Evaluating Performances of Attention-Based Merge Architecture Models for Image Captioning in Indian Languages

Rahul Tangsali, Swapnil Chhatre *, Soham Naik, Pranav Bhagwat, and Geetanjali Kale

Department of Computer Engineering, SCTR's Pune Institute of Computer Technology, Pune, India;
Email: rahuul2001@gmail.com (R.T.), nsoham01@gmail.com (S.N.), gvkale@pict.edu (G.K.),
pranav221b@gmail.com (P.B.)
*Correspondence: swapchhatre5@gmail.com (S.C.)

***Abstract*—Image captioning is a growing topic of research in which numerous advancements have been made in the past few years. Deep learning methods have been used extensively for generating textual descriptions of image data. In addition, attention-based image captioning mechanisms have also been proposed, which give state-of-the-art results in image captioning. However, many applications and analyses of these methodologies have not been made in the case of languages from the Indian subcontinent. This paper presents attention-based merge architecture models to achieve accurate captions of images in four Indian languages- Marathi, Kannada, Malayalam, and Tamil. The widely known Flickr8K dataset was used for this project. Pre-trained Convolutional Neural Network (CNN) models and language decoder attention models were implemented, which serve as the components of the merge-architecture proposed here. Finally, the accuracy of the generated captions was compared against the gold captions using Bilingual Evaluation Understudy (BLEU) as an evaluation metric. It was observed that the merge architectures consisting of InceptionV3 give the best results for the languages we test on, the scores discussed in the paper. Highest BLEU-1 scores obtained for each language were: 0.4939 for Marathi, 0.4557 for Kannada, 0.5082 for Malayalam, and 0.5201 for Tamil. Our proposed architectures gave much higher scores than other architectures implemented for these languages.**

***Keywords*—image captioning, Recurrent Neural Networks (RNN), Long Short-Term Memory Unit (LSTM), GRU, Pre-trained Convolutional Neural Network (CNN) models, Indian languages**

I. Introduction

With the influx of vast amounts of image data from numerous sources worldwide, there exists the need to have robust mechanisms for extracting valuable insights from the same. The concept of description generation for visual data comes into the picture. There are numerous applications where there is a need for further research into this concept. For example, blind people worldwide cannot comprehend their surroundings visually. Analysis has been carried out regarding the same as well [1], and this is where the concept of textual description generation comes into play. Other use cases of the idea include summarizing long videos for scientific and investigative research, deriving insights from a large set of images, etc. In this research, the focus is limited to the captioning of static image data only.

Image captioning is the process of describing the content of an image using text. Thus, image captioning research lies at the intersection of computer vision and natural language processing. The main aim behind image captioning is to make the model generate a descriptive caption of the image, almost as good as a human can describe it. Therefore, the task at hand is not only to develop the relevant words but also to keep in mind the order in which they need to be generated. Otherwise, we might end up with many meaningful but mixed-up words. Recent approaches to image captioning are good at predicting the context of the pictures and not just describing what is in them.

Marathi is an Indo-Aryan language spoken primarily by the Marathi people of Maharashtra, India. It is one of India's 22 scheduled languages, with 83 million speakers recorded in 2011. Marathi is the tenth most spoken language in the world, and it has the third highest number of native speakers in India. The Dravidian family of languages, consisting of Kannada, Tamil, Malayalam, and Telugu, is a family of languages spoken in the Indian Peninsula; Kannada, Tamil, and Malayalam are spoken by around 152 million people across the world. Most of the research in image captioning is done for high-resource languages, from communication within a team to documentation and presenting the study to the world. However, the same research scale isn't observed for low-resource languages, especially those in the Indian subcontinent, even though they are spoken by a considerably massive group of people across the globe. This serves as a motivation to perform research in low-resource Indian languages and achieve pivotal accuracies in the same.

This paper presents a merge architecture approach with ap-plied attention by combining image feature extraction

Manuscript received February 6, 2023; revised March 23, 2023; accepted April 15, 2023.

models such as InceptionV3, Residual Networks-50 (ResNet50), and Visual Geometry Group-16 (VGG16) along with language models such as vanilla Recurrent Neural Network (RNN), Long Short-Term Memory Unit (LSTM), and Gated Recurrent Unit (GRU). These models are permuted within them- selves, and each feature extraction model is combined with each language model, thus resulting in nine different systems for each of the four languages (Marathi, Kannada, Malayalam, Tamil). The BLEU metric is used to evaluate the accuracy of the generated captions against the gold captions present. Accomplished results are achieved with the attention model compared to other image captioning approaches implemented in the past for Indian languages.

