Test and Investigation of Video Learning Project

By Tuan Nghia Nguyen Khaled Kandil

**Abstract:**

This paper is conducted by taking the already developed framework of Hierarchical Temporal Memory (HTM) which is used to Learn Videos, then putting it under various conditions by modifying different parameters to test the model’s performance.

Keywords—HTM, Video Learning, frame, HTM configurations

1. **Introduction:**

The project is based on the project of HTM Video Learning, which is also originated from the HTM Project with the ideas of Sequence Learning. However, instead of scalars, inputs of images were used. The Video Learning program is already finished but it’s performance was not thoroughly tested for all of the functions and features. Therefore, the main goal of the project is to test the software with various parameters, inputs then document the results. Finally, based on these experiments, different aspects of the program will be concluded to determine if the direction of this Video Learning project was good enough, with the possibility of an improvement proposal.

1. **Methods:**

The aforementioned project of Video Learning with HTM functions by using Temporal Memory to learn binary representation of videos (sequence of bit-arrays, with each bit-array represents 1 frame). This whole experiment was conducted with the input of randomly chosen videos to ensure the project’s objectivity in order to evaluate the video learning process’s performance accurately. Firstly, the video’s input folder path were dragged into of the program. After the period of run time, the result of the learning process was determined by how well the trained model could predict the next frame of the video based on what it had learned, when a specific frame was assigned to the model. Finally, an image (or a frame) was provided to the trained model. The model then attempted to recreate a video with proceeding frame after the input frame. The model’s performance was assessed by two aspects: the accuracy and the elapsed time of the learning process. These aspect’s data were obtained by experimenting with two types of input videos:

Simple Shape Video inputs:The first experiment was running the program with many set of simple videos, in which only basic shapes (a black circle, rectangle and triangle) moving around on a white background. These videos could be found in the path: Our Project\HTMVideoLearning\VideoLibrary\AngleLibrary. All of them have the same duration length of 2 seconds. The difference between these sets is the angle that these shape moves. By changing the parameter in the python code, different set of videos with different moving angle (varies from 0 to 360 degree) can be generated.

