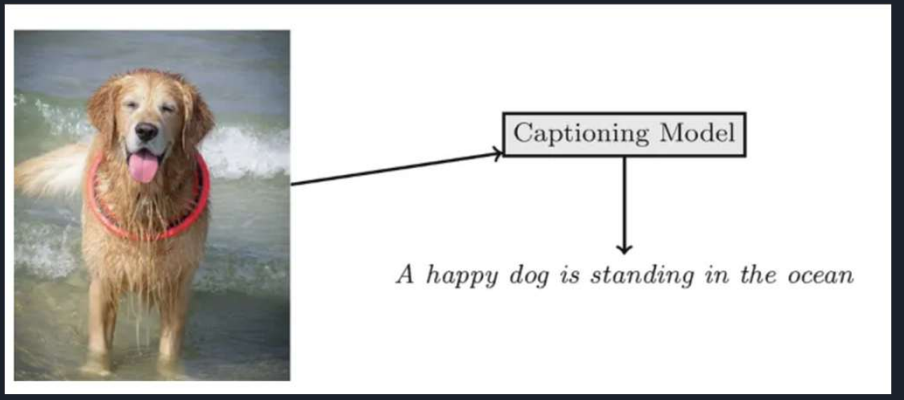
Generating Captions for Images using CNN and LSTM

Presented by:

Aditi Akhilesh
Amrita Prakash
Shivani Bapuji Vhatkar
Vaishnavi Pravin Apsingkar

Introduction



- What is Image captioning?
  - Combines Computer vision and Natural Language Processing
  - Recognizes image context and describes it in natural human language (ex. English)
  - In simple words, converting image into text
- Applications
  - Automatic content creation
  - Accessibility tools for the visually impaired
  - Enhancing search engines for image search

Objective

- Building a model for generating accurate, coherent captions for images
- Understanding CNN (Convolutional Neural Network) and LSTM (Long Short-Term Memory) models
- Using CNN for feature extraction and LSTM for sequence generation

Dataset

- Flickr_8k dataset is used
- Includes
  - 8091 images
  - Each image has 5 textual captions
- Lightweight dataset
- Less computational cost
- Small but effective for prototyping.

Data Preprocessing

Image Data:

- Resized to 224x224 pixels for VGG16.
- Features extracted and stored using pickle for efficiency.

Text Data:

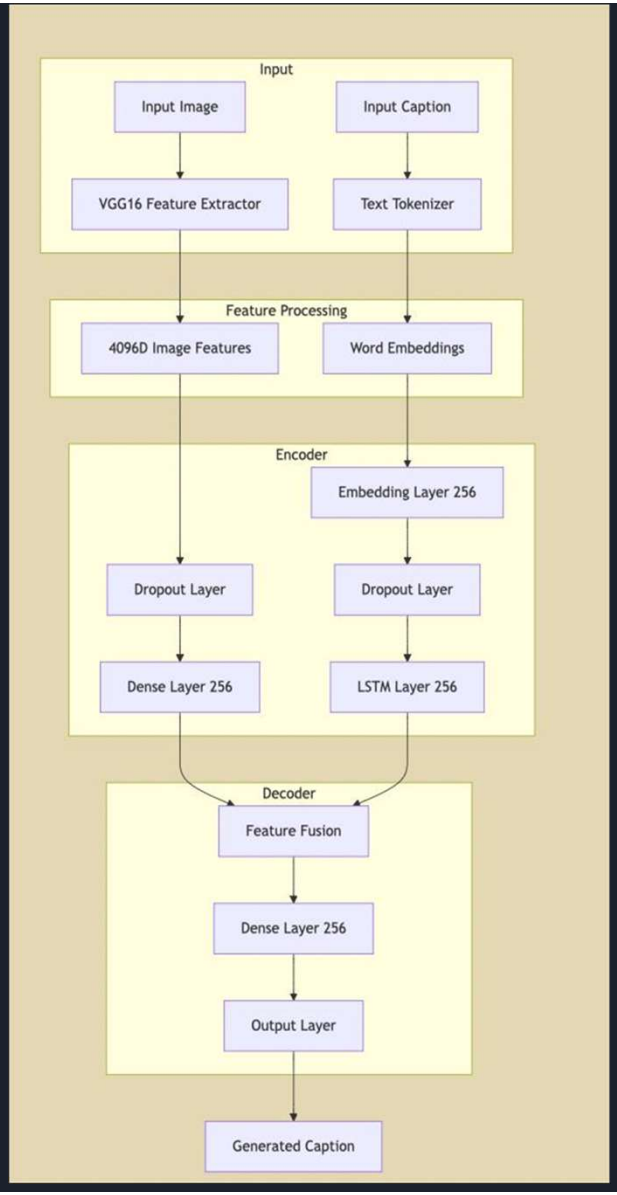
- Cleaned captions: Removed punctuation, digits, and converted to lowercase.
- Added start/end tokens: "startseq ... endseq".
- Tokenized captions : Converting text captions into numerical representations.

