IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

VISUAL CONTENT CAPTIONING: A CNN AND TRANSFORMER BASED MODEL

[1] Mr.T. Upender, [2] S. BhagyaShree [3] L. Jatin [4] S. Deepthi

[1] Assistant Professor, [2, 3, 4] UG Student

[1,2,3,4] Department of Computer Science and Engineering,

[1,2,3,4] CMR College of Engineering & Technology, Hyderabad, Telangana, India

Abstract: During the past few years, the fields of computer vision and natural language processing has drawn more attention in the issue of automatically generating descriptive sentences for images, which is known as image captions.An image caption is a written description of an image that is widely used in programs that require information from images in a text format.Understanding the image's semantics and being able to construct descriptive sentences with the right structures are necessary to generate image captions. Rapid progress in AI (Artificial Intelligence) has led numerous researchers to focus on the area of image captioning. By utilising advanced deep- learning techniques, large datasets, and computer power, a model can be built to efficiently generate captions. The model is trained to generate captions that describe the input image. To achieve this objective, the Encoder - Decoder model utilises two advanced deep learning algorithms: CNN (Convolutional Neural Network) and Transformers. Initially, feature extraction is performed, followed by generation of captions. For this task, a pre-trained CNN model, EfficientNetB0 is employed.The Flickr_8k dataset, which contains 8000 images and five different captions for each image, is used to train the model.

Keywords **- Caption generator, Transformers, CNN (Convolutional Neural Network), EfficientNetB0, Encoder - Decoder.**

I.INTRODUCTION

The human capacity for effortless verbal description of a scene is a wonder of language and thought. In the domain of artificial intelligence, it is extremely difficult to replicate this ability in machines. In an effort to close the gap between the linguistic and visual domains, image captioning—the process of automatically generating textual descriptions of images— has become a potent tool at the junction between natural language processing (NLP) and computer vision. The term "image caption generator" implies that our goal is to design the most effective mechanism that can produce grammatically and semantically correct captions for images. It has many different kinds of applications, including social media, image indexing, editing advice, applications for the blind, and other NLP applications. Utilising the Flickr 8k dataset, which comprises approximately 8000 sample photos with five captions each, we constructed the deep neural network and machine learning techniques based model. The process is divided into three stages: using Convolutional Neural Network to extract high-level visual features from the input image; using attention mechanisms in the Transformer Encoder to process the encoded visual feature information in parallel; and using the Transformer Decoder to generate the caption word by word. Transformers and CNN (Convolutional Neural Network) are the two effective tools utilised in this method of image captioning.

II. Related Work

A. Image Caption Generation using Attention Mechanism :

Visual attention to English datasets has been used by a large number of researchers in the past. Two main forms of attention been implemented in encoder-decoder research: captioning images or videos. Semantic attention, or attention to words, is the term used to describe the first type of attention. The focus on images is the subject of spatial attention, which is the second type of attention. A visual attention model was first introduced in research on image captioning by Xu et al. [12]. They either used "soft" pooling, which takes the spatial qualities' average and gives attentive weights to each variable, or "hard" pooling, which determines the region that is more likely to be attended.Additionally, when viewing the network, CNN's Channel-wise Attention and Spatial Attention were used [13]. In addition, Chen et al. [14] made use of visual attention when writing the picture captions. Additionally, to link the visual feature with the visual concepts and produce the picture description, a semantic attention model was employed in RNNs [15].

B. Image Caption Generation in Different Languages :

The attention-based mechanism is modified for caption generation, but most research for caption generation was carried out in English because most of the datasets are written in that language [1]. For the encoder portion of the captioning model, like the ConvNet, most of the studies used the VGG-16 [2]. But for the visual feature and BiLSTM, several researchers also used the pre-trained models like AlexNet [3], [9], or Residual Network (ResNet) [3]. Aside from English, other datasets were also created for other languages, including Chinese [4], [5], Japanese Yoshikawa [6], Arabic [10], Bahasa Indonesia in [11] (custom dataset that combines Flickr30k and MS COCO), Indonesian Flickr30k [10], and the FEEH-ID Flickr8k's dataset [811].

