Dear Reviewers:

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

**Reviewer #1:**

1. This paper need a major revision. The current expression of the novel ideas are not clear**.** Quantization is a commonly used technique for FPGA based implementations and multi-PE is also not totally novel. I believe the systematic design of the paper should be a good one, but it is not convincible enough now.

**Response:**

Thank for your recognition to systematic design of our paper. Our main contributions are: a) we design a novel layer-by-layer pipeline which actually pipelining the layer in CNN model; b) we take hybrid bit-width to quantize our model. We mainly focus the difference between the data before normalization (intermediate data) and the data after normalization (feature data), based on discussion about such difference, we devise corresponding bitwise schemes; c) we implement sufficient performance comparison experiments between our accelerator and other hardware platforms.

All these contents are rearranged in our paper, together with necessary hardware design details. Thank you for your suggestions and we have made revision.

1. The 'level' in the paper is used wrongly, it should be 'layer' according to the architecture of the CNNs. Please double check all the scientific naming of the key words and revise them.

**Response:**

Thank you for reminding us and we are very for such mistakes. We have corrected all these errors.

1. Fig. 1 is not referred and also not helpful to the main context.

**Response:**

We have removed this figure and thanks for your reminding.

1. It is not clear through the entire paper why the CNNs show advantage to the temporal models on the temporal data, which is the speed data. Please further extend is and explain it clearly.

**Response:**

We have removed this figure and thanks for your remainding.

1. The idea that Fig.2 is presenting is too simple, should not use a figure but just explanation is enough to understand.

**Response:**

We have replaced this figure with description in text.

1. Does the 'level-by-level' pipeline actually pipelining the 'layer'? if so, it should be 'layer-by-layer' pipelining. Again, please verify the correctness of all the names of the proper nouns.

**Response:**

Our accelerator does work in “layer-by-layer” mode and we have modified all improper scientific names. Thanks for your suggestions.

1. Fig. 10 is hard to understand, how about using the bitwidth as the x-axis?

**Response:**

We have modified this figure’s x-axis into (IRDB, NRDB) in Fig.8 (corresponding to old version paper’s Fig.10) in our revised version paper. In this figure, the x-axis (IRDB, NRDB) just means the combination of intermediate result decimal bit-width and normalization result decimal bit-wise as you suggest. We think it is a better way to demonstrate our experiment. Thanks for your suggestion!

**Reviewer #2**

1. The two compared baselines are all published in 2015, which is already five years. I encourage the authors to compare with more recent and state-of-the-art FPGA implementations to clear show the advances of this paper.

**Response:**

Thanks for your suggestions! We have added comparison with recent and state-of-art FPGA implements into our paper. In Sec. 5.2 of our revised paper, we compare our design with recent fully-binarized CNN accelerators based on FPGA and sparse RNN’s FPGA accelerators. The detail is shown in Table.5b and Table.5c. As shown that, our accelerator has less accuracy loss and higher computing performance than full-binary CNN architectures. When compared with RNN-based accelerators, ours outperforms in computing latency.

1. Also, the quantization has been studied widely, what are the differences in this paper?

**Response:**

Thank you for reminding us and we are very for such mistakes. We have corrected all these errors.

1. How does the FPGA implementation compared against other ASIC solutions?

**Response:**

The comparison between ASIC solutions and our FPGA implement is illustrated in Table.5a. We choose serval typical ASIC designs which are similar with us in binary weight and quantization. Our solutions can have better computing performance.