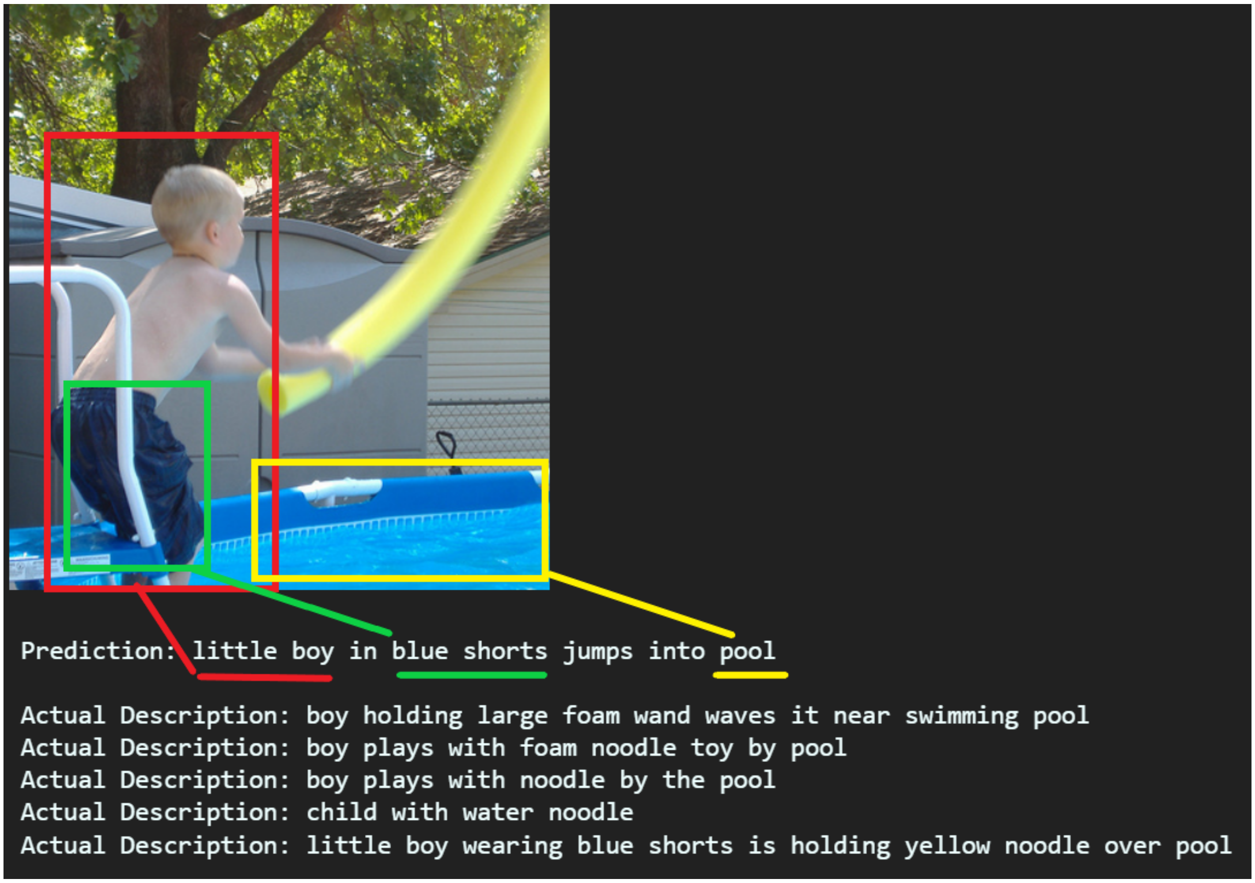
IMAGE CAPTIONING

Image Captioning is the process of generating textual description of an image. It uses both **Natural Language Processing** and **Computer Vision** to generate the captions. The idea is to replace the encoder (RNN layer) in an encoder-decoder architecture with a deep **convolutional neural network** (**CNN**) trained to classify objects in images.

