Birla Institute of Technology and Science

Thesis

Developing An Algorithm For Autonomous Navigation Of Drones

Author:
Akshaya Agrawal(2013B5AA0559H)

Supervisors:
Dr. S.N Omkar
&
Prof. BVVSN Prabhakar Rao

A thesis submitted in partial fulfillment of the requirements of BITS F422T Thesis

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE
May 2018

Declaration of Authorship

I, Akshaya Agrawal, declare that this Thesis titled, 'Developing An Algorithm For Autonomous Navigation Of Drones' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: ____________________

Date: 14/05/2018

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE

Abstract

Developing An Algorithm For Autonomous Navigation Of Drones

by Akshaya Agrawal

An Algorithm for autonomous navigation of drones has been proposed based on a specially designed Neural Network known as Spatial CNN. What is Spatial CNN has been discussed briefly along with the other assumptions and procedure followed to design this network. . .

Acknowledgements

First and foremost, I would like to thank my guide Prof. S.N Omkar Sir for giving me the opportunity to work on this project. He has been extremely supportive throughout the steep learning curve, and without his continuous inputs and extensive insights, this project would not have been continued. I am especially grateful to him for providing me with all the necessary resources to carry out the thesis works. Also, I would extend my gratitude to Mr. Saumya Kumar Saxena and Mr. Navaneeth Krishnan for their constant inputs and advises.

I would also like to thank Prof. BVVSN Prabhakar Rao for being my co-supervisor and encouraging me to undertake this interdisciplinary work. Without you, this wouldn't have been possible. . . .

Contents

List of Figures

Chapter 1

Introduction

Convolutional neural networks (CNNs) are made by stacking up convolutional layers which are defined by convulations with specific dimensional Kernels. CNN potrays the capability of extracting semantics from raw pixels. Hence, a major problem of road detection has been selected for further exploration. Significant amount of work has been done in this field, but a lot is still left to be explored. The major unsolved issues like problems created while detecting roads due to sharp shadows, differential lighting condition from morning to evening, artificial lighting on roads and congested roads have been taken into consideration. A new model of Spatial CNN to detect roads has been designed and discussed here.

Chapter 2

Autonomous Navigation of drones

2.1 Methodology

FIGURE 2.1: Major issues with road Detection

Traditionally researchers have modeled neural network structures based on MRF (Markov Random Fields) or CRF (Conditional Random Fields). This models the spatial relationship. Recently, mean field algorithm has been implemented along with neural networks so as to combine MRF or CRF with CNN. The output is normalized by applying softmax operation. Message passing is implemented by applying channel wise convolution that has large kernels. The kernel weights are optimized in the training process. Finally, after applying compatibility transform by implementing a 1 x 1 convolution layer and adding unary potentials the complete process is iterated N times to generate the final output.

Accounting for real time autonomous driving, message passing technique that has been used traditionally is computationally very expensive and hard to be used. This is because each pixel here receives information from all other pixels. Also, these methods are implemented on the output of Convolutional neural network but, top layers should be used to model the spatial relation.

Hence, to find solution to these issues Spatial CNN (SCNN) algorithm has been proposed. Here, the word 'spatial' is not synonymous to 'spatial convolution'. It points to the spatial propagation of information through the designed convolutional neural network (CNN) structure.

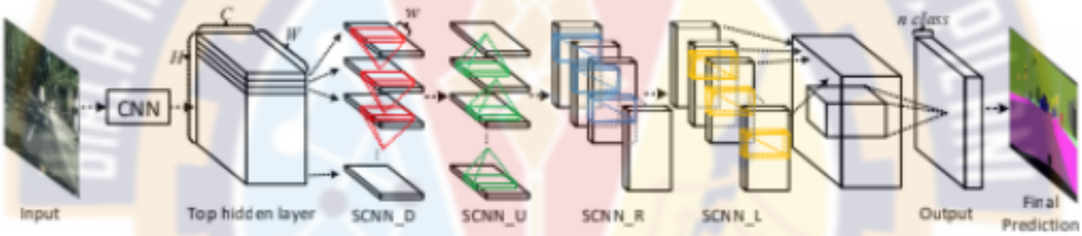


FIGURE 2.2: Convolutional layers in Spatial CNN

In order to understand this let us consider how a SCNN is applied on a 3-D tensor. Let us take a 3-D tensor of size C x H x W, where C, H and W are the number of channels, rows and columns respectively. Now split the tensor into H slices and send the first slice of this as an input to the convolutional layer with a kernel C of size C x w, where w is the width of the kernel. Traditionally, the output of a convolutional layer is fed as an input to the next layer, but here it is added to the next slice and then again this new slice goes through the same convolutional layer. Once the entire tensor goes through one convolutional layer the output is fed as an input to next convolutional layer. A spatial convolutional neural network comprises of multiple such layers along with Relu, Maxpool and fully connected layers.In a fully connected layer neurons have connections to all activations in the previous layer.

