

A Reconfigurable Streaming Deep Convolutional Neural Network Accelerator for Internet of Things

Li Du, *Member, IEEE*, Yuan Du, Yilei Li and Mau-Chung Frank Chang, *Fellow, IEEE*

The main contribution of this paper include:

1. A CNN accelerator design using streaming data flow to achieve optimal energy efficiency.
2. An interleaving architecture to enable parallel computing for multiple output features without SRAM input bandwidth increment.
3. A methodology to decompose large-sized filter computation to be many small-sized filter computation, achieving high reconfigurability without adding additional hardware penalty.
4. A supplementary pooling block that can support pooling function while the main engine serves for CNN computation.
5. A prototype design with FPGA verification, which can achieve a peak performance of 152 GOPS and energy efficiency of 434 GOPS/W.

A. Filter Decomposition

To minimize the hardware resource usage, a filter decomposition algorithm is proposed to compute any large kernel-sized($>3*3$) convolution through using only $3*3$ -sized CU.

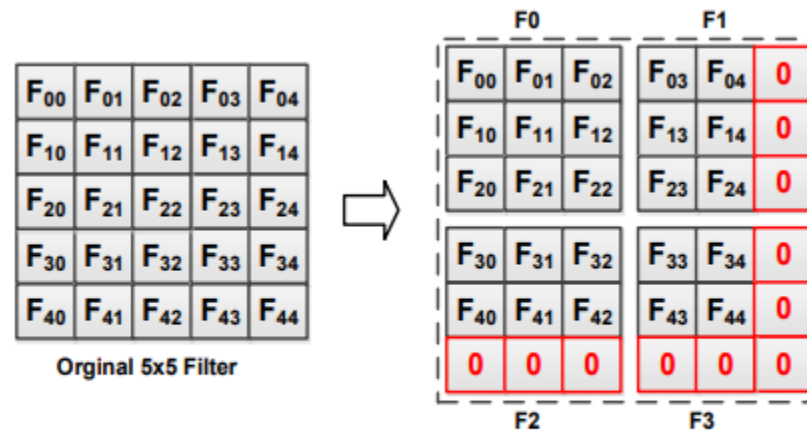


Fig.5 An 5x5 Filter decomposed into four 3x3 sub filter. F0, F1, F2, F3's shift address are (0,0), (0,3), (3,0), (3,3).

A. Filter Decomposition

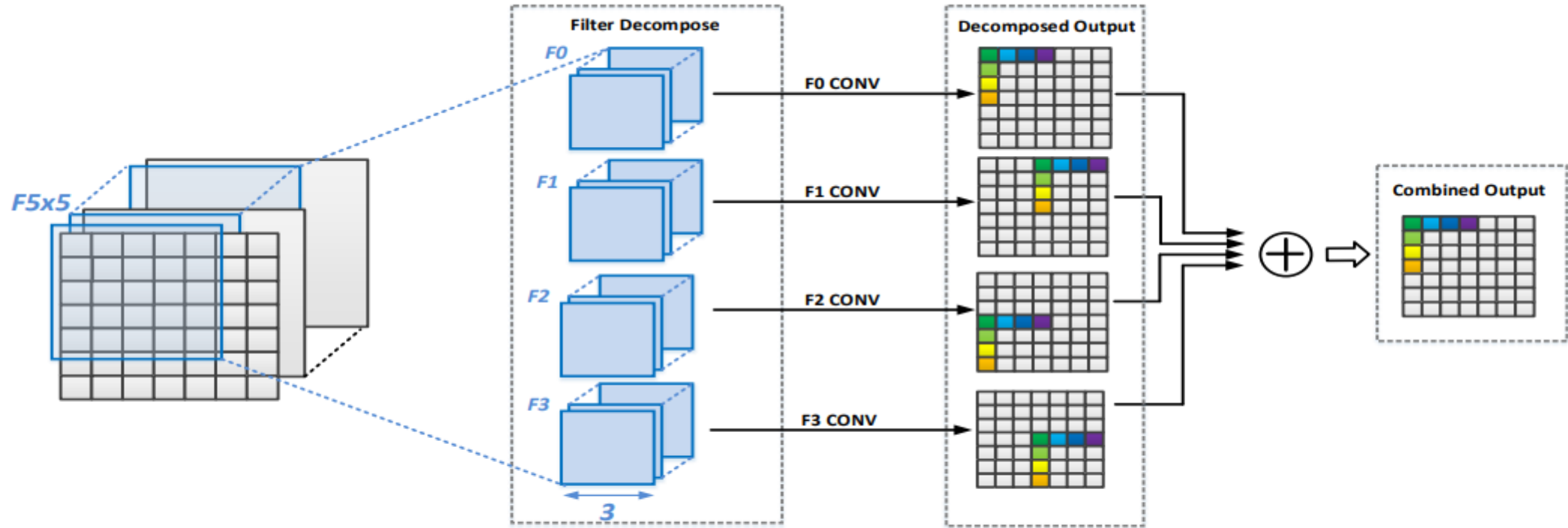
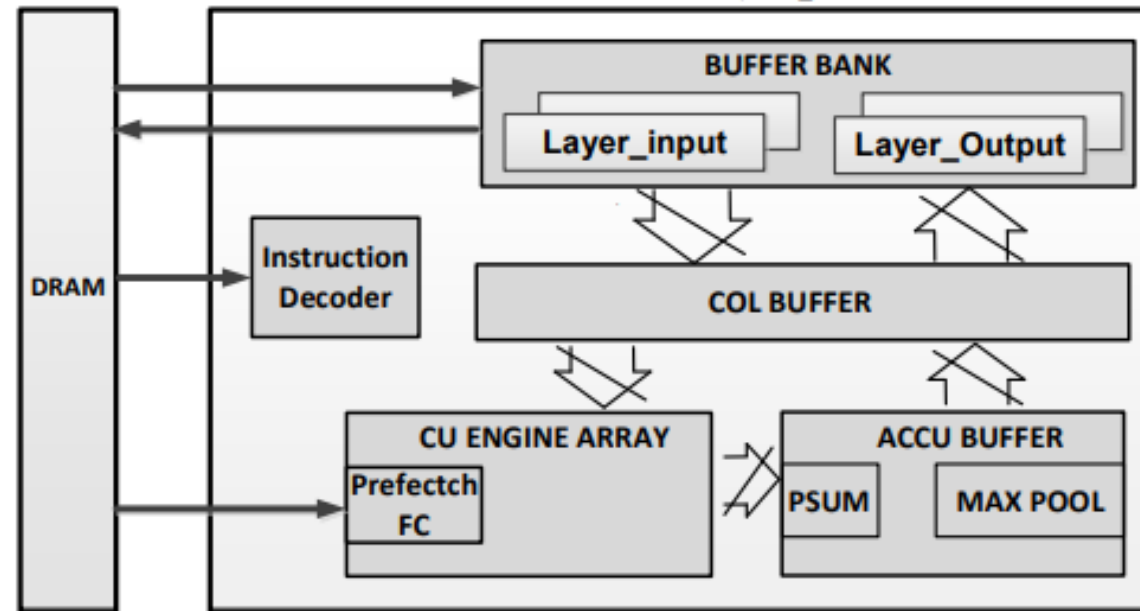


