Image Captioning

1) Problem Statement

Image Captioning is the task of generating a human-like description for given images. It is easy for a human to glance at the image and describe the image in an appropriate sentence. However, it is a complex task for a computer program to take an image as input and generate a meaningful and syntactically correct sentence. The complex task can be achieved by combining Computer vision with Natural Language Processing (NLP) in Deep Learning to solve this problem.

The real-world scenarios where this can be useful are as follows:

1) **Aiding the Visually Challenged** - The Image Captioning solution can help the visually challenged to see the image through the captions. The product can help then traveling on the roads by first converting the scene to text and then text to voice
2) **Self-Driving Cars** – Automatic driving is the biggest challenge and proper captioning of the scene around the car can be a boost to the self-driving system.
3) **Image Search** – Automatic captioning can help search engines like Google Image search to first convert the image to a caption and a search can be performed based on the caption
4) **Web Development** – It is a good practice to describe any image that is dynamically appearing on the web page that can be read or heard instead of just seen. This makes web content accessible.
5) **Security and Surveillance** – Describing events and objects in security camera footage to aid in identifying security threats or incidents in real-time. This can help in raising alarm as soon as there is any malicious activity going on.

2) Data

The COCO (Common Object and Context) dataset contains 120K images with respective captions. It is a very diverse dataset with multiple captions for an image.

https://cocodataset.org/#download

3) Metrics

The evaluation metric for image captioning is the BLEU score. The BLEU score is a number between zero and one that measures the similarity of the machine-generated captions to a set of reference captions. The score considers the matching n-grams between the reference and the generated captions.

4) Data preprocessing

a) **Image Preprocessing** – The JPEG-encoded images are decoded into a Tensor array. The image is resized to 299 x 299 dimensions for sending into the pre-trained CNN model. Image augmentation like flipping the image, increasing the contrast, and image rotations are randomly applied to increase the capability of predictions.
b) **Text Preprocessing** – The captions for the images are cleaned by removing special characters and unwanted spaces, and converting the uppercase text to lowercase. Adding **<start>** and **<end>** tags to the start and end of each caption. Then the captions are tokenized using text vectorization.

