IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p-ISSN: 2278-8727, Volume 26, Issue 6, Ser. 3 (Nov. – Dec. 2024), PP 15-24
www.iosrjournals.org

Leveraging AI and Computer Vision for Generating Precise E-Commerce Product Descriptions

Radhika Bansal[1], Karishma Jain[1], Tejaswi Tyagi[1], Roli Bansal[2#]
[1]*Department of Artificial Intelligence and Data Sciences, Indira Gandhi Delhi Technical University for Women, Delhi, India*
[2]*Department of Computer Science, Keshav Mahavidyalaya, University of Delhi, Delhi, India*
[#]*Corresponding Author*

Abstract:
With the exponential growth of e-commerce, there comes the need for compelling and detailed product visibility descriptions that improve the customer experience in online marketplaces. The proposed work addresses the inefficiencies of the manual creation of such descriptions by developing an AI-powered system capable of generating precise and multilingual product descriptions directly from images. Our system automatically extracts visual features by incorporating advanced computer vision techniques and natural language processing models and converts them into rich, human-like textual descriptions. The proposed approach is based on the use of state-of-the-art pre-trained deep learning models, ResNet50, and LSTMs, thus offering high accuracy, semantic relevance, and scalability in terms of captions generation. Multilingual support via automatic translation is also a part of the solution, providing access to global markets and allowing sellers to reach out to linguistically diverse audiences. This innovation simplifies the listing process for sellers, empowering them to manage large inventories efficiently while improving product visibility and accessibility. It delivers more intuitive and trustworthy shopping experiences for customers by offering clear and descriptive product information. By addressing the critical challenges posed by quality inconsistency, time inefficiency, and localization, the proposed work establishes a new level of intelligent, inclusive, and scalable product listing solutions that transform the online marketplace into a much more dynamic and accessible one. Extensive experiments with acceptable BLEU and BERT scores validate that the proposed system can generate descriptions that follow customer expectations in quality and context correctness.

Keywords: *E-commerce; Computer Vision; Natural Language Processing (NLP); Bilingual Evaluation Understudy (BLEU) Score; Bidirectional Encoder Representations from Transformers (BERT) Score; ResNet50; VGG16; VGG19; Long Short-Term Memory (LSTM); Multilingual-support.*

Date of Submission: 11-12-2024 Date of Acceptance: 21-12-2024

I. Introduction

The e-commerce sector has witnessed exponential growth over the last decade, fueled by advancements in digital technology and changing consumer behavior. Online marketplaces [1] like Amazon, eBay, and Flipkart now cater to millions of customers across the globe, with products ranging from everyday essentials to specialized niche items. In this competitive e-commerce environment, the need for high-quality product descriptions is more critical than ever. Accurate and detailed descriptions not only enhance product visibility on search engines but also improve customer engagement and satisfaction. However, manually crafting such descriptions is a time-consuming and labor-intensive process, often resulting in inconsistencies and insufficient detail. For sellers managing large inventories, this inefficiency can lead to missed opportunities, lower sales, and a poor overall shopping experience for customers. The challenges become even more significant in multilingual and diverse markets, where product descriptions must cater to a global audience. Effective translation and localization are essential to ensure accessibility and relevance across regions. For customers, a lack of detailed or accurate product descriptions results in confusion, poor decision-making, and dissatisfaction. Searching for specific products becomes challenging without properly structured and descriptive listings. The motivation for this research arises from these pressing challenges faced by both sellers and customers in the e-commerce landscape.

To overcome these challenges, AI-based solutions combining computer vision [2] and Natural Language Processing (NLP) have emerged as promising tools. Computer vision techniques enable systems to extract attributes from product images like colors, and objects, while NLP [3] models transform these attributes into coherent and descriptive text. Despite progress in these technologies, gaps remain in their application to e-commerce, particularly regarding accuracy, multilingual translation, and integration with seller platforms. This

work focuses on bridging these gaps through a system that automates product description generation, leveraging state-of-the-art AI models such as Visual Geometry Group 16 (VGG16) [4], Visual Geometry Group (VGG19) [5], and ResNet50 [6]. By comparing these models' performance, we identify the most effective approach for extracting product attributes and generating high-quality, multilingual descriptions.

The primary contribution of this research is to develop an AI-powered system that automates the generation of detailed and high-quality product descriptions from product images by:

- Creating a suitable dataset containing fashion product images and their corresponding descriptions.
- Developing a deep learning-based system that combines computer vision and natural language processing to generate detailed product descriptions.
- Evaluating the performance of the proposed system using standard evaluation metrics - BLEU Score [7] which measures the accuracy of generated captions compared to ground truth and BERT Score [8] which captures semantic similarity between generated and actual captions.

