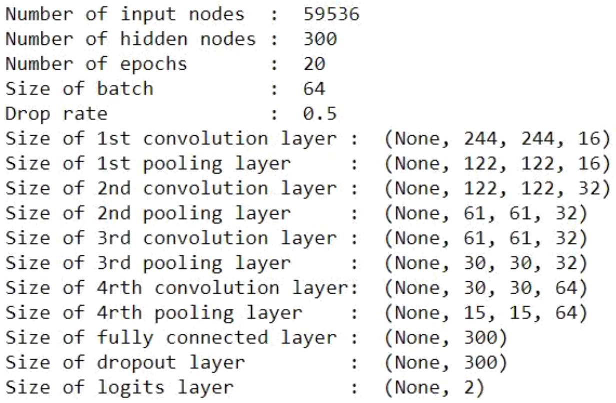
Image classification using Convolutional Neural Networks

Aim: To classify images into 'bus' and 'car' category.

Description: The project is implemented in python using TensorFlow framework.

helper_functions.py contains functions which performs preprocessing on the training data. rand_batch creates batches of random samples from training data

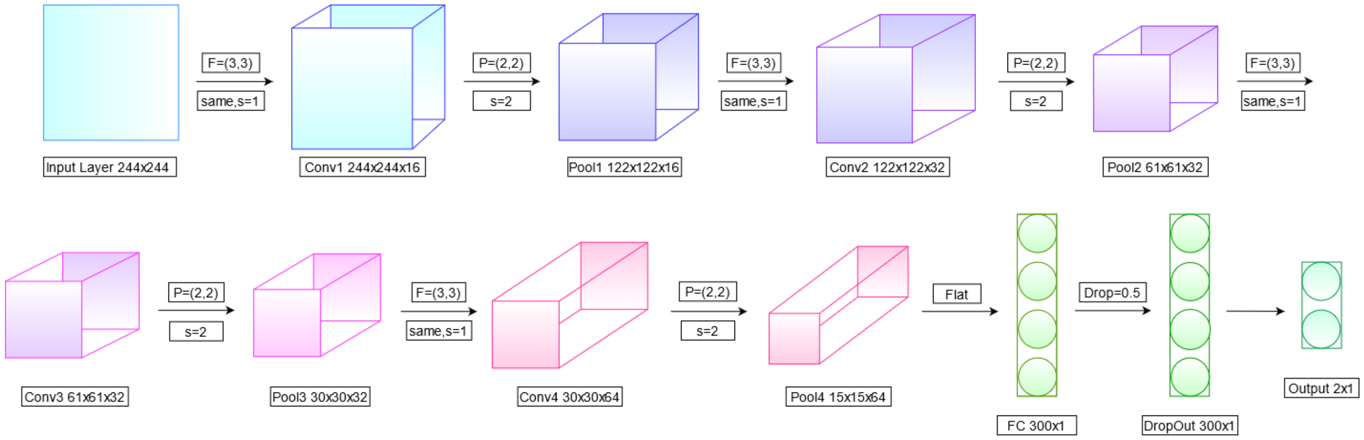
main.py contains implementation of Convolutional Neural Network.

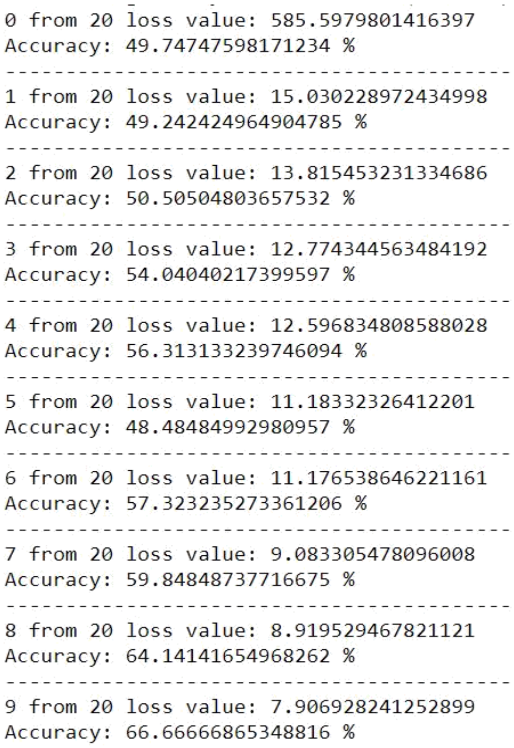


```
Number of input nodes  :  59536
Number of hidden nodes :  300
Number of epochs       :  20
Size of batch          :  64
Drop rate              :  0.5
Size of 1st convolution layer :  (None, 244, 244, 16)
Size of 1st pooling layer     :  (None, 122, 122, 16)
Size of 2nd convolution layer :  (None, 122, 122, 32)
Size of 2nd pooling layer     :  (None, 61, 61, 32)
Size of 3rd convolution layer :  (None, 61, 61, 32)
Size of 3rd pooling layer     :  (None, 30, 30, 32)
Size of 4rth convolution layer:  (None, 30, 30, 64)
Size of 4rth pooling layer    :  (None, 15, 15, 64)
Size of fully connected layer :  (None, 300)
Size of dropout layer         :  (None, 300)
Size of logits layer          :  (None, 2)
```

CNN Architecture

Architecture of the CNN is as follows:





```
0 from 20 loss value: 585.5979801416397
Accuracy: 49.74747598171234 %
------------------------------------------
1 from 20 loss value: 15.030228972434998
Accuracy: 49.242424964904785 %
------------------------------------------
2 from 20 loss value: 13.815453231334686
Accuracy: 50.50504803657532 %
------------------------------------------
3 from 20 loss value: 12.774344563484192
Accuracy: 54.04040217399597 %
------------------------------------------
4 from 20 loss value: 12.596834808588028
Accuracy: 56.313133239746094 %
------------------------------------------
5 from 20 loss value: 11.18332326412201
Accuracy: 48.48484992980957 %
------------------------------------------
6 from 20 loss value: 11.176538646221161
Accuracy: 57.323235273361206 %
------------------------------------------
7 from 20 loss value: 9.083305478096008
Accuracy: 59.84848737716675 %
------------------------------------------
8 from 20 loss value: 8.919529467821121
Accuracy: 64.14141654968262 %
------------------------------------------
9 from 20 loss value: 7.906928241252899
Accuracy: 66.66666865348816 %
```

Results

Architecture Details:

Optimizer: AdamOptimizer ()

Cost function: softmax_cross_entropy_with_logits ()

Results: Aforementioned CNN is trained on balanced dataset of 1400 images of buses and cars. Results shows test accuracy of 66.66% at 9th epoch.

This CNN model can be trained for classification of more than 2 categories. This CNN model can be trained for similar classification problems by adding problem specific class encodings to the load_data function.

Accuracy can be increased by training the CNN longer or hyperparameters search.