Fig. 6 Filter decomposition technique to compute a 5x5 filter on the 7x7 image. The Filter is decomposed into $F0$, $F1$, $F2$, $F3$, generating four sub-images. The sub-images are summed based on their filter's shift address. Same color's pixels will be added together to generate the corresponding pixels in the output image.

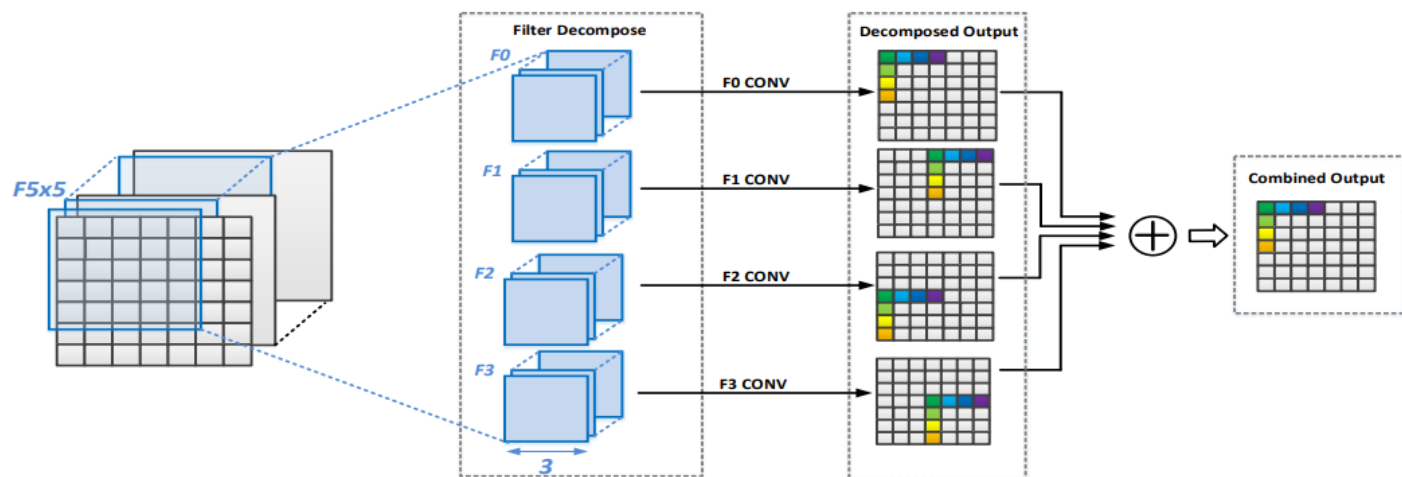
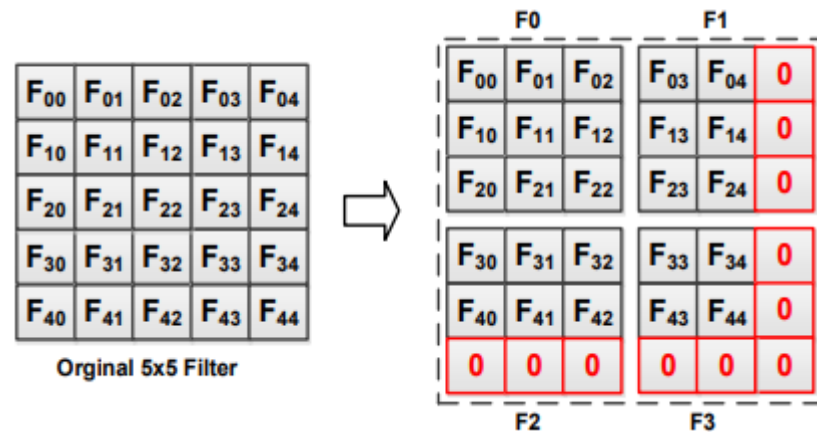
Overall Architecture of the CNN Accelerator