Example of text preprocessing:

- Before: "A child in a pink dress climbing stairs."
- After: "startseq a child in a pink dress climbing stairs endseq"

Model architecture

- Encoder-decoder structure
- 2 pre-processing layer.
- Encoder-Dense Layer: Most important 256 features captured.
- LSTM: captures sequential relation among words in caption.
- Decoder-Dense Layer: Studies pattern from combined feature.



Implementation of CNN

- Pre-trained model VGG16 is used for feature extraction
- VGG16:
  - 16 layer deep convolutional network
  - Trained on Imagenet dataset with 1000 classes
  - Classification layer removed.
- Steps:
  - Load VGG16 with pre-trained weights
  - Restructure the model to output from penultimate layer (4096 features)
  - Preprocess the images
    - Resizing to 224x224 pixels
    - Convert to numpy array and normalize
  - Feed the images into CNN to extract feature vectors

```
# Load the VGG16 model
vgg_model = VGG16()
vgg_model = Model(inputs=vgg_model.inputs, outputs=vgg_model.layers[-2].output)

# Extract features from images
features = {}
for img_name in tqdm(os.listdir(IMAGES_DIR)):
    img_path = os.path.join(IMAGES_DIR, img_name)
    image = load_img(img_path, target_size=(224, 224))
    image = img_to_array(image)
    image = image.reshape((1, image.shape[0], image.shape[1], image.shape[2]))
    image = preprocess_input(image)
    feature = vgg_model.predict(image, verbose=0)
    image_id = img_name.split('.')[0]
    features[image_id] = feature
```

Implementation of LSTM

- LSTM is type of RNN (Recurrent Neural Network) model
- LSTM:
  - Ideal for handling sequential data
  - Used to generate caption word by word based on features of image and the previous words
  - This layer has 256 neurons
- Steps:
  - Takes input from Embedding layer and passes it to Dropout layer
  - Selects the word with maximum probability.
  - Output is combined with image features using add().

Experimental Setup

```
inputs1 = Input(shape=(4096,))
fe1 = Dropout(0.5)(inputs1)
fe2 = Dense(256, activation='relu')(fe1)

inputs2 = Input(shape=(max_length,))
se1 = Embedding(vocab_size, 256, mask_zero=True)(inputs2)
se2 = Dropout(0.5)(se1)
se3 = LSTM(256)(se2)

decoder1 = add([fe2, se3])
decoder2 = Dense(256, activation='relu')(decoder1)
outputs = Dense(vocab_size, activation='softmax')(decoder2)

model = Model(inputs=[inputs1, inputs2], outputs=outputs)
model.compile(loss='categorical_crossentropy', optimizer='adam')
```

Dataset Split: 90% training, 10% testing.

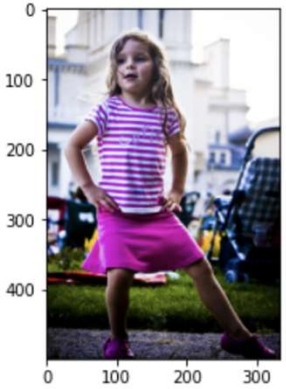
Training Details:

- Batch size: 64.
- Epochs: 25.
- Optimizer: Adam(Adaptive Moment Estimation)
    - To minimize the loss function during training.

For loss function, we have used categorical cross - entropy

Outputs

```
---------------------Actual---------------------
startseq child in all pink is posing nearby stroller with buildings in the distance endseq
startseq little girl in pink dances with her hands on her hips endseq
startseq small girl wearing pink dances on the sidewalk endseq
startseq the girl in bright pink skirt dances near stroller endseq
startseq the little girl in pink has her hands on her hips endseq
--------------------Predicted--------------------
startseq girl in pink dances in parade endseq
```

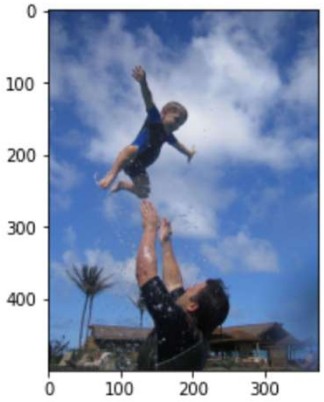


Outputs

```
--------------------Actual--------------------
startseq crowd watching air balloons at night endseq
startseq group of hot air balloons lit up at night endseq
startseq people are watching hot air balloons in the park endseq
startseq people watching hot air balloons endseq
startseq seven large balloons are lined up at nighttime near crowd endseq
-------------------Predicted-------------------
startseq crowd of people watching hot air balloons at night endseq
```

Outputs

```
--------------------Actual--------------------
startseq man in wetsuit is throwing baby wearing wetsuit up into the air endseq
startseq man in wetsuit is throwing toddler up in the air and is ready to catch him endseq
startseq man in water throwing little boy up in the air and waiting for him to come down so he can catch him endseq
startseq the man is in the pool and throwing small boy into the air endseq
startseq "while water droplets fly man throws little boy up in the air ." endseq
--------------------Predicted--------------------
startseq man is throwing his board up into the air endseq
```



