AN EFFICIENT METHOD FOR GENERIC DSP IMPLEMENTATION OF DILATED CONVOLUTION

Harinarayanan.E.V, Sachin Ghanekar

Cadence Design Systems, Pune, India
{hariev, ghanekar}@cadence.com

ABSTRACT

Dilated convolution is a well-known technique used in neural networks algorithms in AI/ML applications to increase receptive-field under analysis. Dilated convolution layer has an inherent property of capturing wider context in an image and long-term temporal characteristics in an audio signal. In this paper we propose a scheme that allows efficient/generic implementation of 2D dilated convolution and stride on typical DSPs where the instruction sets are well tuned for standard 1D and 2D filtering and convolution operations. The paper analyzes and morphs the basic structures of dilated convolution computations using a decomposition method similar to polyphase decomposition to map it to a friendly and efficient standard convolution operation. The method also naturally extends to include *stride* as a feature for dilated convolution. Using this scheme, we publish results on Tensilica's HiFi5 platform achieving a computational cycle reduction in the order of 30X and memory reduction in the order of 150X against a standard implementation for dilation. We have also made the code available as a part of Cadence's NN Library git hub code base on HiFi5 processor.

***Index Terms*—** CNN, 2D convolution, dilation, stride

1. INTRODUCTION

The use of CNNs has shown promising results in the areas of computer vision, image recognition, contextual analysis, image segmentation and so on. Since the introduction of Alexnet in 2012 the CNNs were explored with various architectural modification and different such CNN based networks were proposed. Some of the notable networks include GoogleNet[2], Resnet[5], VGG[6], dilated convolution, one such variant used for semantic segmentation resulted in improved accuracy as noted by Fisher Yu and Vladen Koltun[4]. The main idea behind exploration of dilated convolution in image application is its exponential expansion of the receptive field without loss of resolution or coverage. Dilated convolution finds applications in other fields including speech. In Wavenet [8] the concept of dilated convolution with the exponential increase of depth and to cover multiple timesteps enables to capture the higher order long term characteristics of the signal. In this paper, we concentrate on the implementation aspect of dilation/stride and thereby explore a generic method to efficiently implement the inference section of a 2D dilated convolution without any constraint on the size of the input or filter or dilation factor or stride. In this proposal an existing flexible and efficient 2D convolution implementation with stride support forms the basic building block to implement 2D-dilated convolution with stride. The 2D-dilated convolution is equivalently represented as several smaller 2D convolutions with appropriate matrix slicing and re-ordering. The proposed method's computational complexity is immune to dilation factor and hence the overall complexity of the implementation remains same irrespective of dilation factor. The same scheme used for dilation is also extended to support stride feature. A joint scheme to efficiently implement both dilation and stride in a single framework is proposed.

2. DILATED CONVOLUTION AND STRIDE

2D convolution applies a 2D filter on a 2D input to produce an output which is 2D in shape. Let, $I(x_1, y_1)$ be a real valued 2D input and $K(x_2, y_2)$be a real valued 2D kernel then the convolution is defined as

$$Y(x_3, y_3) = \sum_{x_1+x_2=x_3} \sum_{y_1+y_2=y_3} I(x_1, y_1) * K(x_2, y_2) \quad (1)$$

The above equation is a generic definition for 2D convolution. At the edges of the input matrix the output values of $Y(x_3, y_3)$ would be computed through partial overlapped convolutions between input and kernel. Generally, the values of output where there is a complete overlap between input and kernel are used for further computations. For complete overlapping cases the above equation can be written as,

$$Y(x_3, y_3) = \sum_{0 \le x_2 < K_h} \sum_{0 \le y_2 < K_w} I(x_2 + x_3, y_2 + y_3) * F(x_2, y_2) \quad (2)$$

Where, $0 \le y_3 < I_w - K_w + 1$ and $0 \le x_3 < I_h - K_h + 1$. $\{I_w, I_h\}$ represent the input size in width and height dimensions respectively. Similarly, $\{K_w, K_h\}$ represent the kernel size in width and height dimensions respectively. The filter F is a flipped version of K in both height and width dimensions and this is an equivalent representation of convolution expressed in terms of correlation. Such representation of 2D convolution is used in some of the neural network libraries for ex., TensorFlow™ Lite Micro uses conv2D under a similar definition. Dilated convolution uses skipped values of input to compute the output. Dilated convolution is mathematically represented as given in (3).

