Tamil Online Recognition

Points of discussion:

Strokes:
The Tamil Dataset provided is at the stroke level. A stroke is defined as the pixels traversed when the pen came down to the point the pen was lifted while writing. Thus it provides a sequence of strokes for the Tamil word the writer wrote.
The Tamil word in question is provided in the Groundtruth.xls as sequence of symbols it comprises. More than stroke can contribute to the same symbol. Just to be clear here, the number of stokes and the number of symbols for a given word are generally NOT the same. There can be 5 strokes for a Tamil word comprising 3 Symbols.
The creators of the dataset have selected 2000 optimal words to represent the symbols in the language. Then writers manually wrote these words repeatedly to obtain a total of 15000 words.

Methodology:
The classification the authors have pursued is essentially at the at the symbol level. It has been done according to two different models. Lexicon-Free and Lexicon-Driven models. the first generates a central HMM by concatenating the symbol HMMs (details not discussed) and the second follows a two stage methodology, first it uses a recurrent HMM for predicting a symbol at a time to give a sequence and then checks the possible words formed using the symbols independent of the order in which the symbols were transcribed, in what the authors call a bag of Symbols setup (much like a Bag-of-words). The Lexicon free performs better in the case of Tamil as discussed in their paper.

My implementation:

The Dataset :
It is described at the Stroke level. Each stroke begins with Pen_Down signal followed by a sequence of positions followed by a PEN_UP.
My methodology was to append these to the inputs for symbols for the words. An added column for the status of the pen has 0's and 1's along with the x and y position. The 1 is for start of stroke and the 0 otherwise.

Division of data for the training and testing sets.
I used a simple approach, dividing the dataset such that the one instance of each of the 2000 words is used for testing and all other instances are in the training set There are at least 5 instances of each word.
I have not used any Val file at this stage. The authors have used ten-fold validation for training the model.

Representation of the symbols
An Issue is the representation of the symbols in the target strings. There are two categories we use WORD Target String and just Target string

For example, for string "abcd"
The word would be abcd and the target stings [a,b,c,d]
Now, in on line the model I have used earlier and even now is to use the same for both these categories. The reason is the output is a list of symbols that can be later looked up for representation. For now, word and target strings are : [43, 113, 45] so on. Meaning it is composed of the 43rd, 113rd and 45th symbols.

Training Time.
The training time is huge. Each epoch takes ~21 minutes, which was unexpected considering that the dataset is smalled than 100MB. But I have a few theories on this as we could possibly discuss later. This is when I was only using data fraction of about 1/2.
I have since then set the data fraction to 1. I believe this is due to number of weights of the network, which are a humongous 448835! This about 5 times the weights initialized for the action recognitions tasks, even on UCF videos! The reason I suspect for this large number of weights is the output symbols being 114 and large input feature set.
This leads to another issue. The total Tamil symbols discussed the authors are 152 and they claim the 2000 words are optimal to represent each one. But I checked and only 114 are covered.

Note : I have not used any normalization in the initial symbol dataset. That will be in the next step and I am hoping for an improvement due to this.