# Test Case: 4

As we have started testing with the newer version of Video Learning Project, we have figured out some issues regarding the dataset that we have created. The dataset we have used is bigger than which were used to test the performance of the HTM Model (newer version of code). We have tried all the following configurations by keeping the **frameHeight, frameWidth** and **maxCycles** parameters unchanged (in previous reports we have tested with these parameters).

1. **GetPredictedInputValues** = **1**

ColumnDimensions = 1024 Runtime: 24 hours (forcefully stopped)

CellsPerColumn = 16

1. **GetPredictedInputValues** = **1**

ColumnDimensions = 1024 Runtime: 36 hours (forcefully stopped)

CellsPerColumn = 40

1. **GetPredictedInputValues** = **1**

ColumnDimensions = 1024 Runtime: 18 hours (forcefully stopped)

CellsPerColumn = 80

1. **GetPredictedInputValues** = **1**

ColumnDimensions = 2048 Runtime: 14 hours (forcefully stopped)

CellsPerColumn = 16

1. **GetPredictedInputValues** = **1**

ColumnDimensions = 2048 Runtime: 27 hours (forcefully stopped)

CellsPerColumn = 40

1. **GetPredictedInputValues** = **1**

ColumnDimensions = 2048 Runtime: 36 hours (forcefully stopped)

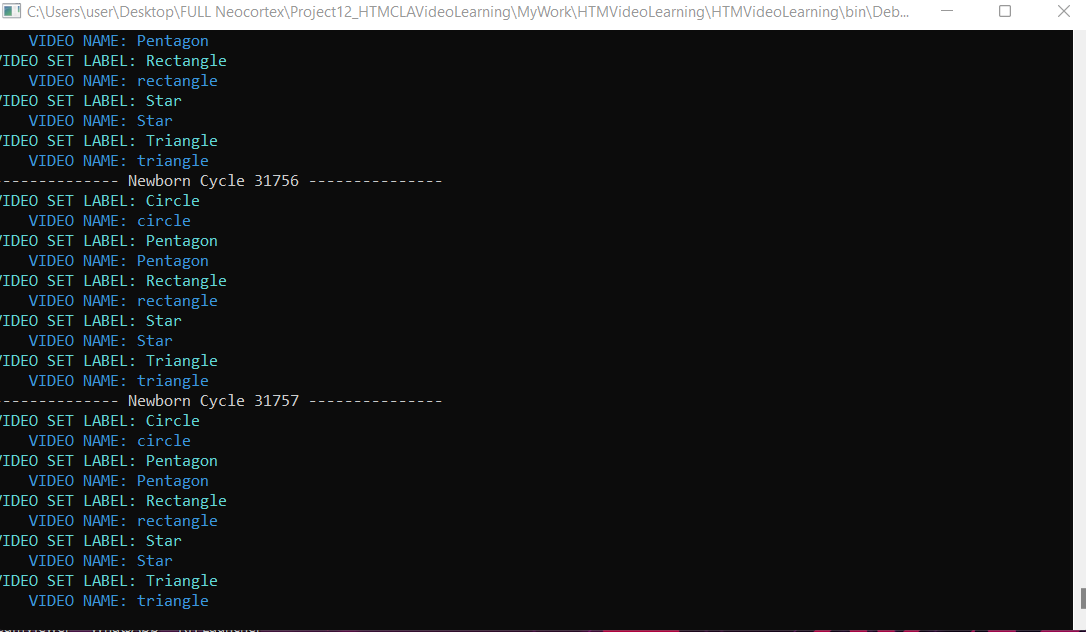
CellsPerColumn = 80

1. **GetPredictedInputValues** = **1**

ColumnDimensions = 4096 Runtime: 18 hours (forcefully stopped)

CellsPerColumn = 40

For these above-mentioned configurations, we could not pass through the newborn cycles state. We have used circle, rectangle, pentagon, and triangle videoset for these experiments. Machine kept generating newborn Cycles, so we must force fully shutdown the process. The following screenshot depicts the problem which has been described here.



We have figured out that as we are using significantly larger videoset the parameter **GetPredictedInputValues** should be larger than 2 because it means how many predictions it makes for a group of active cells. In our previous configurations the value of this specific parameters was set to 1 which means it only takes the first with highest similarities. So we have changed the parameter **GetPredictedInputValues** and ran our experiments with 4 videosets (circle, rectangle, pentagon and triangle). We have tried the following configurations.

1. **GetPredictedInputValues** = **3**

ColumnDimensions = 1024 Runtime: 18 hours (forcefully stopped)

CellsPerColumn = 16

1. **GetPredictedInputValues** = **3**

ColumnDimensions = 1024 Runtime: 24 hours (forcefully stopped)

CellsPerColumn = 80

1. **GetPredictedInputValues** = **3**

ColumnDimensions = 2048 Runtime: 12 hours (forcefully stopped)

CellsPerColumn = 40

1. **GetPredictedInputValues** = **3**

ColumnDimensions = 2048 Runtime: 24 hours (forcefully stopped)

CellsPerColumn = 80

1. **GetPredictedInputValues** = **3**

ColumnDimensions = 4096 Runtime: 24 hours (forcefully stopped)

CellsPerColumn = 80

From configurations **8** to **12,** the model passed through the newborn cycles but when it started predicting the frames of pentagon videoset in the training phase it stuck. The model could not pass through the pentagon videoset in the training phase. **We have figured out that the problem was in the pentagon videoset.** **While I had created the pentagon videoset there were a lot of frames which were similar to each other. But when the video reading started it specified different frame keys with same pattern. Afterwards, in prediction phase it checked for the key to a different frame in time with the same pattern which resulted in a right prediction but the wrong frame key.** So, we have now changed our created videoset **pentagon** by changing the frame duration from 0.4s to 0.2s while creating the video. Which results in the significant mitigation of frames which are similar to each other, and we have also decreased the whole duration of the pentagon video. **We would like to mention one thing that the model does not perform well if we try to train the model with larger videos (in terms of duration) because in a larger video it may contain frames which are similar to each other.** The following screenshot depicts the problem we had faced while we tried to train the HTM model with old pentagon videoset(after 18 hrs the model is still in predicting **pentagon** frames ).