$$Y(x_3, y_3) = \sum_{0 \le x_2 < K_h} \sum_{0 \le y_2 < K_w} I\begin{pmatrix} x_2 * d_h + x_3 , \\ y_2 * d_w + y_3 \end{pmatrix} * F(x_2, y_2) \tag{3}$$

Where, d_h *and* d_w represent dilation factors in height and width dimensions respectively. The new limits on the range of output Y is, $0 \le y_3 < I_w - K_{w\_D} + 1$ and $0 \le x_3 < I_h - K_{h\_D} + 1$; Where, $K_{w\_D} = K_w + (K_w - 1) * (d_w - 1)$; $K_{h\_D} = K_h + (K_h - 1) * (d_h - 1)$; Note that the output dimensions of dilated convolution is $(I_h - K_{h\_D} + 1)$ X $(I_w - K_{w\_D} + 1)$ as against $(I_h - K_h + 1)$ X $(I_w - K_w + 1)$ in case of a standard 2D convolution. To reduce the output dimension further, it is common to introduce stride factors in 2D dilated convolution equation as below as in (4)

$$Y(x_3, y_3) = \sum_{0 \le x_2 < K_h} \sum_{0 \le y_2 < K_w} I\begin{pmatrix} x_2 * d_h + x_3 * s_h , \\ y_2 * d_w + y_3 * s_w \end{pmatrix} * F(x_2, y_2) \tag{4}$$

In (4), the number of output points are reduced as a result of accounting stride in the equation, $0 \le y_3 < \lfloor (I_w - K_{w\_D})/s_w \rfloor + 1$ and $0 \le x_3 < \lfloor (I_h - K_{h\_D})/s_h \rfloor + 1$ where, $\{s_w, s_h\}$ are the stride in width and height dimensions respectively and $\lfloor \ \rfloor$ represent floor operator.

2.1. Basic Structure

In the definition of 2D convolution in (2), (3) and (4) there exist corner cases and assumptions which are addressed here. In (2) the range of y_3 is stated as per the limits, $0 \le y_3 < I_w - K_w + 1$. The general assumption here is input dimension i.e., I_w is greater than or equal to kernel dimension i.e, K_w. This is indeed true for all practical use-cases. The above assumption is true in both width and height dimension. This assumption also holds for dilated and dilated-with-stride convolution. For cases where this is not true, the input is assumed to be appropriately zero padded in the required dimension i.e., width / height to satisfy the above statement. The dimensions of input i.e., I_w& I_h herein considered such that $I_w \ge K_w$ and $I_h \ge K_h$. The same is true for dilated convolution $I_w \ge K_{w\_D}$ and $I_h \ge K_{h\_D}$.

3. A SCHEME FOR DILATED CONVOLUTION

One way to visualize dilated convolution is to dilate the kernel itself by injecting zeros in the kernel matrix by a factor of dilation and perform standard convolution. This method is referred as atrous convolution or convolution with holes in literature [12] and is also a generalized implementation method for dilated convolution. This method is investigated as a part of various AI/ML networks [13][14]. We refer to this method as zero-injection (ZI) method herein. Dilated convolution or convolution with dilated kernel both yield same result. ZI method helps in visualizing a framework to implement dilated convolution. [14] explains a method where the dilated convolution is split into normal convolution and re-interlaced. Mathematical extension of the method along with a unified framework support for stride is explained in the further sections. Let dilated convolution in (3) be represented as given in (5).

$$Y = I \ (\emptyset_{d_h, d_w}) \ F \tag{5}$$

Where, $\emptyset_{d_h, d_w}$ represent dilated convolution operator along with dilation factors. Consider (6) for convolution, this represents a sub-set of final output points as given in (3). The input points that participate also is a subset of the total input. Equation (6) forms the basis for modelling a dilated convolution into number of smaller standard convolutions.

$$Y_{n_1 n_2 d_h, d_w}(x_n^{\sim}, y_n^{\sim}) = \sum_{0 \le x_2 < K_h} \sum_{0 \le y_2 < K_w} I\begin{pmatrix} n_1 + d_h * (x_2 + x_n^{\sim}) , \\ n_2 + d_w * (y_2 + y_n^{\sim}) \end{pmatrix} * F(x_2, y_2) \tag{6}$$