II. Literature Review

Image captioning has found its applications in a variety of real-world use cases. There has been a recent increase in scholarly interest regarding image captioning. Many approaches have been implemented to generate photo captions to achieve state-of-the-art accuracies. In the initial stages of research, approaches proposed depended upon extreme constraints applied to the datasets, with straightforward descriptions generated for the same. Kojima *et al.* proposed concept hierarchies of actions to generate captions for objects present in an office setting [2]. Even Hede et al. presented a method of using object dictionaries and language models to describe images of objects in a clutter-free background [3]. These methods, however, had limited scalability and applications to other scenarios. Template-based and retrieval-based approaches had also been studied [4–10]. Template-based approaches have fixed templates with some empty spaces for captioning. In these approaches, different objects, properties, and actions are first detected, then gaps in the template are filled. Retrieval- based methods retrieve captions from an existing caption set. They first find images which are visually similar to their annotations from the training dataset, also called candidate captions. Then, the caption for the input image is selected from the captions pool [11]. In both of these approaches, however, there is little flexibility in applying them to real-world situations.

With the advent of deep learning since 2010, numerous deep learning-based approaches have been discussed and put forth [12]. Although deep neural networks have been widely applied today to solve the task of creating image captions, different methods may be based on other frameworks. Some of the fundamental research involves works, etc. [13–15]. Multi-modal image captioning was further introduced, which considers images and textual modalities. In this approach, the features are extracted and quantified from the input image. Then, they are mapped to a shared space along with the word features, where the next word of the caption is predicted based on the image features and the previously generated words. Relevant research includes works, etc. [16, 17]. Encoder-decoder based approaches were proposed later, which were aided by attention, later on, to give state-of-the-art accuracies in caption generation [18–21]. As part of this, a neural encoder first encodes an image in an intermediate representation, which is then taken by the decoder to generate the output word-by-word [22].

Indian languages have slowly been focused upon as subjects for image captioning research. Kumar *et al.* [23] worked on creating a custom dataset for image captioning in Tamil by translating captions from the MSCOCO dataset in English to Tamil and then experimented with several multi-modal architectures to provide captions of images directly in Tamil for any given input image [23, 24]. Dwarampudi *et al.* [25] used InceptionV3, a pre-trained CNN model for extracting the image features, which are then processed using pre-trained fastText word embeddings of Telugu on a Recurrent Neural Network of Longshort-term memory architecture [25–27].

More researchers from the NLP and computer vision communities are now involved in this study on image captioningin Indian languages.

III. Proposed Architecture

A. Dataset Description

The Flickr8K dataset is used for training purposes. The dataset is open-source and widely used for image captioning purposes. Flickr8K dataset consists of 8092 images, each image consisting of five distinct English captions describing it. The dataset images are chosen such that no well-known persons or locations appear but instead portray various genericevents and scenarios.

The images are not of fixed dimensions but have either a width or a height of 500 pixels. The captions are one sentence each, either in a phrasal or full-sentence format. First, we translated the English captions into the Indian languages we plan to work on, i.e., Marathi, Kannada, Malayalam, and Tamil. Then, we used them and the original images to constitute our final raw dataset.

B. Dataset Preparation

The English captions were translated into the following four languages: Marathi, Kannada, Malayalam, and Tamil using the Google Translate API[1] officially provided by Google. The final dataset consisted of the images folder and separate CSV files for each language, containing the captions to the corresponding image IDs.

For every natural language processing task, pre-processing of text is essential. The CSV file was converted into a map data structure, with each image ID mapping to five different captions. Since the text would further be passed to a sequence- to-sequence language model (a simple RNN, LSTM, or GRU), it is essential to set special tokens at the start and end of each caption and tokenize the entire caption sentence. “<start>” and “<end>” tokens are defined for this purpose. For the decoding algorithm where the language model is used, the <start> token acts as an instruction for the decoder to start decoding, as it needs a

[1] https://cloud.google.com/translate/

very first state to predict the next first token. The <end> token is an instruction to stop generating more tokens.

Once the <start> and <end> tokens are added, the inbuilt tokenizer is used, which is provided by Torchtext[2], to tokenize the captions. The mapping function is created for converting each token to its quantified version, and by using the same, the token sequences are converted to vectors. Any token which is not present in the mapping function would be labeled as "<UNK>" (unknown token). Furthermore, padding is applied to the text vectors generated, which pads the vectors to the size of the largest vector in the batch [28]. Padding helps set all the input sequences to equal lengths so they can be passed toa language model.

C. Proposed System

The entire pipeline for the project has been built using PyTorch and its modules [29]. The below content describes the pipeline after the caption data has been preprocessed. The entire dataset is split into training and validation datasets, with a validation split of 0.1. A random seed is considered to get consistent results on the systems [30].

Once the text captions in a particular language are tokenized and vectorized, the images of varying dimensions are resized to dimensions 299×299 and are then normalized to get the transformed images [31]. Next, these images are subjected to feature extraction. These features would be vectors that contain information about the essential parts that define the image. Pre-trained Convolutional Neural Network (CNN) models are used [32]. Since modern CNN models have millions of parameters that define them, training them from scratch takes time and effort. Using pre-trained models trained on a considerable amount of data for a related task resolves this problem very easily. In feature extraction, the final layer weights of the pre-trained model are updated, from which predictions are derived. The three pre-trained models used in this research are InceptionV3 [26], ResNet50, and VGG16 [26, 33, 34]. This part of the system is the encoderarchitecture.