II. Related Work

Wang, Hou, Liu, et al. [9], propose a statistical framework for generating coherent and fluent product descriptions based on product attributes. Initially, sentence-level templates are extracted and aligned with attributes, and then candidates are ranked to produce an optimal description. "Templated knowledge" is used to ensure coherence and "structured knowledge" is to understand attribute dependencies and select relevant attributes. This approach achieved a BLEU (bilingual evaluation understudy) score of 0.154, showing a significant improvement over the baseline model on the "Computers & Tablets" category on Amazon, containing 25,375 samples with paired descriptions and attribute lists. Limitations of this study are that it relies on structured text data and does not work on image data, it depends on aligned templates and product attributes, limiting flexibility for purely visual inputs, and it lacks integration with neural networks.

Wang, Zhang, and Yu [10], use various benchmark datasets such as MS COCO, Flickr8k/30k, PASCAL 1K, and others, to evaluate caption generation models. They employ hand-crafted feature-based approaches (e.g., CRF-based and maximum likelihood models) and neural encoder-decoder models (e.g., CNN [12] paired with RNNs or LSTMs for feature extraction and caption generation). The primary focus is the attention mechanisms that dynamically focus on different parts of an image or text input to improve caption accuracy. Evaluation metrics such as BLEU, METEOR, CIDEr, and ROUGE, with deliberate attention reaching a BLEU-4 score of 0.375. However, it lacks focus on product-specific details and doesn't address extracting attribute-value pairs directly from images.

Cui, Yang, et al. [11] utilized a large-scale dataset of labeled images from Fine-Grained Visual Categorization (FGVC) tasks to evaluate model performance, specifically focusing on domain-specific categories such as bird species and car models. The proposed method introduced a novel "Knowledge Transfer" framework for domain adaptation, where a pre-trained network is fine-tuned to assess models on small-scale datasets. The results demonstrated improved evaluation accuracy and better generalization across diverse datasets, highlighting the effectiveness of the approach in handling domain-specific image captioning tasks. However, this method relies heavily on pre-training, which may not generalize well to drastically different domains without extensive fine-tuning.

Cai, Liu, et al. [12] (2024), introduce a newly collected dataset of fine-grained product images, specifically designed for attribute extraction tasks. This dataset addresses challenges such as varying lighting conditions and complex backgrounds, which often hinder accurate attribute extraction. The paper proposes a multi-task learning model that integrates Convolutional Neural Networks (CNNs) with Graph Neural Networks (GNNs) to extract and correlate attributes from product images effectively. The proposed model achieves state-of-the-art results in attribute extraction, demonstrating a significant improvement in accuracy over baseline models. The model's performance may degrade when applied to datasets with unseen attributes or significantly noisier data.

Jafar, Ghneim, et al. [13] (2022), introduce an attention-based Encoder-Decoder framework for image captioning, leveraging two methods of feature extraction. The first method employs an image classification CNN, specifically Xception, while the second utilizes an object detection module (YOLOv4) to extract object-level features. The model is evaluated on the MS COCO and Flickr30k datasets. While the approach effectively enhances feature extraction, it falls short in capturing deeper relationships between objects within the image, limiting its ability to fully mimic human image understanding.

Rinaldi, Russo, et al. [14] (2023) present a method that integrates three deep neural networks—Mask R-CNN, YOLOv3, and RetinaNet—to enhance image captioning. These models are utilized for object detection, and their outputs are combined using a hierarchical classification approach to improve detection accuracy. The combined results are then passed to a captioning module, which generates textual descriptions of the images. This approach demonstrates improved precision and detection accuracy compared to using a single model, leading to more accurate image captions. The method achieves significant gains in mean Average Precision

(mAP) across various images. However, the generated captions adhere to a rigidly structured format, which restricts natural language flexibility and limits the fluency needed to reflect human-like descriptions.

Farhadi, Young, Hejrati, et al. [15] (2010) introduce a model that generates descriptive sentences for images by analyzing detected objects, actions, and scenes. The approach combines visual and semantic processing to map images to structured sentences, enabling the system to produce concise and descriptive captions. The authors evaluate the method on popular datasets like PASCAL, Flickr8k, and custom datasets with annotated objects and scenes. Results indicate that the model outperforms earlier methods in terms of human-evaluated description accuracy, particularly for simpler image contexts. However, limitations include challenges in producing complex narratives or capturing deeper relationships between objects. The model also struggles with ambiguity or multi-layered interpretations present in certain images, which can affect the quality of the generated descriptions.

Papineni, Roukos, et al. [16], introduce the BLEU (Bilingual Evaluation Understudy) score, a widely used metric for assessing machine translation quality. The BLEU score evaluates translations by comparing machine-generated text to one or more reference translations, focusing on n-gram precision to measure word overlap while incorporating a brevity penalty to address excessively short outputs. The authors demonstrate that BLEU correlates well with human judgments of translation quality, providing a fast, inexpensive, and language-independent alternative to manual evaluations. However, BLEU is less effective for translations requiring contextual understanding or nuanced language, as it primarily focuses on surface-level word matches without considering deeper semantic meaning or fluency.