Where, $0 \le x_n^{\sim} < \lceil (I_h - n_1)/d_h \rceil - K_h + 1$ and $0 \le y_n^{\sim} < \lceil (I_w - n_2)/d_w \rceil - K_w + 1$; $0 \le n1 < d_h$ and $0 \le n2 < d_w$. The above equation represents a subset of output values w.r.t., (3) for a specific value of $n1, n2$ and $\lceil \ \rceil$ represent ceil operation. The above equation can be modified to be suitably represented as regular convolution as given in (7). Equation (7) is similar to (3) and can equivalently be represented as a simple convolution. The modified input $I_{n_1 n_2 d_h d_w}$ in (7) is a subset of the original input I in (3). The bridge equation between these two inputs is given in (8).

$$Y_{n_1 n_2 d_h, d_w}(x_n^{\sim}, y_n^{\sim}) = \sum_{0 \le x_2 < K_h} \sum_{0 \le y_2 < K_w} I_{n_1 n_2 d_h d_w}\begin{pmatrix} x_2 + x_n^{\sim}, \\ y_2 + y_n^{\sim} \end{pmatrix} * F(x_2, y_2) \tag{7}$$

$$I_{n_1 n_2 d_h d_w}(x, y) = I(n_1 + d_h * x, n_2 + d_w * y) \tag{8}$$

Where, $0 \le x < \lceil (I_h - n_1)/d_h \rceil$ and $0 \le y < \lceil (I_w - n_2)/d_w \rceil$. All the values of the output i.e., $Y(x_3, y_3)$ in (3) can equivalently be represented using $Y_{n_1 n_2 d_h, d_w}(x_n^{\sim}, y_n^{\sim})$. The bridge equation between (3) and (7) is as below,

$$Y(n_1 + x_n^{\sim} * d_h, n_2 + y_n^{\sim} * d_w) = Y_{n_1 n_2 d_h, d_w}(x_n^{\sim}, y_n^{\sim}) \tag{9}$$

Equation (7) (8) (9) are the basic set of equations for computation of dilation without incurring any of the extra cost of zero multiplication arising from a zero-injection method. With this approach the problem of 2D dilated convolution is expressed as a collective set of smaller 2D standard convolutions.

4. A SCHEME FOR STRIDE

The skipped values of convolved output by a pre-defined factor is given by stride. The skip factor is the stride factor. This section describes how the above dilation equations (7)(8)(9) are modified with the constraints of the stride parameter and thereby, observe the modified equations for their mathematical properties to map them on an efficient implementation scheme. Equation (10) represents stride output $Y_{s_h s_w}$ from a fully convolved output Y. Computing all the values of output and skipping indices by stride factor post-computation would be a sub-optimal solution. We propose a solution within the framework of dilation. From (10) and (9) it could be observed that only the co-ordinate values which satisfy (11) are the output points of interest.

$$Y_{s_h s_w}(x, y) = Y(s_h * x, s_w * y) \tag{10}$$

$$Y(s_h * x, s_w * y) = Y(n_1 + x_n^{\sim} * d_h, n_2 + y_n^{\sim} * d_w) \tag{11}$$

Or equivalently the co-ordinate points of interest from (11) which represent the strided output are as defined in (12),

$$< x, y > = < \frac{n_1 + x_n^{\sim} * d_h}{s_h}, \frac{n_2 + y_n^{\sim} * d_w}{s_w} > \tag{12}$$

Problem statement:
For a given offset pair $< n_1, n_2 >$ and the value pair $< x_n^{\sim}, y_n^{\sim} >$ as given in (7) resulting in positive integer values of $< x, y >$ as given in (12) are the co-ordinates of interest. Note: It can be observed that $x_n^{\sim}$ and $y_n^{\sim}$ are two independent co-ordinates and does not need a joint solution.