Further, the Bahdanau attention model is defined for the system, which is a three-layered model [35]. This model is then incorporated into the decoder part of the system. The encoder layer of the attention model has a size equal to the length of the feature vectors from the images. With each hidden state generated by the language model, the image feature and the hidden state are passed to the attention model. The attention mechanism focuses on the relevant part of the image by using the hidden state as the context. The output of the attention model, which is the representation of the image filtered such that only relevant parts of the image remain, is used as an input to the language model. The language model predicts a new word and returns the next hidden state. This process continues until the <end> token is encountered, and the caption generation process is completed. Simple RNN, LSTM, and GRU are used as the different language modelsfor the system [36–38].

The system is hence called a merge architecture, as it combines both the encoded form of the input image with the encoded form of the text description generated so far. Each of the three pre-trained CNN models is combined with the three language models, thus resulting in nine different merge-architectures for each language. Since the experiment is carried out in Marathi, Kannada, Malayalam, and Tamil languages, there are 36 models trained in total for this project. Fig. 1 shows the proposed system architecture.

D. Pre-trained CNN Models

1) VGG16: VGG stands for Visual Geometry Group, a group of Oxford scholars that created this architecture. Simoyan et al. suggested it in 2014 [34]. It is one of the most used CNN architectures for computer vision applications. It is pre-trained on a subset of the ImageNet dataset, a collection of over 14 million images belonging to 22,000 categories [39]. It contains 16 convolutional layers and is more complicated than other typical CNN models. However, it has a highly consistent architecture. VGG16 includes a large number of parameters—approximately 138M—which contributes to the model's high complexity. The VGG16 design starts with two convolutional layers, then a max-pooling layer, which is then again followed by two convolutional layers and another max-pooling layer [40]. Three consecutive dense layers then follow the architecture. The VGG16 model's last layer has an output dimension of 512, from which input image features may be extracted and used further.

2) InceptionV3: The InceptionV3 model is part of the Inception family of architectures, frequently employed for image data-related applications [41]. The InceptionV3 architecture was first introduced in 2015. It is the third edition of the Inception CNN model by Google [42], initially instigated during the ImageNet Recognition Challenge. The InceptionV3 model was significantly modified compared to previous models, including generous dimension reduction accompanied by factorization into smaller convolutions, spatial factorization into asymmetric convolutions, the utility of auxiliary classifiers, and efficient grid size reduction. The InceptionV3 model is pre-trained on nearly a million images from the ImageNet dataset. The InceptionV3 model has 42 layers, somewhat more than the preceding Inception V1 and V2 models. In terms of efficiency, the model is outstanding. The image features could be extracted from the linear layer at the end of the architecture, which has a dimension of 2048.

ResNet50: ResNet models are the foundation for many computer vision applications and get trained via deep residual learning. ResNet50 is a 50-layer CNN that was trained on the ImageNet dataset. ResNet was a game changer because it allowed researchers to train intense neural networks with 150+ layers. The ResNet50 model was introduced in 2015 [33]. ResNet architectures aid in the resolution of the vanishing gradient problem in classic CNNs, in which gradient values are scarcely

[2] https://pytorch.org/text/stable/index.html

modified during backpropagation in training. ResNet employs a "skip connection" strategy, in which the original input is appended to the output of the convolutional block without requiring gradient descent. A skip connection is a direct connection that skips across some model levels. The diagram above illustrates the skip mechanism. Without the skip mechanism, input "X" would be multiplied by the layer weights, followed by a bias term. A ResNet block is diagrammatically represented in Fig. 2.

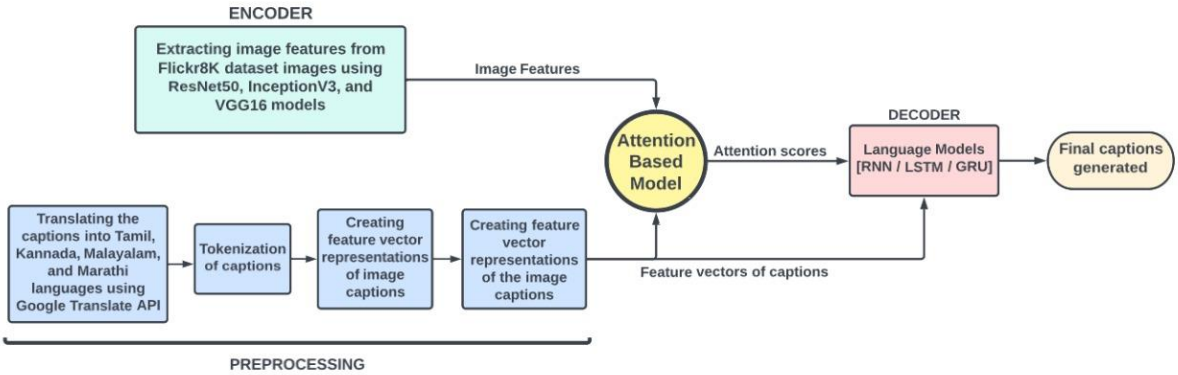

Figure 1. Proposed system architecture.

