CS772 Project (Image Captioning)

https://github.com/RamkrishnaKamble/ImageCaptioning

Laxman Desai, 190020066
Jos Katiyare, 19020054
Ramkrishna Kamble, 190020094

5-May-2022

Problem Statement (1/2)

The goal is to create a system that combines the power of RNN, CNN and FFNN. You will have a two stage DNN, wherein the first stage is a CNN processing an image and an RNN/Transformer processing the caption of the image. The FFNN will take outputs of CNN and RNN and will give the verdict as a value between 0 and 1 (both included), expressing the degree of consistency between the image and the caption (1- consistent, 0-inconsistent).

Problem Statement (2/2)

For example, if the image is that of a tiger chasing a deer, the caption of "a peaceful scene of nature" is inconsistent with the picture. On the other hand, the picture of a long line of people can have many consistent captions- (a) Crowd eagerly waiting for a ticket to the cricket stadium, or (b) Hungry people in food-line during covid, or (c) Students waiting in queue for an admission form, but not (d) Snow-flakes falling from the sky.

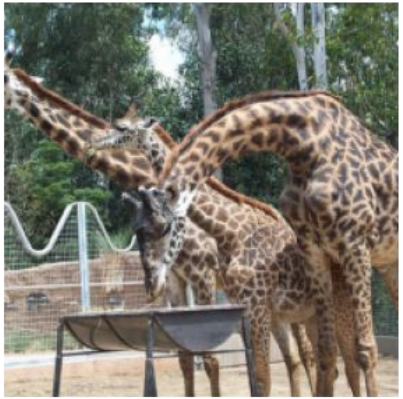
Dataset Discussion

Visualising dataset

Training dataset images

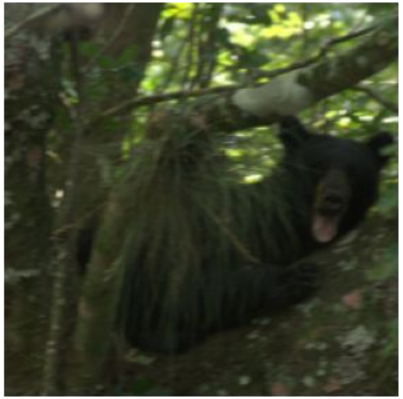
Caption:
an airplane monument placed beside of a road
.

Caption:
a number of giraffes near one another



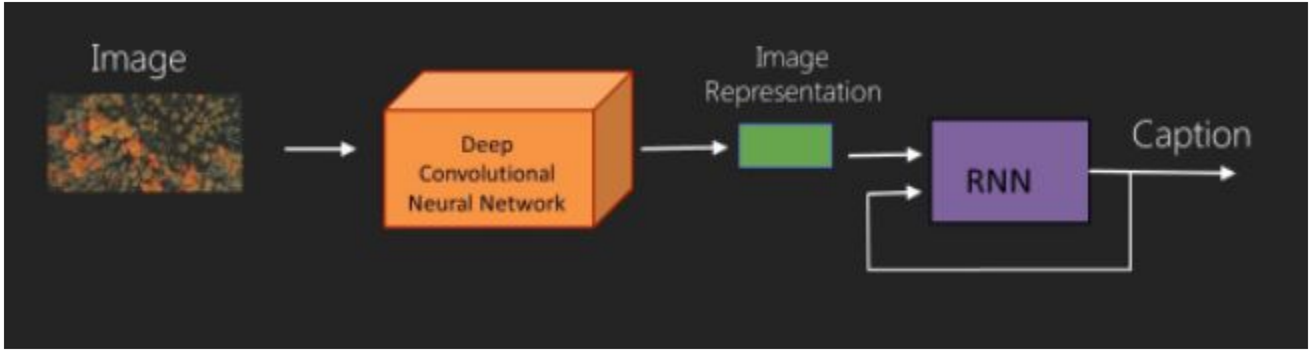
Caption:
an array of pictures of a family and
different food selections .

Caption:
what better place for a bear than up
a tree ?



Details of the FFNN N/W

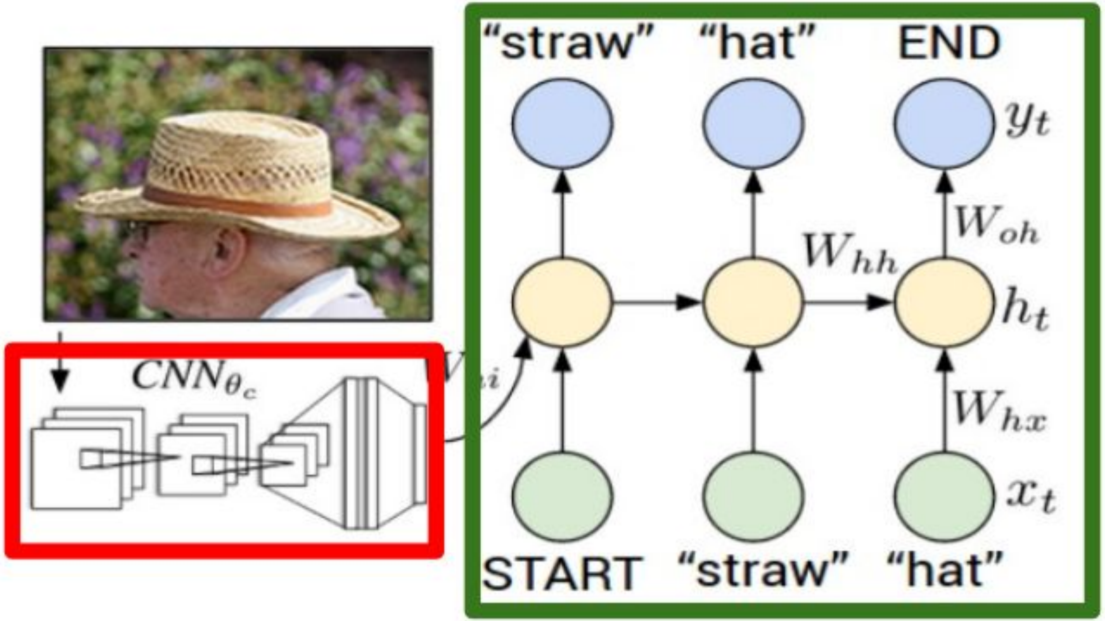
- Hidden Layers - 512
- Embedding dimension - 512
- Different Hyper parameters - Number of hidden layers, epochs, embedding dimensions