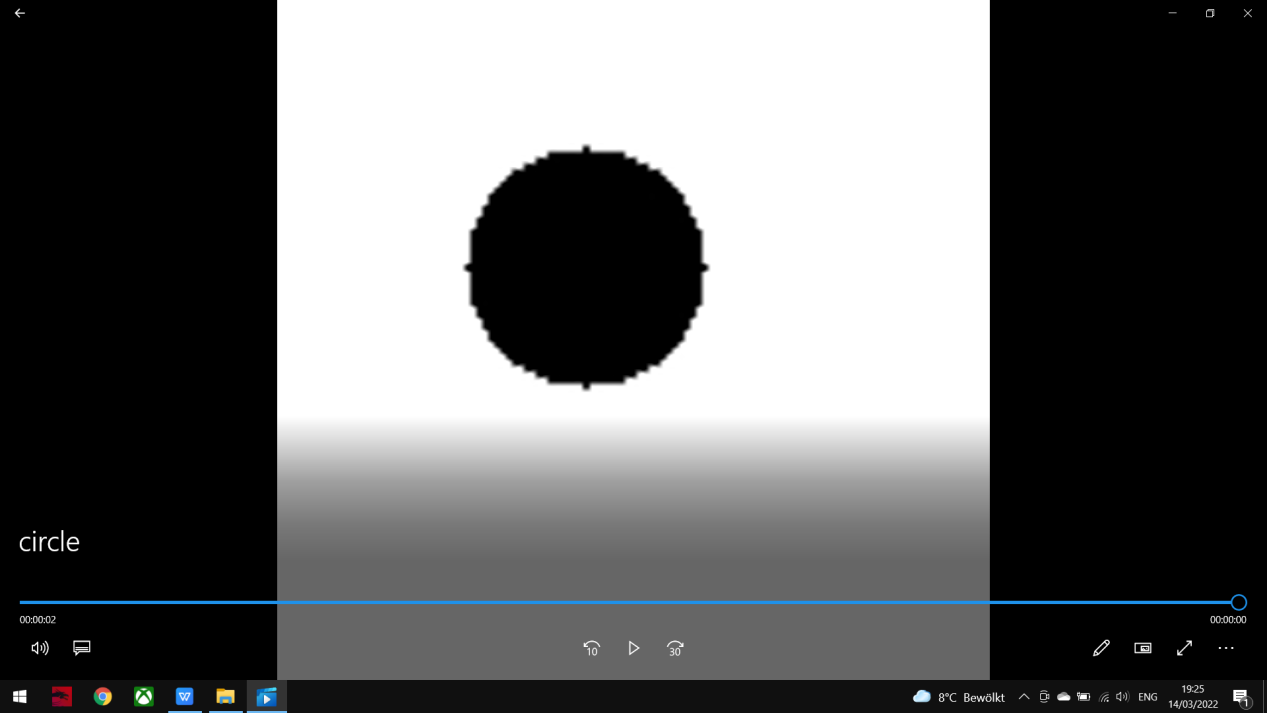


Figure 1a: Circle Video

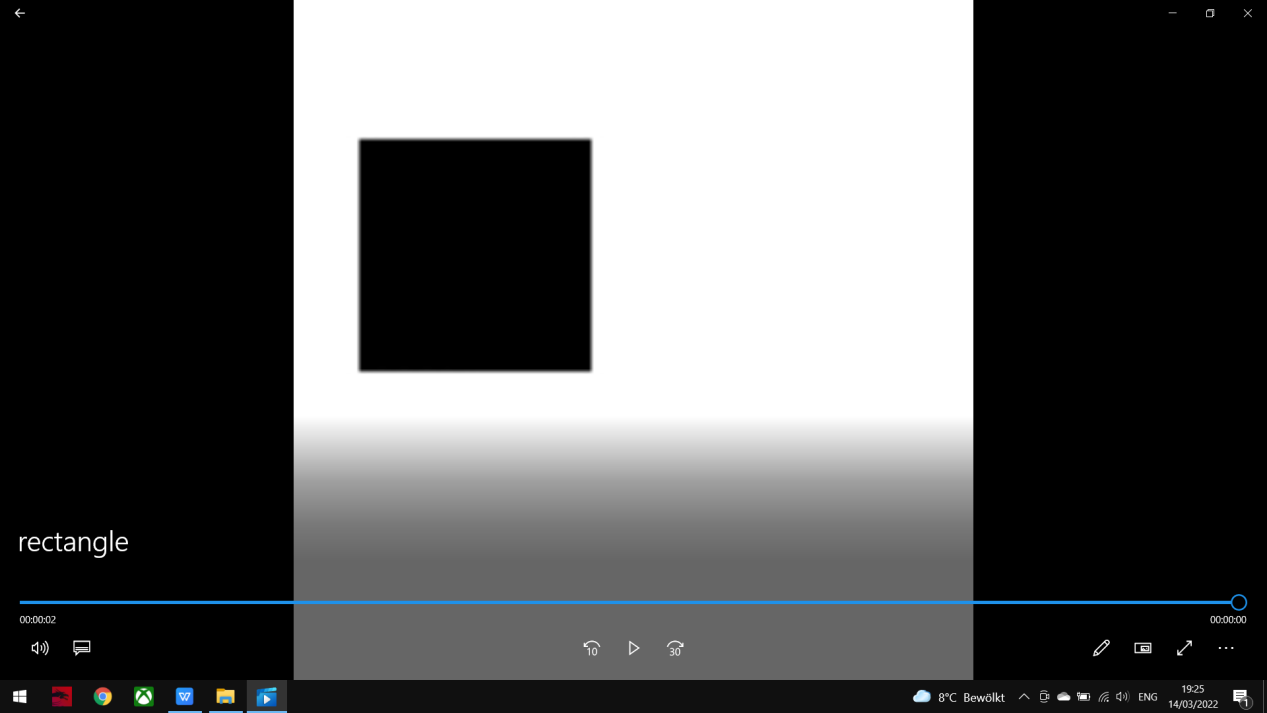


Figure 1b: Rectangle Video

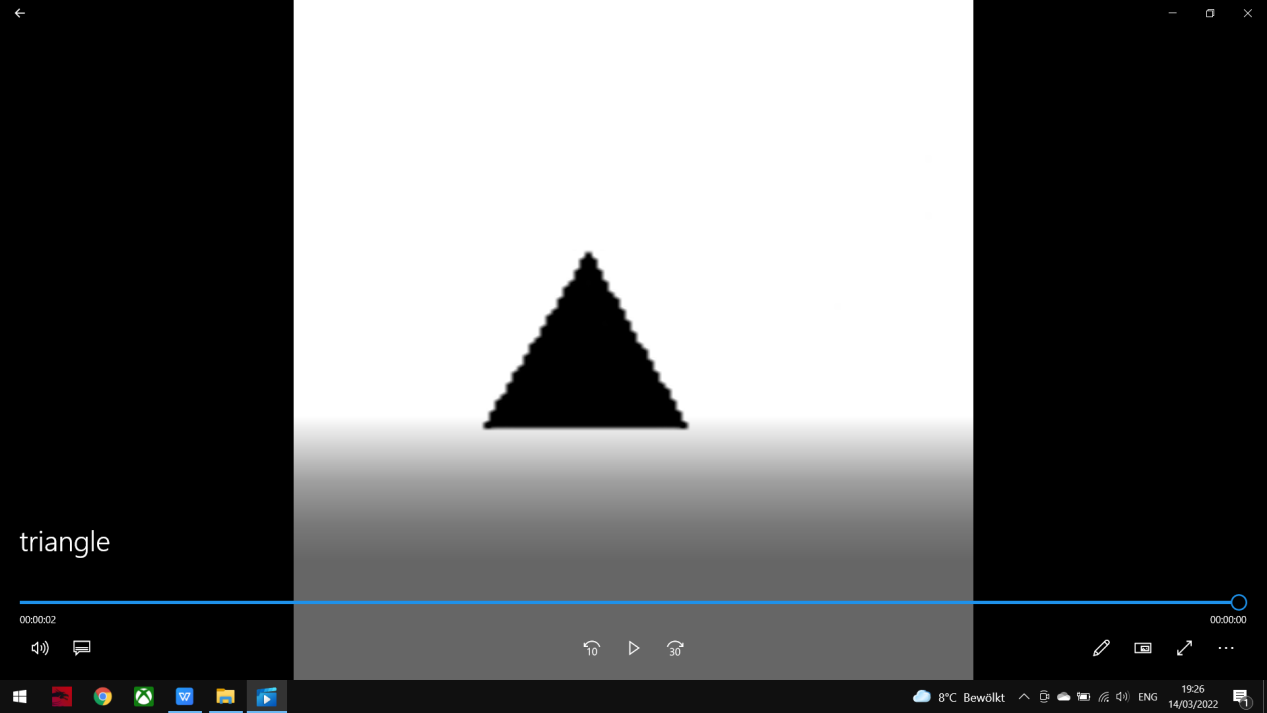


Figure 1c: Triangle Video

Youtube Video input: For other parameters, a random music video on Youtube (“Never Gonna Give You Up” by Rick Astley is used in this case) is edited down to a small 5-second video which is used as an input for the learning process of the program. The editing is done by an online editor called Kapwing, and was put in the following path: GroupSEProject\HTMVideoLearning\VideoLibrary\YoutubeVideo. This video was put under many different conditions, such as by changing the video’s resolutions or the HTM configurations of the program, including Cells per column, Global Inhibition, Max Boost, etc. Another Video input was used (“Frankfurt Video”) edited first into 11 seconds with 184 frames which was a lot of patterns to take care of though for Run1 testing resolution was done in a reasonable time, the video was edited down again into 5 seconds with 85 frames, but it took a very long time to learn such video so Youtube Video was carried on for testing on Run2.