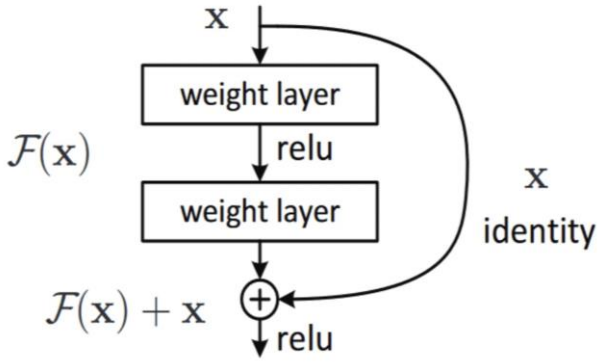

Figure 2. Residual block of ResNet.

E. Language Models

1) Simple RNN: Recurrent Neural Networks (RNN) are basic architectures for working with sequential data such as text, audio files, time-series data, and so on [43]. RNN generates the following output by utilizing prior information in the sequence. The RNN begins with one piece of input data and predicts the next bit of data in the sequence.

On the other hand, the basic design of vanilla RNN suffers from vanishing gradient—as the RNN processes more steps, it suffers from vanishing gradient more than different neural network architectures. RNN thus finds it challenging to learn to store information across several timesteps. As a result, the hidden state in vanilla RNN is continually rebuilt.

2) GRU and LSTM: To overcome the problem of vanishing gradient problem, a gated version of RNN was introduced, called Gated Recurrent Unit (GRU). GRU uses a memory cell to store the activation value of previous words in the long sequence. The gates in GRU control the flow of information in the network. Gates can learn which inputs in the sequence are essential and store their information in the memory unit. They can pass the information in long sequences and use them to make predictions. Gates are neural networks, where each gate has its weights and biases. The components of GRU include an update gate, a reset gate, a candidate cell, and a final cell state.

LSTM stands for Long-Short Term Memory unit. LSTM is very similar to GRU and is also intended to solve the vanishing gradient problem in vanilla RNN. In LSTM, there are two more gates besides GRU- forget gate and output gate.

IV. Experimetnal Detials

The models were trained by building them using the PyTorch library. They were trained on the NVIDIA Tesla K80 GPU (graphics processing unit), which has a 13GB RAM capacity.

In the case of the encoder, the encoded features are extracted from the pre-final layer of the pre-trained CNN model. Its output vector size is 2048 in the case of InceptionV3 and ResNet50, whereas it is 512 in the case of VGG16. For the decoder part of the merge architecture model, we set the size of the embedding layer to 300. This embedding layer is defined before the attention model in the decoder. The input vector size of the attention layer was set to 128, whereas the size of the decoder's language model (either a simple RNN, LSTM, or GRU) was set to 512. The output vector obtained from the decoder was size 256, converted into the corresponding textualcaption.

The models are trained on 12 total epochs, with about 3.5 h of training for each model. The learning rate was set to 0.0003 [44]. The cross-entropy loss function is used to determine the loss obtained at each epoch during training [45]. The objective function was to minimize the cross-entropy loss for an image-caption pair during training. The Adam optimizer was used for enhancing training performance, with corresponding beta coefficients set to 0.9 and 0.98 [46].

During training, with each input image-caption pair passed, the gradients were set to zero, and the loss

was calculated to update the gradient weights. Next, backpropagation was applied to update the weights of the decoder language model and apply the optimizer at each step for better convergence [47].

It was found that the above-mentioned custom architecture dimensions and model hyperparameters gave the best possible results; hence the corresponding results are presented in this paper. The BLEU scores obtained after the testing of the models are presented further.

V. Results

The generated captions' accuracy is assessed using the widely used BLEU metric [48]. BLEU stands for Bilingual Evaluation Understudy. The BLEU technique is used to assess the quality of language generation. The BLEU metric is simple to comprehend and calculate and is independent of the source and target languages. The higher the BLEU, the better are the translations. BLEU scores are examined between 0 and 1, with a value around 0.7 being a nearly perfect score.

Based on the value of "n" selected while considering n-grams for the BLEU score, the BLEU scores were determined as BLEU-1, BLEU-2, BLEU-3, and BLEU-4. An n-gram is a group of words appearing in a specific window, where 'n' denotes the window size. To determine the number of matches, BLEU compares the n-grams of the candidate and the reference translations. These matches don't depend on the positions in which they take place. Depending upon the length of the text to be evaluated, the BLEU variation can be chosen. As we are dealing with shorter one-line texts in this case, BLEU-1 metric would be preferred. Other BLEU variations are evaluated as well for the reader to get an understanding.