Shinde, Gawde, and Paradkar [17] (April 2021) explore a framework that employs VGG16, a convolutional neural network (CNN), to extract image features, which are then fed into an LSTM (Long Short-Term Memory) network to generate sequential captions. The methodology combines visual features with caption text in an LSTM layer, allowing the model to produce context-aware captions that align with the image content. The approach performs well in generating factual descriptions of images, effectively capturing basic details like objects and their relationships. However, the model struggles to produce captions that convey feelings or interpret complex situations, limiting its ability to reflect deeper semantic understanding or emotional nuances. The system is evaluated using popular datasets such as Flickr and COCO, which are widely used benchmarks for image captioning tasks.

Chauhan, Beniwal, and Arya [18] (2023) present a CNN-LSTM model for generating captions from images, evaluated on widely used datasets such as MS COCO, Flickr 8K, and Flickr30K. The proposed methodology utilizes a Convolutional Neural Network (CNN), specifically VGGNet, to extract high-level image features. These extracted features are then passed to a Long Short-Term Memory (LSTM) network, which processes them sequentially to generate textual captions based on learned patterns from the training data. The model performs well in creating factual descriptions, and effectively identifying objects and basic relationships within the images. However, it struggles to produce more nuanced captions that capture sentiment or complex contextual details, limiting its ability to convey deeper meaning or emotional tones. The focus on factual accuracy highlights its strengths for straightforward caption generation but exposes limitations when addressing ambiguous or intricate visual content.

Thus, the literature survey on image captioning and product description generation highlighted several gaps in existing methodologies such as: generating product descriptions from structured input text or text-based attributes rather than from product images directly, not catering to a global audience by incorporating automatic translation capabilities, missing product-specific nuances by not focusing on capturing fine-grained product details such as color, object, etc. The proposed work has tried to address these gaps by focusing on generating descriptions from images and adding multilingual support.

III. Material And Methods

This section presents the proposed system for automatic product description generation for e-commerce platforms. The detailed block diagram of the proposed methodology is depicted in Figure 1 below. Each step is explained in detail in the following subsections:

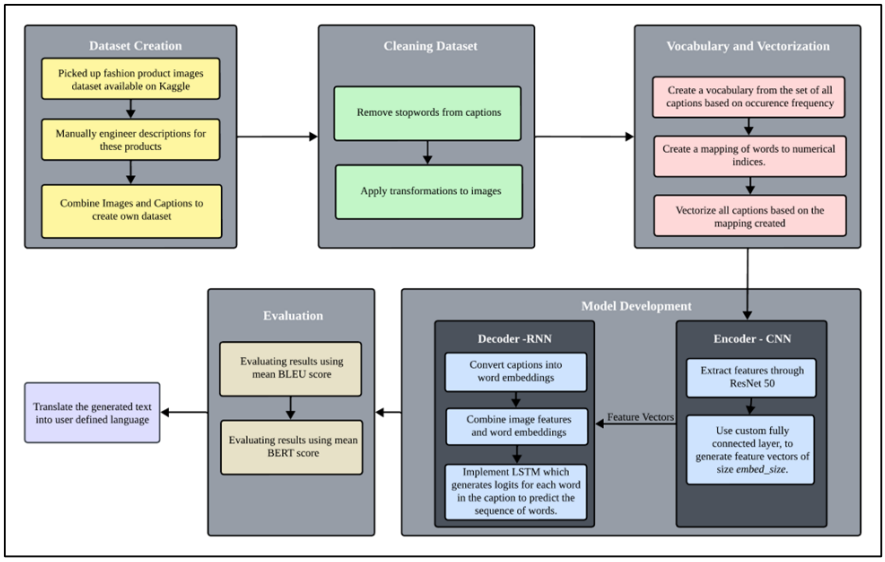

Figure no. 1: Proposed Methodology

Dataset Creation

For this work, we created a custom dataset using images sourced from the Kaggle dataset [19] to obtain e-commerce product images. We manually curated descriptions for each product, compiling a final dataset in an image-caption format. This step was necessary because existing datasets were not specifically tailored to e-commerce products. To align with our objective, we built a dataset consisting of approximately 9,000 images and their corresponding captions.

Cleaning and Transforming the Dataset

The next step in our process is cleaning the:

1. Caption data: The two main steps that we employ are: removing punctuation from the input text and removing words with fewer than 3 characters.
2. Image Data: For input images, the following default transformations are applied using the transform function of torchvision [20]: each image is resized to (224, 224) which is the input size for CNNs like ResNet and converted to tensors. Pixel values are normalized with mean and standard deviation.

Vocabulary and Vectorization

The next step is creating vocabulary from the dataset, which means taking all the captions of the dataset and splitting them into individual words or subwords. After this, the frequency of each word is calculated to help figure out which words occur more frequently throughout the dataset. The most common words are kept in the vocabulary and infrequent words are deleted from the vocabulary to eliminate noise within the vocabulary. Thus, the model will not closely fit the rare terms, preventing overfitting which will then fail to meaningfully improve learning.

We create the vocabulary through the unique mapping of words to an integer ID. The two dictionaries used are `stoi` (string-to-integer mapping) and `itos` (integer-to-string mapping) so that the model can turn the text into numerical representations for processing. Special tokens are then used for specific cases. Some of the important tokens [21] include <PAD> used to pad shorter sentences, <SOS> to mark the beginning of a sentence, <EOS> a sentence closer, and <UNK> in words not found in the vocabulary.