C. Image Caption Using CNN and RNN :

[16] In this research, deep learning algorithms are utilised for image caption generation .Using this technique, natural phrases that ultimately describe the image are produced. Recurrent Neural Network (RNN) and CNN (Convolutional Neural Network) make up this paradigm. Sentence creation is done with RNN, and feature extraction from images is done with CNN. The model is trained so that when an input image is supplied, it produces captions that pretty much describe the image. Various datasets are used to assess the model's accuracy, further the smoothness or command of language that the algorithm learns from picture descriptions. These tests demonstrate that the model often provides precise descriptions for the input image.

D. Image Caption Using CNN and LSTM :

[3]In this paper, a deep learning approach to model implementation for image captioning is presented. The model is separated into three phases, first being the image feature extraction. Using the Xception model, the features of the images were extracted during this phase. The dataset considered was Flickr_8k. The Sequence Processor, which handles text input by functioning as a word embedding layer, was the second stage. The rules for extracting the required features are contained in the embedded layer, which uses masking to ignore the other values. Subsequently, the network will be connected to LSTM So that it can caption images. Decoder was the final and third phase that was examined. In this stage, the model will apply a method to combine the image input extractor phase and the sequence processor phase. Next, it will be sent to neural layers, and the final output—the Dense layer—will generate the words needed for the caption over the language that was created from the typed information gathered from the Processor phase that is in the sequence.

III. Methods And Experimental Details

A. Dataset

In this analysis, the standard Flickr8K dataset was used. With 8,092 images and five distinct captions that clearly describe the important entities and events, it's a new benchmark collection for sentence-based image description and search. Which we divide into 1,529 validation images and 6,114 training image. The photographs in the dataset were picked by hand from six distinct Flickr groups, and they primarily show a range of situations and scenarios rather than any famous persons or places.

B. System Design

CNN are specialized deep neural networks that can process the input data as a two-dimensional matrix. It is primarily used to classify images and determine whether they depict a bird, a plane, Superman, etc. It extracts significant features from photos by scanning them top to bottom and left to right, then combines those features to categorize the images. Additionally, it can manage images that have undergone perspective adjustments, translation, rotation, and scaling. Where as, **Transformers** learns long-range relationships between extracted features and generated words by using multi- head attention mechanisms and self-attention. This enables the model to identify complex relationships and context in the picture, which results in captions that are more precise and reasonable. Its non-sequential structure allows it to process all image features in parallel, significantly increasing caption generation speed over recurrent neural networks (RNNs) such as LSTMs. Additionally, it adapts to various image formats and caption styles with ease.

Since EfficientNetB0 uses less memory and processing power in order to attain equivalent or better outcomes, it's used as a pre-trained model for our image captioning. Good performance, flexibility, and ideal for resource-constrained applications (due to it's pre-trained weights it can be easily integrated into image captioning pipelines, which offer a solid basis for capturing high-level image features that can then be processed further by a decoder architecture to produce captions).

The following are the primary steps that comprise the overall workflow:

1. Data Preparation :

Before raw data is processed and analyzed, it must first be cleaned and transformed. It is a crucial stage before processing that frequently entails reformatting data, repairing data, and merging missing information sets to enhance data—that is, the operation of erasing or correcting inaccurate, manipulated, inaccurately formatted, copies, or from a dataset.Then the dataset was prepared for validation and training resulting in the input data for the CNN approach with transfer learning techniques during the training procedure.

2. Text and Image Pre-Processing :

- We first download and preprocesses the Flickr8k dataset, which is made up of pictures with captions that go with them.
- The text data is preprocessed by eliminating special characters, changing all characters to lowercase, and removing excessively long or short captions.
- It also preprocesses the image data, converting into a floating- point format, resizing, and normalizing the images.

3. Model Architecture :

Constructing a captioning system with the following two primary parts:

- **Image Feature Extractor:** This CNN is pre-trained to extract features from pictures.
- **Transformer-based Caption Generator:** Given the features extracted from the images, this sequence-to-sequence
  model creates captions using a Transformer architecture.

4. Training Model :

A dataset used to train an ML algorithm is called a training model. It is made up of matching sets of input data that affects both the output and sample output data.

5. Caption Generation :

Prediction is the result of an algorithm, trained on one historical dataset and used to forecast the probability of a specific outcome. In this case, the algorithm is creating a caption for the image.