Table I presents the detailed results of this research. From the table, it could be inferred that the InceptionV3 model gives the best results out of all the three models used for image feature extraction in the encoder. For Kannada, the proposed systems give slightly low scores compared to the other languages. This could be probably due to lower morpho-logical richness in Kannada than in other languages, or less accurate alignment of the image and the textual features during decoding. Considering the language model's perspective, the BLEU scores depend upon the feature extraction model taken and the language under consideration. For each language, the highest BLEU-1 scores achieved are 0.4939 for Marathi, 0.4557 for Kannada, 0.5082 for Malayalam, and 0.5201 for Tamil. In many of the systems, GRU tends to overperform, while in many others, LSTM and RNN perform better as well.

TABLE I. Results Obtained on the Test set for the 36 Distinct Models We Trained. These 36 Models Are Designed by Permuting between the Pre-trained CNN Models for Encoding the Images (InceptionV3, ResNet50, VGG16) and Language Models Used as the Decoder (RNN, GRU, LSTM). The Table Contains the Bleu Scores Calculated for Different Numbers of N-Grams, Ranging from 1 to 4

Language	BLEU metric	InceptionV3			ResNet50			VGG16		
		RNN	GRU	LSTM	RNN	GRU	LSTM	RNN	GRU	LSTM
Marathi	BLEU-1	0.493454	0.489715	**0.493954**	0.485828	0.479079	0.460888	0.455181	0.454941	0.451353
	BLEU-2	0.303109	0.300092	**0.303258**	0.295132	0.292759	0.276512	0.267041	0.26906	0.269321
	BLEU-3	0.224492	0.224068	**0.226731**	0.215801	0.216506	0.20194	0.19258	0.196008	0.195112
	BLEU-4	0.119534	0.118297	**0.122775**	0.112342	0.111992	0.104357	0.095123	0.10091	0.099294
Kannada	BLEU-1	**0.455757**	0.446693	0.421288	0.41574	0.442111	0.421155	0.413686	0.427966	0.39391
	BLEU-2	**0.245872**	0.237989	0.22459	0.219931	0.234022	0.218452	0.245587	0.220381	0.197347
	BLEU-3	**0.165322**	0.157846	0.147863	0.145532	0.156406	0.142489	0.129734	0.143746	0.125423
	BLEU-4	**0.072618**	0.065687	0.058655	0.058664	0.066156	0.053827	0.05677	0.05949	0.046696
Malayalam	BLEU-1	**0.508195**	0.5019	0.491015	0.500999	0.497043	0.489659	0.468711	0.484288	0.47791
	BLEU-2	**0.323874**	0.318976	0.311843	0.319866	0.310551	0.307649	0.289639	0.302339	0.298257
	BLEU-3	**0.228079**	0.220121	0.215865	0.226553	0.210826	0.211061	0.194421	0.20465	0.200624
	BLEU-4	**0.114529**	0.109077	0.10578	0.116873	0.099497	0.102869	0.092689	0.098265	0.096308
Tamil	BLEU-1	0.511552	**0.520094**	0.500551	0.489099	0.483312	0.501862	0.480982	0.497276	0.4962
	BLEU-2	0.353912	**0.35799**	0.341703	0.329613	0.328573	0.342492	0.293452	0.334753	0.333903
	BLEU-3	0.266309	**0.268588**	0.252919	0.237803	0.241367	0.249035	0.231097	0.249078	0.23694
	BLEU-4	0.145903	**0.148346**	0.137636	0.125393	0.127485	0.131501	0.124461	0.135299	0.120517

Note: Scores for best results obtained for each language are marked in bold.

VI. Comparison with Previous Architectures

To measure the effectiveness of the proposed model, the obtained results were compared with the results published in the paper [49], which implements an architecture that is like the proposed architecture, but without an attention mechanism present. The paper implements image captioning in the Tamil language on the Flickr30K dataset, with a CNN model for feature extraction and an LSTM as the language model. Even though that architecture was trained on a larger dataset, our attention-based approach gives better results. We take our VGG16+LSTM based merge architecture for this comparison. Table II illustrates this comparison.

TABLE II. Statistics of the Dataset Used for Supervised Training

	CNN+LSTM	VGG16+LSTM with attention (Our architecture)
BLEU-1	0.370611	**0.4962**
BLEU-2	0.217844	**0.333903**
BLEU-3	0.160439	**0.23694**
BLEU-4	0.077670	**0.120517**

VII. Sample Input and Captions Generated

We have depicted the captions generated for a few models, for the Fig. 3. Table III illustrates the captions generated for the respective models.