Once the vocabulary is created, vectorization translates the textual data into a form that is numerically convenient for processing in machine learning. All tokens in the sentence are replaced by corresponding IDs from the vocabulary. If there is no corresponding token from the vocabulary then it will be replaced with the ID of the <UNK> token. Therefore, the model does not fail when the word is not known to the model.

The problem of unequal lengths of sentences is solved using padding, so all the sequences will be of equal length. The shorter sentences compared to the maximum length will be padded with `<PAD>` tokens. These padded, numerical sequences are then passed through embedding layers of the model which converts the numerical IDs into dense vectors. These vectors help in capturing semantic relationships between words and enable the model to understand their meaning and context better.

Model Development

The model comprises two main building blocks namely EncoderCNN (Encoder Convolutional Neural Network) and DecoderRNN (Decoder Recurrent Neural Network). The encoder extracts high-level features from product images while the decoder combines the visual context with the word embedding of the true captions to iteratively predict the next sequence of words in the captions.

The EncoderCNN extracts meaningful high-level visual features from input images using a pre-trained CNN [22] ResNet50. The original fully connected layer of ResNet50 is replaced with a new linear layer with parameters that include the input size of the previous layer and the dimensionality of the output feature vector, which serves as the image embedding. This is done because instead of classifying images into categories, we want a dense feature representation of the image that can be passed to the decoder. ReLU function [23] is used for activation, introducing non-linearity to the output. Dropout [24] is also employed to randomly set 50% of the neurons to 0 during training to prevent overfitting. A single forward pass of this encoder accepts a batch of images as tensors and outputs a tensor of shape *(batch_size, embed_size)* representing the encoded image features.

The Decoder (RNN) [25] generates a sequence of words (captions) based on the image features provided by the Encoder and true captions. We convert input word indices (captions) into dense word embeddings of size *embed_size*. Each word is represented as a vector in a continuous space. A Long Short-Term Memory (LSTM) network processes sequential data (captions) word by word by taking the parameters like *embed_size*: input size for the LSTM (word embedding size), *hidden_size*: dimensionality of the LSTM's hidden state, *num_layers*: number of stacked LSTM layers for learning more complex patterns. The Linear Layer maps the LSTM's hidden state to the vocabulary space while Dropout adds regularization to the embedding layer. We use the ADAM optimizer [26] together with the loss function to update the model weights. This model architecture, comprising of the encoder-decoder, can be visualized with the help of Figure 2 below. One forward pass of the decoder involves:

1. Embedding Captions: Converts input captions (word indices) to embeddings using the embed function.
2. Concatenating Features: Appends the image features (from Encoder) at the start of the caption sequence
3. LSTM Processing: Feeds the concatenated sequence into the LSTM.
4. Linear Projection: The LSTM's output is passed through the linear layer to produce vocabulary-sized logits at each time step.
5. Output: A tensor of shape *(sequence_length + 1, batch_size, vocab_size)* containing predicted word distributions.