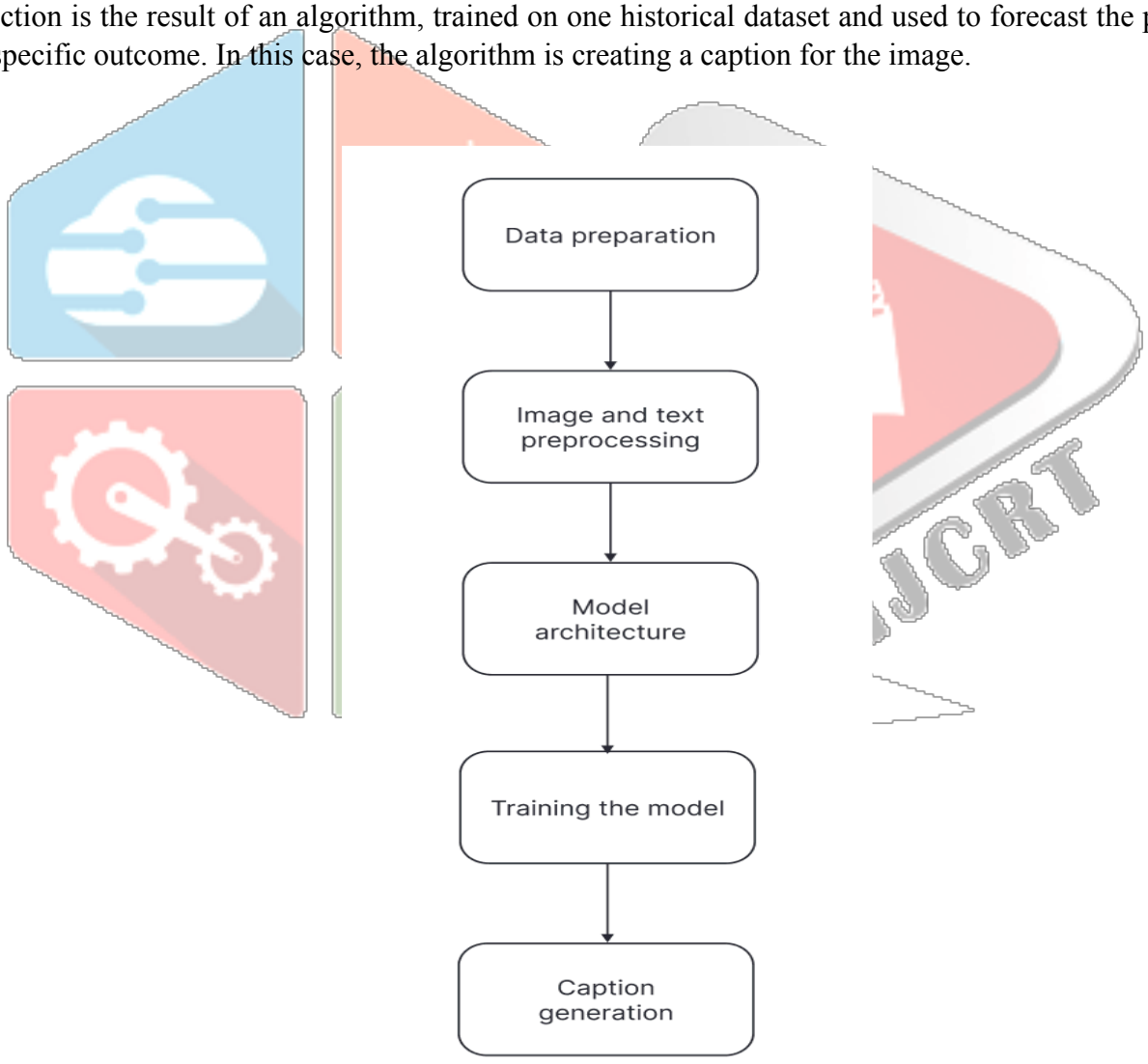

Fig 1: Proposed Model Flowchart

C. CNN - Transformer Architecture :

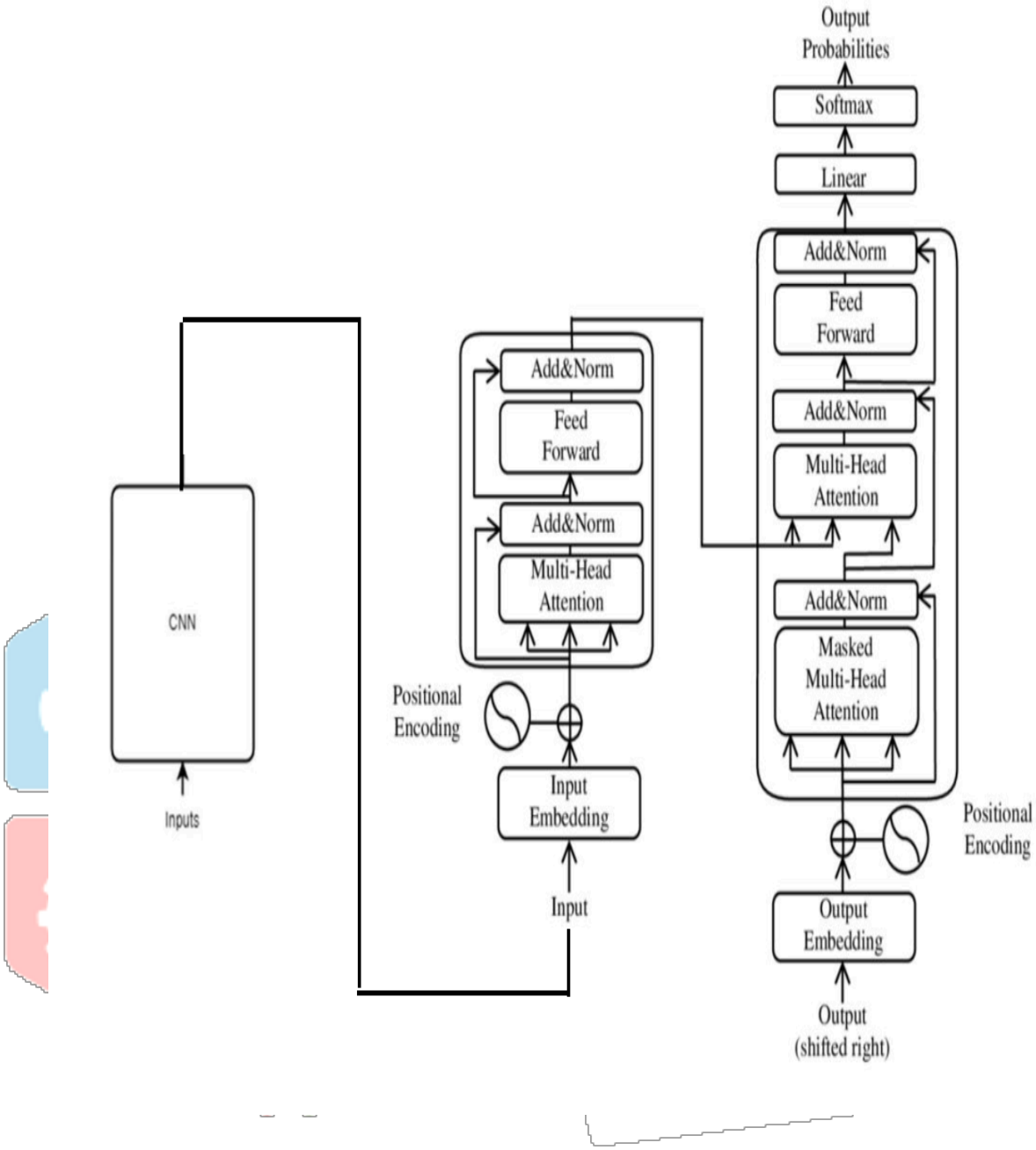

Fig 2 : CNN - Transformer Encoder-Decoder based Architecture

1. Feature Extraction with EfficientNetB0 :

- The retrieved feature map from EfficientNetB0 must first be flattened before it's loaded into the Transformer encoder.
- In addition positional encoding components are added to the flattened feature vector to depict the relative positional locations since the Transformer uses positional information when processing sequences. This aids the model understanding the relative locations of various parts within the image.
- The Transformer encoder looks at many portions inside the feature vector concurrently using a multi-head attention technique. This enables the model to represent relationships between various components in the image and also long-range dependencies.

