You've commented on the studies from the literature that are looking at similar problems, and in particular you've commented on the lack of implementational details. How much of a barrier has that been to you in your own work, and in delivering a comparison with the results of the other authors?

CNN is originally designed for picture recognition rather than signal events classification. For CNN method described in several papers that have a good performance in onset detection, because the lacking of detail from the previous paper in data preparing or pre-processing, we can only prepare our own data by imitating the picture data processing and feed it to the CNN model.

Actually I did want to repeat the first paper's result since the author use a public music database. And I actually find this database (See http://www.cp.jku.at/people/boeck/ISMIR2012.html) and downloaded it. I created absolutely the same structure from Böck's paper[1] and do the evaluation. However, after modifying many times I still can't achieve the author's result as 88.5% accuracy. The system usually trained badly, or accuracy stop at around 70% which is different than author's description. So finally I can only assume that I miss some important detail or misunderstand the paper's data preparation method that haven't mentioned by the author which is hard for me for further comparison.

How much additional training data do you think you would need to achieve the level of result that you're looking for? Is this practical / achievable in the field?

I'm currently given 14 files contains around 700 annotations of onset for training and validating, considering a "balanced" dataset, so there will be around 1400-1500 contains both onset and non-onset data for training, which just approach the minimum requirement of CNN-based model training. I'm considering to combined the dataset used in Böck's paper [1](a large data but contains many instruments pieces, not only string instruments) and my current data to prepare around 10,000 data for training, which will usually generate a more acceptable and average level result.

What was the biggest challenge that you faced while delivering your project, and how would you approach the problem if you were to do this project again?

I spent a lot of time on modifying the model structure(activation function, dropout), change loss function, and justify the optimizer and it's learning rate and momentum. For example, I changed the learning rate in range of 0.05 to 0.001, and momentum changes from 0.9 down to 0.2. I tried many combinations of function such as "relu" follows "sigmoid", "tanh" with "sigmoid" and so on, just want to find the combination that lead to better result. I think this kind of "Tuning" experience will be very helpful if I'm going to do similar projects at machine area in the future.