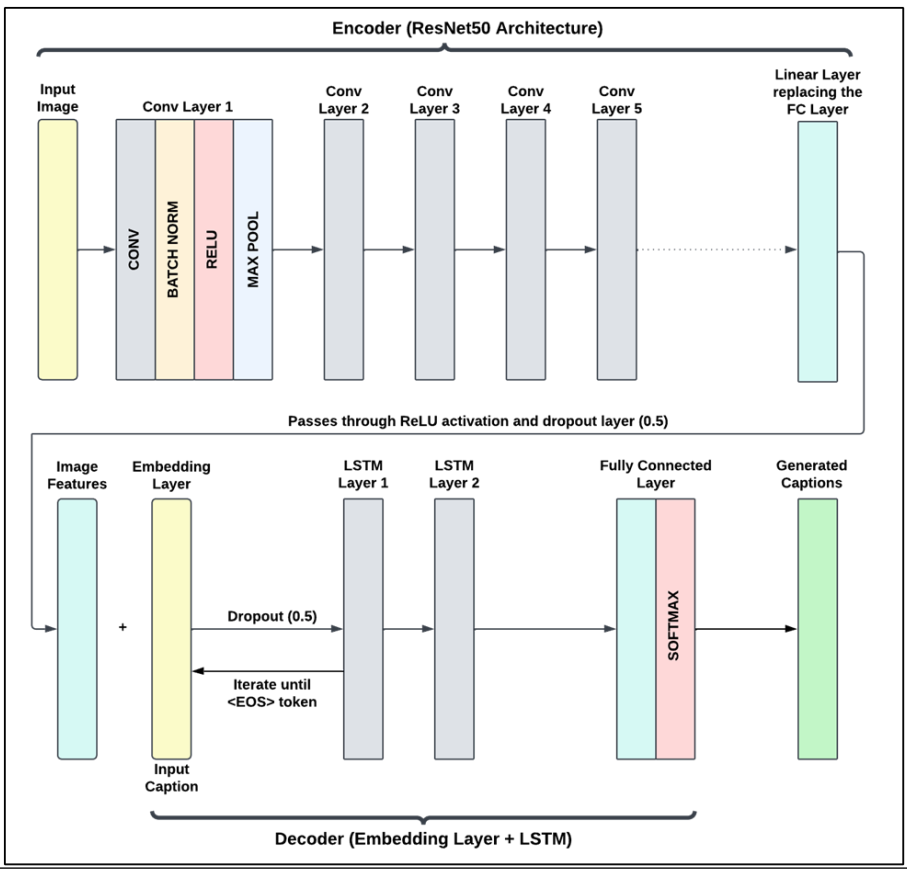

Figure no. 2: Proposed Model Architecture

Translation

Finally, when a user submits an image for a product description, the system also asks them to input their preferred language for the description. Once the proposed model generates the description, it uses the GoogleTrans library [27] to translate it to the desired language before sending the result.

IV. Results And Discussion

In this work, we explored various approaches to enhance the accuracy of our model and its evaluation metrics. Specifically, we compared the performance of the model using different feature extraction methods, including VGG16, VGG19, and ResNet, combined with LSTM for caption generation. To evaluate the results, we used the BERT Score, which measures the semantic similarity between predicted and reference captions, and the BLEU score, which evaluates the n-gram overlap between them. Notably, a BLEU score of 0.3 or higher is considered decent and indicative of high-quality translations. Table 1 below summarizes the results of our experiments. It can be clearly seen that the proposed model performed best when coupled with ResNet50 for feature extraction, achieving a BLEU score of 0.513 and a BERT score of 0.808.

Model used for feature extraction	BLEU Score	BERT Score
VGG16	0.182	0.312
VGG19	0.147	0.547
ResNet50	0.513	0.808

Table no. 1: Shows BLEU and BERT Scores obtained by different CNN models

As discussed in section III under "Dataset Creation", some examples of records in our dataset with fashion product images are highlighted in Figure 3. The dataset contains 9,000 image-caption pairs and includes a wide variety of fashion products like clothing, accessories, shoes, etc.

Figure no. 3: Curated Dataset

After cleaning the captions by removing punctuations and words less than 3 characters as discussed in section III under vocabulary and vectorization, we print the 50 most and least frequently occurring words in the vocabulary, results are shown in Figure 4 below.

