Car Speed Estimation using Contextual Convolutions

Varun Menon (vk2148), Astha Gupta (ag7982), Utkarsh Kumar (uk2012)

New York University

Abstract. In this paper, we aim to estimate the speed of a moving vehicle. The input to the model is a dash camera video feed and the output is the predicted speed. We use Gunnar Farneback optical flow as feature extractor and train a baseline Convolutional Neural Network model and a Contextual Convolution model with the same parameters. Our results show that contextual convolutions improve the performance of the speed detection system.Our work is available at: **https://github.com/alchemi5t/Contextual-speed-detection**.

Keywords: Contextual Convolutions, Deep Learning, Neural Networks

1 Introduction

Estimation of the speed of a moving vehicle using it's dash camera video feed serves as an important stepping stone in the challenge of developing autonomous vehicles. We attempt to solve this problem using Optical Flow and the novel method of Contextual Convolutions.

The use of Convolutional Neural Networks in image recognition tasks has revolutionized the field of Computer Vision. The ability of CNNs to learn features automatically have enabled their applications in many disciplines. In standard CNN architectures, the receptive field of the kernel is fixed since only one type of kernel with fixed size and dilation rate is used to perform convolutions. However, these architectures do not have the ability to integrate multi-scale contextual information that is vital for visual perception as shown by [1].

The data set used for out project was taken from Comma.Ai's Speed Challenge (https://github.com/commaai/speedchallenge). The data consists of a video consisting of 20400 frames, shot at 20 frames per second.

2 Related Work

2.1 Optical Flow

Optical Flow is a method used to estimate the apparent motion between the pixels of two or more images. Using optical flow, one can infer the movement

of real-world objects using video-based or image-based data. The concept uses the pixel intensity as its core feature. Since it's inception, optical flow has been utilised in a multitude of applications in the field of Computer Vision. It has enabled us to track the movement of multiple objects between various frames of a video.

Many techniques exist for the calculation of optical flow. In our study, we have made use of the Gunnar Farneback method [2]. The Gunnar Farneback is a dense optical flow algorithm, meaning that the changes in the intensities of all the pixels in an image are noted. First, the pixel intensity changes between two frames are highlighted. The flow vectors are then used to calculate the magnitude and direction of the motion. Finally, the magnitude of motion and the hue values are used to estimate the angle for the movement.

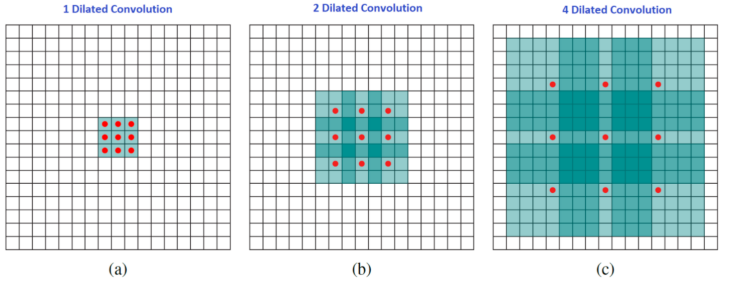
2.2 Dilated Convolution

Dilated convolution can be thought of as the standard discrete 2D convolution with padded kernels (pixel skipping). They are use to increase the effective receptive field of a convolution operation without increasing the number of parameters. Dilation=2 means the weights in the kernel matrix have a padding of (Dilation-1) around them, in this case, a padding of 1.

Let l be a dilation factor and let $*_l$ be defined as the dilated convolution operator. Dilated Convolution can then be defined as,

$$(F *_l k)(\mathbf{p}) = \sum_{\mathbf{s}+l\mathbf{t}=\mathbf{p}} F(\mathbf{s})k(\mathbf{t})$$

Fig. 1. Dilated kernels with 1, 2 and 4 levels of dilation shown by [3]



2.3 CoConv

Recently, [1] proposed contextual convolution (CoConv) which serves as a direct replacement of the standard convolution that can be used at any stage in

CNN architectures. The authors apply different dilation factors to sets of kernels within one convolution block: D = d1, d2, d3, . . . , dn. At every convolution layer, each set of dilated kernels allows the network to build over different levels of context. Kernels with lower dilations help with understanding local features vs kernels with larger dilations which help build a global understanding of the input volumes.