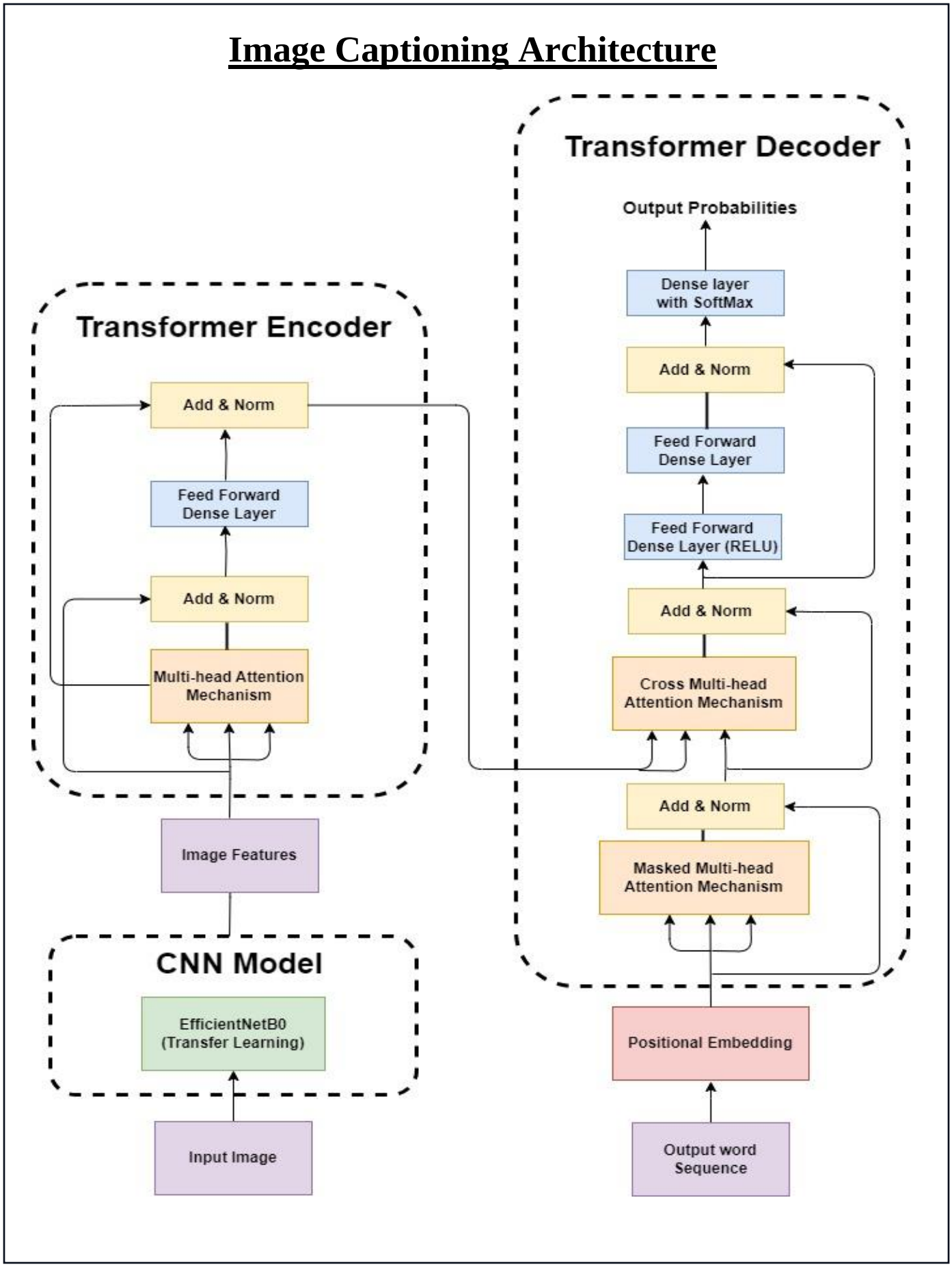
   Finally, the preprocessed image and text are randomly divided into train and validation datasets.

5) Modeling

The architecture of the Image Captioning model involves various deep neural network models like the CNN (Convolutional Neural Network) model for image feature extraction, the Transformer encoder to encode the input images, and the Transformer decoder for decoding the output captions for the image. Eventually, the output from the encoder is sent to the decoder, and output probabilities are generated from the decoder.

Notebook for Image Captioning Model

Image Captioning Architecture



a) CNN Model

The Pre-trained model called EfficientNetB0 is trained on ImageNet for image classification which will be for transfer learning for extracting the image features. Based on this model, a new Convolutional Neural Network (CNN) model will be created for feature extraction. There are many other pre-trained models available for image object recognition. The model will identify the objects in the image and generate their probabilities. The pre-trained model has been used since it saves training time, gives better performance of neural networks, and will not need a lot of data for training. The images are sent to the EfficientNetB0 model and it gives the output as probabilities divided into various categories of individual objects found in the image. For example, if there is an image of a man holding a pen. The model outputs the probabilities of the man and the pen along with other categories. Since for the Image Captioning model, we do not need the probabilities rather extracted features are required, the top SoftMax layer in the CNN model is not taken. The output of the CNN model will contain various features from the image and it will be sent to the Transformer for captioning.

