 28, 256)       0
block4_conv1 (Conv2D)        (None, 28, 28, 512)       1180160
block4_conv2 (Conv2D)        (None, 28, 28, 512)       2359808
block4_conv3 (Conv2D)        (None, 28, 28, 512)       2359808
block4_pool (MaxPooling2D)   (None, 14, 14, 512)       0
block5_conv1 (Conv2D)        (None, 14, 14, 512)       2359808
block5_conv2 (Conv2D)        (None, 14, 14, 512)       2359808
block5_conv3 (Conv2D)        (None, 14, 14, 512)       2359808
block5_pool (MaxPooling2D)   (None, 7, 7, 512)         0
flatten (Flatten)            (None, 25088)             0
fc1 (Dense)                  (None, 4096)              102764544
fc2 (Dense)                  (None, 4096)              16781312
=================================================================
Total params: 134260544 (512.16 MB)
Trainable params: 134260544 (512.16 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
None
```

FIGURE 2

II. YOLOv4 Model

The object detection model known as YOLOv4(you only look once at version 4) is intended for real-time processing. This code predicts bounding boxes and associated classes for objects in the input photos using YOLOv4. Weights are loaded from the "yolov4.weights" file and the YOLOv4 model is initialized. The "coco.names" file is used to specify the class names.

© Author(s). This work is peer-reviewed, openly published, and permanently archived
This article is openly accessible and reusable with proper attribution.

International Journal of Science, Strategic Management and Technology
Volume **02** Issue **02** February-2026 | ISSN: **3108-1762 (Online)** | Impact Factor: 3.8
An International, Peer-Reviewed, Open Access Scholarly Journal Indexed in recognized academic databases

III. Encoder Model

The encoder model receives as inputs the characteristics extracted by the 4096- dimensional vector VGG16 and the variable-dimensional vector YOLOv4, which is based on the number of objects observed. It uses a thick layer with ReLU activation and a dropout layer to handle the VGG16 features. Analogously, an LSTM layer and an embedding layer with dropout process the YOLOv4 data. The output from these two paths is then combined via the addition of elements. It performs dropout for regularization (fe1) using the VGG16 model's output features(inputs1). Then, the dimensionality is reduced to 256(fe2) using a dense layer with ReLU activation. High- level features are extracted from the image in this section.

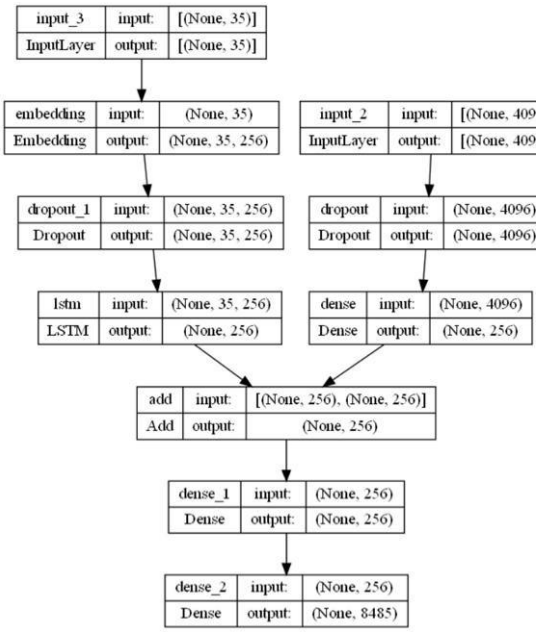


FIGURE 3

Decoder Model

The combined features from the encoder are processed by the model's decoder section. The next word in the sequences' probability distribution is produced by a dense layer with ReLU activation and a final dense layer with softmax activation. The input is transformed into dense vectors of a predetermined size using an embedding layer(yolo_embedding). Dropout is used to become regular. The sequential input is then processed by an LSTM layer, which also records contextual data(yolo_lstm).

The encoder and decoder parts are merged using an element-wise addition('add'). Further dense layers are applied to generate the final output. The Softmax activation function in the output layer predicts the following word in the sequence.

In the training, the Keras tokenizer class is used to tokenize text data. Maximum caption length and vocabulary size are established. A data generator is used to produce training data in batches. After encoding, sequences are divided into input- output pairs. As inputs and outputs, features(VGG16 and YOLOv4) and related sequences are ready. The Model is compiled using the Adam Optimizer with categorical cross-entropy loss. Training takes place over a certain number of epochs. The data generator produces batches for model fitting at the end of each epoch

RESULTS AND DISCUSSION:

Our code is developed by using the VGG16 model to extract high-level features from the Flickr8k dataset which is essential for converting unprocessed pixel data into a format. We have preprocessed by converting all characters to lowercase, unnecessary digits, and special characters are removed, and have tokenized the preprocessed test using Keras Tokenizer to produce organized vocabulary. We developed a model that combines the power of VGG16 features with the LSTM layer along with the YOLOV4 model. Using BLEU scores, we evaluated our model on a specific test set to determine its efficacy that has concentrated on both unigrams(BLEU-1) and bigrams(BLEU-2).