- A context vector, which is the Transformer encoder's output, contains a summary of the crucial data that the model has extracted from the picture- — essentially the overall meaning and relationships within the image.
- The output dimensions are indicated as H x W x C, where H, W stand for the feature map's height and width, respectively (usually lower than the original image) and C stand for number of feature channels that capture different facets are indicated in the image content.

2. Transformer Encoder :

- Transformer encoder takes the flattened feature map from EfficientNetB0 as its input.
- Positional encoding, besides the flattening of feature vector, is done to the sequence processed by the Transformer due to its positional basis. In the process, it becomes more convenient for the model to comprehend the image the way different elements occupy it.
- The Transformer encoder uses multi-head attention method to focus on different feature vector parts at the same time. Consequently, the network have the ability to depict and it can capture interrelation and long-range dependency between various image parts.
- The Transformer encoder derives the image's most crucial data, which is represented by a context vector. This vector captures the meaning of the picture including the relationships.

3. Transformer Decoder :

- The Transformer decoder consists a context vector containing a concatenation of encoder output and a precisely "start of sentence" token.
- Concerning lexicon, a number representing each word vector is held responsible for the semantic information of the words along with their relationships.
- The attention-based mechanism of decoder cover context vector and words that are previously generated. By means of this mechanism the decoder ensures that each word carries different weight. This permits the model to generate the word following the caption based on the details gotten from the image and the words used in the sentence before.
- Image caption is made by Transformer decoder by building a word sequence step by step. On every step a single word is forecasted. It includes two parts: the current vector in context and the past ones. They assist you to make up a forecast and choose the correct word.

Our approach is distinctive to RNN, as we input all the sentences at once into the decoder, while the latter one feeds the words to the model sequentially. The main plus point of this parallelization to the prior architectures including RNN/LSTM and GRU is that the new architecture trains faster.

IV. RESULTS AND DISCUSSION

The integration of CNNs with Transformer architecture, which offers a robust framework for generating descriptive and contextually relevant captions from visual content, could significantly advance the area of image captioning. Thanks to continuous research and innovation, both natural language processing and computer vision are set to witness new developments and applications.

1. Image Feature Extraction :

Used EfficientNetB0 pre-trained model with weights frozen to preserve learned features.Produces a flattened vector representing the image feature.

2. Transformer Encoder :

Processes the extracted image features using a single TransformerEncoderBlock, encoding them into a fixed-length representation appropriate for caption creation.

3. Transformer Decoder :

Creates captions by utilizing multiple TransformerDecoderBlocks. Every block takes care of the previously generated tokens and encoded image features. During prediction, it makes use of attention mechanisms to concentrate on relevant areas of the image and the previously created caption.

4. Combined approach :

Individual architectures are outperformed by CNN-Transformer models. With datasets like MS-COCO and others, they are able to achieve state-of-the-art results on image captioning.

We discovered that our implemented model can produce reasonable image descriptions in proper English sentences during the validation phase. Enough information resides in the generated captions to describe the objects or elements in the picture. Our success has been aided by the system's accuracy and loss.

Regarding the validation image, the model produces good captions; however, it can always be enhanced by experimenting with different CNN configurations, adjusting hyper parameters, fine tuning and training on bigger datasets.

Since the approach consists of many phases, such as downloading the dataset, pre-processing images and text, vectorizing the text data, model building, model training, validation, and caption generation. The following figure will illustrate the steps involved in putting the plan into action and achieving good outcomes :