Outputs

```
----------------------Actual----------------------
startseq dog jumps into the pond endseq
startseq dog leaping into the water near motorboat endseq
startseq light colored dog jumping off the back of boat into the water endseq
startseq tan dog jumping off of boat into the water endseq
startseq the yellow dog is jumping off of ship into the water endseq
--------------------Predicted--------------------
startseq dog is running through the water endseq
```

Outputs

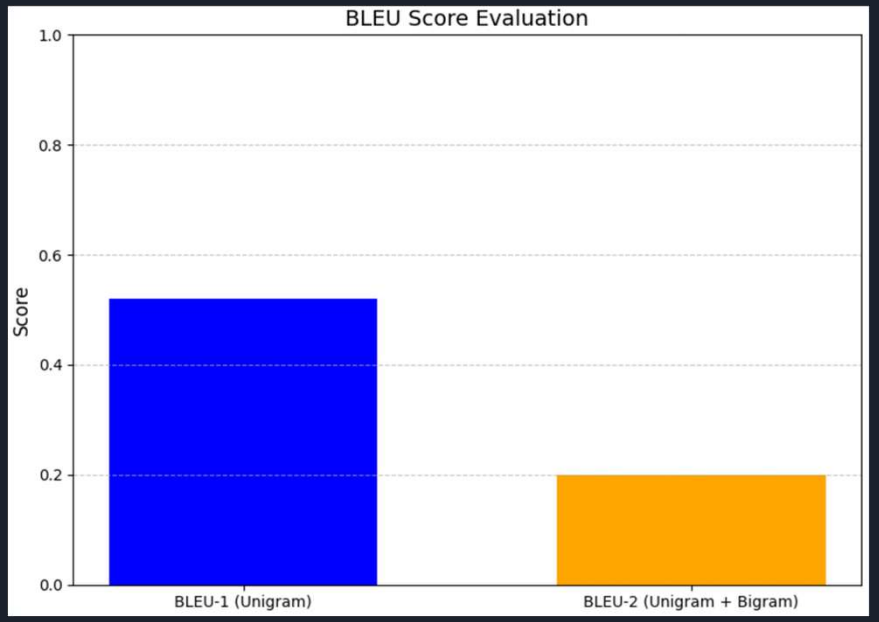
```
---------------------Actual---------------------
startseq boy goes down an inflatable slide endseq
startseq boy in red slides down an inflatable ride endseq
startseq boy is sliding down in red shirt endseq
startseq child going down an inflatable slide endseq
startseq "a young boy sliding down an inflatable is looking off camera ." endseq
--------------------Predicted--------------------
startseq a child slides down an orange slide endseq
```

Evaluation

BLEU (Bilingual Evaluation Understudy) Score measures the quality of text generated by comparing it to one or more reference texts.

- BLEU-1 (Unigram): Measures the overlap of individual words.
- BLEU-2 (Unigram + Bigram): Considers both individual words and pairs of consecutive words.

Tested with test dataset(10%)



Evaluation

Criterion	Average Score (Out of 5)	Percentage
Relevance	3.7	74%
Fluency	4.7	94%
Detail	3.5	70%
Novelty	2.9	58%
Overall Quality	3.5	70%

Process:

- Selected few images from the test dataset.
- Conducted an evaluation based on human feedback by asking specific questions.

Evaluation Criteria:

- Relevancy: How closely does the caption match the content of the image?
- Fluency: Is the caption grammatically correct and natural?
- Detail: Does the caption provide sufficient information about the image?
- Novelty: Is the caption unique and non-repetitive?

Link

Challenges and limitations

- Training on larger datasets requires significant time and resources.
- Fine-tuning pre-trained CNNs for improved feature extraction.
- BLEU score improvement through better tokenization and hyperparameter tuning.

Discussion and future work

- Train on larger datasets (e.g., Flickr30K dataset, MSCOCO).
- Experiment with advanced architectures (e.g., Transformers).
- Integrate real-time captioning for live feeds.

Personal Takeaways

- Technical Insights:
  - Improved understanding of computer vision and NLP techniques.
  - Learned to build encoder-decoder models and use VGG16 for feature extraction.
  - Gained experience in text preprocessing and model evaluation using BLEU scores.

- Problem-Solving Strategies:
  - Developed effective methods to break down complex tasks.
  - Efficiently accessed resources through documentation and forums.
  - Refined model performance through iterative testing.

Conclusion

- Successfully built an image caption generator using CNN and LSTM.
- Demonstrated the ability to describe images with reasonable accuracy.
- Encouraging results with BLEU score and meaningful captions.

Any questions?

Thank you!

References

- Flickr8k Dataset:
  https://github.com/jbrownlee/Datasets/releases
- Pre-trained VGG16 Model:
  Simonyan, K., & Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition

  https://arxiv.org/abs/1409.1556
- BLEU Score:
  Papineni, K., Roukos, S., Ward, T., & Zhu, W. J. (2002). BLEU: a method for automatic evaluation of machine translation

  https://aclanthology.org/P02-1040/
- NLTK Documentation for BLEU Scoring:
  https://www.nltk.org/_modules/nltk/translate/bleu_score.html