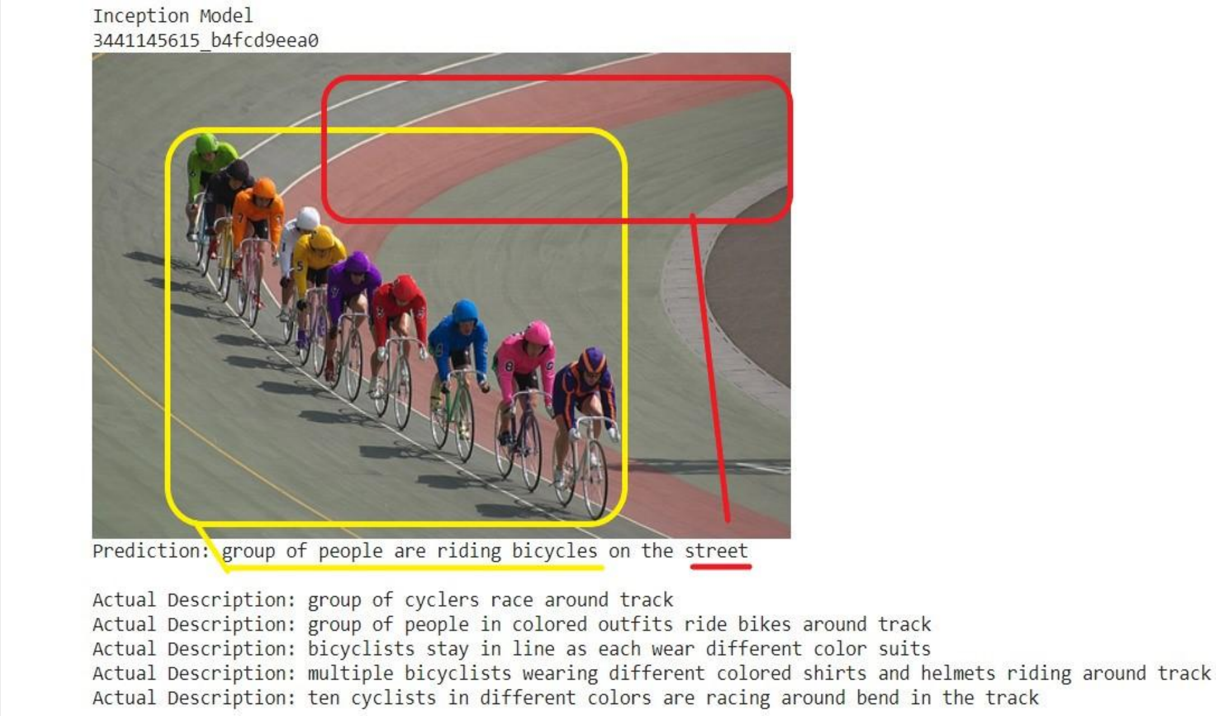
Applications

- Help for blind people by generating captions of images of the view they see.
- Search photos by searching their generated captions. (Ex. Google Photos)
- Provision of captions to provide HTML header and "alt" attribute to improve Search Engine scoring of the page for search terms related to content of the movie or image(in Web Development).
- Use in virtual assistants.
- In Social Media (Ex. Facebook auto detects friends from images of posts we share and notify those friends)

Observations (Test Dataset)

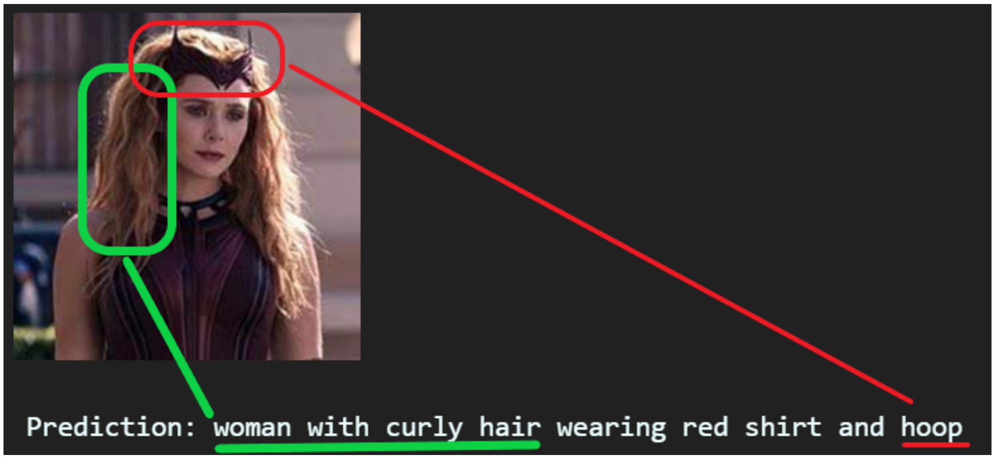


```
Prediction: little boy in blue shorts jumps into pool

Actual Description: boy holding large foam wand waves it near swimming pool
Actual Description: boy plays with foam noodle toy by pool
Actual Description: boy plays with noodle by the pool
Actual Description: child with water noodle
Actual Description: little boy wearing blue shorts is holding yellow noodle over pool
```

