

Project Report: ECS 271 001 FQ 2023

Enabling Visual Accessibility: Leveraging Deep Learning Methodologies for Image Captioning

Team members: Shantanu Milind Joshi (Student ID: 922815849), Tejas Shrikant Patil (Student ID: 922828661), Amritanand Sudheerkumar (Student ID: 922796395)

1. Problem Statement:

The widespread inclusion of graphical content in the digital world is a significant barrier for the visually impaired. This project addresses the accessibility challenges faced by individuals with visual impairments when interacting with web content containing images. Specifically, the focus is on generating descriptive captions for images, making the visual information accessible through natural language understanding.

2. Motivation:

Accessibility is an important factor in an inclusive and equal society. The digital world, with its vast repository of information and opportunities, should be accessible to all, regardless of physical limitations. For individuals with visual impairments, the ability to interpret visual information is crucial for their personal, social, and professional development. As per Cornell Disability Statistics[1], more than one million individuals of all ages in the United States reported Visual disability. By solving this problem, we can enhance the lives of millions of individuals, providing them with equal access to the vast amount of visual content available on the web. Additionally, we can promote independence and enrich virtual experiences for these individuals.

3. Dataset:

- The Flickr8k dataset is used to train the models. It is a collection of 8,091 images from the image hosting website Flickr, and their corresponding captions.
- Each image from Flickr had been annotated with 5 captions each by human judges. Thus, making a total of 40,455 captions.
- The images in the dataset cover a wide range of topics, including people, animals, objects, and scenes.

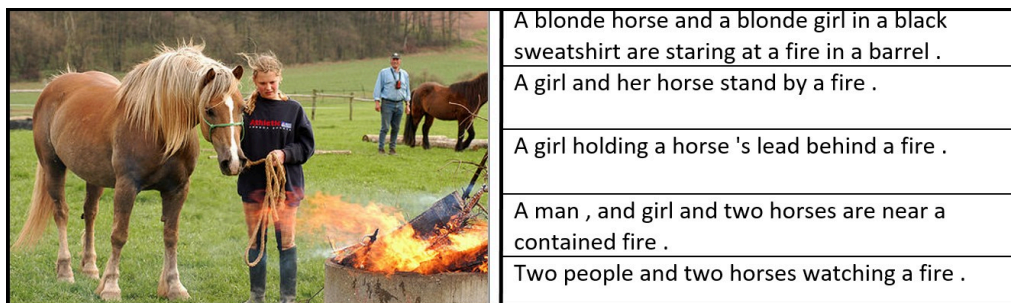


Fig. 1. Sample Image and corresponding captions

4. Methodology and Architecture:

The proposed solution leverages a combination of deep-learning algorithms to generate captions. Traditional approaches[2] follow Inject Architecture, where the Encoder layer extracts features and the

Decoder layer generates words in the caption. Taking inspiration from these traditional Encoder-Decoder models, we implemented an advanced version known as Merge Model Architecture[3].

4. a. Merge Architecture:

The Merge architecture[3] combines the features extracted from the Images as well as the Text. The architecture can be seen as a combination of 2 Encoder layers and 1 Decoder layer:

4.a. i. Encoder 1 - CNN (Convolutional Neural Network):

Convolutional Neural Networks are used to extract relevant features from the images and act as Image Encoder. We used 5 pre-trained CNN models namely: VGG16, VGG19, Resnet50, InceptionV3, and Xception. This helped us to perform a comparative analysis and obtain a model with the best accuracy.

4.a. ii. Encoder 2 - LSTM (Long Short-Term Memory):

LSTMs have become popular for sentence prediction tasks[4]. But in our case, we are using them just to generate the encoding of the caption sequence. Hence, LSTM in our architecture carries purely linguistic information. Additionally, we apply a combination of **text preprocessing** techniques like stop word removal, special character removal, tokenization, embedding, and numeric value removal to the captions.

4.a. iii. Decoder:

The Decoder layer consists of a Dense and Softmax. It is used to generate the next word in the sequence. The final output caption is generated by this layer.