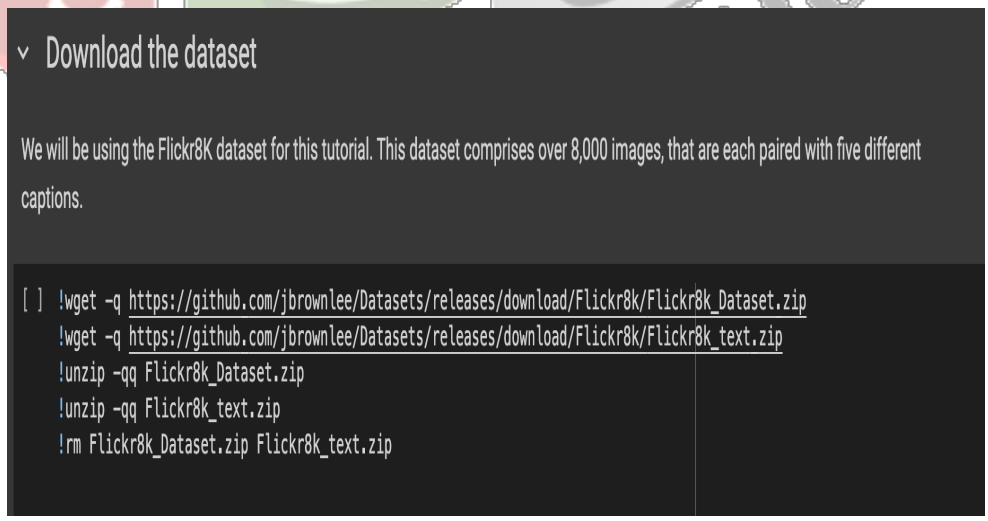

Fig 3: Downloading Flickr_8k dataset

We'll use the `TextVectorization` layer to vectorize the text data, that is to say, to turn the original strings into integer sequences where each integer represents the index of a word in a vocabulary. We will use a custom string standardization scheme (strip punctuation characters except `<` and `>`) and the default splitting scheme (split on whitespace).

ChatGPT

```
def custom_standardization(input_string):
    lowercase = tf.strings.lower(input_string)
    return tf.strings.regex_replace(lowercase, "[%s]" % re.escape(strip_chars), "")


strip_chars = "!\"#$%&'()*+,-./:;<=>?@[\]^_`{|}~"
strip_chars = strip_chars.replace("<", "")
strip_chars = strip_chars.replace(">", "")

vectorization = TextVectorization(
    max_tokens=VOCAB_SIZE,
    output_mode="int",
    output_sequence_length=SEQ_LENGTH,
    standardize=custom_standardization,
)
vectorization.adapt(text_data)

# Data augmentation for image data
image_augmentation = keras.Sequential(
    [
        layers.RandomFlip("horizontal"),
        layers.RandomRotation(0.2),
        layers.RandomContrast(0.3),
    ]
)
```

Fig 4 : Image, Data pre - processing

```
for line in caption_data:
    line = line.rstrip("\n")
    # Image name and captions are separated using a tab
    img_name, caption = line.split("\t")

    # Each image is repeated five times for the five different captions.
    # Each image name has a suffix `#(caption_number)`
    img_name = img_name.split("#")[0]
    img_name = os.path.join(IMAGES_PATH, img_name.strip())

    # We will remove caption that are either too short to too long
    tokens = caption.strip().split()

    if len(tokens) < 5 or len(tokens) > SEQ_LENGTH:
        images_to_skip.add(img_name)
        continue

    if img_name.endswith("jpg") and img_name not in images_to_skip:
        # We will add a start and an end token to each caption
        caption = "<start> " + caption.strip() + " <end>"
        text_data.append(caption)

        if img_name in caption_mapping:
            caption_mapping[img_name].append(caption)
        else:
            caption_mapping[img_name] = [caption]
```

Fig 5 : Text Vectorization

```
Epoch 1/30
96/96 [==============================] - 100s 819ms/step - loss: 12.1854 - acc: 0.4694 - val_loss: 15.0604 - val_acc: 0.4134
Epoch 2/30
96/96 [==============================] - 71s 737ms/step - loss: 12.1044 - acc: 0.4706 - val_loss: 15.1099 - val_acc: 0.4146
Epoch 3/30
96/96 [==============================] - 70s 727ms/step - loss: 11.8490 - acc: 0.4775 - val_loss: 15.0904 - val_acc: 0.4124
Epoch 4/30
96/96 [==============================] - 69s 715ms/step - loss: 11.6006 - acc: 0.4818 - val_loss: 15.1300 - val_acc: 0.4133
<keras.src.callbacks.History at 0x7fb9f1c27100>
```