A. Filter Decomposition

$$\begin{aligned}
 F_{3K}(a, b) &= \sum_{i=0}^{3K-1} \sum_{j=0}^{3K-1} f(i, j) \times I_i(a + i, b + j) \\
 &= \sum_{i=0}^{K-1} \sum_{j=0}^{K-1} \sum_{l=0}^2 \sum_{m=0}^2 f(3i + l, 3j + m) \times I_i(a + 3i + l, b + 3j + m) \\
 &= \sum_{i=0}^{K-1} \sum_{j=0}^{K-1} F_{3_i_j}(a + 3i, b + 3j)
 \end{aligned} \tag{5}$$

$$\begin{aligned}
 F_{3_i_j}(a, b) &= \sum_{m=0}^2 \sum_{l=0}^2 f(3i + l, 3j + m) \times I_i(a + 3i + l, b + 3j + m) \\
 0 \leq i < K - 1; 0 \leq j < K - 1;
 \end{aligned} \tag{6}$$



The Output Size of Convolution

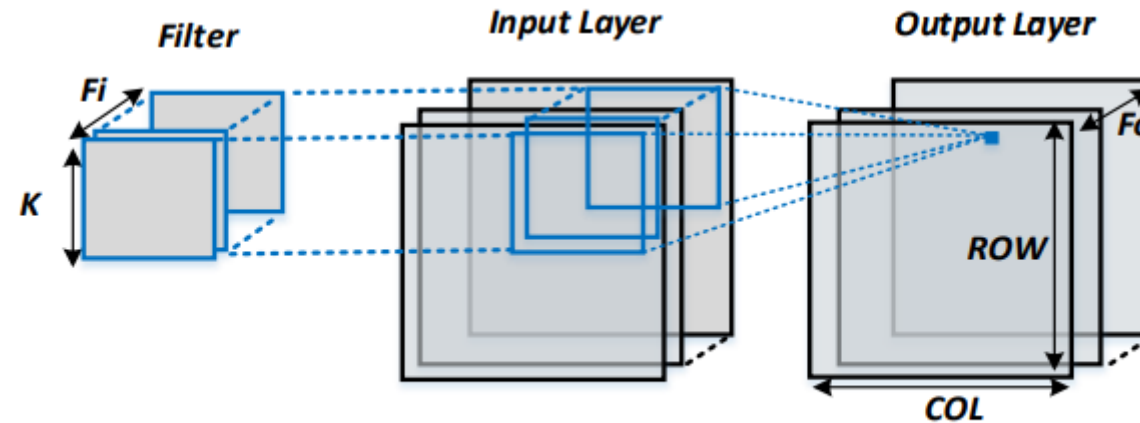
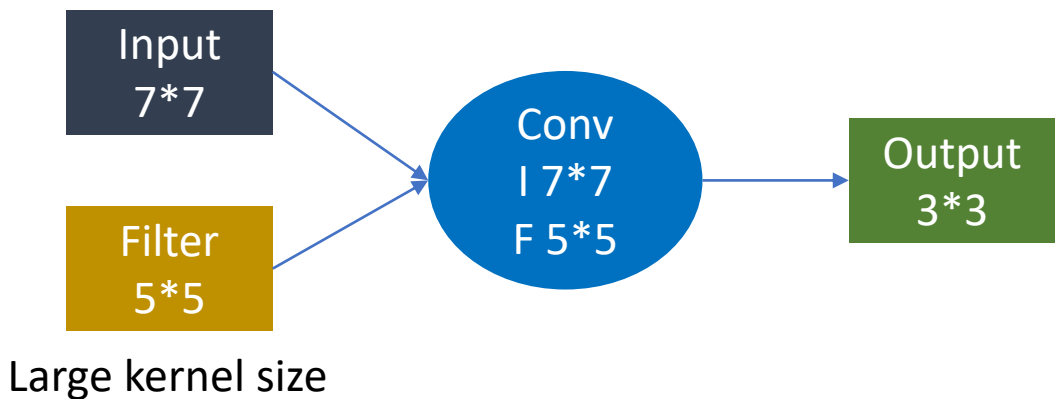