b) Transformer-based Encoder

The image features that were extracted from the CNN model are sent as inputs to an encoder to generate new representations. The inputs first go through a self-attention layer. The layer creates three vectors (query, key, and value vectors), calculated by multiplying the embedding by the matrices weights from the training process. The self-attention layer adds Multi-Head Attention to enable the model to focus on different positions. Attention is given to a particular part of the image. Layer normalization is done to help normalize the outputs to make them compatible with the original inputs (residue connection), which allows the preservation of important information and gradients. Then it goes through a fully connected layer like a Dense layer and again to Layer Normalization. The output is then sent to the Decoder

c) Positional Embedding Layer

Transformers do not have an inherent knowledge of order or position. They would take the input sequence as a Bag of Words, which may be indistinguishable. So before passing the text features as inputs to the decoder, they are converted into token embeddings, and positional information is added to each token. By doing so, the model can effectively decode both the content and the position of tokens in the input sequence, enabling it to capture positional relationships and dependencies in the data. Two embedding layers for token embedding and one for positional embedding are created. The token embedding layer maps the tokens to dense vectors, while the positional embedding layer maps positions within the sequence of dense vectors.

d) Transformer-based Decoder

The decoder is more complex to implement. It generates the output one by one while consulting the representation generated by the encoder. The decoder has a positional embedding layer and a stack of layers.

The positional embedding first goes through masked self-attention. The self-attention works by generating high scores for those words that are related in a sentence, thus capturing the contextual/semantic meaning of the sentence. For example, given the sentence: “I Love Deep Learning”, the words Deep and Learning are closely related, and the vector that corresponds to the word “Learning” will incorporate more context to the word “Deep”. The Casual Masking is done by applying masking on future positions in the self-attention calculation. The causal attention mask ensures that each token can only attend to its previous positions and itself during self-attention, preventing information flow from future positions to past positions. The decoder’s self-attention layer can only attend to earlier positions in the output sequence.

After the self-attention layer, there is a residual connection with the Layer Normalization. Then it is passed to the cross multi-head attention. The output of the top encoder is transformed into a set of attention vectors used in the “encoder-decoder cross self-attention” layer, enabling the decoder to focus on appropriate places in the input sequence. The data then goes through Layer Normalization, a Dense layer with the ‘RELU’ activation to avoid the vanishing Gradient problem, Dense Layer, Layer Normalization, and finally a SoftMax layer to generate probabilities of the words in the vocabulary.

e) Training the Model

The model combines the feature extractor from the CNN model, the encoder, and the decoder to generate the captions for images. When we call the model for training, it will receive the image and caption pairs. The model also calculates the loss and the average accuracy (by comparing the true labels and the predicted labels). While training, it will monitor the model's validation loss to gauge its performance. By defining an Early Stopping callback, the training will stop if the model does not improve its performance for three consecutive epochs (i.e. the model is overfitting).

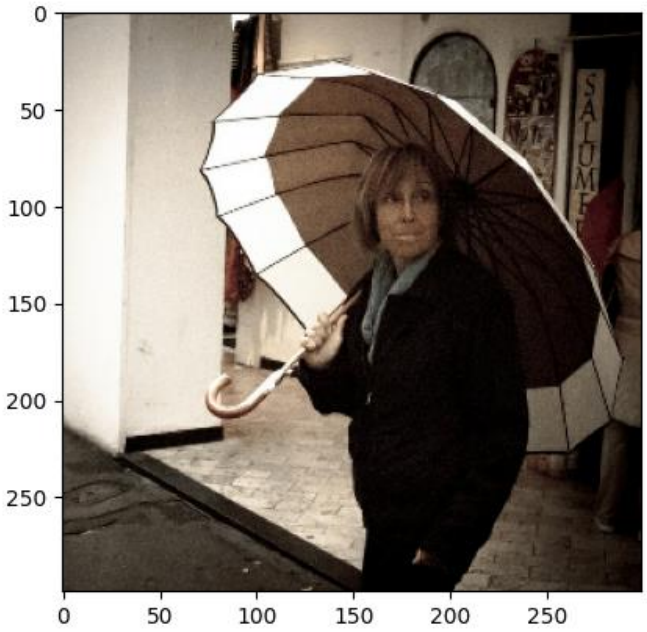
6) Generating Captions

The trained model is finally used to predict the caption for an image. During the prediction process following steps are done –