Fig 6 : Training, Validation Phase

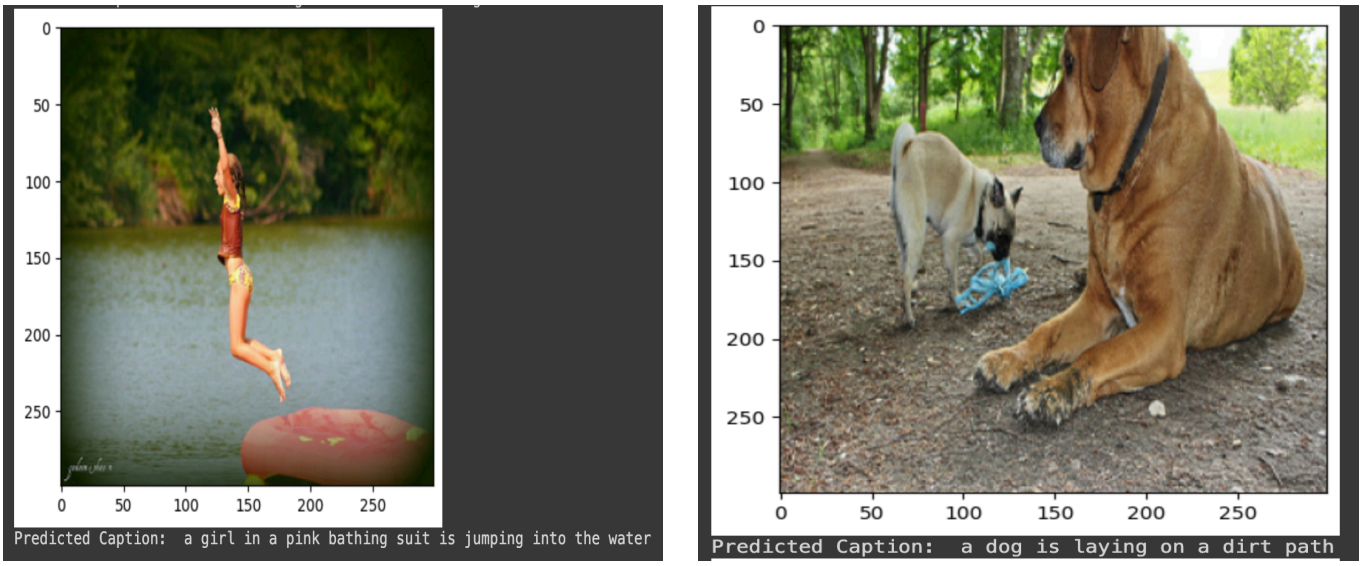

Fig 7 : Image Caption Generation

V. CONCLUSION

The image captioning model proposed in this research paper utilizes the benefits of Transformers and CNNs. It is a challenging task to generate a suitable and grammatically correct caption in a natural language (human), involving concepts from neural networks and natural language processing. CNN and Transformer both did well in identifying a relationship between objects in the images by working together in co-ordination. This analysis uses the Flickr_8k dataset, which has 8000 images and five different captions mapped to each image, for the model's training. The model extracts rich visual features from images by using a CNN (EfficientNetB0) that has already been trained. A Transformer encoder then uses these features to extract contextual relationships and long-range dependencies from the image.Lastly, a Transformer decoder, conditioned on both the encoded image features and previously generated tokens, generates captions word-by-word. Based on the findings, this may be utilised to create captions for several images at once in real time. But there are a few issues with this work as well, which can be fixed in the future. Future research on bigger datasets and fine-tuning of the hyper parameters is possible and should produce better outcomes. After preprocessing, the MS COCO and Flickr30k datasets can be used for this purpose. In a similar manner, different CNN models can be used for this tasks, like InceptionV3, Xception, ResNet, etc., can be explored. Additionally, this research can be expanded to include video captioning and multi-lingual captioning as well.

VI. REFERENCES

[1] Convolutional image captioning, J. Aneja, A. Deshpande, and A.G. Schwan, Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., pp. 5561–5570, 2018, doi: 10.1109/ CVPR.2018.00583.

[2] "Deep Neural Network-Based Visual Captioning," S. Liu, L. Bai, Y. Hu, and H. Wang, MATEC Web Conference, vol. 232, pp. 1–7, 2018, doi: 10.1051/matecconf/201823201052.