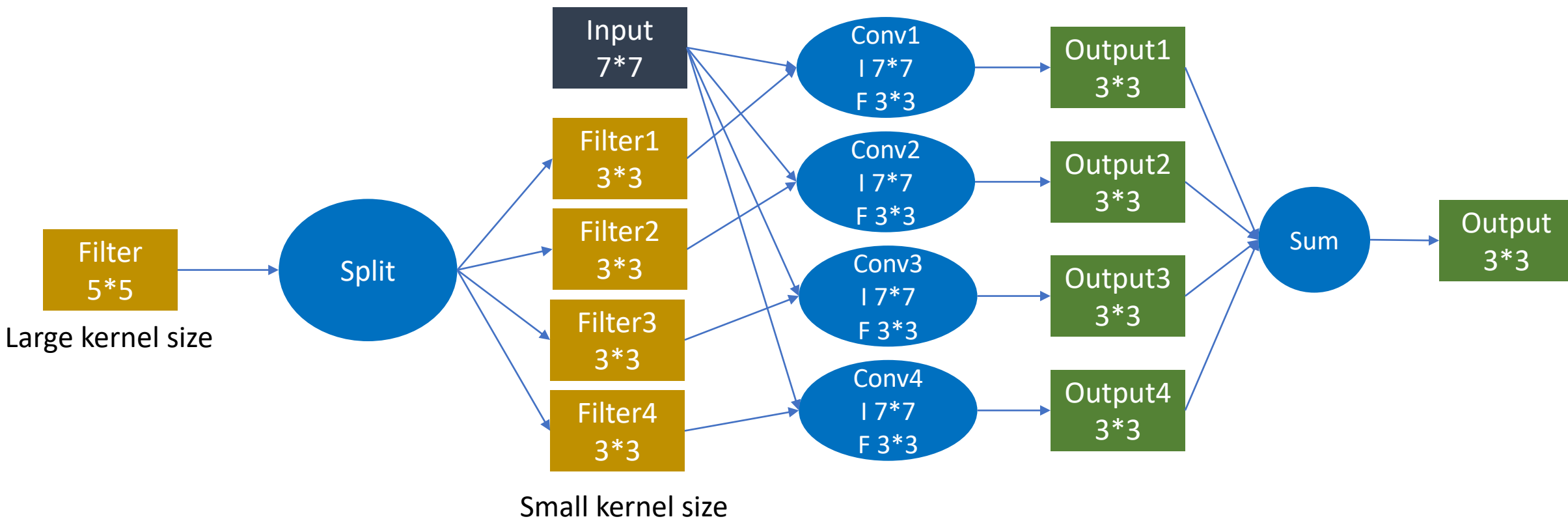
Fig.1 Example of computation of a CNN layer.

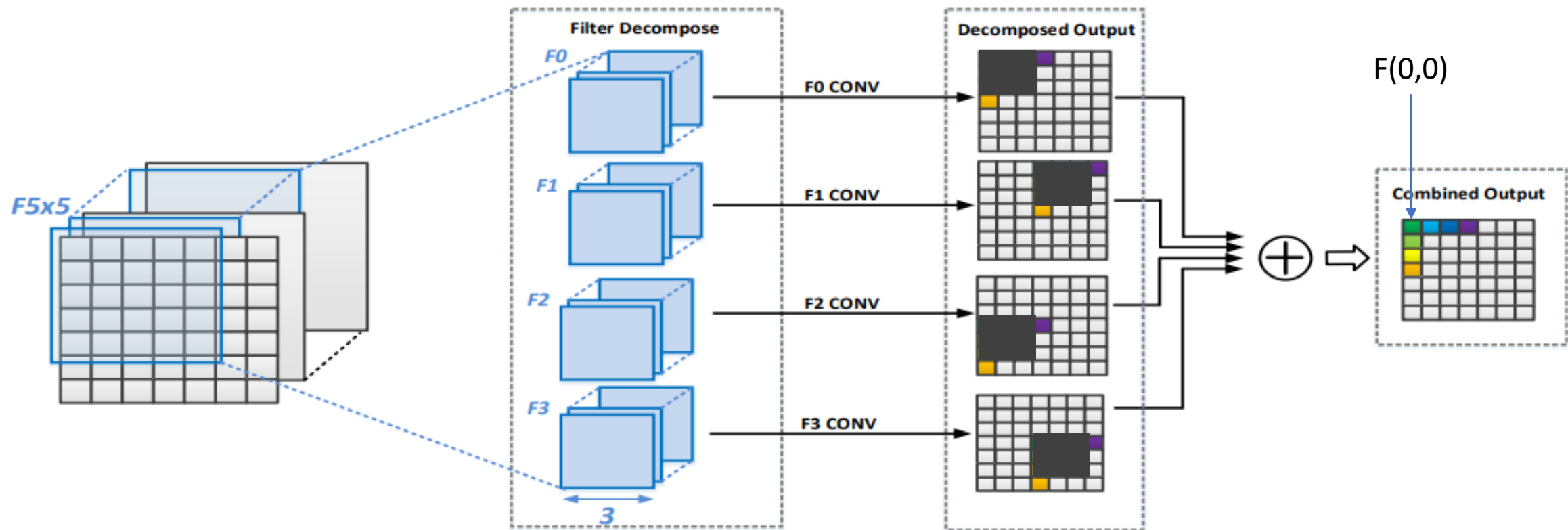
Output height = (Input height + padding height top + padding height bottom - kernel height) / (stride height) + 1