Alternate problem statement:

- *Statement1*: For a given offset pair $< n_1, n_2 >$ find the minimum value of co-ordinates $< x_n^{\sim}, y_n^{\sim} >$ say, $< x_{n\_min}^{\sim}, y_{n\_min}^{\sim} >$ such that (12) is satisfied
- *Statement2*: From the initial values $< x_{n\_min}^{\sim}, y_{n\_min}^{\sim} >$ for a given $< n_1, n_2 >$ find successive value of $< x_n^{\sim}, y_n^{\sim} >$ which satisfy (12)

Solution:

- *Postulate*: Assume Statement1 is true i.e., for a given $< n_1, n_2 >$ there exist a $< x_{n\_min}^{\sim}, y_{n\_min}^{\sim} >$ such that (12) is satisfied. We try to find solution for Statement2 based on this postulate i.e., to find the successive values of $< x_n^{\sim}, y_n^{\sim} >$ after the initial value $< x_{n\_min}^{\sim}, y_{n\_min}^{\sim} >$..
- *Inference:* Let the next successive value of $< x_n^{\sim}, y_n^{\sim} >$ after $< x_{n\_min}^{\sim}, y_{n\_min}^{\sim} >$ satisfying (12) be $< x_{n\_min}^{\sim} + \Delta_x, y_{n\_min}^{\sim} + \Delta_y >$. Inserting these values in (12) results in (13).

$$< x, y > = < \frac{n_1 + (x_{n\_min}^{\sim} + \Delta_x) * d_h}{s_h}, \frac{n_2 + (y_{n\_min}^{\sim} + \Delta_y) d_w}{s_w} > \tag{13}$$

$$= < \frac{n_1 + (x_{n\_min}^{\sim} * d_h)}{s_h} + \frac{\Delta_x * d_h}{s_h}, \frac{n_2 + (y_{n\_min}^{\sim} * d_w)}{s_w} + \frac{\Delta_y * d_w}{s_w} >$$

Based on the postulate statement both co-ordinates $< \frac{n_1 + (x_{n\_min}^{\sim} * d_h)}{s_h}, \frac{n_2 + (y_{n\_min}^{\sim} * d_w)}{s_w} >$ are integer components. Therefore, for $< x, y >$ to be integer, $\frac{\Delta_x * d_h}{s_h}$ & $\frac{\Delta_y * d_w}{s_w}$ should result in integer values. It can be deduced that minimum value of Δ_x & Δ_y to contribute an integer value is,

$$\Delta_x, \Delta_y = \frac{s_h}{GCD(s_h, d_h)}, \frac{s_w}{GCD(s_w, d_w)} \tag{14}$$

Based on (14) it can be concluded that if there exist a minimum value $< x_{n\_min}^{\sim}, y_{n\_min}^{\sim} >$ the successive values to satisfy (12) would be at $< x_{n\_min}^{\sim} + \Delta_x, y_{n\_min}^{\sim} + \Delta_y >$. Extending the same logic it can be stated, there exists valid values at every step of Δ_x & Δ_y from the initial value.

Conclusion: From (14) for a given n_1, n_2 if there exist $x_{n\_min}^{\sim}$, $y_{n\_min}^{\sim}$ **then Δ_x, Δ_y is the (new) stride value for the convolutions on the sub-matrices**. Equation (7) $Y_{n_1 n_2 d_h, d_w}$ would now accommodate stride $Y_{n_1 n_2 d_h, d_w s_h s_w}$. Also, from (13) it can be observed that the equality holds good with a periodicity factor of Δ_x & Δ_y iff, $< x_{n\_min}^{\sim}, y_{n\_min}^{\sim} >$ exist. Implying, it is sufficient to check (12) by running values iteratively from $0\ to\ \Delta_x - 1$ & $0\ to\ \Delta_y - 1$ for $x_n^{\sim}\ and\ y_n^{\sim}$ respectively, to check if the first value $< x_{n\_min}^{\sim}, y_{n\_min}^{\sim} >$ does exist. In this iterative process if the equality does not hold for either of $< x, y >$ for a given $< n_1, n_2 >$ then that convolution output $Y_{n_1 n_2 d_h, d_w}$ need not be computed as none of its values contribute towards dilated-stride output. Such omissions of submatrix calculation saves considerable computation from an implementation perspective. Also to note, in the input $I_{n_1 n_2 d_h, d_w}$ the first points of convolution begins at co-ordinate index $< x_{n\_min}^{\sim}(n_1), y_{n\_min}^{\sim}(n_2) >$ (an index of n_1 & n_2 are added as they are different for each sub-matrix) and all values in the co-ordinates lesser than this is not of interest as it does not contribute towards the final output. This requires modification in the input slicing mechanism. The input sliced matrix $I_{n_1 n_2 d_h, d_w}$ is realtered to accommodate stride related changes before performing sub-matrix convolution $I_{n_1 n_2 d_h, d_w s_w s_h}$. Some interesting cases arise when dilation and stride are co-prime to or are factors of each other. These examples are covered in plots shown in Fig. 1.