[3] ACM Int. Conf. Proceeding Ser., vol. 01052, pp. 1–7, 2018, doi: 10.1145/3240876.3240900, H. Shi, P. Li, B. Wang, and Z. Wang, "Using reinforcement learning for image description".

[4] Fluency-guided cross-lingual image captioning, W. Lan, X. Li, and J. Dong. Proceedings of the 2017 ACM Multimed Conference, MM 2017; doi: 10.1145/3123266.3123366; pp. 1549–557).

[5] "Adding Chinese captions to images," by X. Li, W. Lan, J. Dong, and H. Liu ACM Int. Conf. Multimed. Retr., pp. 271–275, 2016, doi:10.1145/2911996.2912049; ICMR 2016 - Proc.

[6] Y. Yoshikawa, Y. Shigeto, and A. Takeuchi, "STAIR captions: Building a comprehensive dataset of Japanese image captions," ACL 2017, the 55th Annual Meeting of the Association for Computer-Assisted Linguistics, Proceedings of the Conference (Long Papers, vol. 2, pp. 417–421, 2017, doi: 10.18653/v1/ P17-2066.

[7] H.A. Almuthuzini,T.N.Alyahya, andH. Benhidour, an automatic Arabic image captioning using RNN.LSTM based language models and CNN Int. J. Adv. Comput. Sci. Appl., vol. 9, no. 6, pp. 67–73, 2018, doi: 10.14569/ IJACSA.2018.090610.

[8] The Eighth International Conference on ICT, "Indonesia Image Captioning: Adaptable Attention Generation,"". ICoICT 2020, 10.1109/ ICoICT49345.2020.9166244, M. R. S. Mahadi, A. Arifianto, and K. N. Ramadhani.

[9] Wang, C., Yang, H., & Meinel, C. (2018, April 25). Image Captioning with Deep Bidirectional LSTMs and Multi-Task Learning. ACM Transactions on Multimedia Computing, Communications, and Applications, 14(2s), 1–20. https://doi.org/10.1145/3115432

[10] A. Arifianto, A. Nugraha, and Suyanto, Indonesian language : text caption generation using cnn-gated RNN model," 7th Int. Conf. Inf. Commun. Technol. ICoICT 2019, pp. 1–6, 2019, doi: 10.1109/ ICoICT.2019.8835370.

[11] Research presented by Mulyanto et al. (2019) in their study on automatic image caption generation in Indonesian. The authors leverage a CNN-LSTM model and introduce the FEEH-ID dataset for training and evaluation purposes.IEEE Int. Conf. Comput. Intell. Virtual Environ. Meas. Syst. Appl. CIVEMSA 2019 - Proc., 2019, doi: 10.1109/ CIVEMSA45640.2019.9071632.

[**12]** K. Xu and colleagues, "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention," 32nd International Conference on Machine Learning (ICML 2015), vol. 3, pp. 2048–2057, 2015.

[13] "SCA-CNN: Spatial and channel wise attention in convolutional neural networks (CNN) f or image captioning," by L. Chen and colleagues Proc. - 30th IEEE Conf. Vis. Pattern Recognition in Computing , CVPR 2017, vol. 2017-Janua, pp. 6298– 6306, 2017, doi: 10.1109/CVPR.2017.667.

[14] "Show, Attend, and Tell : Attribute-driven attention strategy for describing pictures," by H. Chen, G. Ding, Z. Lin, S. Zhao, and J. Han doi: 10.24963/ijcai.2018/84. IJCAI Int. Jt. Conf. Artif. Intell., vol. 2018-July, pp. 606–612, 2018.

[15] "Image captioning with lexical attention," by Q. You, H. Jin, Z. Wang, C. Fang, and J. Luo 2016, pp. 4651–4659 in Proc. IEEE Comput. Soc. Conference. Comput. Vis. Pattern Recognit., vol. 2016-Decem, doi: 10.1109/CVPR.2016.503.

[16] Chetan Amritkar and Vaishali Jabade, 14th International Conference on Computing, Communication Control, and Automation (ICCUBEA), IEEE, 2018, 978-1-5386-5257-2/18, "Image Caption Generation via Deep Learning Technique" .

[17] 2020 Ali Ashraf Mohamed: CNN and LSTM Image captioning.