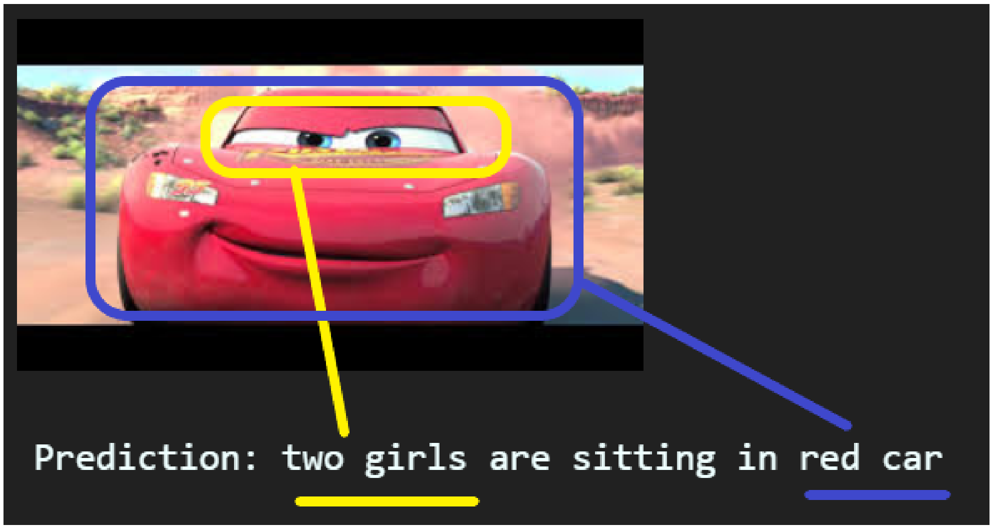


```
Inception Model
3441145615_b4fcd9eea0
Prediction: group of people are riding bicycles on the street

Actual Description: group of cyclers race around track
Actual Description: group of people in colored outfits ride bikes around track
Actual Description: bicyclists stay in line as each wear different color suits
Actual Description: multiple bicyclists wearing different colored shirts and helmets riding around track
Actual Description: ten cyclists in different colors are racing around bend in the track
```

