5.1. *Experimental set up*

To evaluate the performance of the proposed approach, we have considered some well-established kernels.[5] The designs are implemented in Verilog Hardware Description Language and synthesized using the *Cadence Register Transfer Level* (*RTL*) *compiler*. In order to analyze the circuit area, power and delay, we have synthesized the proposed designs using 180 nm *technology library* of the Cadence RTL compiler. In addition to this, we have implemented the proposed designs on *Xilinx Artix-7 FPGA device* to understand the impact of the proposed approach on the FPGA architecture. To this end, all the implementations have been realized using *Vivado Design Suite* for Xilinx FPGA device.

The proposed designs realize a single convolution output, however, in many applications such as image processing, neural networks, multiple convolution outputs are generated from a given image by recursively applying the convolution process. For this purpose, we have considered multiple images from the USC-SIPI image database[38] and partitioned each corresponding image matrix into a smaller sub-matrices as discussed in Sec. 4.1. The size of each sub-matrix is chosen as 3×3. To do this, we have implemented the matrix partitioning scheme using Python programming language. All the experiments have been conducted on a computer with an Intel(R) Core(TM) i5-6200U CPU 2.40 GHz and 8 GB RAM.

5.2. *Results and discussions*

5.2.1. *Comparisons with baseline structure*

Table 1 reports the resulting circuit area (*Area* (*in* nm^2)), power consumption (*Power* (*in* mW)) and critical path delay (*Delay* (*in* ns)) of the baseline design and the proposed design for each kernel. Note that we have considered the circuit structure shown in Fig. 2 as the baseline. Figure 7 compares the design metrics (i.e., area, power and delay) with respect to 180 nm technology node of the baseline and the proposed designs. For better representation and clarity, we have denoted few kernels by a single representative in Fig. 7 as those kernels have same circuit area, power consumption and delay. For example, the kernels, *prewitt_h, prewitt_v* are denoted as *prewitt, sobel_h, sobel_v* are denoted as *sobel*, all the eight *kirsch* kernels are indicated as *kirsch* and *horizontal_line, vertical_line* and *line_p45, line_m45* are represented by *horizontal, vertical* and *line_45*, respectively.

The results clearly show that, using the proposed idea, area, power consumption and delay can substantially be reduced if the kernel functionality are considered during the design of convolution operation over general baseline structure. On average, reductions of approximately 88% in area, power consumption and approximately 31% in critical path delay are obtained, while, in the best case, reductions of more than 93% in area, more than 95% in power consumption and more than 44% are achieved as can be seen in Figs. 7(a)–7(c), respectively.

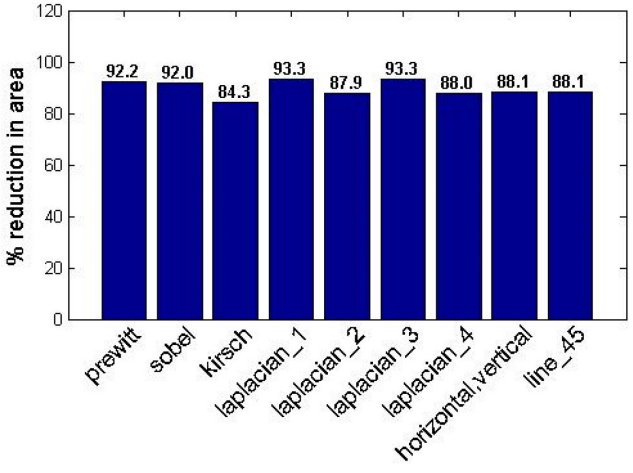
Table 1. Area, power and delay of baseline and proposed designs.