Mathematical expression: let us consider a 3-D kernel tensor K with element K(i,j,k) being the weight between an element in ith channel of the last slice and an element in the jth channel of the current slice. It has an offset of k columns between 2 elements. Let us assume the elements of input 3-D tensor X as X(i,j,k),

$$X'_{i,j,k} = \begin{cases} X_{i,j,k}, & j = 1 \\ X_{i,j,k} + f\Big(\sum_{m}\sum_{n} X'_{m,j-1,k+n-1} \\ \qquad \times K_{m,i,n}\Big), & j = 2, 3, ..., H \end{cases}$$

FIGURE 2.3: Mathematical expression for spatial convolution

where i, j, and k indicate indexes of channel, row and column respectively. Then the forward computation of SCNN is given by:

where f is a nonlinear activation function as ReLU. The X with superscript 0 denotes the element that has been updated. The convolution kernel weights are shared across all slices. Therefore, we can say that SCNN is a kind of recurrent neural network. Also, SCNN is direction specific. The SCNN module has 4 suffixes 'D','U','R','L' i.e downward, upward, rightward and leftward respectively.

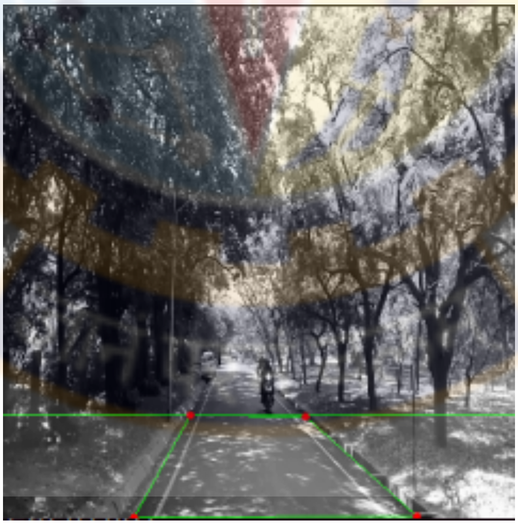


FIGURE 2.4: Sample input image with expected output

Four corner points of the road at fixed height from the bottom of the image are selected. These are the expected output features of the designed neural network. The feature points are selected in this fashion because 4 points at fixed height can define a regular trapezium. And the any road can be defined as a trapezium from a fixed height.

2.2 Results

FIGURE 2.5: Current prediction of points

Unfortunately, results are not as expected yet. Hence, currently trials are been made to train only the x co-ordinates of the feature points.

2.3 Convolutional neural networks

Convolutional neural networks are made from neurons that have learnable weights and biases. Each neuron is feed with a set of inputs which are summed over learnable weights, added with bias and passed through activation function to give the output. It has an overall loss function and an optimiser. First let us start with convolution.

2.3.1 Convolution

If we have a 5*5*3 filter and we slide it over the image to give the dot product of corresponding elements and sum then up the process is called performing convolution on the image with filter 5*5*3.

A Convolutional network is a sequential model of multiple layers stacked together. Each Convolution layer has ReLU activation function. Every convolution layer is followed by a Max pooling layer. Finally after subsequent number of ConV2D and MaxPooling2D the output is flattened. After this, multiple fully connected layer is applied to give the final predicted output.

2.3.2 Max Pooling 2D

Max pooling is a sample-based discretization process. The main objective of adding a Max pooling layer in network is to down sample the output of previous layer ultimately resulting to reduce the dimensionality. For this a max filter is used.