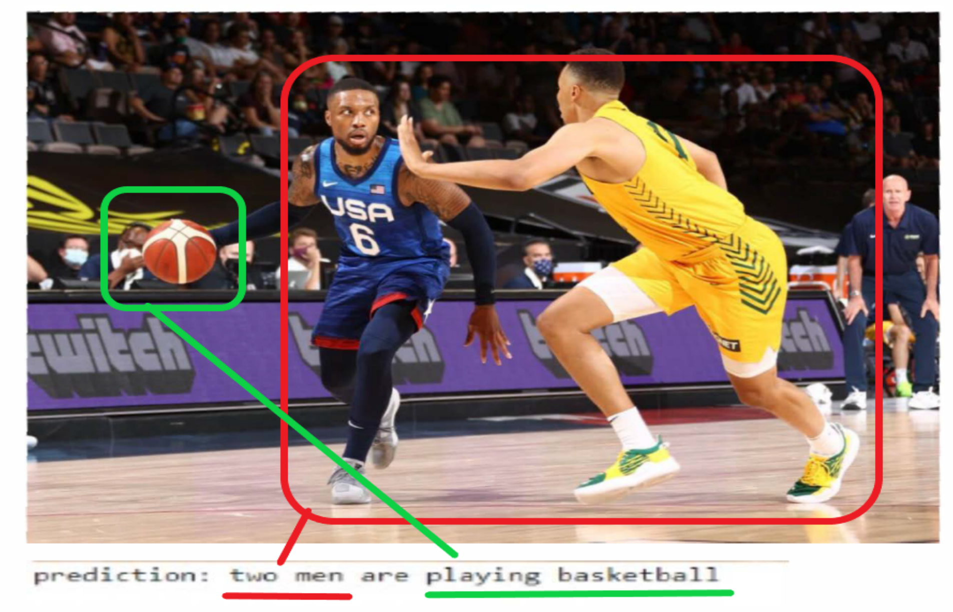
Locally Tested Images Not in Dataset



Prediction: woman with curly hair wearing red shirt and hoop



Prediction: two girls are sitting in red car



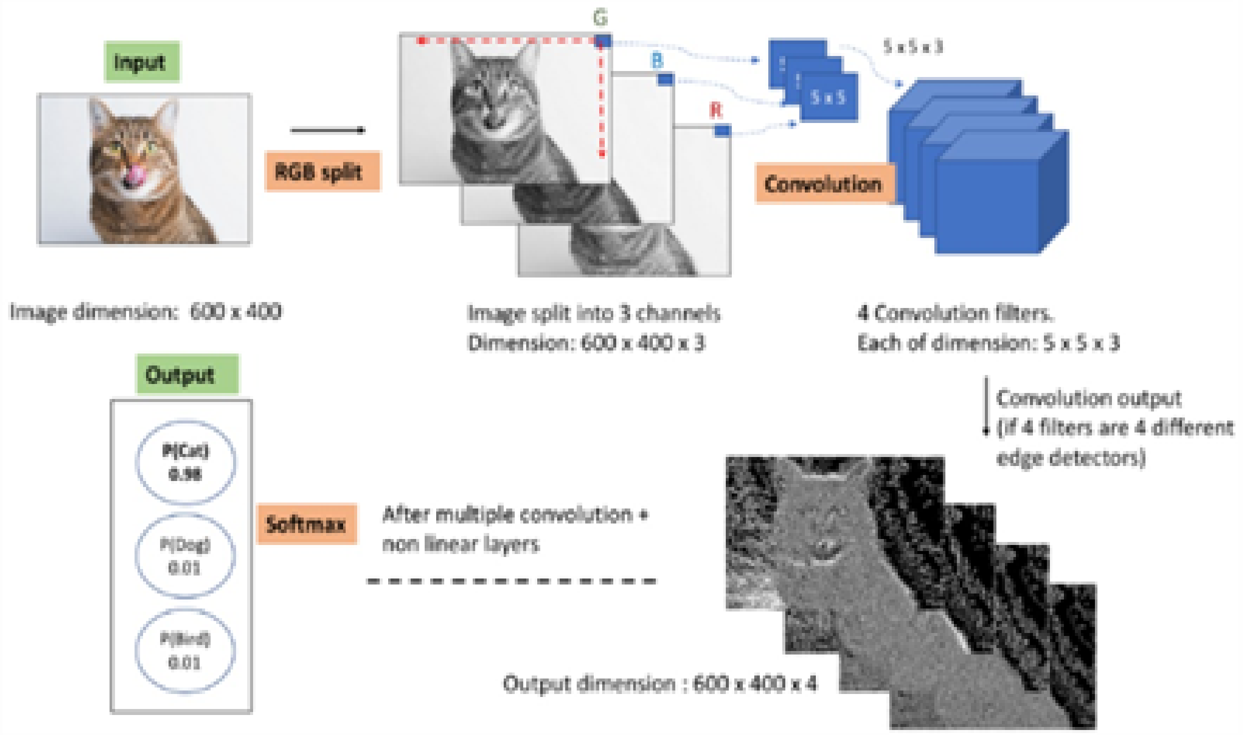
prediction: two men are playing basketball

Convolutional Neural Networks

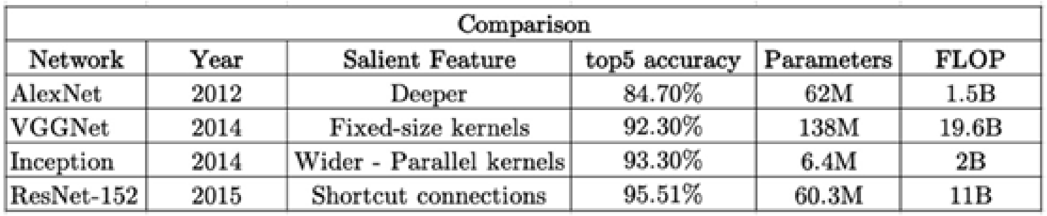
Image classification/recognition, object detection and localisation are the three important computer vision problems.

The machine must be able to learn patterns like vertical, horizontal edges, round shapes and other patterns of an image. Convolution, Pooling and RELU are the three operations that are performed multiple times on an input matrix of pixels representing an image.

Convolution is done by convolving with n filters which is then treated with pooling layers like max-pool / average-pooling and then with nonlinear activation functions like RELU. Finally the fully connected layer is passed to a softmax layer which predicts the probability of each object in the given image.