A picture containing text, tree, sign, road

Description automatically generated  
Figure 2: Frankfurt Video



Figure 2a: Youtube Video

Video’s configuration [1]:

int frameWidth = 18;

int frameHeight = 18;

ColorMode colorMode = ColorMode.BLACKWHITE;

Default HTM configuration [2]:

private static HtmConfig GetHTM(int[] inputBits, int[] numColumns)

{

HtmConfig htm = new(inputBits, numColumns)

{

Random = new ThreadSafeRandom(42),

CellsPerColumn = 30,

GlobalInhibition = true,

//LocalAreaDensity = -1,

NumActiveColumnsPerInhArea = 0.02 \* numColumns[0],

PotentialRadius = (int)(0.15 \* inputBits[0]),

//InhibitionRadius = 15,

MaxBoost = 10.0,

//DutyCyclePeriod = 25,

//MinPctOverlapDutyCycles = 0.75,

MaxSynapsesPerSegment = (int)(0.02 \* numColumns[0]),

//ActivationThreshold = 15,

//ConnectedPermanence = 0.5,

// Learning is slower than forgetting in this case.

//PermanenceDecrement = 0.15,

//PermanenceIncrement = 0.15,

// Used by punishing of segments.

};

return htm;

}

For the HTM configuration experiment, each below parameter is modified independently, while the others remained the same as the default HTM Configuration. The value of N/A indicates the experiment required too long an elapsed time without reaching a specific result, so they were cancel. Additionally, only Run2 was applied for this testing. Because Run1 (with Max Cycle = 1000 so the model could reach proper accuracy) required too much run time and there would not be enough time to experiment with various HTM configuration parameters. Additionally, Run2 is superior since the key used for learning is generated from the FrameKey List previousInputs of the current Video, in which one frame information is associated with the information of the whole frame sequence (Video). Compared to Run1 that predicts the frame one by one, the error of the learning process is reduced. Therefore, testing with only Run2 would be a much more optimal choice.

1. **Result:**
2. **Angle Experiments with Simple Shape inputs:**

Run1 was only run with Max Cycle = 10 to see the difference between each case of the experiment, since elapsed time of Run1 with Max Cycle = 1000 took much longer for each case than Run2. This is also the reason that the average accuracy for Run1 is much lower than Run2.

**Run1: Max Cycle = 10**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Angle** | 36 | 72 | 108 | 144 | 180 |
| **Accuracy** | 0.73 | 0.61 | 0.73 | 0.75 | 0.64 |
| **Elapsed time** | 7min | 5min | 7min | 6min | 15min |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Angle** | 216 | 252 | 288 | 324 | 360 |
| **Accuracy** | 0.65 | 0.69 | 0.69 | 0.64 | 0.6 |
| **Elapsed time** | 5min | 6min | 6min | 7min | 17min |

**Run2: Max Cycle = 1000**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Angle** | 36 | 72 | 108 | 144 | 180 |
| **Accuracy** | 0.93 | 0.91 | 0.88 | 0.88 | 0.88 |
| **Elapsed time** | 7min | 10min | 7min | 11min | 11min |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Angle** | 216 | 252 | 288 | 324 | 360 |
| **Accuracy** | 0.75 | 0.71 | 0.77 | 0.74 | 0.8 |
| **Elapsed time** | 5min | 6min | 6min | 7min | 12min |

1. **1. Resolution Experiments with Youtube Video input:**

**Run1: Max Cycle = 10**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Resolution** | 18x18 | 30x30 | 40x40 | 50x50 | 50x25 |
| **Accuracy** | 0.58 | 0.68 | 0.78 | 0.69 | 0.59 |
| **Elapsed time** | 4min | 10min | 21min | 37min | 16min |

**Run2: Max Cycle = 1000**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Resolution** | 18x18 | 30x30 | 40x40 | 50x50 | 50x25 |
| **Accuracy** | 0.92 | 0.98 | 0.90 | 0.90 | 0.98 |
| **Elapsed time** | 6min | 7min | 12min | 7min | 8min |