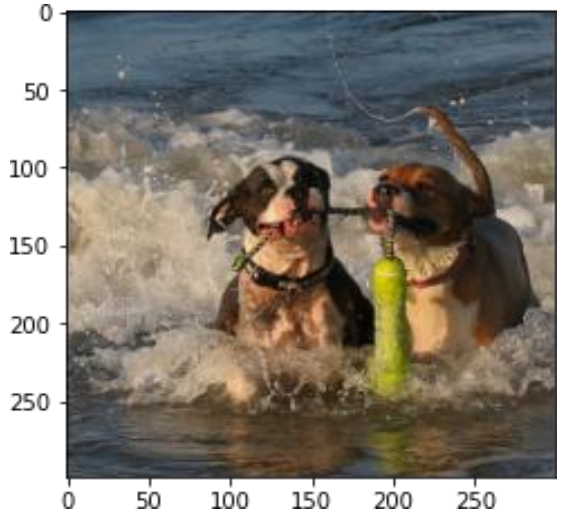

Figure 3. Flickr8K image for English caption: "Two dogs are playing in the water."

TABLE III. Captions Obtained Using InceptionV3 + RNN, VGG16 + RNN, InceptionV3 + LSTM, ResNet50 + RNN, and ResNet50 + LSTM Models

Model	Language	Generated caption
Inceptionv3 + RNN	English	Two dogs are playing in the water.
	Marathi	दोन कुत्रे पाण्यात खेळत आहेत
	Kannada	ಎರಡು ನಾಯಿಗಳು ನೀರಿನಲ್ಲಿ ಆಟವಾಡುತ್ತಿವೆ
	Malayalam	രണ്ട് നായ്ക്കൾ വെള്ളത്തിൽ കളിക്കുന്നു
	Tamil	இரண்டு நாய்கள் தண்ணீரில் விளையாடுகின்றன
VGG16 + RNN	English	Two dogs are playing in the water.
	Marathi	दोन कुत्री काठी काढत पाण्यात आहेत
	Kannada	ಎರಡು ನಾಯಿಗಳು ನೀರಿನಲ್ಲಿ ಕೋಲನ್ನು ಪಡೆಯುತ್ತಿವೆ
	Malayalam	രണ്ട് നായ്ക്കൾ ഒരു വടി വീണ്ടെടുക്കാൻ വെള്ളത്തിലാണ്
	Tamil	இரண்டு நாய்கள் ஒரு குச்சியை மீட்டெடுக்கும் தண்ணீரில் உள்ளன
Inceptionv3 + LSTM	English	Two dogs are playing in the water.
	Marathi	सर्फमधील दोन कुत्रे एकाच काठीला धरून आहेत
	Kannada	ಸರ್ಫ್‌ನಲ್ಲಿ ಎರಡು ನಾಯಿಗಳು ಒಂದೇ ಕೋಲನ್ನು ಹಿಡಿದಿವೆ
	Malayalam	സർഫിൽ ഒരേ വടിയിൽ പിടിച്ചിരിക്കുന്ന രണ്ട് നായ്ക്കൾ
	Tamil	அலைச்சலில் இரண்டு நாய்கள் ஒரே குச்சியைப் பிடித்துக் கொண்டிருக்கின்றன
ResNet50 + RNN	English	Two dogs are playing in the water.
	Marathi	पाण्यात दोन कुत्रे काठीवर भांडत आहेत
	Kannada	ನೀರಿನಲ್ಲಿ ಎರಡು ನಾಯಿಗಳು ಕೋಲಿನಿಂದ ಜಗಳವಾಡುತ್ತಿವೆ
	Malayalam	വെള്ളത്തിലിരിക്കുന്ന രണ്ട് നായ്ക്കൾ ഒരു വടിയുമായി പൊരുതുന്നു
	Tamil	தண்ணீரில் இரண்டு நாய்கள் ஒரு குச்சியால் சண்டையிடுகின்றன
ResNet50 + LSTM	English	Two dogs are playing in the water.
	Marathi	दोन कुत्री काठीने पोहत आहेत
	Kannada	ಕೋಲಿನಿಂದ ಈಜುತ್ತಿರುವ ಎರಡು ನಾಯಿಗಳು
	Malayalam	രണ്ട് നായ്ക്കൾ വടിയുമായി നീന്തുന്നു
	Tamil	இரண்டு நாய்கள் குச்சியுடன் நீந்துகின்றன

VIII. Conclusion and Future Work

Thus, the performance of various attention-based merge architecture models is evaluated for image captioning in Indian languages. It can be inferred that the InceptionV3 model gives the best results compared to ResNet50 and VGG16. The combination of InceptionV3 and RNN gives the best results for Kannada and Malayalam languages, whereas the combination of InceptionV3 and LSTM gives optimum results for Marathi and Tamil languages. We believe that we have set an initiation to image captioning research in low-resource Indian languages. With more state-of-the-art CNN pre-trained models underway, we plan to experiment with them in our merge architectures. We plan to leverage our architectures by further finetuning them with augmented datasets such as Flickr30K and using computationally powerful GPUs for effective learning. We also plan to apply better preprocessing and tokenization techniques to our caption data for better results.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