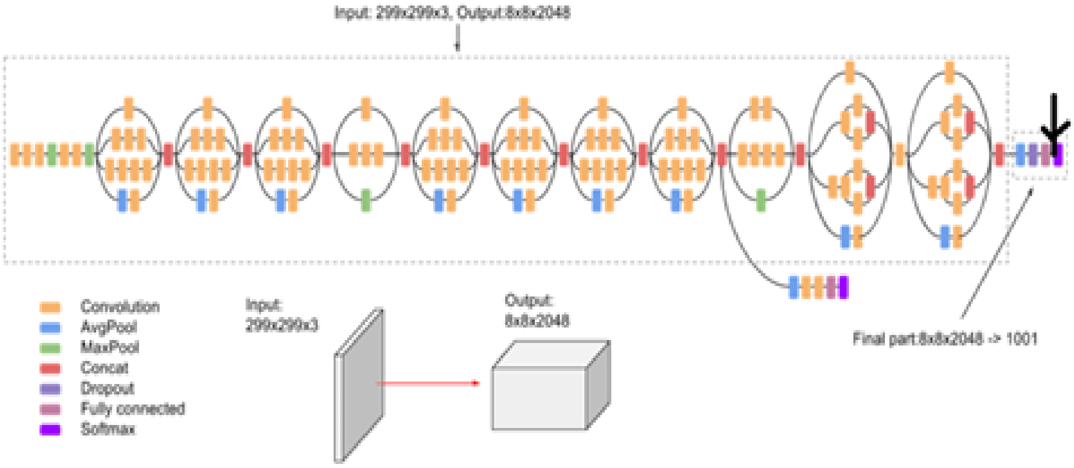
Some of the classic networks to extract features and implement the above are AlexNet, VGG, LeNet-5, Inception, ResNet.



Comparison					
Network	Year	Salient Feature	top5 accuracy	Parameters	FLOP
AlexNet	2012	Deeper	84.70%	62M	1.5B
VGGNet	2014	Fixed-size kernels	92.30%	138M	19.6B
Inception	2014	Wider - Parallel kernels	93.30%	6.4M	2B
ResNet-152	2015	Shortcut connections	95.51%	60.3M	11B

Source

In this project we have used two networks - InceptionV3 and ResNet152 which are CNN models trained on the imagenet dataset. The pretrained weights used for these networks were already trained on 1000 different classes.



At this point we extract the 2048 - length vector by removing the last softmax layer which performs 1000-class classification

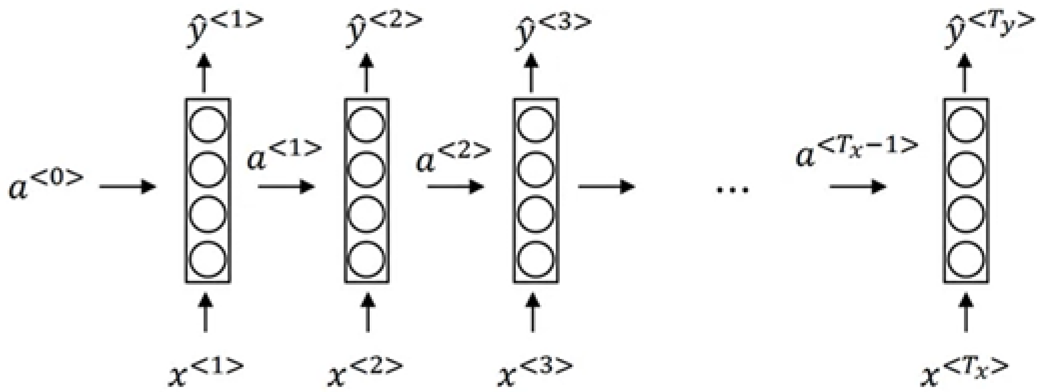
Normally, the CNN’s last layer is the softmax layer, which assigns the probability that each object might be in the image. But if we remove that softmax layer from CNN, we can feed the CNN’s rich encoding of the image into the decoder (language generation RNN) designed to produce phrases

Inceptionv3 - 2048 length feature vector

Resnet 152 V2 - 2048 length feature vector

Recurrent Neural Networks (RNN)