**2.2. Resolution Experiments with Frankfurt Video input:**

**Run1: Max Cycle = 10**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Resolution** | 18x18 | 30x30 | 40x40 | 30x40 |
| **Accuracy** | 0.28 | 0.33 | 0.31 | 0.25 |
| **Elapsed time** | 19min | 29min | 46min | 36min |

**Run2: Max Cycle = 1000**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Resolution** | 18x18 | 30x30 | 40x40 | 30x40 |
| **Accuracy** | 0.86 | 0.5 | 0.43 | 0.82 |
| **Elapsed time** | 35min | 45min | 48min | 50min |

1. **HTM Configuration Experiments with Youtube Video input:**

**Cells per column:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Parameter** | 30 | 40 | 60 | 80 | 100 |
| **Accuracy** | 0.81 | 0.88 | 0.82 | 0.90 | 0.84 |
| **Elapsed time** | 6min | 7min | 12min | 7min | 8min |

**Switch Global Inhibition = false:**

**Accuracy:** 0.88

**Elapsed Time:** 40min

**Enable LocalAreaDensity:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter** | -1 | 0 | 1 |
| **Accuracy** | 0.85 | 0.91 | N/A |
| **Elapsed time** | 14min | 22min | N/A |

**NumActiveColumnsPerInhArea with modified constant:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Constant** | 0.01 | 0.02 | 0.03 | 0.04 |
| **Accuracy** | 0.95 | 0.86 | 1 | 1 |
| **Elapsed time** | 16min | 10min | 25min | 30min |

**PotentialRadius:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Constant** | 0.15 | 0.2 | 0.3 | 0.1 |
| **Accuracy** | 0.86 | 0.81 | 0.99 | 0.99 |
| **Elapsed time** | 10min | 11min | 32min | 14min |

**Enable InhibitionRadius:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Parameter** | 40 | 50 | 60 | 70 | 80 |
| **Accuracy** | 0.92 | 0.82 | 0.85 | 0.90 | 0.84 |
| **Elapsed time** | 18min | 13min | 21min | 16min | 15min |

**MaxBoost:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Parameter** | 10 | 20 | 30 | 40 | 50 |
| **Accuracy** | 0.81 | 0.88 | 0.81 | 0.81 | 0.85 |
| **Elapsed time** | 6min | 22min | 20min | 15min | 10min |

**Enable DutyCyclePeriod:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Parameter** | 50 | 60 | 70 | 80 | 90 |
| **Accuracy** | 0.86 | 0.89 | 0.89 | 0.81 | 0.96 |
| **Elapsed time** | 10min | 38min | 19min | 12min | 19min |

**Enable MinPctOverlapDutyCycles:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Parameter** | 0.5 | 0.75 | 1.0 | 1.25 | 1.5 |
| **Accuracy** | 0.86 | 0.82 | 0.81 | 0.86 | 0.85 |
| **Elapsed time** | 10min | 21min | 20min | 8min | 13min |

**MaxSynapsesPerSegment with modified constant:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Constant** | 0.02 | 0.03 | 0.04 | 0.05 | 0.1 |
| **Accuracy** | 0.82 | 0.88 | 0.84 | 0.81 | 0.82 |
| **Elapsed time** | 7min | 13min | 28min | 12min | 18min |

**Enable ActivationThreshold = 10: N/A**

**Enable ActivationThreshold = 20: N/A**

->Only with ActivationThreshold = 15

**Enable ConnectedPermanence:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter** | 0.5 | 0.75 | 1.5 |
| **Accuracy** | 0.97 | 0.89 | 0.84 |
| **Elapsed time** | 25min | 36min | 27min |