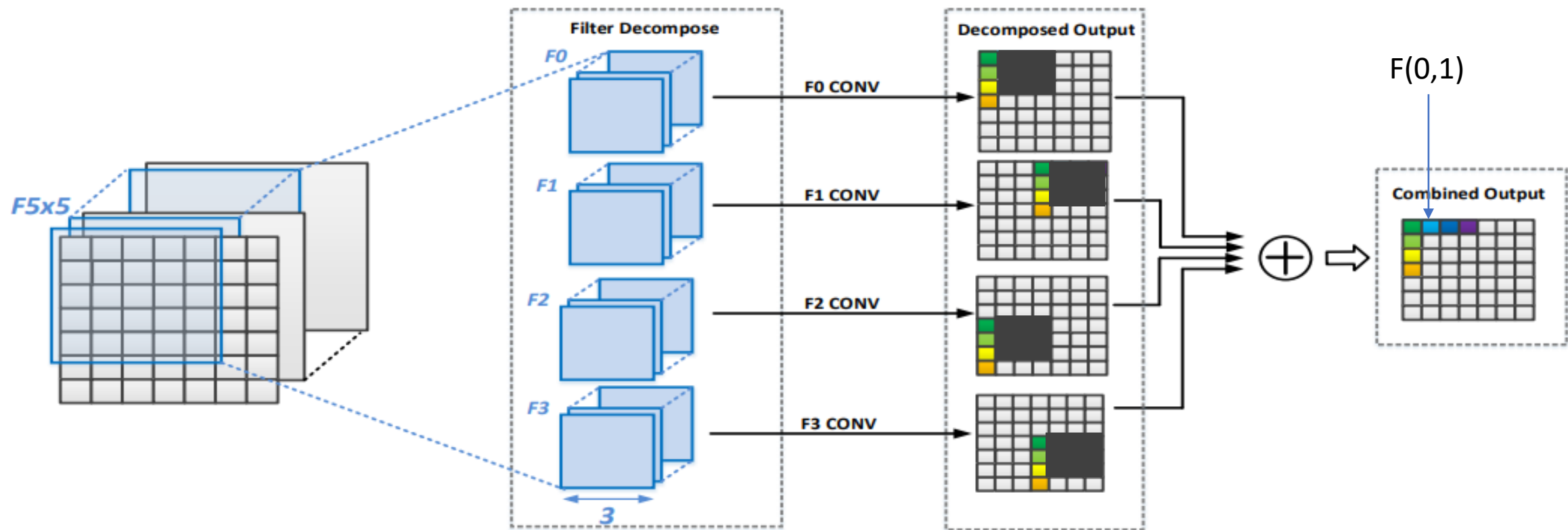
Output width = (Output width + padding width right + padding width left - kernel width) / (stride width) + 1

