Image Caption Generator

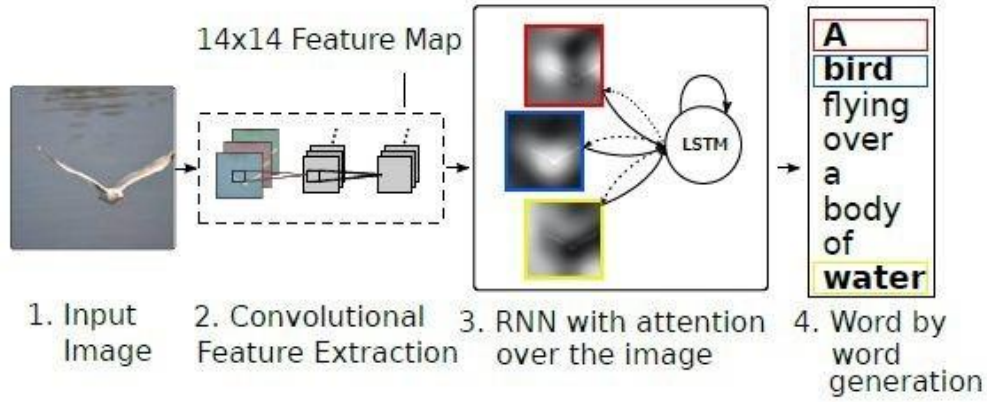
TEAM MEMBERS

MANEESHA Y(1005-21-733-033)

NIHARIKA RODDA(1005-21-733-043)

Problem Statement:

- Implementing model for generating descriptive captions for images, focusing on salient regions and objects in the image.
- This can be done using Convolutional Neural Networks (CNN) and Long short-term memory (LSTM) together



Dataset:

- Flickr8k Dataset:
  - 8091 images
  - 5 captions for each image.

- Caption1: closeup of white dog that is laying its head on its paws
- Caption 2: large white dog lying on the floor
- Caption 3: white dog has its head on the ground
- Caption 4: white dog is resting its head on tiled floor with its eyes open
- Caption 5: white dog rests its head on the patio bricks

FLOWCHART


INPUT IMAGE
FEATURE EXTRACTION(14*14)
CNN
• VGG–16 MODEL
LSTM
NLP
• BLEU SCORE
GENERATED CAPTIONS

Flow of the project

a. Cleaning the caption data

b. Extracting features from images using VGG-16

c. Merging the captions and images

d. Building LSTM model for training

e. Predicting on test data

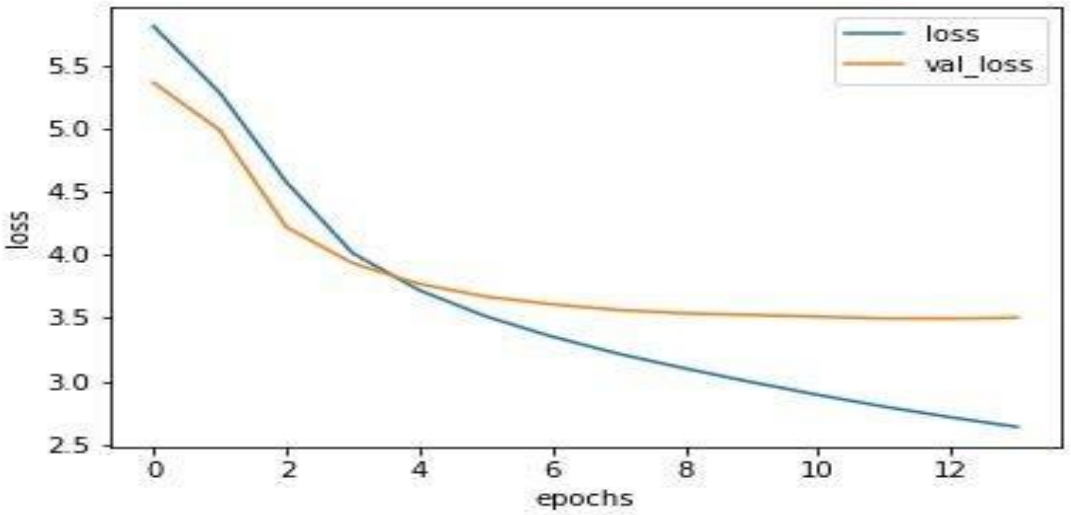
f.Evaluating the captions using BLEU scores as the metric

Steps to follow:

- **1. Cleaning the captions**
- This is the first step of data pre-processing. The captions contain regular expressions, numbers and other stop words which need to be cleaned before they are fed to the model for further training. The cleaning part involves removing punctuations, single character and numerical values. After cleaning we try to figure out the top 50 and least 50 words in our dataset.
- **2. Adding start and end sequence to the captions**
- Start and end sequence need to be added to the captions because the captions vary in length for each image and the model has to understand the start and the end.
- 3. **Extracting features from images**
- After dealing with the captions we then go ahead with processing the images. For this we make use of the pre-trained VGG-16 weights.
- Instead of using this pre-trained model for image classification as it was intended to be used. We just use it for extracting the features from the images. In order to do that we need to get rid of the last output layer from the model. The model then generates **4096** features from taking images of size (224,224,3).
- 4. **Viewing similar images**
- When the VGG-16 model finishes extracting features from all the images from the dataset, similar images from the clusters are displayed together to see if the VGG-16 model has extracted the features correctly and we are able to see them together.