4.1. *R*e-ordering stride output: Post sub-matrix convolution

Re-ordering of dilation output without considering stride is represented in (9). Based on the above conclusion, the sub-matrix convolution as given in (7) will be implemented with a stride factor $\frac{s_h}{GCD(s_h,d_h)}$ & $\frac{s_w}{GCD(s_w,d_w)}$ for those n_1&n_2 pairs where $x_{\tilde{n}\_min}$ & $y_{\tilde{n}\_min}$ exist. The output, post-convolution must be re-ordered. Equation (15) gives the re-ordering formula and it needs to be computed only for values of n_1&n_2 for which $x_{\tilde{n}\_min}(n_1)$ & $y_{\tilde{n}\_min}(n_2)$ exist. Observing (15), the right side of the equation $Y_{n_1 n_2 d_h, d_w s_h s_w}$ are result of sub-matrix convolutions. Examining the indices on the left of equality i.e., Y, the height component index i.e., $\frac{h_{offset}(n_1)+i_1 * R_h}{s_h}$ has two parts i.e., $\frac{h_{offset}(n_1)}{s_h}$ and $\frac{i_1 * R_h}{s_h}$. The first component is a constant offset component for a given sub-matrix with index n_1. The second component is an output stride component. An efficient implementation of standard 2D convolution to accommodate output stride would save all the cost involved in this re-ordering step.

$$Y\left(\frac{h_{offset}(n_1) + i_1 * R_h}{s_h}, \frac{w_{offset}(n_2) + i_2 * R_w}{s_w}\right) = Y_{n_1 n_2 d_h, d_w s_h s_w}(i_1, i_2) \tag{15}$$

$$\begin{gathered} Where, \\ n_1 = 0,1,2, \dots, d_h - 1 \\ n_2 = 0,1,2, \dots, d_w - 1 \\ h_{offset}(n_1) = n_1 + x_{\tilde{n}\_min}(n_1) * d_h \\ w_{offset}(n_2) = n_2 + y_{\tilde{n}\_min}(n_2) * d_w \\ R_h = \frac{s_h}{GCD(d_h, s_h)} * d_h \\ R_w = \frac{s_w}{GCD(d_w, s_w)} * d_w \end{gathered}$$

5. RESULTS

We perform our evaluation in a framework compatible with TensorFlow™ Lite Micro (TFLM) data format where, the input representation of matrices are in (N,H,W,C) default format. 'H', 'W' & 'C' represent height, width and channels. 'N' represents number of matrices. While applying convolution between an input and kernel matrix with channels, every point multiplication in 2D (2) is translated to a dot product in the channel dimension. In case of dilation and stride, the channel components are not affected by dilation and stride factors. The proposed method of decomposition is a new technique to encompass dilation and stride and is compared with base ZI method for varying factors of dilation and stride. In ZI method, the kernel is dilated and only the output points of interest are computed for a chosen stride factor. The cycles and memory required for dilating the kernel is not considered in the results reported. Both the methods are implemented on Cadence Tensilica HiFi5 processor. A DSP friendly circular buffer-based design as mentioned in [15] is used for implementing standard convolution. All the performance specification measurements are carried out on a cycle-accurate HiFi5 simulator assuming zero memory wait states. This is equivalent to running with all code and data in local memory. Such a model is chosen to decouple the effects of memory architectures due memory penalty and cache misses on performance numbers. The HiFi5 core is used with Neural Network extension along with Xtensa tools with a moderate level of intrinsic optimization for the kernels, with further scope of optimization for specific use-cases.

Dilation Factor	Decomposition Method		Zero Insertion (ZI) Method	
(Stride=2)	Cycles ($X10^6$)	Scratch Memory (KB)	Cycles ($X10^6$)	Scratch Memory (KB)
2	1.94	6.14	4.89	20.20
4	2.01	3.14	10.01	36.33
8	2.14	1.64	26.54	68.58
16	2.32	0.89	69.70	133.08

Table 1. Cycles and Memory comparison between Proposed decomposition method and ZI method

