Dear Reviewers:

Thank you for the reviewers’ comments. We have studied comments carefully and have corrected. The responses are as follows:

**Reviewer #1:**

1. The abbreviation "BNN" (in Section I, paragraph 1, page 1, column 2) is not explained. In addition, in Section I, the author mentioned both binarized convolutional neural network (BCNN) and binarized neural networks (BNN). What are the differences between these two DNNs? Considering that the BinaryNet model used in this paper only contains convolutional layers, and does not contain layers such as LSTM and GRU, should it be more appropriate to be a BCNN?

**Response:**

**B**inarized **n**eural **n**etworks (BNN) is a generic term, which contains many kinds of neural networks, such as binarized recurrent neural networks and binarized convolutional neural networks. The BinaryNet model used in the paper only contains convolutional layers and FC layers. It will be more accurate to call it a BCNN model. For clarity, we have modified Section 1 with more accurate abbreviations. At the beginning of Section 2, to maintain the custom of expression, we have stated that BNN is equivalent to BCNN in subsequent parts of the paper.

1. Add some explanations or references for "The results demonstrate that our design improves performance at least twice as much as previous works." (Section I, paragraph 1, page 3, column 2).

**Response:**

Thanks for your suggestions and we have revised the article.

1. Perhaps it would be better to add some explanations to present more clearly the characteristics of the strategy in this "We proposed a strategy to optimize resource utilization." (Section I, paragraph 3, page 3, column 2).

**Response:**

We have modified the article and explained the detailed strategy.

1. Typo: "usually has there channels of RGB" instead of "usually has three channels of RGB" (Section II, paragraph 1, page 4, column 1); "Througput (GOPS)" instead of "Throughput (GOPS)" in Table 4.

**Response:**

We are very sorry for our mistakes. After carefully checking, we found many grammatical errors, and have modified the manuscript accordingly.

1. Figure 6 should move forward to section 3.4, and before section 3.5.

**Response:**

We have revised the article.

1. In Table 4, the authors compared the performance and resource utilization of the proposed accelerator using different number of PEs. Perhaps it would be better to give the power breakdown of the proposed accelerator implementations with different configurations. Taking the implementation-1 with 1PE configuration and the implementation-4 with 4PE configuration as examples, the comparison results in Table 4 show that: the kLUT used in implementation-4 (78kLUT) is 1.9 times that of implementation-1 (41kLUT); the DSP used in implementation-4 (43) is 3.3 times that of implementation-1 (13); the throughput achieved by implementation-4 (2630 GOPS) is 4 times that of implementation-1 (658 GOPS); However, the power consumption of implementation-4 (10.8 W) is only 1.2 times that of implementation-1 (9.2 W). Why different hardware configurations have a great impact on hardware resource overhead, but have little impact on hardware power consumption?

**Response:**

As we said in Section 6.2, we find that both resource efficiency and power efficiency are steadily increasing with task-level parallelism. The main reason for this is that different configurations use the same peripheral framework (such as DDR), so when the task-level parallelism is very low, the acceleration logic is much smaller than the framework which consumes most resources. Most of the power consumption is caused by the peripheral framework (DDR and control logic). When the parallelism increased, the power consumption of the peripheral framework was almost no change. Therefore, different hardware configurations have a great impact on resource and a little impact on power consumption.

**Reviewer #2:**

1. What was the bottleneck of previous work? Why can't their design scale on large FPGA?

**Response:**

Most of the previous works (such as Zhao et al.[1] and FBNA[2]) generally used smaller FPGA chips with little on-chip memory, which could not store all the weights and intermediate results in on-chip memory. They frequently access off-chip memory during the computation process, which resulted in delays. If they are scaled to a large FPGA, there will be an obvious mismatch between memory usage and bandwidth, resulting in a huge waste of on-chip storage and compute resources. Meanwhile, our design makes the best use of these valuable resources. FP-BNN[3] used a large FPGA but it folded the network, which simplified the hardware and decreased resource efficiency since different layers have different requirements. Our design is to fully expand the network and optimize the pipeline at a fine-grained level, which makes the hardware more complex but more efficient.

**References:**

1. Zhao, Ritchie, Weinan Song, Wentao Zhang, Tianwei Xing, Jeng-Hau Lin, Mani Srivastava, Rajesh Gupta, and Zhiru Zhang. 2017. "Accelerating binarized convolutional neural networks with software-programmable FPGAs." In Proceedings of the 2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays, 15-24.
2. Guo, Peng, Hong Ma, Ruizhi Chen, Pin Li, Shaolin Xie, and Donglin Wang. 2018. "FBNA: A Fully Binarized Neural Network Accelerator." In 2018 28th International Conference on Field Programmable Logic and Applications (FPL), 51-513. IEEE.
3. Liang, Shuang, Shouyi Yin, Leibo Liu, Wayne Luk, and Shaojun Wei. 2018. 'FP-BNN: Binarized neural network on FPGA', Neurocomputing, 275: 1072-86.
4. Why is the design scalable? I suggest the authors supplement it explicitly in the paper.

**Response:**

Our design can be implemented with different combinations of multi-levels parallelisms. The parallelisms consist of task-level, layer-level, loop-level, and operation-level parallelism as described in Section 3.3.1, Section 3.4, and Section 5.2. As described in Section 6.1, the reason for verifying scalability only by changing the task-level parallelism is that cur-rent layer-level, loop-level, and operation-level parallelism are the best combinations obtained using the design space exploration algorithm for the XCKU115 FPGA chip. With this combination, compute (LUT) and memory (BRAM) resources can get similar utilization, which means that we do not waste any resource due to a bottleneck of one resource. On the other hand, the use of simpler models for different datasets also reflects our control of loop-level parallelism. We have made some changes to the article to make that clearer.

We look forward to hearing from you regarding our submission. We would be glad to respond to any further questions and comments that you may have.