Design	Area (in nm^2)	Power (in mW)	Delay (in ns)
Baseline	123191	9.32	5.88
prewitt_h	9650	0.56	3.26
prewitt_v	9650	0.55	3.26
sobel_h	9836	0.53	3.27
sobel_v	9836	0.58	3.27
kirsch_n	19381	1.62	4.74
kirsch_nw	19381	1.6	4.74
kirsch_w	19381	1.59	4.74
kirsch_sw	19381	1.59	4.74
kirsch_s	19381	1.6	4.74
kirsch_se	19381	1.64	4.74
kirsch_e	19381	1.59	4.74
kirsch_ne	19381	1.62	4.74
laplacian_1	8253	0.43	3.24
laplacian_2	14853	1.01	3.83
laplacian_3	8230	0.47	3.29
laplacian_4	14760	1.03	3.88
horizontal_line	14609	0.94	3.86
vertical_line	14609	0.95	3.86
line_p45	14609	0.96	3.86
line_m45	14609	0.96	3.86

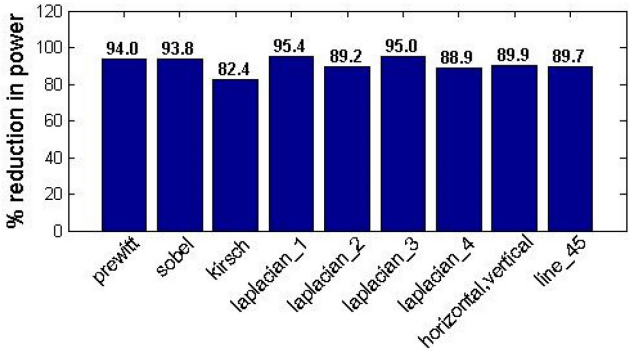
Table 2 reports the hardware resources i.e., the number of Look-up tables (*LUT count*) and input-output pins (*Pin count*) of Xilinx Artix-7 FPGA Device utilized by the baseline design and the proposed kernel-based designs. Figure 8 compares the baseline and the proposed kernel-based designs with respect to LUT and pin counts of a FPGA device.

The results clearly confirm that, using the proposed idea, hardware resources of the FPGA device can substantially be reduced if we consider the kernel functionality over general baseline structure during the hardware realization of convolution operation. More precisely, on average, reductions of approximately 93% in LUT count and approximately 54% in pin count are obtained, while, the best case scenario shows reductions of approximately 96% in LUT count and approximately 68% in pin count can be achieved as shown in Figs. 8(a) and 8(b), respectively.

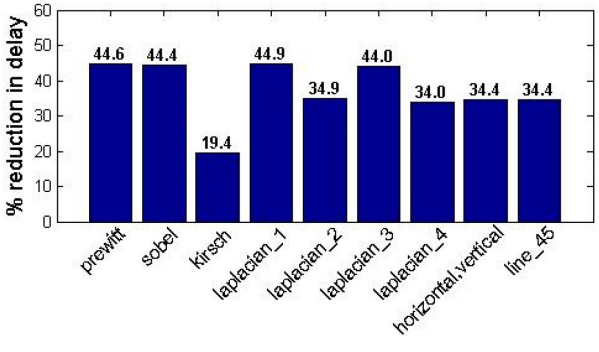
The Xilinx Artix-7 FPGA device has total of 134,600 LUTs and 500 pins available to realize any design. The numbers in Table 2 indicate that a fraction of the available resources has been used by the baseline and the proposed kernel-based designs. We have shown the percentage utilization of LUTs and pins by the baseline and kernel-based designs in Figs. 9(a) and 9(b), respectively. It is obvious that the baseline design utilizes more resources as compared to the kernel-based designs. It can be observed that only less than 1% LUTs have been utilized in all the cases. This utilization is obtained for the realization of single convolution operation, but in practice, multiple convolution outputs are needed to be generated which is possible if