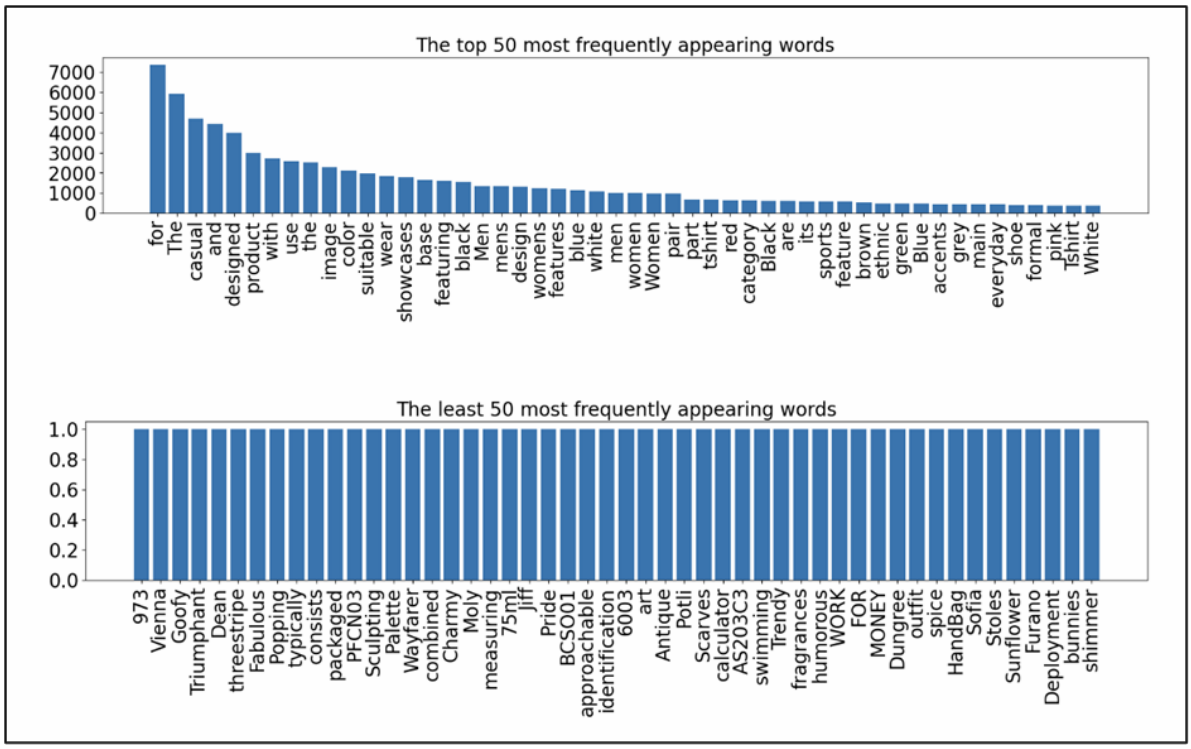

Figure no. 4: Caption Analysis

The layer summary of the encoder and decoder discussed in section III under "model development", designed using a Convolutional Neural Network model and an LSTM model respectively are shown in Figure 5.

```
==========================================================================================================================================
Layer (type:depth-idx)                                  Input Shape               Output Shape              Param #                   Trainable
==========================================================================================================================================
CNNtoRNNWithDummy                                       [32, 3, 224, 224]         [51, 32, 3792]            --                        True
├─EncoderCNN: 1-1                                       [32, 3, 224, 224]         [32, 256]                 --                        True
│    └─ResNet: 2-1                                      [32, 3, 224, 224]         [32, 256]                 --                        True
│    │    └─Conv2d: 3-1                                 [32, 3, 224, 224]         [32, 64, 112, 112]        9,408                     True
│    │    └─BatchNorm2d: 3-2                            [32, 64, 112, 112]        [32, 64, 112, 112]        128                       True
│    │    └─ReLU: 3-3                                   [32, 64, 112, 112]        [32, 64, 112, 112]        --                        --
│    │    └─MaxPool2d: 3-4                              [32, 64, 112, 112]        [32, 64, 56, 56]          --                        --
│    │    └─Sequential: 3-5                             [32, 64, 56, 56]          [32, 256, 56, 56]         215,808                   True
│    │    └─Sequential: 3-6                             [32, 256, 56, 56]         [32, 512, 28, 28]         1,219,584                 True
│    │    └─Sequential: 3-7                             [32, 512, 28, 28]         [32, 1024, 14, 14]        7,098,368                 True
│    │    └─Sequential: 3-8                             [32, 1024, 14, 14]        [32, 2048, 7, 7]          14,964,736                True
│    │    └─AdaptiveAvgPool2d: 3-9                      [32, 2048, 7, 7]          [32, 2048, 1, 1]          --                        --
│    │    └─Linear: 3-10                                [32, 2048]                [32, 256]                 524,544                   True
│    └─ReLU: 2-2                                        [32, 256]                 [32, 256]                 --                        --
│    └─Dropout: 2-3                                     [32, 256]                 [32, 256]                 --                        --
├─DecoderRNN: 1-2                                       [32, 256]                 [51, 32, 3792]            --                        True
│    └─Embedding: 2-4                                   [50, 32]                  [50, 32, 256]             970,752                   True
│    └─Dropout: 2-5                                     [50, 32, 256]             [50, 32, 256]             --                        --
│    └─LSTM: 2-6                                        [51, 32, 256]             [51, 32, 512]             3,678,208                 True
│    └─Linear: 2-7                                      [51, 32, 512]             [51, 32, 3792]            1,945,296                 True
==========================================================================================================================================
Total params: 30,626,832
Trainable params: 30,626,832
Non-trainable params: 0
Total mult-adds (G): 136.96
==========================================================================================================================================
Input size (MB): 19.27
Forward/backward pass size (MB): 5749.90
Params size (MB): 122.51
Estimated Total Size (MB): 5891.67
==========================================================================================================================================
```

Figure no. 5: Encoder-Decoder Summary

Figure 6 depicts the epoch summary of the LSTM model implemented in the decoder as explained in section III under "model development". This is followed by Figure 6 which shows the graph of the number of epochs vs the training and validation loss of this model.