**Enable PermanenceDecrement = 0.15:**

**Accuracy:** 0.84

**Elapsed Time:** 11min

**Enable PermanenceIncrement = 0.15:**

**Accuracy:** 0.81

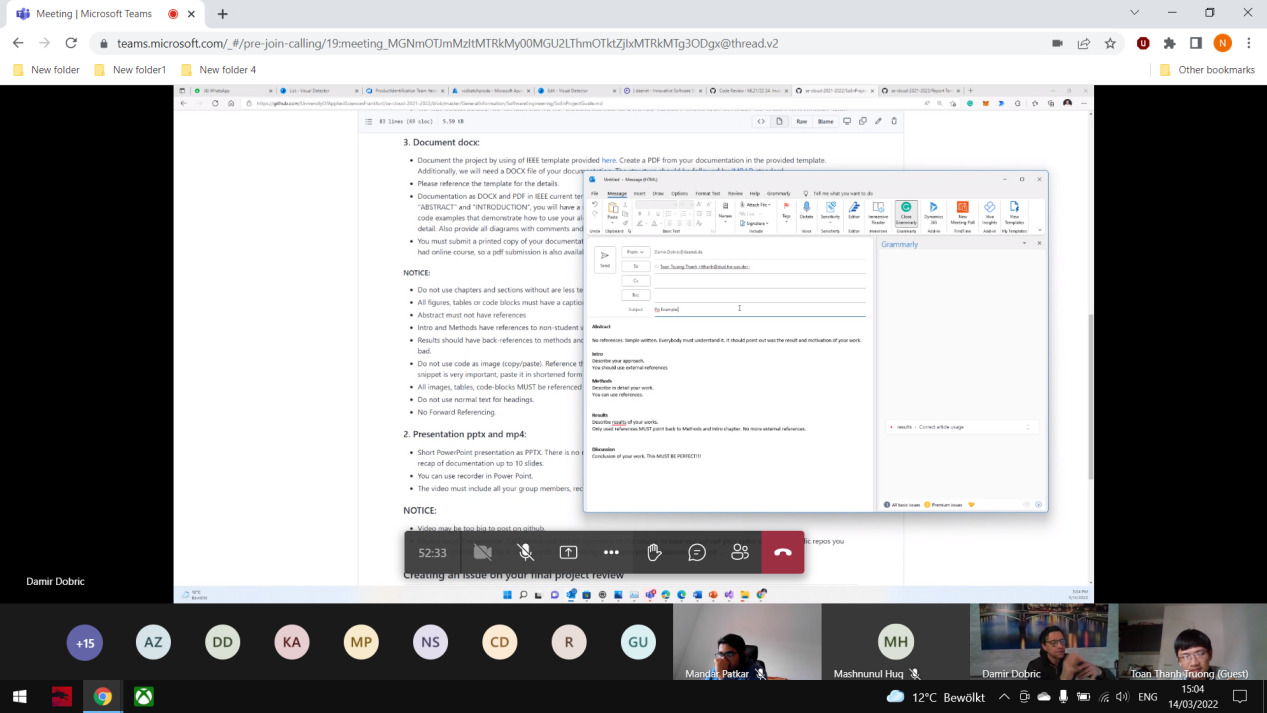
**Elapsed Time:** 14min

**Enable both PermanenceIncrement and PermanenceDecrement:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameters** | 0.15 | 0.2 | 0.25 | 0.3 |
| **Accuracy** | 0.96 | 0.97 | 0.95 | 0.95 |
| **Elapsed time** | 17min | 31min | 12min | 13min |

1. **Discussion:**

Overall, the experiments produced satisfactory results. Their accuracy has the mean value of approximately 0.88, with the lowest value was 0.81, which is decent. Noticeably, a number of modifications in certain parameters results in exceptional accuracy values, some of which were even able to reach 0.99. The elapsed time for these experiments varies from 4 minutes to 46 minutes. Unconventionally, it is observed that high accuracy was not always accompanied by the long elapsed time and vice versa.



1. **Reference:**

Frankfurt Video: https://www.youtube.com/watch?v=V2wofDTu48g&t=112s&ab\_channel=Exploropia

Trimming video:

https://express.adobe.com/tools/trim-video/#