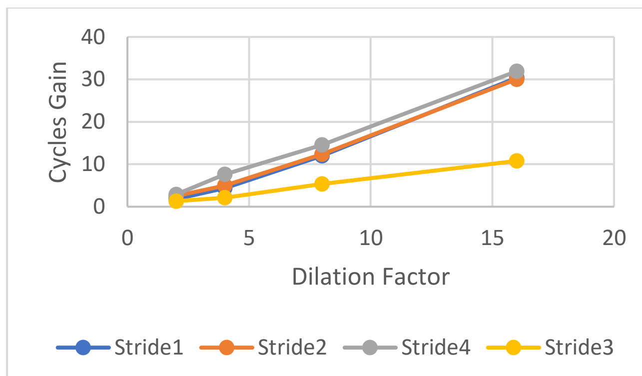

Fig. 1. Measured ratio of cycle complexity between Decomposition and ZI method

Use-cases of dilation factors include {2,4,8,16} and stride factors {1,2,3,4}. Table 1. records cycles and memory complexity of both the methods for various dilation factors with stride=2. In decomposition method, irrespective of dilation factor, the number of cycles consumed does not vary much because the overall computational complexity of the proposed algorithm is independent of dilation factor. On the other hand, the ZI method's complexity is dependent on the product of dilation factor and kernel dimension. Fig.1. plots the ratio of cycles(Gain) between the two methods for various dilation factors with fixed stride value. An improvement of 30X can be observed for dilation factor of 16. The scratch memory requirement [15] also reduces for increasing dilation factor, by ~150X (133.08/0.89 = 149.5) for dilation factor of 16 as in Table 1., this is because the decomposition method reduces larger convolutions into smaller std. convolution. The above discussed implementation of dilation and stride for 2D convolution is also made available in GitHub as a part of Cadence's Neural Network Library on HiFi5 processor [11].

12. REFERENCES

[1] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton., "ImageNet Classification with Deep Convolutional Neural Networks", *Advances in neural information processing systems,* 2012

[2] Szegedy, Christian, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich., "Going deeper with convolutions", *In Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1-9, 2015

[3] M. P. Véstias, "A survey of convolutional neural networks on edge with reconfigurable computing", *Algorithms,* vol. 12, no. 8, https://www.mdpi.com/1999-4893/12/8/154, 2019

[4] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions", in *ICLR*, 2016

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, "Deep Residual Learning for Image Recognition", *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016

[6] K. Simonyan and A. Zisserman "Very deep convolutional networks for large-scale image recognition", *arXiv preprint* arXiv:1409.1556, 2014

[7] S. Mehta, M. Rastegari, A. Caspi et al., "ESPNet: efficient spatial pyramid of dilated convolutions for semantic seg-mentation", *arXiv*, https://arxiv.org/abs/1803.06815, 2018

[8] van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., and Kavukcuoglu, K. "Wavenet: A generative model for raw audio", *arXiv preprint* arXiv:1609.03499, 2016

[9] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller. "Striving for simplicity: The all convolutional net", *arXiv preprint* arXiv:1412.6806, 2014

[10] Yanxiong Li , Mingle Liu , Konstantinos Drossos , Tuomas Virtanen., "Sound Event Detection Via Dilated Convolutional Recurrent Neural Networks", *in ICASSP*, 2020

[11] Dilated convolution implementation as a part of Cadence's HiFi5 based Neural Network Library: https://github.com/foss-xtensa/nnlib-hifi5/blob/master/xa_nnlib/algo/kernels/cnn/hifi5/xa_nn_conv2d_std_sym8sxasym8s.c

[12] M. Holschneider, R. Kronland-Martinet, J. Morlet, and P. Tchamitchian. "A real-time algorithm for signal analysis with the help of the wavelet transform", *in Wavelets*, pages 286–297. Springer, 1989.

[13] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *arXiv,* preprint arXiv:1606.00915, 2016.

[14] Zhengyang Wang and Shuiwang Ji, "Smoothed Dilated Convolutions for Improved Dense Prediction", *IEEE transactions on pattern analysis and machine intelligence, vol. XX, NO. X,* April 2019

[15] Ananda Sarangaram Tharma Ranga Raja, Prasad Nikam, ND Divyakumar, Himanshu Singhal, Vijay Pawar, Sachin P. Ghanekar, SYSTEMS AND METHODS OF BUFFERING AND ACCESSING INPUT DATA FOR CONVOLUTION COMPUTATIONS United States Patent and Trademark Office application 16/433,533.