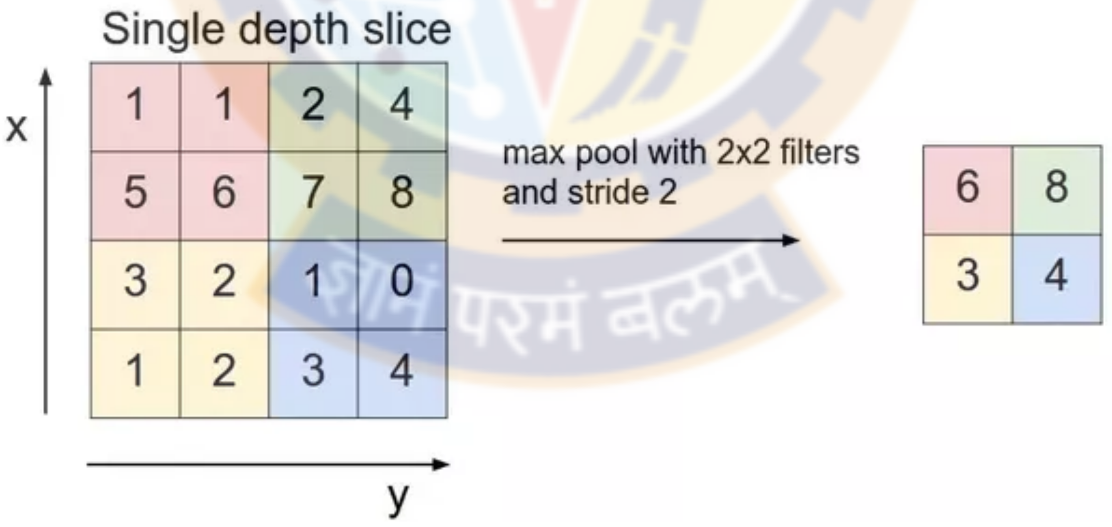


FIGURE 2.6: Mathematical expression for spatial convolution

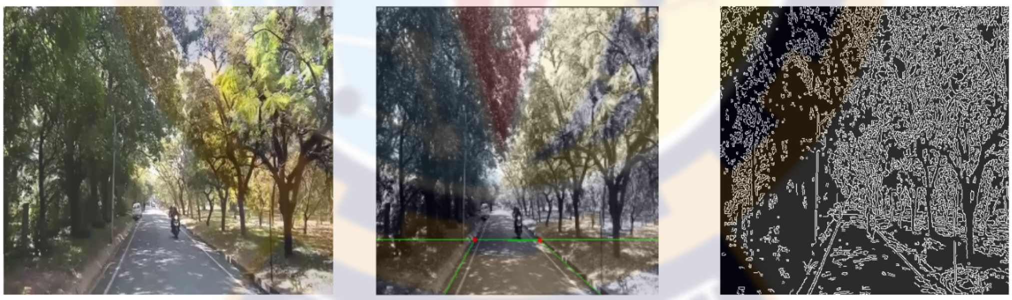
if we assume a 4x4 matrix as our initial input and filter of kernel 2x2. Also, stride is of 2 (i.e the (dx, dy) for stepping over our input will be (2, 2)) and so as to specify non-overlapping regions. For each of the regions represented by the filter, we will take the max of that region and create a new, output matrix where each element is the max of a region in the original input.

2.4 Architecture

First a batch normalization is performed on input images. Next Conv2D with 8 filters of 2x2 kernel size is applied on each image. After this there is a Max pooling layer of 2x2 kernel size. Now, again a conv2D with 16 filter and max pooling with 2x2 kernel is applied. Next 3 layers are dropout layer, conv2D layer with 32 filters and con2D layer with 32 filters again. Next is flattening layer which is followed by 3 fully connected layers of output size 256, 128 and 20.

Later loss is calculated using mean squared loss function. Now comes an important concept of Back propagation which is an algorithm for finding gradient of loss function. This gradient is then used in optimization algorithms like Adam, Adagrad, Gradient Descent to update the weights for training. Here, I have used Adam optimization algorithm for training the weights

2.5 Training Input



RGB Image Grayscale Image Canny Edge Detection Image

FIGURE 2.7: Mathematical expression for spatial convolution

The images are resized to a 400x400x3. Now, canny's edge detection algorithm is used to detect edges of the image after converting it to a grayscale image. This 400x400x(batchSize) size image is sent as an input array.

2.6 Output

10 points on each side of the road are selected to detect the shape of the road in a particular frame. Co-ordinates of these 10 points are the expected output from this model. Hence, in the size of predicted output array will be 20x20x(batchSize).

2.6.1 Adam Optimizer

Adam stands for Adaptive Moment Estimation. In this optimization algorithm, running averages of both the second moments of the gradient and the gradient itself are calculated. And later the weights are updated based on these calculated parameters.

2.7 Results

FIGURE 2.8: Mathematical expression for spatial convolution

The current accuracy of my prediction model is 0.53. It is able to detect the edges of road and fit a trapezoid on road.

2.8 Future Work

This model has to be trained on extremely congested road. Also, its real time efficiency and performance is to be tested.

2.9 Conclusion

The shape of road can be detected with an accuracy of 0.5 for autonomous navigation of drones.

Chapter 3

Bibliography

- Aly, M. 2008. Real time detection of lane markers in urban streets. In Intelligent Vehicles Symposium, 2008 IEEE, 7–12. IEEE.
- TuSimple. 2017. Tusimple benchmark. http://benchmark.tusimple.ai//
- Simonyan, K., and Zisserman, A. 2015. Very deep convolutional networks for large-scale image recognition. In ICLR.
- Chu, X.; Ouyang, W.; Wang, X.; et al. 2016. Crf-cnn: Modeling structured information in human pose estimation. In NIPS.
- Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L.2009. Imagenet: A large-scale hierarchical image database. In CVPR.