(a) % reduc. in area



(b) % reduc. in power



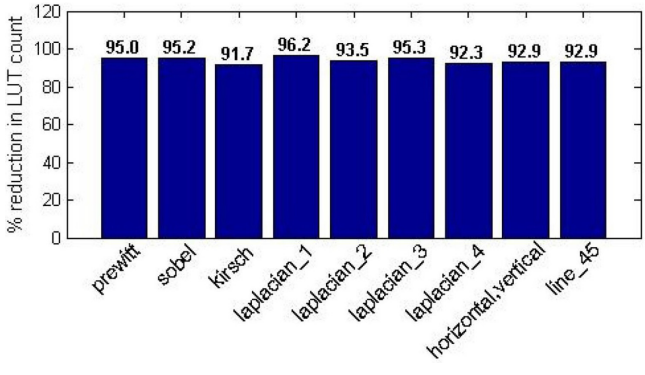
(c) % reduc. in delay

Fig. 7. Comparisons of area, power and delay.

Table 2. Utilization of FPGA resources by baseline and proposed designs.

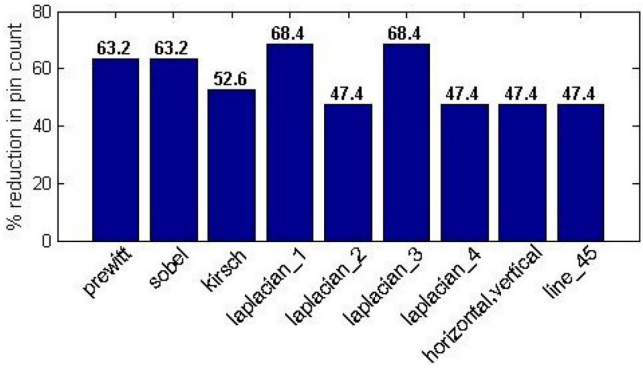
	Xilinx Artix-7 FPGA device	
Design	LUT count	Pin count
Baseline	768	152
prewitt_h	38	56
prewitt_v	38	56
sobel_h	37	56
sobel_v	37	56
kirsch_n	64	72
kirsch_nw	64	72
kirsch_w	64	72
kirsch_sw	64	72
kirsch_s	64	72
kirsch_se	64	72
kirsch_e	64	72
kirsch_ne	64	72
laplacian_1	29	48
laplacian_2	50	80
laplacian_3	36	48
laplacian_4	59	80
horizontal_line	54	80
vertical_line	54	80
line_p45	54	80
line_m45	54	80

we use multiple copies of the single circuit realizing a single convolution operation. The allows parallel computations of several convolution outputs by utilizing the full potential of the available LUTs in the FPGA devices. However, the maximum number of outputs that can be computed in parallel is limited to the number of pins available in the corresponding FPGA devices. This is evident from Fig. 9(b) which



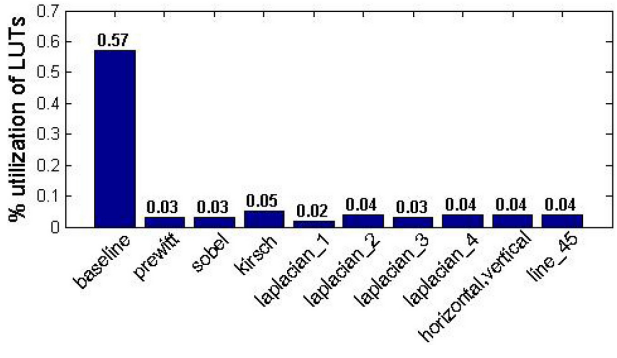
(a) % reduc. in LUT count

Fig. 8. Comparisons of LUT and pin counts of FPGA.