System implementation

Describing images

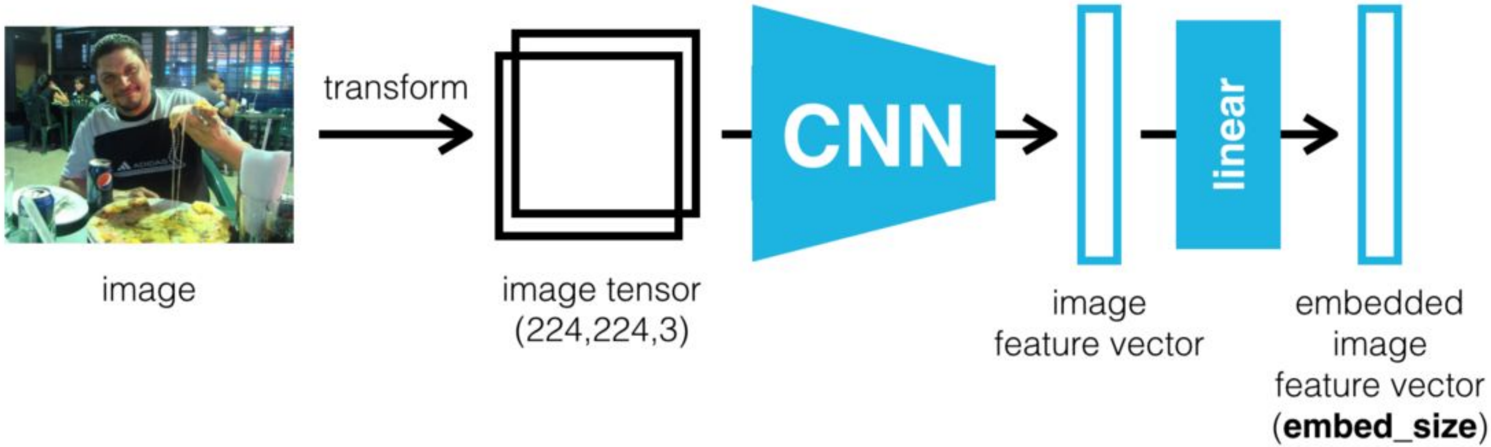
Recurrent Neural Network



Convolutional Neural Network

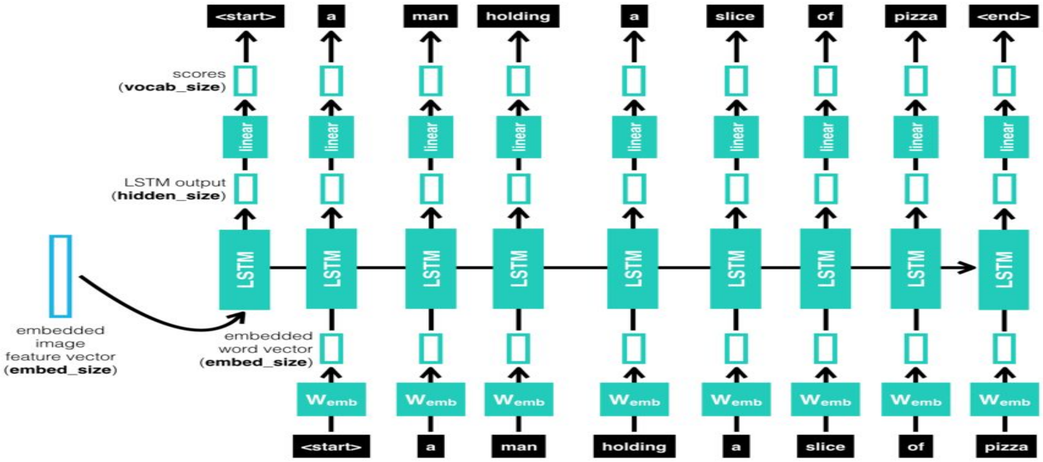
Details of the CNN N/W

As mention we usne CNN, layer as encoder layer. We first normalize image using mean and std deviation of imagenet dataset. After that we use VGGnet architecture as CNN to get feature vector. After that we pass it through linear layer to get feature vector of embedding size (512).



Details of the LSTM N/W

LSTM here is used as a decoder to predict words using feature vector as weights. Final image vector we get from encoder is passed an initial state. After that word embedding is passed with previous hidden state to get word embeddings which is used to predict final sentence(caption)



Training details (hyper-parameters)

- How many epochs?

  3
- What is the learning rate?

  1e-3
- Loss function

  Criterion we use here is categorical cross entropy loss.