CoConv enable the neural network to learn features with different levels of context at each branch. This is done while the parameter count is preserved. The study shows that this leads to improvement in the network's ability to learn global features, leading to better detection, recognition and generation capabilities.

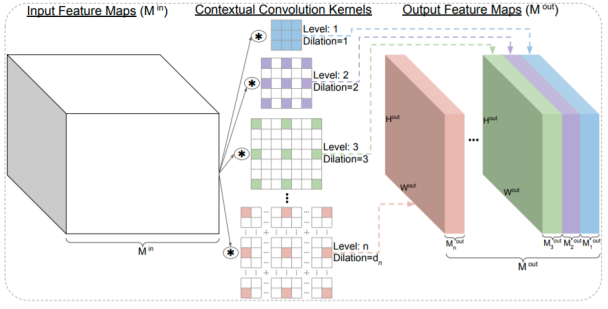


Fig. 2. Contextual Convolutions

3 Method

3.1 Pre-Processing Steps

We read the input video frame by frame, resize it to 240x320 and store frames to process it further. These RGB frames are converted to grayscale to compute the dense optical flow using Gunnar FarneBack technique. In this method, we consider all the pixels in the input pair of images and compute the intensity changes between the pair. The computed flow is then converted to HSV and then RGB for better interpretability. The final RGB representation of the optical flow is used as the input to our Convolutional Neural Network.

3.2 Baseline

We use Nvidia's CNN architecture as described in [4]. The first 4 layers are convolution layers followed by 2D maxpooling layers. All convolution layers have

3x3 kernels and 16, 32, 48 and 64 output channels respectively. This is followed up with a sequence of dense layers each with 1164, 100, 50, 10 and 1 hidden units each. We use the baseline model available at(https://github.com/MahimanaGIT/SpeedP

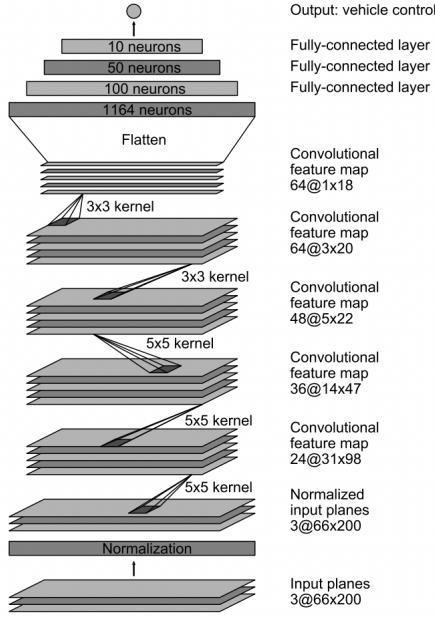


Fig. 3. Baseline Architecture, Nvidia CNN

3.3 Proposed Network Architecture

As proposed by [1], we use CoConv as a replacement layer for the standard convolution in the Baseline Architecture. We equally split the output channels for each layer by 4 and have dilation rates 1, 2, 3 and 4 for each branch. After computing the output for each branch, we concatenate the results and feed it forward to the next layers. We have maintained the kernel size and therefore the model size and number of parameters are preserved. We build this model using keras and make sure we preserve the model parameters and size to have a fair comparison.

4 Results

The results of our experiments show that Contextual Convolutions outperform vanilla convolutions in the estimation of vehicle speed. This is consistent with

our assumption that the addition of contextual information improves model performance for vision tasks.

Fig. 4 shows the Training loss curve and validation loss curve comparison between Nvidia's vanilla CNN architecture and our contextual variant of Nvidia's CNN.