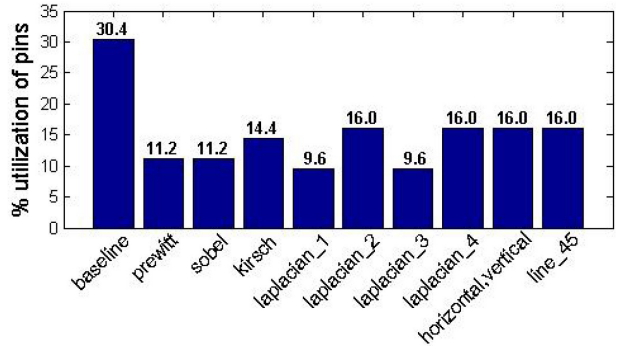


(b) % reduc. in pin count

Fig. 8. (*Continued*)



(a) % LUT utilization

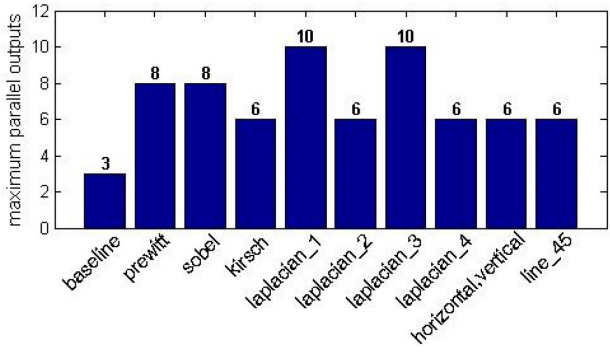


(b) % pin utilization

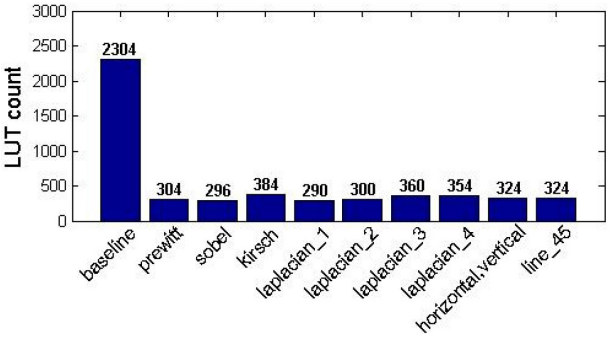
Fig. 9. Comparisons of FPGA resource utilization for single convolution output.

shows that the baseline design realizing a single convolution operation utilizes more than 30% pins. In contrast, the proposed kernel-based designs, on average, utilize approximately 14% pins, while the best case scenario reports not more than 10% pin utilization. These results allow us to compute the number of outputs that can be generated in parallel in the Artix-7 FPGA device.

Based on the number of pins used for single convolution output, we have obtained the maximum number of outputs that can be realized in parallel using the baseline and the proposed kernel-based designs. The results in Fig. 10(a) show that the baseline design can only realize 3 outputs in parallel, whereas, the kernel-based designs allow us to realize up to 10 outputs in parallel. This parallelism comes with an increase in the number of LUTs and the number of pins as shown in Figs. 10(b) and 10(c), respectively. The increase in LUT count and pin count obviously leads to an increase in the resource utilization. Figure 11(a) indicates that the baseline architecture utilizes 1.7% of the available LUTs, while, the proposed kernel-based designs utilize up to 0.29% of the available LUTs. This means, the LUT utilization is

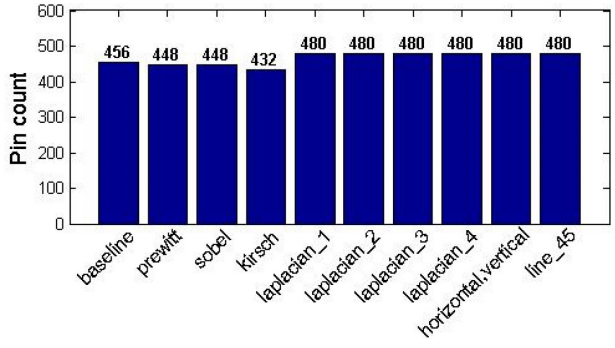


(a) max. parallel output



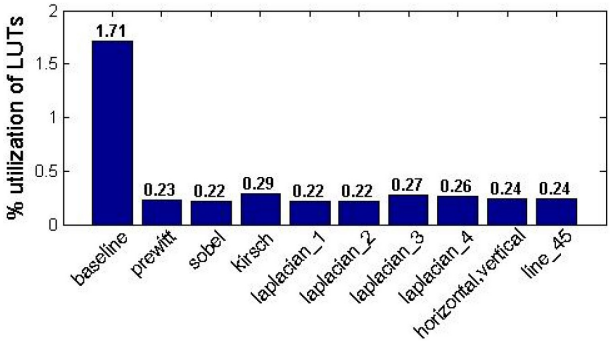
(b) LUT count

Fig. 10. LUT and pin counts for parallel convolution outputs.