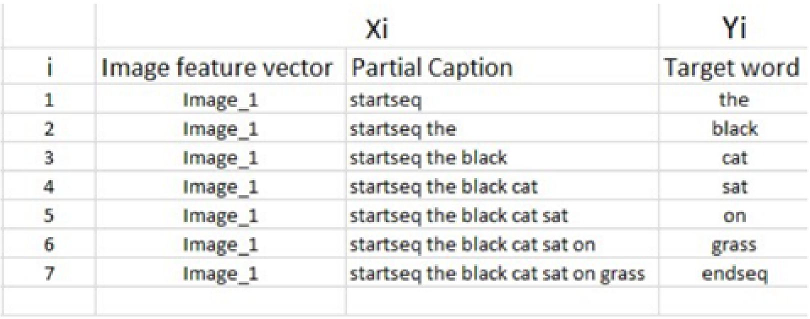
The second step in this project is to generate captions.. In image captioning the input is an image and the output is sequence of words, so we use sequence models such as Recurrent Neural Networks to solve this problem



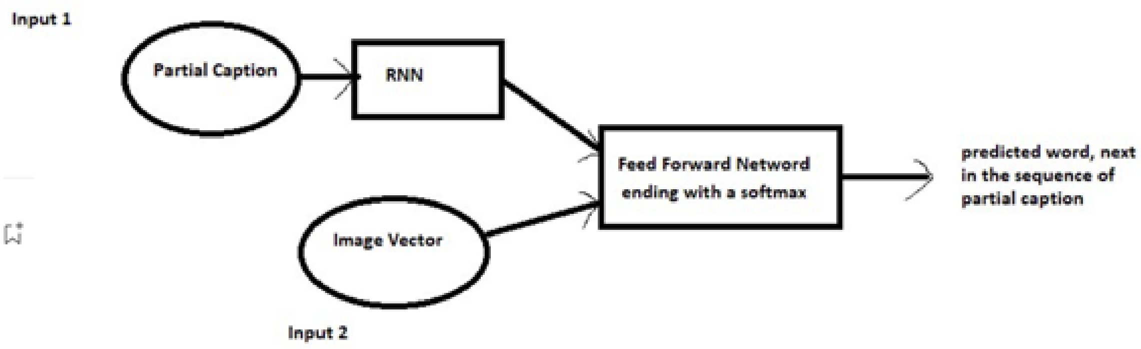
Language Modelling

We construct a vocabulary of words from the training dataset. Each word is assigned an index corresponding to its position in the list.

In Image Captioning we have two inputs, feature vector of the image+ partial caption.



	Xi		Yi
i	Image feature vector	Partial Caption	Target word
1	Image_1	startseq	the
2	Image_1	startseq the	black
3	Image_1	startseq the black	cat
4	Image_1	startseq the black cat	sat
5	Image_1	startseq the black cat sat	on
6	Image_1	startseq the black cat sat on	grass
7	Image_1	startseq the black cat sat on grass	endseq