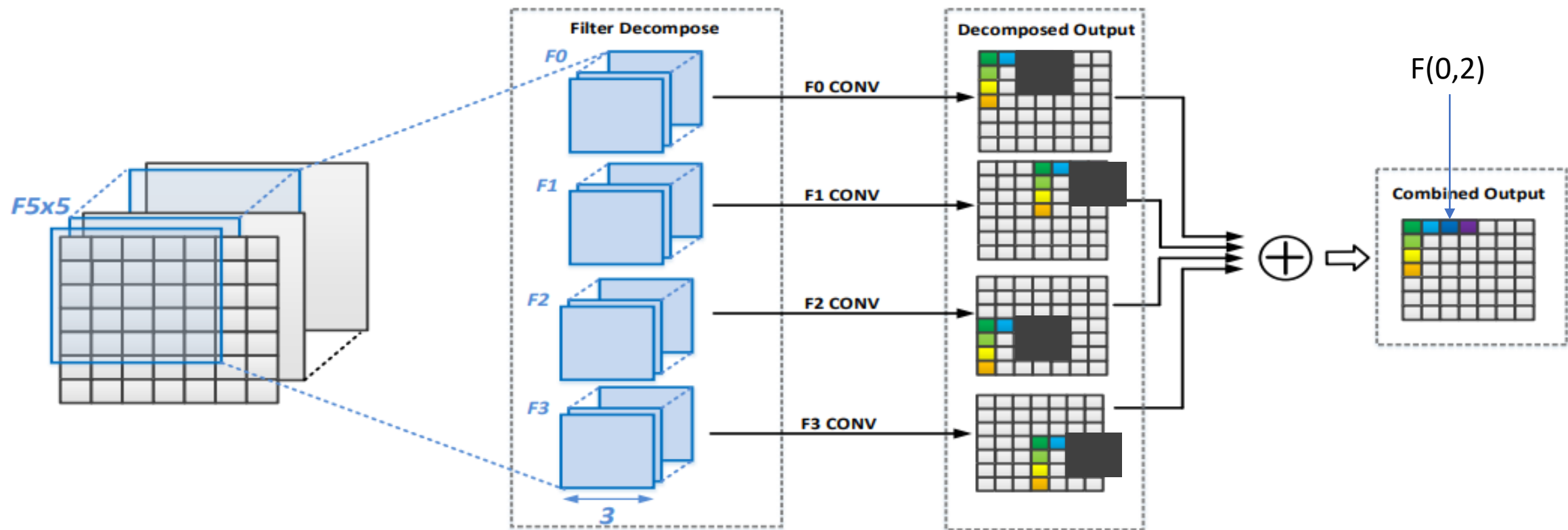


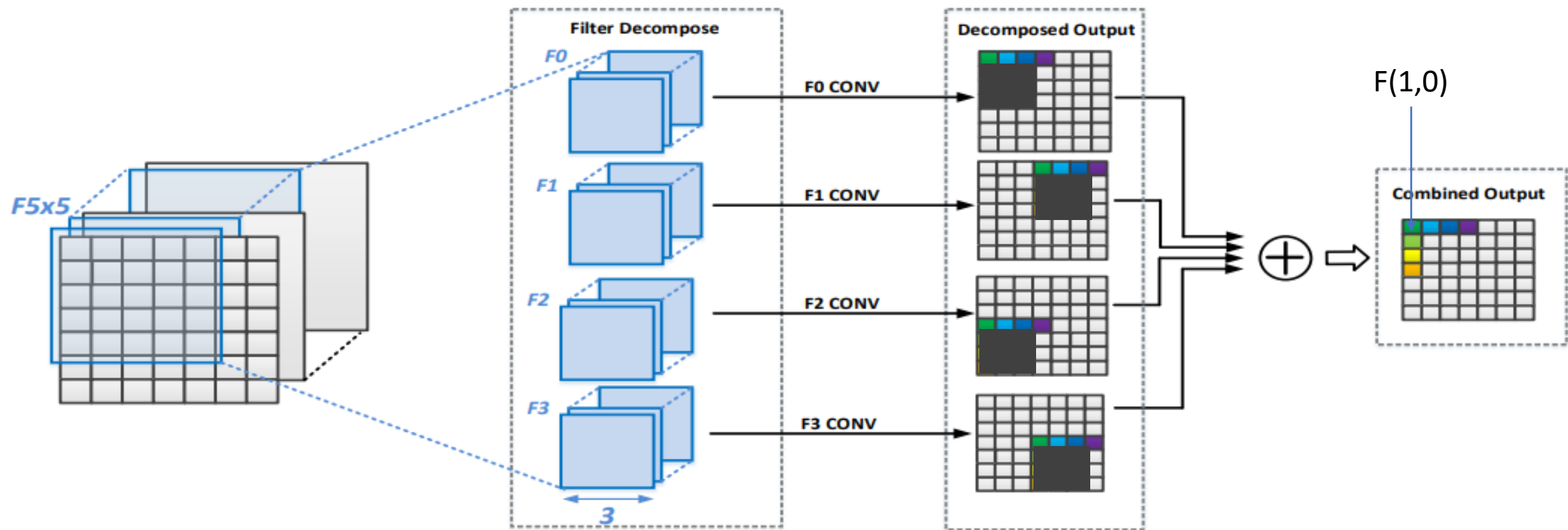
To decompose large kernel-sized computation to many parallel small kernel-sized computations.