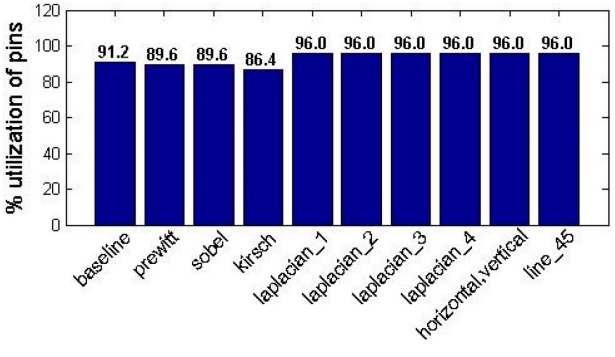


(c) Pin count

Fig. 10. (*Continued*)



(a) % LUT utilization



(b) % pin utilization

Fig. 11. Comparisons of FPGA resource utilization for parallel convolution outputs.

still significantly small in all the cases. However, the pin utilization has reached the maximum as can be seen from Fig. 11(b), where all the designs realizing multiple outputs in parallel have used more than 86% of the available pins. All the results confirm that the proposed kernel-based designs realize single or multiple convolution output(s) with smaller circuit complexity as compared to the baseline design.

5.2.2. *Comparisons with existing works*

We have compared our proposed scheme with the existing dataflow architectures[22,24,28] for matrix convolution operations. We have implemented all the approaches (the existing ones and the proposed one) in the Xilinx Artix-7 FPGA device and the resource utilization of the prior and proposed schemes are summarized in Table 3. The results clearly demonstrate the advantages of the proposed scheme over the existing ones. More precisely, the architecture reported in Ref. 24 allows one to compute at most 3 outputs in parallel which results in 96.2% pin utilization. In contrast, the architectures reported in Refs. 22 and 28 utilize 84.8% and 85.4% of available pins, respectively, and can compute at most 4 outputs in parallel. Unlike pin utilization which is maximum, the LUT utilization remain very small for all the existing architectures (i.e., 1.87% in case of Ref. 24, 2.2% in case of Ref. 22 and 2.09% in case of Ref. 28).

Nevertheless, the proposed kernel-based scheme allows higher parallelism with minimum LUT utilization as can be seen in Fig. 10 and Table 3. While utilizing 2% of the available LUTs, the existing schemes[22,24,28] can compute 3 or 4 outputs in parallel, our approach, utilizing less than 1% of the available LUTs, can compute 6–10 outputs in parallel, thereby accelerating the convolution operations over any input image with minimum hardware utilization. This clearly establishes the

Table 3. Comparisons with existing works.

			Resource utilization of Artix-7 FPGA device			
		Maximum parallel output	LUT count	Pin count	% LUT utilization	% pin utilization
Existing approach	Ref. 24	3	2517	481	1.87	96.2
	Ref. 22	4	2960	424	2.2	84.8
	Ref. 28	4	2814	427	2.09	85.4
Proposed approach	prewitt	8	304	448	0.23	89.6
	sobel	8	296	448	0.22	89.6
	kirsch	6	384	432	0.29	86.4
	laplacian_1	10	290	480	0.22	96
	laplacian_2	6	300	480	0.22	96
	laplacian_3	10	360	480	0.27	96
	laplacian_4	6	354	480	0.26	96
	horizontal,vertical	6	324	480	0.24	96
	line_45	6	324	480	0.24	96

Table 4. Execution times of the baseline and proposed kernel-based circuits for images.

