Image Captioning using hybrid CNN-LSTM Model

- By Team A - Batch 35B - EvoAstra Internship Program

Introduction

This project aims to generate descriptive captions for the given images by implementing a hybrid Convolutional Neural Network (CNN) - Long-Short Term Memory (LSTM) models.

The project involves essential steps like:

- Image Pre-Processing
- Captions Cleaning and Tokenization
- Image Features Extraction
- Model Building, Training and Evaluation
- Fine-Tuning
- Model Testing

Importing Modules and Image Pre-processing

Importing Dataset and Modules

- Loaded the Flickr8k dataset, which contains 8,091 images of different scenarios and a captions text file which contains 5 captions per each image
- Imported necessary libraries like Numpy, Matplotlib, Tensorflow, Pickle, etc. for model development

Image Pre-processing

- Resized the images to 224 × 224 format for the model to understand better
- Reshaped the image data to pre-process in RBG format image
- Split the image name from the extension to load only the image name

Features Extraction

VGG16 Model

- Loaded the VGG16 model excluding the final classification layer, only to extract features from image
- Extracted the important features of all the images
- Stored the extracted features using Pickle module for quicker use and future purposes without re-extraction

Captions Cleaning and Tokenization

01

Load Captions

- Loaded the captions text file, which contains 5 captions per each image
- Removed the extension from image ID

02

Pre-process Text

- Pre-processed the captions like deleting digits, special characters, additional spaces
- Added start and end tokens to the beginning and end of each caption

03

Map to Integers

- Mapped each word to a unique integer using a tokenizer
- Stored all the captions into a list

Train/Test Split

Dataset Division

Split the dataset into 80% for training and 20% for testing purposes.

Batch Generator

Defined a batch generator which includes a padding sequence to normalizes the size of all captions to the maximum size for better results

Model Building

Model Architecture - Tensorflow

Built the model using Tensorflow encoder- decoder architecture which include techniques like:

- Taking output shape of the extracted features from VGG16 and processed captions as input layer
- Adding Dropout layers to drop a fraction of data to prevent overfitting
- Using activation functions like Rectified Linear Unit (ReLU) to learn complex patterns in the data and Softmax for probabilistic classification function
- Using Categorical Cross Entropy and Adam optimizer which automatically adjusts the learning rate of the model
- Summarized and plotted the model for numerical and visual representation of the model architecture

Layer (type)	Output Shape	Param #	Connected to
text (InputLayer)	(None, 35)	0	–
image (InputLayer)	(None, 4096)	0	–
embedding (Embedding)	(None, 35, 256)	2,157,312	text[0][0]
dropout (Dropout)	(None, 4096)	0	image[0][0]
dropout_1 (Dropout)	(None, 35, 256)	0	embedding[0][0]
not_equal (NotEqual)	(None, 35)	0	text[0][0]
dense (Dense)	(None, 256)	1,048,832	dropout[0][0]
lstm (LSTM)	(None, 256)	525,312	dropout_1[0][0], not_equal[0][0]
add (Add)	(None, 256)	0	dense[0][0], lstm[0][0]
dense_1 (Dense)	(None, 256)	65,792	add[0][0]
dense_2 (Dense)	(None, 8427)	2,165,739	dense_1[0][0]

Total params: 5,962,987 (22.75 MB)

Trainable params: 5,962,987 (22.75 MB)

Non-trainable params: 0 (0.00 B)

Model Training and Evaluation

Model Training

- Trained the model over 20 epochs
- Added a back-propagation layer to decrease loss over each epoch
- Saved the trained model for future use

Model Evaluation

- Evaluated the model using Bilingual Evaluation Understudy (BLEU) scores
- Obtained BLEU scores between 0.2 - 0.5

Model Testing

- Tested the model on an image and visualized the results
- First prints the actual captions of the image then prints a predicted caption of the image to understand how better our model has performed.

Conclusion

This project successfully demonstrates an end-to-end image captioning system built using a hybrid CNN–LSTM architecture. Through a structured workflow, ranging from image preprocessing and caption cleaning to feature extraction with VGG16, model construction, training, and evaluation, the system is able to generate meaningful textual descriptions for images.

Overall, the project highlights the effectiveness of combining CNN-based image feature extraction with LSTM-based language modelling for automated image caption generation, which can be used for real-world applications such as helping visually impaired people, Content generation, Product description in E-Commerce website, etc.