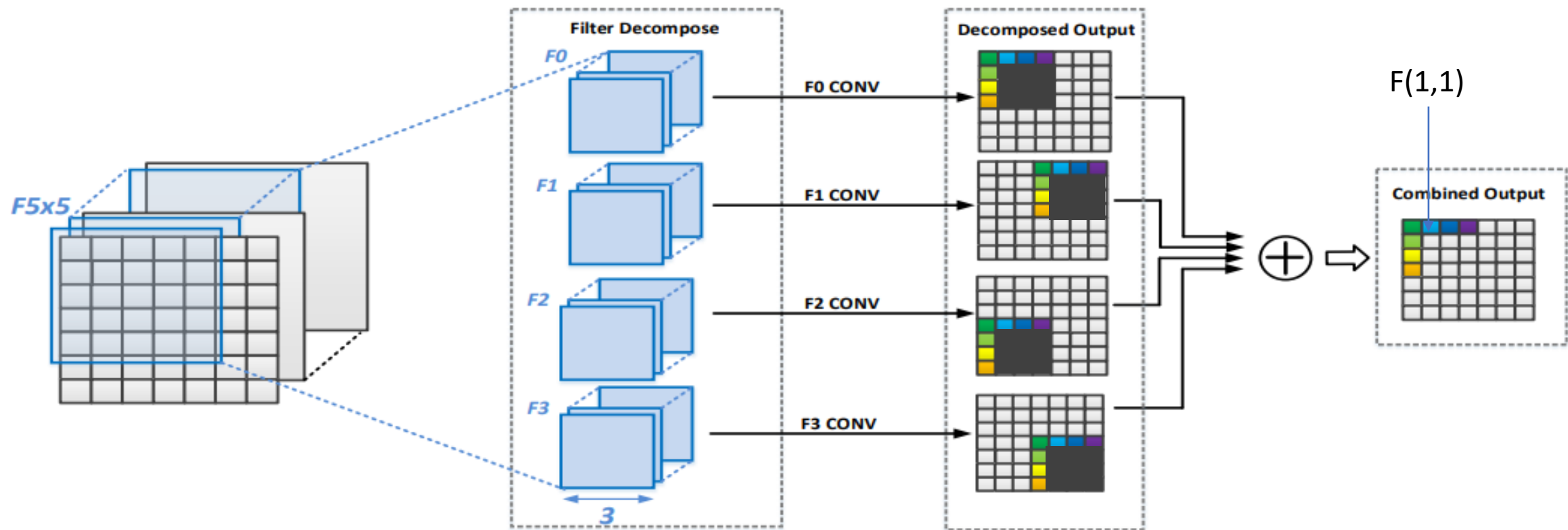


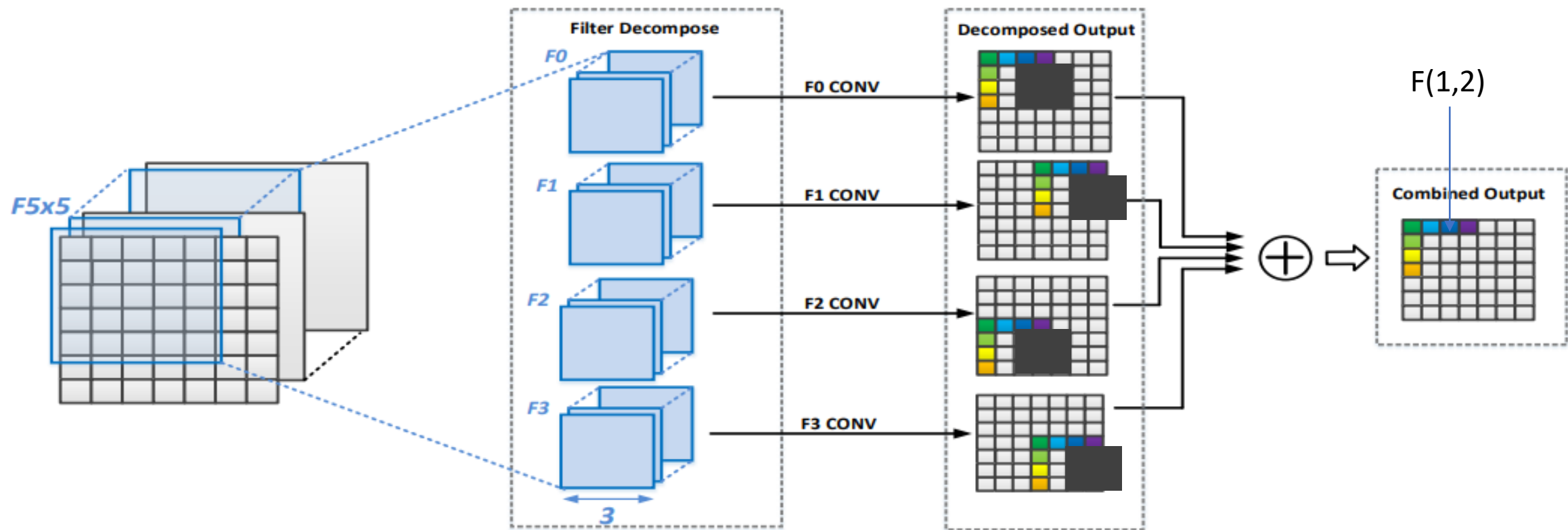


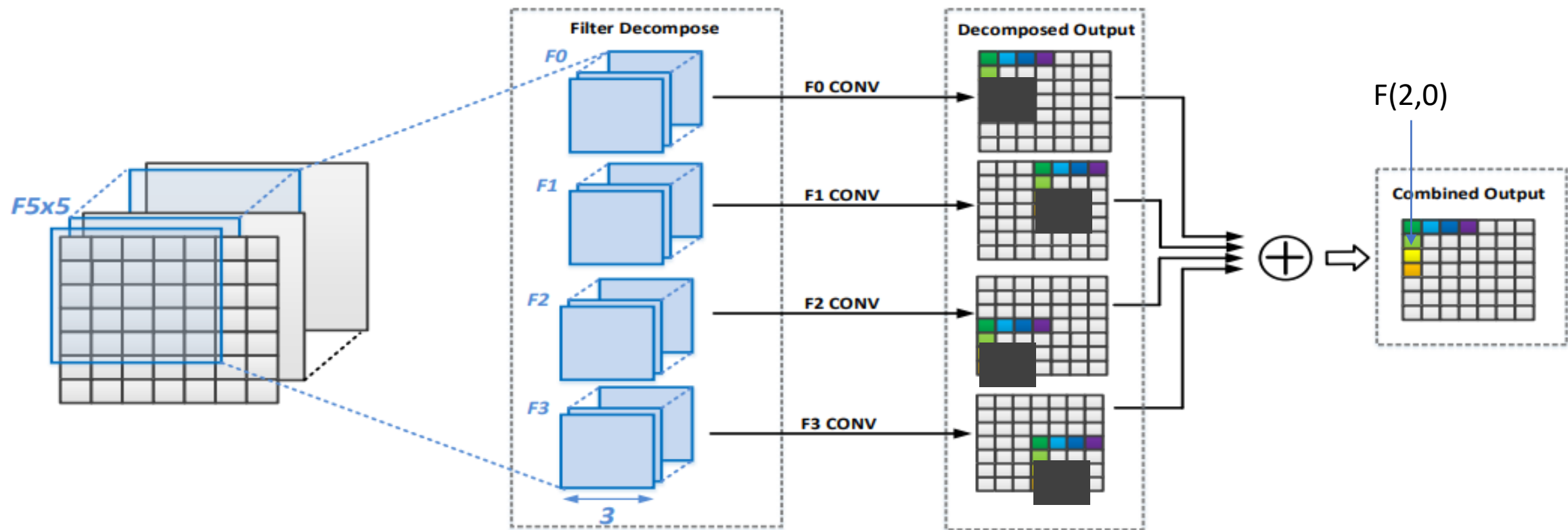