```
BLEU-1: 0.541891
BLEU-2: 0.314490
```

FIGURE 4

The model effectively showcased its ability to produce captions for fresh pictures. We usedthe picture "244443352_d7636e1253.jpg" and "56494233_1824005879.jpg" as examples and displayed captions that were really in the dataset and captions that our model predicted

© Author(s). This work is peer-reviewed, openly published, and permanently archived
This article is openly accessible and reusable with proper attribution.

Detected image

```
----------Actual----------
startseq boy goes down an inflatable slide endseq
startseq boy in red slides down an inflatable ride endseq
startseq boy is sliding down in red shirt endseq
startseq child going down an inflatable slide endseq
startseq "a young boy sliding down an inflatable is looking off camer
endseq
----------Predicted----------
startseq boy goes down an inflatable slide endseq
```

Generated caption

FIGURE 5

Detected image

```
----------Actual----------
startseq boy dressed in an orange shirt and helmet is riding dirt bike in
the woods endseq
startseq boy wearing helmet on bicycle in wooded area endseq
startseq boy with helmet mountain bikes through the woods endseq
startseq "a child is posing on his mountain bike and wearing helmet ." end
seq
startseq helmeted boy bicyclist on path in the woods endseq
----------Predicted----------
startseq boy dressed in an orange shirt and helmet is riding dirt bike in
the woods endseq
```

Generated caption

FIGURE 6

CONCLUSION:

To sum up, our efforts to develop an image captioning model have produced encouraging outcomes with the help of sequential text generation driven by LSTM and VGG16-based image feature extraction and the Yolo model. Our findings are corroborated by BLEU sores and demonstrated by the generated captions for particular photos. When we reflect on this project phase, it is clear that picture captioning acts as a link between natural

language processing and computer vision. Even though we are proud of what our present model has accomplished, there is still room for improvement and investigation. The model can be improved by exploring more advanced architectures, incorporating attention mechanisms, and expanding datasets through advanced data augmentation techniques.

REFERENCES:

1. Grishma Sharma, Priyanka Kalena, Nishi Malde, Aromal Nair, Saurabh Parkar, Visual Image Caption Generator Using Deep Learning written on April 8, 2019, ResearchGate
2. Janvi Jambhale, Shreeya Sangale, Aarti Avhad, Payal Vairagade, Jameer Kotwal, Image Caption Generator using Convolutional Neural Networks and Long Short-Term Memory, Issue:05/May-2022, IRJMETS

© Author(s). This work is peer-reviewed, openly published, and permanently archived
This article is openly accessible and reusable with proper attribution.

3. Muhammad Abdelhadie Al-Malla, Assef Jafar & Nada Ghneim, Image captioning model using attention and object features to mimic human image understanding, Published:14 February 2022, Springer.
4. Akash Verma, Arun Kumar Yadav, Mohit Kumar, Divakar Yadav, Automatic Image Caption Generation using Deep Learning, Posted Date: June 21st 2022, ResearchSquare
5. Megha J Panicker, Vikas Upadhayay, Gunjan Sethi, Vrinda Mathur, Image Caption Generator, Volume-10 Isuue-3, January 2021, IJITEE
6. Shuang Liu, Liang Bai, Yanil Hu, Haoran Wang, Image Captioning Based on Deep Neural Networks, November 2018, ResearchGate
7. Antonio M. Rinaldi, Cristiano Russo, Cristian Tommasino, Automatic image captioning combining natural language processing and deep neural networks, Volume 18, June 2023, ScienceDirect
8. Taraneh Ghandi, Hamidreza Pourreza, Hamidreza Mahyar, Deep Learning Approaches on Image Captioning: A Review, August 23, 2023, ARXIV
9. Simao Herdade, Armin Kappeler, Kofi Boakye, Joao Soares, Image Captioning: Transforming Objects into Words, Yahoo Research
10. Aishwarya Maroju, Sneha Sri Doma, Lahari Chadarlapati, Image Caption Generating Deep Learning Model, Vol. 10 Issue 09, September-2021, IJERT
11. C. S. Kanimozhiselvi, Karthika V, Kalaivani S P, Krithika S, Image Captioning Using Deep Learning, Date of Conference: 25-27 January 2022, Date added:31 March 2022, IEEE
12. Yajush Pratap Singh, Sayed Abu Lais Ezaz Ahmed, Prabhishek Singh, Neeraj Kumar, Manoj Diwakar, Image Captioning using Artificial Intelligence,2021, IOPScience
13. Parth Kotak, Prem Kotak, Image Caption Generator, Vol. 10 Issue 11, November-2021, IJERT
14. Tejal Tiwary, Rajendra Prasad Mahapatra, An accurate generation of image captions for blind people using extended convolutional atom neural network, Published:15 July 2022, Springer

© Author(s). This work is peer-reviewed, openly published, and permanently archived
This article is openly accessible and reusable with proper attribution.