The idea of this paper was formulated by Rahul Tangsali and Swapnil Chhatre. Rahul Tangsali and Soham Naik worked on the collecting the data and translating the captions from English to Indian languages using Google Translate API. Swapnil Chhatre and Pranav Bhagwat tested various feature extraction models for preprocessing of images. Rahul Tangsali successfully implemented the merge-architecture model. Vari- ous combinations of the model were tested for all four Indian languages - Marathi, Malayalam, Kannada, Tamil - by Swapnil Chhatre, Rahul Tangsali, and Soham Naik. Rahul Tangsali evaluated the models using BLEU performance metric. Swap- nil Chhatre and Rahul Tangsali wrote the first draft of the paper. The draft was reviewed by Geetanjali Kale and some modifications were suggested by her. All authors had approved the final version

References

[1] B. Makav and V. Kilic, "A new image captioning approach for visually impaired people," in *Proc. 2019 11th International Conference on Electrical and Electronics Engineering (ELECO)*, 2019, pp. 945–949.

[2] A. Kojima, T. Tamura, and K. Fukunaga, "Natural language description of human activities from video images based on concept hierarchy of actions," *International Journal of Computer Vision*, vol. 50, pp. 171–184, 2002.

[1] P. Héde, P.-A. Moëllic, J. Bourgeoys, M. Joint, and C. Thomas, "Automatic generation of natural language description for images." in *Proc. RIAO'04: Coupling Approaches, Coupling Media and Coupling Languages for Information Retrieval*, 2004, pp. 306–313.

[2] A. Farhadi, M. Hejrati, M. A. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier, and D. Forsyth, "Every picture tells a story: Generating sentences from images," in *Computer Vision—ECCV 2010*, K. Daniilidis, P. Maragos, and N. Paragios, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 15–29.

[3] A. Gupta, Y. Verma, and C. Jawahar, "Choosing linguistics over vision to describe images," in *Proc. of the AAAI Conference on Artificial Intelligence*, vol. 26, no. 1, 2012, pp. 606–612.

[4] R. Mason and E. Charniak, "Nonparametric method for data-driven image captioning," in *Proc. of the 52nd Annual Meeting of* the *Association for Computational Linguistics (Volume 2: Short Papers)*, 2014, pp. 592–598.

[5] P. Kuznetsova, V. Ordonez, T. L. Berg, and Y. Choi, "TreeTalk: Composition and compression of trees for image descriptions," *Transactions of the Association for Computational Linguistics*, vol. 2, pp. 351–362, 2014.

[6] Y. Yang, C. Teo, H. Daume III, and Y. Aloimonos, "Corpus-guided sentence generation of natural images," in *Proc. of the 2011 Conference on Empirical Methods in Natural Language Processing*. Edinburgh, Scotland, UK.: Association for Computational Linguistics, Jul. 2011, pp. 444–454.

[7] S. Li, G. Kulkarni, T. Berg, A. Berg, and Y. Choi, "Composing simple image descriptions using web-scale n-grams," in *Proc. of the Fifteenth Conference on Computational Natural Language Learning*, 2011, pp. 220–228.

[8] Y. Ushiku, M. Yamaguchi, Y. Mukuta, and T. Harada, "Common subspace for model and similarity: Phrase learning for caption generation from images," in *Proc. 2015 IEEE International* Conference *on Computer Vision (ICCV)*, 2015, pp. 2668–2676.

[9] Y. Gong, L. Wang, M. Hodosh, and J. Hockenmaier, "Improving image- sentence embeddings using large weakly annotated photo collections," in *Computer Vision—ECCV 2014*, *Lecture Notes in Computer Science*, vol 8692, D. Fleet, T. Pajdla, B. Schiele, T. Tuytelaars, Eds. Springer, Cham., 2014. https://doi.org/10.1007/978-3-319-10593-2_35

[10] Y. Bengio. (2013). Deep learning of representations: Looking forward. [Online]. Available: https://arxiv.org/abs/1305.0445

[11] A. Karpathy, A. Joulin, and L. F. Fei-Fei, "Deep fragment embeddings for bidirectional image sentence mapping," in *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, Eds., vol. 27. Curran Associates, Inc., 2014.

[12] R. Lebret, P. Pinheiro, and R. Collobert, "Phrase-based image captioning," in *Proc. International Conference on Machine Learning*, *PMLR*, 2015, pp. 2085–2094.

[13] R. Socher, A. Karpathy, Q. V. Le, C. D. Manning, and A. Y. Ng, "Grounded compositional semantics for finding and describing images with sentences," *Transactions of the Association for Computational Linguistics*, vol. 2, pp. 207–218, 2014. https://doi.org/10.1162/tacl a 00177

[14] R. Kiros, R. Salakhutdinov, and R. Zemel, "Multimodal neural language models," in *Proc. International Conference on Machine Learning*, *PMLR*, 2014, pp. 595–603.

[15] J. Mao, W. Xu, Y. Yang, J. Wang, and A. L. Yuille, "Explain images with multimodal recurrent neural networks," arXiv:1410.1090 [cs.CV], 2014.