First step prediction **max(p(y1|startseq))-**

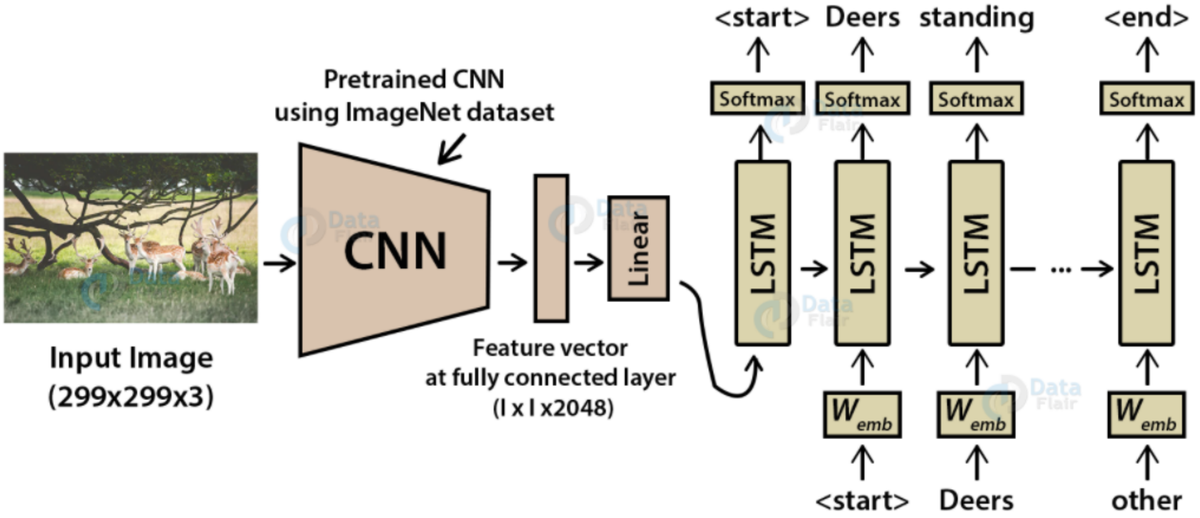
Second step **max(p(y2|startseq the))**

Third step **p(y3| startseq the black)**

……

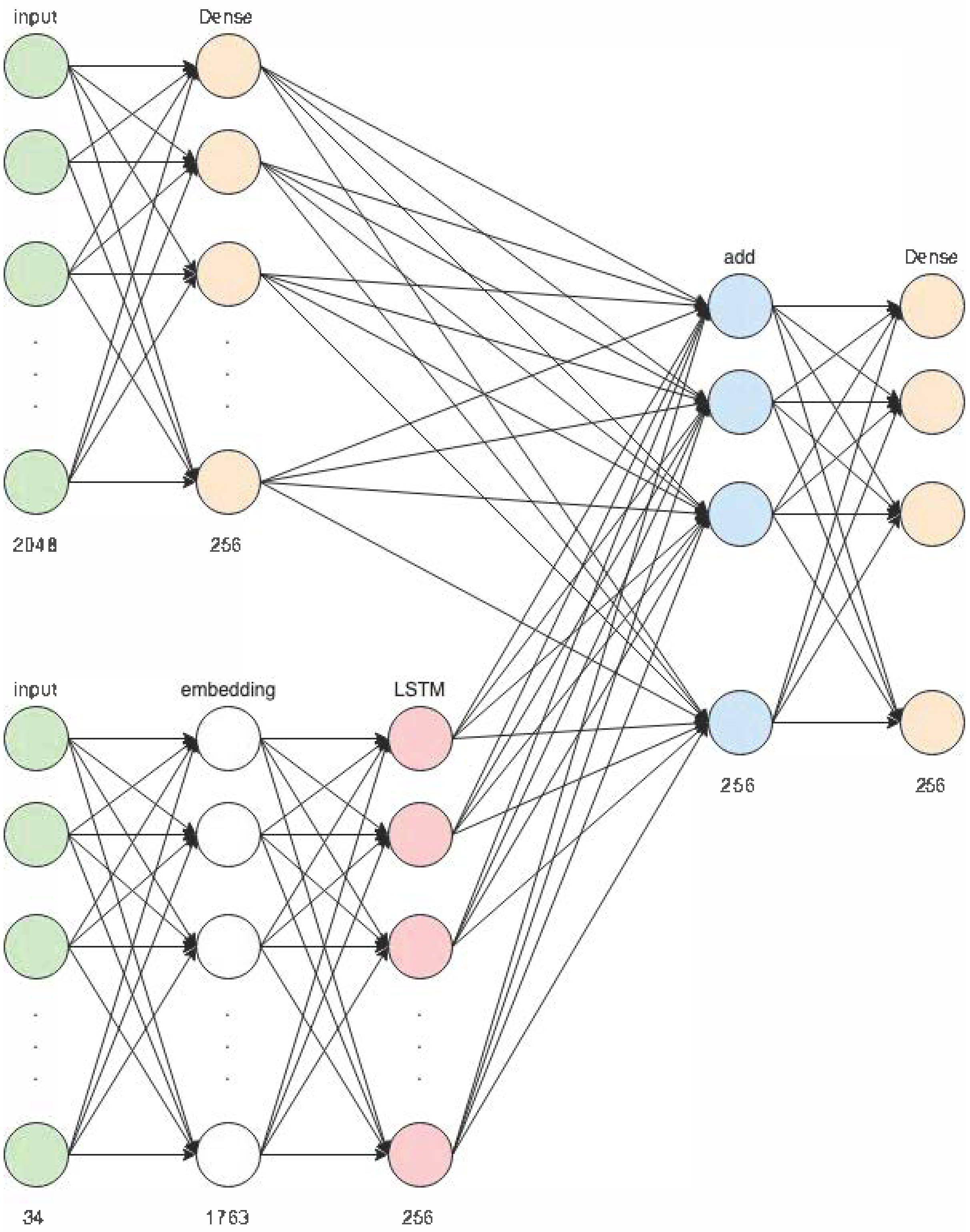
nth step p(yn| startseq the black cat…… grass)