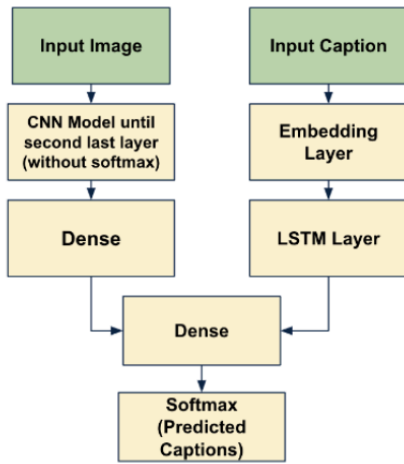


Fig. 2. Merge Architecture

Layer (type)	Output Shape	Param #	Connected to
input_3 (InputLayer)	[(None, 35)]	0	[]
input_2 (InputLayer)	[(None, 4096)]	0	[]
embedding (Embedding)	(None, 35, 256)	2172160	['input_3[0][0]']
dropout (Dropout)	(None, 4096)	0	['input_2[0][0]']
dropout_1 (Dropout)	(None, 35, 256)	0	['embedding[0][0]']
dense (Dense)	(None, 256)	1048832	['dropout[0][0]']
lstm (LSTM)	(None, 256)	525312	['dropout_1[0][0]']
add (Add)	(None, 256)	0	['dense[0][0]', 'lstm[0][0]']
dense_1 (Dense)	(None, 256)	65792	['add[0][0]']
dense_2 (Dense)	(None, 8485)	2180645	['dense_1[0][0]']
Total params: 5992741 (22.86 MB)			
Trainable params: 5992741 (22.86 MB)			
Non-trainable params: 0 (0.00 Byte)			

Fig. 3. Merge Architecture with VGG16 Image Encoder

4. b. Baseline Model:

As our baseline, we are considering VGG16 as Encoder 1 for Image feature extraction. The architecture of the exact baseline models is shown in Fig. 3.

5. Experimental Results:

We conducted the experiments using the five pre-trained models: VGG16, VGG19, InceptionV3, ResNet50, and Xception. For each of these models, the training loss over 20 epochs on GPU, with a batch size of 32, the learning rate of 0.001, and Adam as optimizer was tracked.

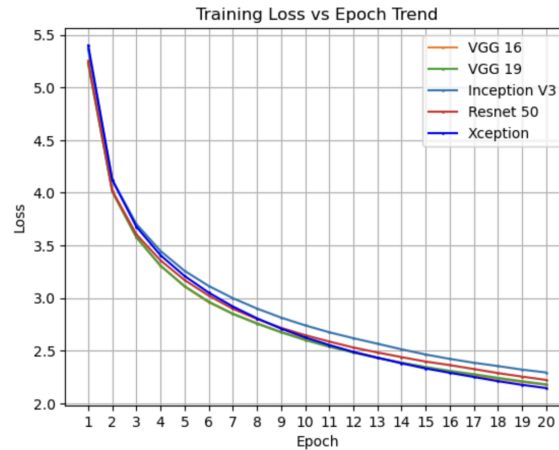


Fig 4. Loss vs Epoch Trend

Initially, the loss values for every model showed a decreasing trend indicating that the model learned from the training data. As the epochs progressed, the rate of decrease diminished, suggesting that the model started moving toward convergence. Despite all the models achieving very low loss values, we found Xception to be the best-performing model relatively, indicating its superior feature-extracting capabilities.

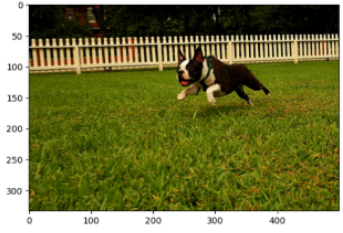
IMAGE	REFERENCE CAPTIONS	GENERATED CAPTIONS
	<ul style="list-style-type: none"> - black and white dog is running in grassy garden surrounded by white fence - black and white dog is running through the grass - boston terrier is running in the grass - boston terrier is running on lush green grass in front of white fence - dog runs on the green grass near wooden fence 	<p>VGG 16: boston terrier is jumping over the fence</p> <p>VGG 19: boston terrier is running through the grass</p> <p>INCEPTION V3: two dogs are playing in the grass</p> <p>RESNET 50: two dogs are playing in yard</p> <p>XCEPTION: black and white dog is running through the grass</p>

Table 1. Reference Captions vs Generated Captions

6. Accuracy Measures:

The Bilingual Evaluation Understudy (BLEU) Score is used for evaluating the performance of the model. The BLEU (Bilingual Evaluation Understudy) Score is a metric that compares the precision of n-grams in the generated text to reference translations to assess the quality of the model-generated text. It ranges between 0 to 1, where the 1 value means the candidate text is more similar to the reference text.