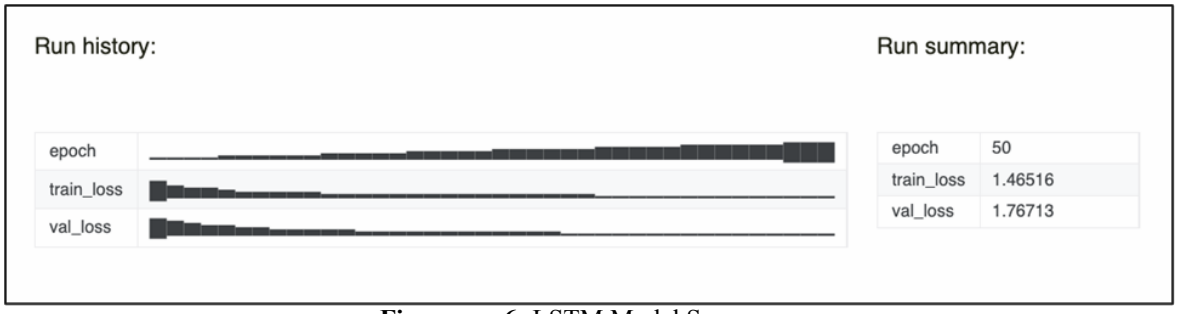

Figure no. 6: LSTM Model Summary

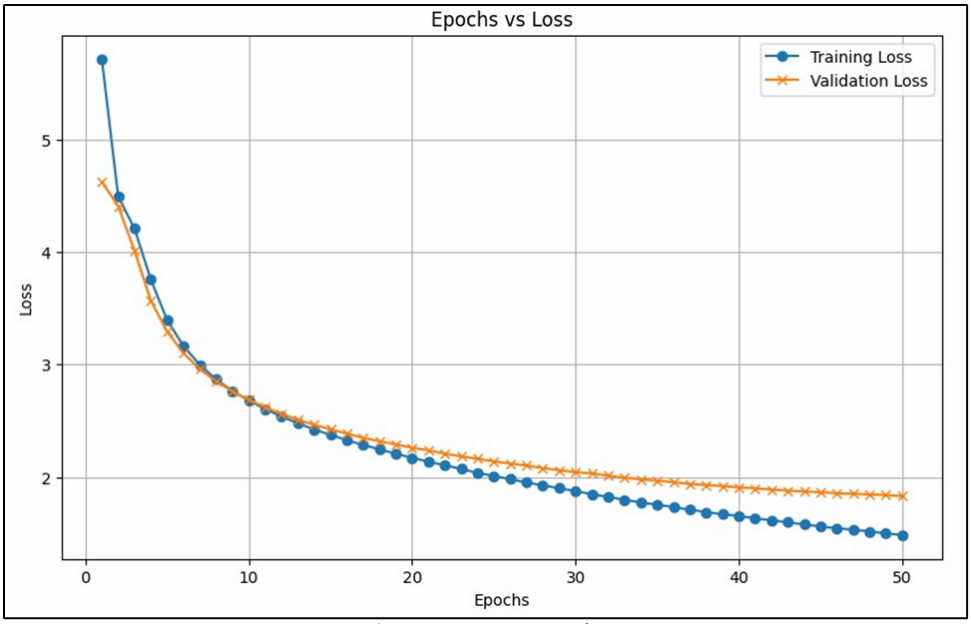

Figure no. 7: Loss Plot

As summarized in Table 1, the results of the BLEU score and BERT score of VGG16, VGG19, and ResNet50 are noted and we observe that ResNet50 gives the best results. Some predictions made by this model are depicted in Figure 8.

Figure no. 8: Generated Captions

As depicted in Figure 8, the generated captions demonstrate the ability of our model to produce high-quality, coherent, and contextually relevant descriptions for products. These results highlight the model's capability to bridge the gap between visual perception and natural language generation.

V. Conclusion

This paper addresses several critical challenges in the landscape of e-commerce by developing an AI-based system for automatically generating rich, precise, and multilingual product descriptions from images. Our solution fills gaps in the manual process of description generation using the best possible deep learning models, including ResNet50 and LSTMs. The system benefits both the seller in terms of managing their huge inventory and enhances the experience of customers by providing rich contextually relevant and globally accessible product descriptions. The experimental results have verified the effectiveness of our approach, where ResNet50 has achieved the best BLEU and BERT scores, which demonstrates its superior capability in extracting fine-grained visual features and generating meaningful captions. Moreover, by integrating translation features, our system extends accessibility to various linguistic markets, where the need for localization within global e-commerce platforms is addressed.

In the future, we can further revolutionize the shopping experience by introducing features that prioritize personalization and user engagement. Additionally, visual search and discovery can be integrated, enabling customers to find products simply by uploading images, making the search process faster, more intuitive, and visually interactive. To build trust and increase purchases, sentiment-driven descriptions may also be generated by analyzing customer reviews and highlighting the features most valued by users, ensuring product descriptions resonate with their needs and preferences.

References

[1]. Etailize. (2024, May 30). Top 20 Marketplaces In India | E-Tailize. E-Tailize. Https://E-Tailize.Com/Blog/Top-20-Marketplaces-In-India/

[2]. Hmrishav Bandyopadhyay (2022). What Is Computer Vision? [Basic Tasks & Techniques]. V7. Https://Www.V7labs.Com/Blog/What-Is-Computer-Vision

[3]. Saadani, T. (2023, December 6). 6 Natural Language Processing Models You Should Know. Medium. Https://Medium.Com/Ubiai-Nlp/5-Natural-Language-Processing-Models-You-Should-Know-836958303ce3

[4]. Qassim, H., Verma, A., & Feinzimer, D. (2018). Compressed Residual-Vgg16 Cnn Model For Big Data Places Image Recognition. 2022 Ieee 12th Annual Computing And Communication Workshop And Conference (Ccwc). Https://Doi.Org/10.1109/Ccwc.2018.8301729

[5]. Bansal, M., Kumar, M., Sachdeva, M., & Mittal, A. (2021). Transfer Learning For Image Classification Using Vgg19: Caltech-101 Image Data Set. Journal Of Ambient Intelligence And Humanized Computing, 14(4), 3609–3620. Https://Doi.Org/10.1007/S12652-021-03488-Z

[6]. Microsoft/Resnet-50 · Hugging Face. (2001, July 16). Https://Huggingface.Co/Microsoft/Resnet-50

[7]. Bleu - A Hugging Face Space By Evaluate-Metric. (N.D.). Https://Huggingface.Co/Spaces/Evaluate-Metric/Bleu

[8]. Bert Score - A Hugging Face Space By Evaluate-Metric. (N.D.). Https://Huggingface.Co/Spaces/Evaluate-Metric/Bertscore