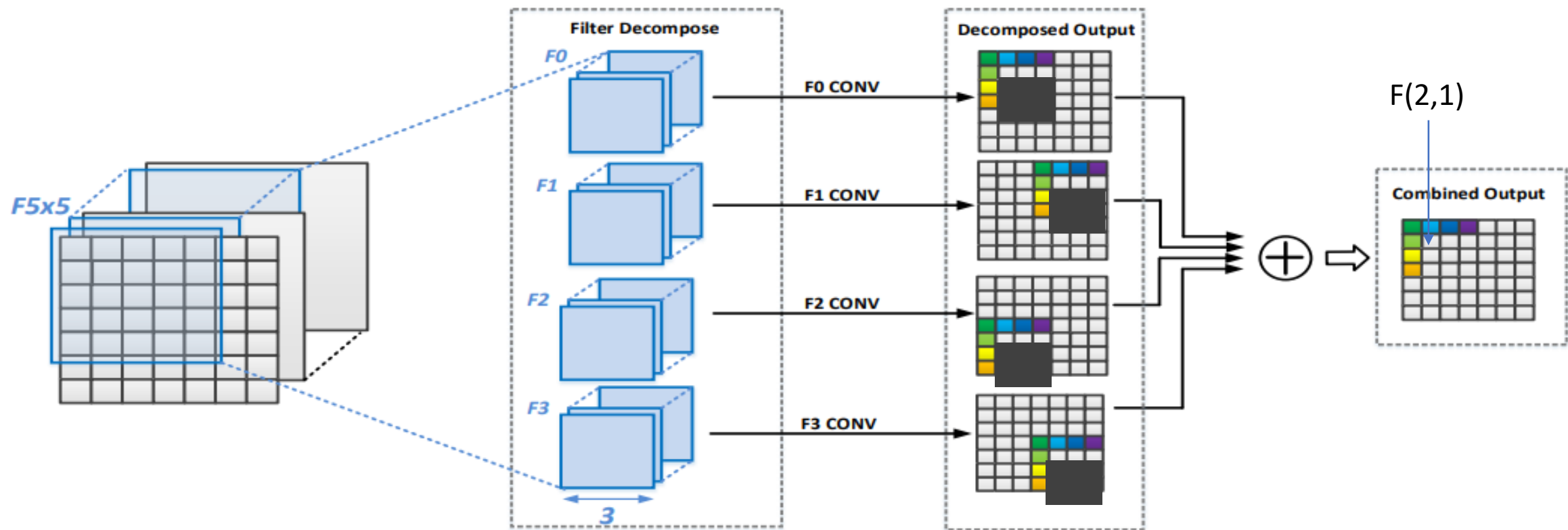


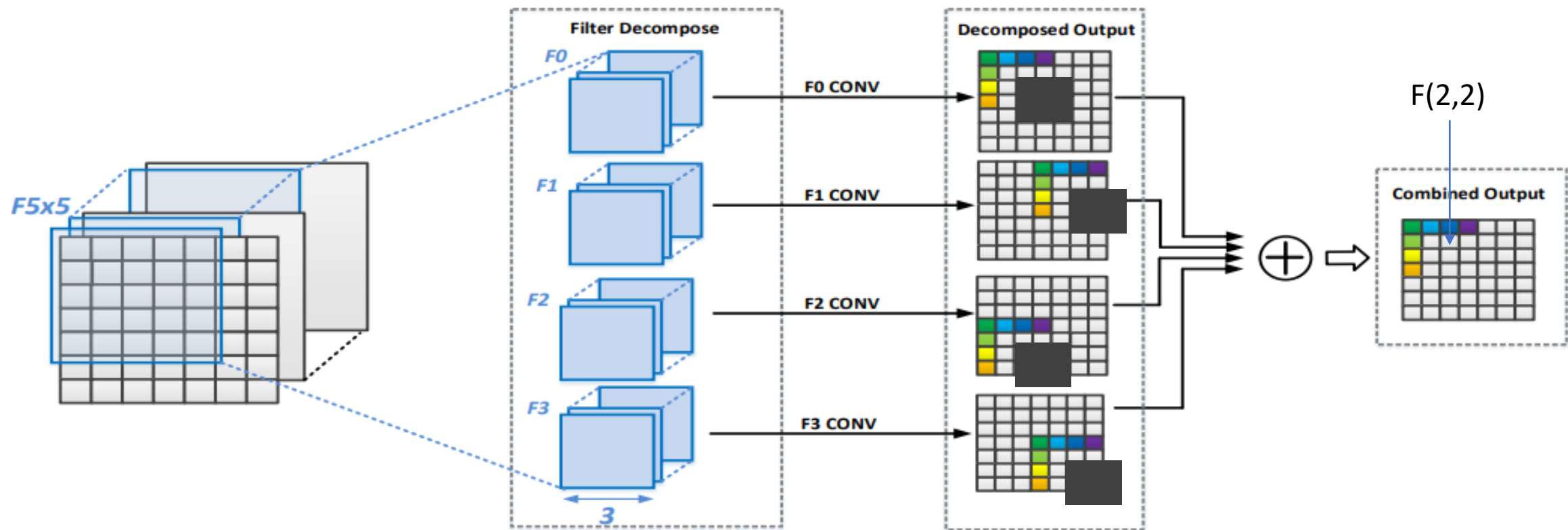












Plan to do