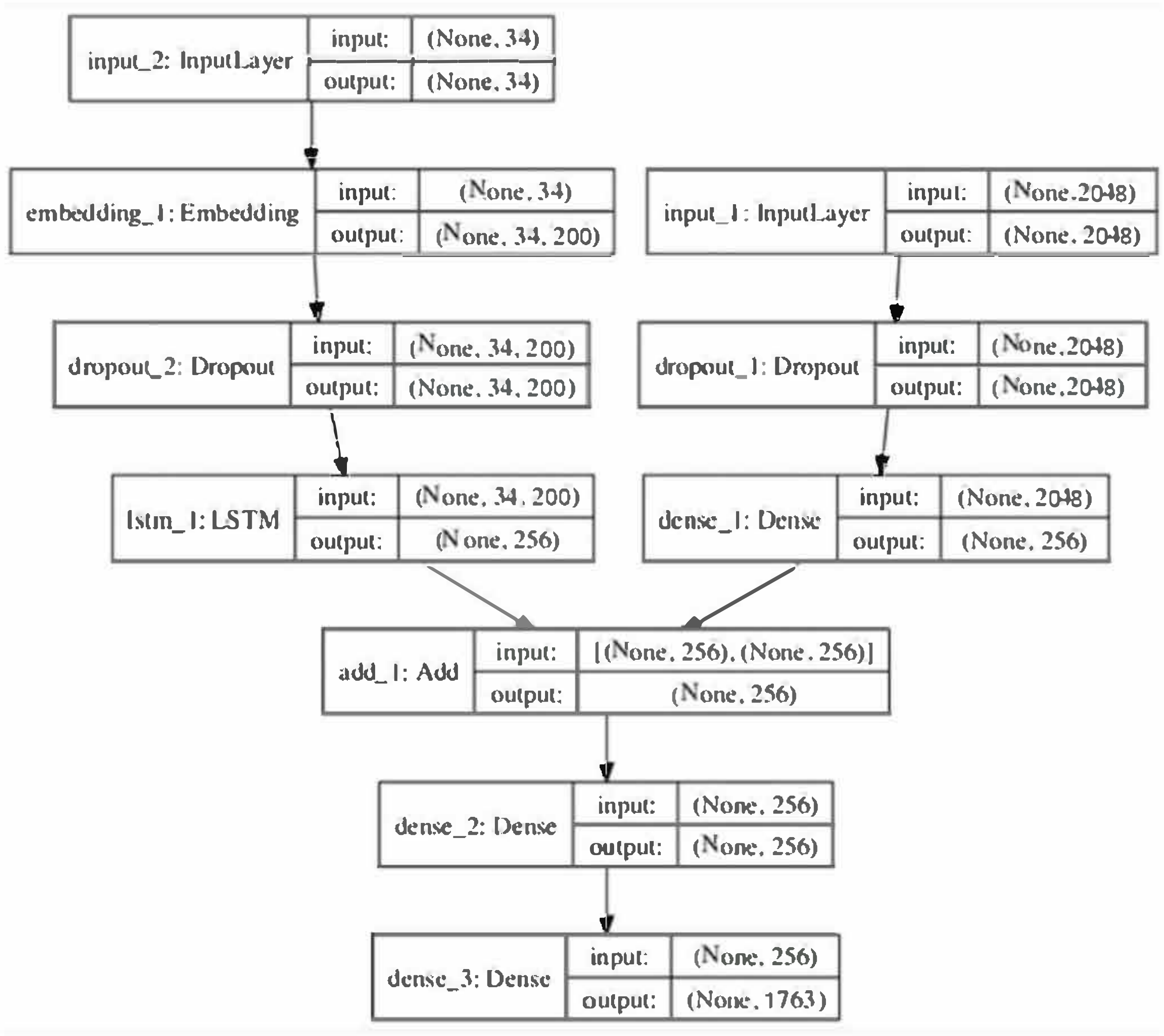
In this project we are using LSTM (**Long short term memory**) which is responsible for generating the image captions. It is a type of RNN which is well suited for sequence prediction problems. Based on the previous text, we can predict what the next word will be. It has proven itself effective from the traditional RNN by overcoming the limitations of RNN which had short term memory. LSTM can carry out relevant information throughout the processing of inputs.



In this project, we are using pre-trained Glove embeddings to feed to LSTM cells.

Model Plot


input
Dense
add
Dense
2048
256
input
embedding
LSTM
256
256
34
1763
256



Improvements

- Training on large dataset (Currently trained on Flickr_8k dataset with only 8k images, can try Flickr_30k dataset or MSCOCO dataset with 120k images), also note training on large dataset may take more than weeks.
- Adding more layers to our LSTM model.
- Training for more epochs and trying different learning rates.