	VGG 16	VGG 19	Inception V3	Resnet 50	Xception
BLEU 1	0.481557	0.489527	0.489213	0.510179	0.512915
BLEU 2	0.284585	0.291600	0.297228	0.308956	0.311626

Table 2. BLEU-1 and BLEU-2 scores for varying Encoder 1 CNN Model

For this project, VGG16 served as the baseline model. All models, including VGG16 demonstrated relatively close BLEU-1 scores. Resnet50 and Xception exhibit slight improvement over the baseline, with Xception scoring the best.

Since Xception performed the best, we tried to further optimize its performance by applying hyperparameter tuning using varying learning rates.

Xception	lr = 0.0007	lr = 0.001	lr = 0.0015	lr = 0.002	lr = 0.005
BLEU 1	0.522568	0.515412	0.492248	0.480784	0.471727
BLEU 2	0.325870	0.321888	0.298757	0.285512	0.278064

Table 3. BLEU-1 and BLEU-2 scores for varying learning rate

The results show the sensitivity of the Xception model to the learning rate, with optimal performance observed at the learning rate = 0.0007. Subsequent increase in the learning rate led to diminishing BLEU scores.

7. Individual Contributions:

Member Name	Tasks and Contributions
Shantanu Milind Joshi	<ol style="list-style-type: none"> 1. Researched and Worked on implementing Merge Architecture 2. Worked on VGG16 Encoder and Xception Encoder 3. Worked on Xception with varied learning rates 4. Worked and Researched on Embedding and LSTM Layer 5. Implemented BLEU Score Analysis 6. Worked on Presentation, Project Report, and Project Proposal 7. Worked on Literature survey and finalizing topic
Tejas Shrikant Patil	<ol style="list-style-type: none"> 1. Researched and Worked on implementing Merge Architecture 2. Worked on VGG19 Encoder and Resnet50 Encoder 3. Implemented Text preprocessing for captions 4. Worked and Researched on LSTM and Embedding Layer 5. Worked on Presentation, Project Report, and Project Proposal 6. Worked on Literature survey and finalizing topic
Amritanand Sudheer Kumar	<ol style="list-style-type: none"> 1. Researched and Worked on the InceptionV3 Encoder Model 2. Refactored code using OOP standards 3. Worked on Presentation, Project Report, and Project Proposal

8. Conclusion:

In conclusion, we successfully developed a model using Merge Architecture to generate captions from the images. We see a significant enhancement in the BLEU-1 Score using Xception: 0.522568 as compared to baseline VGG16: 0.481557. Additionally, the BLEU-2 Score also improved using Xception: 0.325870 in comparison to baseline VGG16: 0.284585. By generating descriptive captions for images using deep learning methodologies, the proposed solution enhances overall web accessibility

9. References:

- [1]"Disability Status Report United States." DisabilityStatistics.org, <https://www.disabilitystatistics.org/>. Accessed: November 9, 2023.
- [2] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and Tell: A Neural Image Caption Generator," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 3156-3164.
- [3] M. Seshadri, M. Srikanth, and M. Belov, "Image to Language Understanding: Captioning Approach", School of Engineering and Applied Science, Columbia University.
- [4] Ghosh, S., Vinyals, O., Strophe, B., Roy, S., Dean, T., & Heck, L. (2016). Contextual lstm (clstm) models for large scale nlp tasks. arXiv preprint arXiv:1602.06291.
- [5] Wikipedia contributors. "BLEU." Wikipedia, The Free Encyclopedia. Wikimedia Foundation, Inc. Available at: <https://en.wikipedia.org/wiki/BLEU>. Accessed [12/6/2023].