Benchmark				total execution time (in ms)									
Name	Size	Sub-matrices	T	Baseline	prewitt	sobel	kirsch	laplacian_1	laplacian_2	laplacian_3	laplacian_4	horizontal, vertical	line_45
Airplane	512×512	260100	0.04	1.53	0.85	0.85	1.23	0.84	1.00	0.86	1.01	1.00	1.00
Barbara	512×512	260100	0.04	1.53	0.85	0.85	1.23	0.84	1.00	0.86	1.01	1.00	1.00
Boat	512×512	260100	0.04	1.53	0.85	0.85	1.23	0.84	1.00	0.86	1.01	1.00	1.00
Cat	733×490	356728	0.03	2.09	1.16	1.17	1.69	1.16	1.37	1.17	1.38	1.38	1.38
Face	768×1024	782852	0.07	4.6	2.55	2.56	3.71	2.54	3.00	2.58	3.04	3.02	3.02
Fruits	512×512	260100	0.04	1.53	0.85	0.85	1.23	0.84	1.00	0.86	1.01	1.00	1.00
Lena	512×512	260100	0.04	1.53	0.85	0.85	1.23	0.84	1.00	0.86	1.01	1.00	1.00
Mountain	480×640	304964	0.17	1.79	0.99	1.00	1.45	0.99	1.17	1.00	1.18	1.18	1.18
Watch	768×1024	782852	0.07	4.6	2.55	2.56	3.71	2.54	3.00	2.58	3.04	3.02	3.02
Average		391988.44	0.06	2.30	1.28	1.28	1.86	1.27	1.50	1.29	1.52	1.51	1.51

fact that the proposed kernel-based scheme is much more advantageous than the existing ones.

5.3. *Impact on applications*

In Sec. 5.2, we have evaluated the circuit complexity of the proposed approach with respect to certain design metrics. But, it is important to evaluate the performance of the proposed kernel-based design scheme on practical cases in terms of execution time. To do this, we have considered convolution operation over an entire image which is frequently conducted in image processing or neural networks.

Table 4 summarizes the total time required to complete the convolution operations for each image when the baseline and the proposed kernel-based circuits are used. The first two columns provide the details of the image benchmarks i.e., name of the benchmark (*Name*) and the size of the benchmark (*Size*). The third column provides the number of sub-matrices that are generated from each considered image. To obtain the sub-matrices, we have applied matrix partitioning scheme described in Sec. 4.1. The run-time (T in CPU seconds) needed to partition the matrix is listed in the fourth column. Herein, we have assumed the size of each sub-matrix to be 3×3 based on the size of the kernels considered in this work. The final columns report the total execution time when baseline and the proposed kernel-based circuits are used for realizing convolution operation over an entire image.

The results indicate that on average, the baseline circuit completes the executions in 2.3 ms whereas, in case of kernel-based circuits the average time of execution lies in the range of 1.27–1.86 ms. This means, in case of the proposed design scheme, the average execution time is significantly less as compared to the general baseline design. Moreover, in the worst cases (i.e., when the image sizes are large for e.g. *Face, Watch*), the baseline design requires 4.6 ms to complete the convolution operations over the corresponding images. In contrast, the worst case time to execute convolution operations using proposed design approach lies in the range of 2.55–3.71 ms. This shows that the proposed approach can complete the convolution over an image much faster than the general baseline approach.

6. Conclusion

In this paper, we have explored the design space of a matrix convolution operation which is a primary operation employed in neural network architectures. To this end, we have introduced alternative design approach which utilizes kernel definitions during the design of circuit structures realizing convolution operations. We have confirmed through experimental evaluations the potential benefits of the proposed scheme which can generate circuits realizing convolutions with less area, power consumption and delay. Moreover, the results of this work motivates the designers to consider the design aspects of the desired functionality to be executed on the