[16] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3156–3164.

[17] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2625–2634.

[18] Y. Pu, Z. Gan, R. Henao, X. Yuan, C. Li, A. Stevens, and L. Carin, "Variational autoencoder for deep learning of images, labels and captions," *Advances in Neural Information Processing Systems*, vol. 29, 2016.

[19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[20] K. Aitken, V. V. Ramasesh, Y. Cao, and N. Maheswaranathan. (2021). Understanding how encoder-decoder architectures attend. [Online]. Available: https://arxiv.org/abs/2110.15253

[21] V. H. V. Kumar and N. Lalithamani, "English to tamil multi-modal image captioning translation," in *Proc. 2022 IEEE World Conference on Applied Intelligence and Computing (AIC)*, 2022, pp. 332–338.

[22] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár. (2014). Microsoft coco: Common objects in context. [Online]. Available: http://arxiv.org/abs/1405.0312

[23] J. M. R. Dwarampudi, D. Rampavan, M. A. S. Sathwik, K. N. Reddy, V. K. Mishra, D. Singh, and A. Agrawal, "Image caption generation in telugu," in *Proc. 2021 7th International Conference on Advanced Computing and* Communication *Systems (ICACCS)*, vol. 1, 2021, pp. 438–443.

[24] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2818–2826.

[25] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. (2016). Enriching word vectors with subword information. [Online]. Available: https://arxiv.org/abs/1607.04606

[26] M. Dwarampudi and N. V. S. Reddy. (2019). Effects of padding on lstms and cnns. [Online]. Available: https://arxiv.org/abs/1903.07288

[27] A. Paszke, S. Gross, F. Massa *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'AlchéBuc, E. Fox, and R. Garnett, Eds. vol. 32, Curran Associates, Inc., 2019.

[28] P. S. Madhyastha and D. Batra, "On model stability as a function of random seed," arXiv:1909.10447 [cs.LG], 2019.

[29] M. S. Yasein and P. Agathoklis, "An image normalization technique based on geometric properties of image feature points," in *Proc. 2007* IEEE *International Symposium on Signal Processing and Information Technology*, 2007, pp. 116–121.

[30] Abdullah and M. S. Hasan, "An application of pre-trained CNN for image classification," in *Proc. 2017 20th International Conference of Computer and Information Technology (ICCIT)*, 2017, pp. 1–6.

[31] K. He, X. Zhang, S. Ren, and J. Sun. (2015). Deep residual learning for image recognition. [Online]. Available: https://arxiv.org/abs/1512.03385

[32] K. Simonyan and A. Zisserman. (2014). Very deep convolutional networks for large-scale image recognition. [Online]. Available: https://arxiv.org/abs/1409.1556

[33] D. Bahdanau, K. Cho, and Y. Bengio. (2014). Neural machine translation by jointly learning to align and translate. [Online]. Available: https://arxiv.org/abs/1409.0473

[34] A. Sherstinsky, "Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network," *Physica D: Nonlinear Phenomena*, vol. 404, 132306, Mar 2020.

[35] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, pp. 1735–1780,1997.

[36] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. [Online]. Available: https://arxiv.org/abs/1412.3555

[37] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proc. 2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[38] J. Nagi, F. Ducatelle, G. A. Di Caro, D. Cires¸an, U. Meier, A. Giusti, F. Nagi, J. Schmidhuber, and L. M. Gambardella, "Max-pooling convo- lutional neural networks for vision-based hand gesture recognition," in *Proc. 2011 IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*, 2011, pp. 342–347.

[39] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov,

[40] D. Erhan, V. Vanhoucke, and A. Rabinovich. (2014). Going deeper with convolutions. [Online]. Available: https://arxiv.org/abs/1409.4842

[41] S. Brin and L. Page. (1998). The anatomy of a large-scale hypertextual web search engine. *Computer Networks*. [Online]. 30. pp. 107–117. Available: http://www-db.stanford.edu/~backrub/google.html

[42] H. Hewamalage, C. Bergmeir, and K. Bandara. (2019). Recurrent neural networks for time series forecasting: Current status and future directions. [Online]. Available: http://arxiv.org/abs/1909.00590

[43] C. Igiri, A. Uzoma, and A. Silas, "Effect of learning rate on artificial neural network in machine learning," *International Journal of Engineering Research*, vol. 4, no. 6, 2021.

[44] Z. Zhang and M. Sabuncu, "Generalized cross entropy loss for training deep neural networks with noisy labels," in *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. vol. 31, Curran Associates, Inc., 2018.

[45] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. International Conference on Learning Representations*, 2014, vol. 12.

[46] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, 1986.

[47] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: A method for automatic evaluation of machine translation," in *Proc. of the 40th Annual Meeting of the Association for Computational Linguistics*. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, Jul. 2002, pp. 311–318.

[48] Image captioning in tamil language with merge architecture. [Online]. Available: http://ir.kdu.ac.lk/handle/345/5209