- **5. Merging the caption with the respective images**
- The next step involves merging the captions with the respective images so that they can be used for training. Here we are only taking the first caption of each image from the dataset as it becomes complicated to train with all 5 of them.
- Then we have to tokenize all the captions before feeding it to the model.
- **6. Splitting the data for training and testing**
- The tokenized captions along with the image data are split into training, test and validation sets as required and are then pre-processed as required for the input for the model.
- **7. Building the LSTM model**
- LSTM model is been used beacuse it takes into consideration the state of the previous cell's output and the present cell's input for the current output. This is useful while generating the captions for the images.
  The step involves building the LSTM model with two or three input layers and one output layer where the captions are generated. The model can be trained with various number of nodes and layers. Various hyperparameters are used to tune the model to generate acceptable captions
- 8. **Predicting on the test dataset and evaluating using BLEU scores**
- After the model is trained, it is tested on test dataset to see how it performs on caption generation for just 5 images. If the captions are acceptable then captions are generated for the whole test data.

- **Fit Model:**
- 6 epochs
- 32 batch size
- Training loss is decreasing as we increase the number of epochs.



- **Evaluation:**
- Bleu score in Python is a metric that measures the goodness of Machine Translation models.
- The BLEU score compares a sentence against one or more reference sentences and tells how well does the candidate sentence matched the list of reference sentences. It gives an output score between 0 and 1.
- A BLEU score of 1 means that the candidate sentence perfectly matches one of the reference sentences.

Table: BLUE score

Metric	VGG16
BLUE-1	0.460179
BLUE-2	0.320112
BLUE-3	0.100245
BLUE-4	0.040572

BLEU SCORE

BLEU Score	Interpretation
< 10	Almost useless
10 – 19	Hard to get the gist
20 – 29	The gist is clear, but has significant grammatical errors
30 – 40	Understandable to good translations
40 – 50	High quality translations
50 – 60	Very high quality, adequate, and fluent translations
> 60	Quality often better than human

1. Break Sentences into N-grams

- **N-grams** are consecutive sequences of n words.

For example:

- Reference: "*A dog is running in the park*"
- Predicted: "*A dog runs in the park*"

For **1-grams** (individual words):

- Reference: `["A", "dog", "is", "running", "in", "the", "park"]`
- Predicted: `["A", "dog", "runs", "in", "the", "park"]`

For **2-grams** (pairs of consecutive words):

- Reference: ["A dog", "dog is", "is running", "running in", "in the", "the park"]
- Predicted: ["A dog", "dog runs", "runs in", "in the", "the park"]

- Reference: `["A dog", "dog is", "is running", "running in", "in the", "the park"]`
- Predicted: `["A dog", "dog runs", "runs in", "in the", "the park"]`

---

2. Count Matches

- Compare the n-grams in the predicted sentence with the reference sentence(s) and count how many match.

Using the above example for **1-grams**:

- Matches: `["A", "dog", "in", "the", "park"]` → 5 matches
- Total n-grams in predicted: 6

For **2-grams**:

- Matches: `["A dog", "in the", "the park"]` → 3 matches
- Total n-grams in predicted: 5

---

3. Precision for Each N-gram

- Calculate the precision for each n-gram size:

$$\text{Precision} = \frac{\text{Number of Matching N-grams}}{\text{Total N-grams in Predicted Sentence}}$$

For **1-grams**:

$$\text{Precision} = \frac{5}{6} = 0.833$$

For **2-grams**:

$$\text{Precision} = \frac{3}{5} = 0.6$$

$$\mathrm{BP} = \begin{cases} 1 & \text{if } c > r \\ e^{(1-r/c)} & \text{if } c \leq r \end{cases}.$$

Then,

$$\mathrm{BLEU} = \mathrm{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right)$$

Result

- Original : playful dog is running through the grass
- Predicted: dog runs on the grass
- (BLEU-1: 0.644123

OUTPUT

Bad Caption

true: little girl covered in paint sits in front of painted rainbow with her hands in bowl

pred: group of people are standing in the air

BLEU: 0.2601300475114445

true: man in hat is displaying pictures next to skier in blue hat

pred: man is playing with the snow

BLEU: 0.2795279274196275 6

true: collage of one person climbing cliff

pred: boy in blue shirt is standing on the beach

BLEU: 0

true: couple and an infant being held by the male sitting next to pond with near by stroller

pred: man in black shirt is standing in front of building

BLEU: 0

true: black dog carries green toy in his mouth as he walks through the grass

pred: black dog is running over the grass

BLEU: 0.24303324868167356

Good Caption

true: black dog and spotted dog are fighting

pred: black and white dog is running in the grass

BLEU: 0.7598356856515925

true: black dog running in the surf

pred: dog is running through the water

BLEU: 0.840896415253714 5

true: man drilling hole in the ice

pred: man in blue shirt is standing on the water

BLEU: 0.7598356856515925

true: man and baby are in yellow kayak on water

pred: man in pink shirt is standing on the water

BLEU: 0.816496580927726