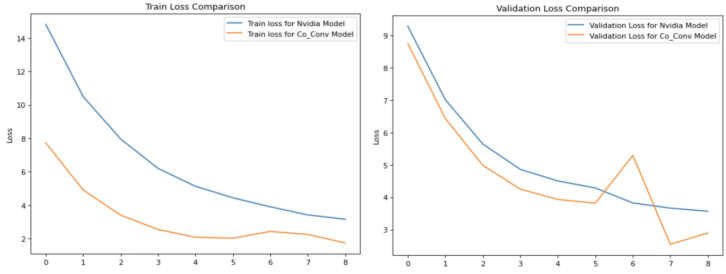


Fig. 4. Nvidia vs CoConv model

With the CoNvidia model, we get the lowest val loss measured by MSE of **2.5** vs Nvidia model whose MSE is **3.56**. Fig 5. shows the model prediction for the pre-processed input (optical flow). The color in the image dictates the direction and magnitude of the flow.

Fig. 5. Ground truth vs Prediction on input optical flow

4.1 Ablation study

To decide the percentage split of the dilated branches in each convolution layer, we ran multiple experiments with varying weights to each dilation rate. We test with standard convolution, CoConv with each dilation rate have equal split (referred to as CoConv model), CoConv with dilation 1 having 50% of the filters, dilation 2 having 25% of the filters and dilation 3 and dilation 4 having 12.5% each (referred to as CoConv_pyramid model). Similarly, we invert the percentage weights of CoConv_pyramid model to evaluate CoConv_Inverse_pyramid as well.

Fig 6. shows that all Contextual variants of the model do better than the Vanilla Nvidia CNN in both training and testing phases. We also note that all Contextual variants are more unstable, possibly due to the excess padding and max pooling not working well together.

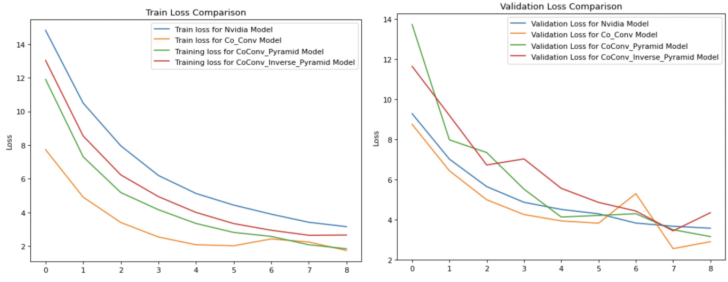


Fig. 6. Comparision of Coconv models and Vanilla Nvidia CNN

Model Name	**MSE**
Baseline	3.56
CoConv	**2.55**
CoConv Inverse Pyramid	3.43
CoConv Pyramid	3.15

Table 1. A comparison of the Mean Squared Errors

5 Discussion

In the future, this study can be extended by using Convolutional Neural Networks to estimate Optical Flow. Such methods provide a highly accurate result, but require large training sets. Recently, SpyNets [5] and FlowNets [6] have shown that such a method would enhance the features by providing a more

granular and specific optical flow computation.

The stability of the model could also be increased by removing maxpooling and having Stride 2 convolutions instead. This was not possible in the current implementations as the project was implemented using keras and keras does not support anything but stride=1 for any dilation rates not equal to 1.

References

1. Duta, I.C., Georgescu, M.I., Ionescu, R.T.: Contextual convolutional neural networks. arXiv preprint arXiv:2108.07387 (2021)
2. Farnebäck, G.: Two-frame motion estimation based on polynomial expansion. Volume 2749. (06 2003) 363–370
3. Yu, F., Koltun, V.: Multi-scale context aggregation by dilated convolutions (2016)
4. Bojarski, M., Testa, D.D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L.D., Monfort, M., Muller, U., Zhang, J., Zhang, X., Zhao, J., Zieba, K.: End to end learning for self-driving cars (2016)
5. Ranjan, A., Black, M.J.: Optical flow estimation using a spatial pyramid network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2017)
6. Fischer, P., Dosovitskiy, A., Ilg, E., Häusser, P., Hazırbaş, C., Golkov, V., van der Smagt, P., Cremers, D., Brox, T.: Flownet: Learning optical flow with convolutional networks (2015)