[9]. Wang, J., Hou, Y., Liu, J., Cao, Y., & Lin, C. (2017, November 1). A Statistical Framework For Product Description Generation. Acl Anthology. Https://Aclanthology.Org/I17-2032/

[10]. Wang, H., Zhang, Y., & Yu, X. (2020). An Overview Of Image Caption Generation Methods. Computational Intelligence And Neuroscience, 2020, 1–13. Https://Doi.Org/10.1155/2020/3062706

[11]. Cui, Y., Yang, G., Veit, A., Huang, X., & Belongie, S. (2018). Learning To Evaluate Image Captioning. Https://Openaccess.Thecvf.Com/Content_Cvpr_2018/Html/Cui_Learning_To_Evaluate_Cvpr_2018_Paper.Html

[12]. Liu, T., Cai, Q., Xu, C., Hong, B., Xiong, J., Qiao, Y., & Yang, T. (2024, March 24). Image Captioning In News Report Scenario. Arxiv.Org. Https://Arxiv.Org/Abs/2403.16209

[13]. Al-Malla, M. A., Jafar, A., & Ghneim, N. (2022). Image Captioning Model Using Attention And Object Features To Mimic Human Image Understanding. Journal Of Big Data, 9(1). Https://Doi.Org/10.1186/S40537-022-00571-W

[14]. Antonio M. Rinaldi, Cristiano Russo, Cristian Tommasino (2023 June). Automatic Image Captioning Combining Natural Language Processing And Deep Neural Networks. Results In Engineering, Volume 18. Https://Www.Sciencedirect.Com/Science/Article/Pii/S2590123023002347

[15]. Farhadi, A., Hejrati, M., Sadeghi, M. A., Young, P., Rashtchian, C., Hockenmaier, J., & Forsyth, D. (2010). Every Picture Tells A Story: Generating Sentences From Images. In Lecture Notes In Computer Science (Pp. 15–29). Https://Doi.Org/10.1007/978-3-642-15561-1_2

[16]. Papineni, K., Jr., Roukos, S., Ward, T., Zhu, W.-J., & Ibm T. J. Watson Research Center. (2002). Bleu: A Method For Automatic Evaluation Of Machine Translation. In Ibm T. J. Watson Research Center. Https://Aclanthology.Org/P02-1040.Pdf

[17]. Omkar Shinde, Rishikesh Gawde, Anurag Paradkar (2021 April). Image Caption Generation Methodologies. International Research Journal Of Engineering And Technology (Irjet), Volume: 08 Issue: 04. Https://Www.Researchgate.Net/Profile/Rishi-G/Publication/351840108_Image_Caption_Generation_Methodologies

[18]. Chauhan, A., Beniwal, A., & Arya, M. D. (2023). Research Paper On Image Caption Generator Using Deep Learning. Journal Of Emerging Technologies And Innovative Research (Jetir), 10(4). Https://Www.Jetir.Org/Papers/Jetir2304298.Pdf

[19]. Fashion Product Images (Small). (2019, April 26). Kaggle. Https://Www.Kaggle.Com/Datasets/Paramaggarwal/Fashion-Product-Images-Small

[20]. Torchvision.Transforms — Torchvision Master Documentation. (N.D.). Https://Pytorch.Org/Vision/0.9/Transforms.Html

[21]. Different Usage Of , And Tokens. (N.D.). Stack Overflow. Https://Stackoverflow.Com/Questions/62464541/Different-Usage-Of-Pad-Eos-And-Go-Tokens

[22]. Liu, Q., & Mukhopadhyay, S. (2018). Unsupervised Learning Using Pretrained Cnn And Associative Memory Bank. 2022 International Joint Conference On Neural Networks (Ijcnn), 01–08. Https://Doi.Org/10.1109/Ijcnn.2018.8489408

[23]. Ide, H., & Kurita, T. (2017). Improvement Of Learning For Cnn With Relu Activation By Sparse Regularization. 2022 International Joint Conference On Neural Networks (Ijcnn), 2684–2691. Https://Doi.Org/10.1109/Ijcnn.2017.7966185
[24]. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A Simple Way To Prevent Neural Networks From Overfitting. Journal Of Machine Learning Research, 15(1), 1929–1958. Https://Jmlr.Csail.Mit.Edu/Papers/Volume15/Srivastava14a/Srivastava14a.Pdf
[25]. Sherstinsky, A. (2020). Fundamentals Of Recurrent Neural Network (Rnn) And Long Short-Term Memory (Lstm) Network. Physica D Nonlinear Phenomena, 404, 132306. Https://Doi.Org/10.1016/J.Physd.2019.132306
[26]. Zhang, Z. (2018). Improved Adam Optimizer For Deep Neural Networks. -, 1–2. Https://Doi.Org/10.1109/Iwqos.2018.8624183
[27]. Googletrans: Free And Unlimited Google Translate Api For Python — Googletrans 3.0.0 Documentation. (N.D.). Https://Py-Googletrans.Readthedocs.Io/En/Latest/