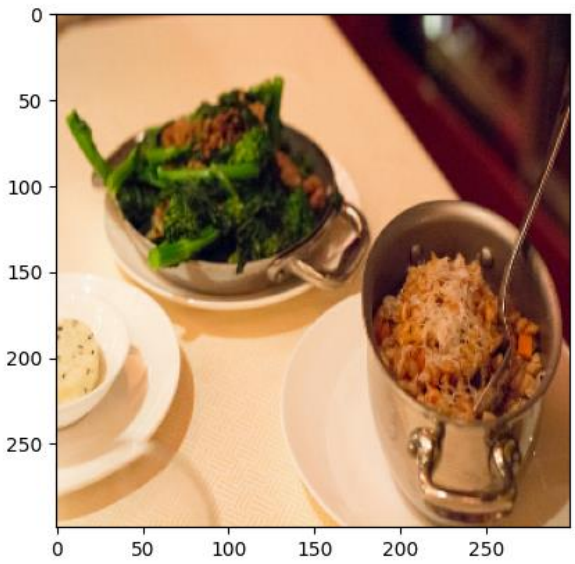
a) Retrieve vocabulary from the training step and map each token position back to their corresponding words in the vocabulary.
b) Select a random image and its image features from the CNN model.
c) Pass the image features to the encoder for encoding.
d) The image features from the encoder and the start of the token <start> which are tokenized using the vectorization layer are sent to the decoder. The output from the decoder is appended with the previous token and it is again sent to the decoder with the image features. The decoder is called iteratively until the maximum sentence length is reached or the end of the token <end> is generated by the decoder
e) Finally, the BLEU score is generated for the predicted captions compared to the reference captions. The score indicates how close the prediction captions are to the reference captions.

Sample predictions

Predicted Caption: a woman holding an umbrella in front of a building
Bleu score: 0.70

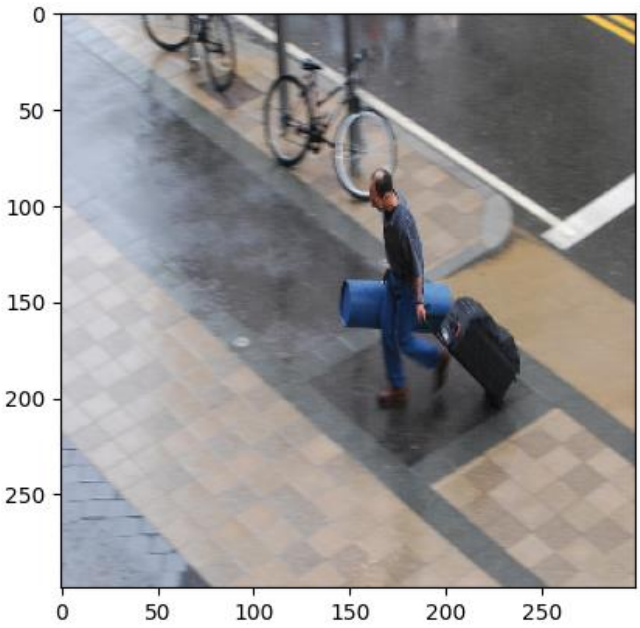


Predicted Caption: a plate of food with a fork on it
Bleu score: 0.20



Predicted Caption: a man is carrying luggage down a street

Bleu score: 0.45



7) Conclusions

The Image Captioning model generated reasonable captions, especially human-readable ones. The caption fits most of the objects in the image and produces a conceptually correct sentence. However, some captions could be improved. The BLEU score metrics show how close the predictions were with the reference captions. BLEU score is tricky to evaluate since an image can have multiple captions and there is no one definite right caption affecting the score generated.

8) Idea for further research

The image caption model can be further improved by increasing the Multi-Head attention which will give more attention to specific parts of the image and text. The diversity of the image and the amount of training data could eventually produce better predictions