# Test Case 2:

1. The existing Video Learning project was trained with three small video data set know as Circle, Triangle and Rectangle. Now we have generated a new video dataset known as **Pentagon**. Parameters were specified in the code and was used to run the **Run1()** function. By keeping all the parameters stagnant (HTM config) we have trained the HTM model with the SP + TM and documented the accuracy after **20** cycles. We have used 4 datasets in this experiment (Circle, Rectangle, Triangle and Pentagon).

Video Set of Label: **Circle reachs accuracy: 45.714285714285715%**

Video Set of Label: **Rectangle reachs accuracy: 31.428571428571427%**

Video Set of Label: **Pentagon reachs accuracy: 67.1875%**

Video Set of Label: **Triangle reachs accuracy: 85.71428571428571%**

**Accuracy in last Cycle (Cycle 19): 57.511160714285715%**

1. As we have trained the model with larger dataset (4 videos), we have increased the maxCycles parameter gradually to check the impact of this parameter on the average accuracy. So, we have changed the maxCycles parameter to 40 cycles and ran the experiment again. The following results were generated after 40 cycles. Here the frameWeight and frameHeight parameters are kept unchanged which is 18.

Video Set of Label: **Circle reachs accuracy: 60%**

Video Set of Label: **Rectangle reachs accuracy: 42.857142857142854%**

Video Set of Label: **Pentagon reachs accuracy: 71.875%**

Video Set of Label: **Triangle reachs accuracy: 100%**

**Accuracy in last Cycle (Cycle 39): 68.68303571428571%**

This results significantly show that if we increase the maxCycles parameter for larger data set than the average training accuracy increases. To validate our hypothesis, we have increased the specified parameters until the average accuracy reaches its saturation.

1. Afterwards we have trained the model with maxCycles = 70. We have noticed that if we train the model with 4 dataset it shows better training accuracy when the maxCycles is 40. The following results were generated after 70 cycles.

Video Set of Label: **Circle reachs accuracy: 51.42857142857142%**

Video Set of Label: **Rectangle reachs accuracy: 45.714285714285715%**

Video Set of Label: **Pentagon reachs accuracy: 67.1875%**

Video Set of Label: **Triangle reachs accuracy: 88.57142857142857%**

**Accuracy in last Cycle (Cycle 69): 63.225446428571416%**

1. Then we trained our model with **40** maxCycles and we have change the parameters **frameWidth** and **frameHeight** to **20**. The following results were generated after 40 cycles.

Video Set of Label: **Circle reachs accuracy: 57.14285714285714%**

Video Set of Label: **Rectangle reachs accuracy: 45.714285714285715%**

Video Set of Label: **Pentagon reachs accuracy: 68.75%**

Video Set of Label: **Triangle reachs accuracy: 88.57142857142857%**

**Accuracy in last Cycle (Cycle 39): 64.33035714285714%**

1. Then we trained our model with **40** maxCycles and we have change the parameters **frameWidth** and **frameHeight** to **22**. The following results were generated after 40 cycles.

Video Set of Label: **Circle reachs accuracy: 51.42857142857142%**

Video Set of Label: **Rectangle reachs accuracy: 34.285714285714285%**

Video Set of Label: **Pentagon reachs accuracy: 71.875%**

Video Set of Label: **Triangle reachs accuracy: 85.71428571428571%**

**Accuracy in last Cycle (Cycle 39): 60.82589285714286%**

1. Then we trained our model with **40** maxCycles and we have change the parameters **frameWidth** and **frameHeight** to **24**. The following results were generated after 40 cycles.

Video Set of Label: **Circle reachs accuracy: 57.14285714285714%**

Video Set of Label: **Rectangle reachs accuracy: 42.857142857142854%**

Video Set of Label: **Pentagon reachs accuracy: 65.625%**

Video Set of Label: **Triangle reachs accuracy: 85.71428571428571%**

**Accuracy in last Cycle (Cycle 39): 62.83482142857143%**

1. So far, we have manipulated three parameters such as maxCycles, frameWeight and frameHeight. If the maxCycles is 40, frameWeight is 18 and frameHeight is 18 then the training accuracy is optimum for these 4 videosets. So, we trained our model with the specified parameters and checked the prediction process with 4 random frames taken form each of the four Converted video sets folder.
2. We took frame no 3 which is Circle\_circle\_3.png and found the result below. In the case of Circle, it did not successfully predicted the next frame.

Predicted nextFrame: Circle\_circle\_2

Predicted nextFrame: Circle\_circle\_3

Predicted nextFrame: Circle\_circle\_22

Predicted nextFrame: Circle\_circle\_3

Predicted nextFrame: Triangle\_triangle\_18

Predicted nextFrame: Triangle\_triangle\_19

Predicted nextFrame: Triangle\_triangle\_20

Predicted nextFrame: Triangle\_triangle\_21

Predicted nextFrame: Triangle\_triangle\_22

Predicted nextFrame: Triangle\_triangle\_23

Predicted nextFrame: Triangle\_triangle\_24

Predicted nextFrame: Triangle\_triangle\_25

1. We took frame no 7 which is Rectangle\_rectangle\_7.png and found the result below. In the case of rectangle, it predicted the next frame successfully. (**twice**)

**Predicted nextFrame: Rectangle\_rectangle\_8**

Predicted nextFrame: Circle\_circle\_1

Predicted nextFrame: Circle\_circle\_2

Predicted nextFrame: Circle\_circle\_3

**Predicted nextFrame: Rectangle\_rectangle\_8**

Predicted nextFrame: Circle\_circle\_1

Predicted nextFrame: Circle\_circle\_2

Predicted nextFrame: Circle\_circle\_3

1. We took frame no 13 which is Triangle\_triangle\_13.png and found the result below. In the case of Triangle, it predicted the next frame successfully. (**Once**)

**Predicted nextFrame: Triangle\_triangle\_14**

Predicted nextFrame: Triangle\_triangle\_30

Predicted nextFrame: Triangle\_triangle\_31

Predicted nextFrame: Triangle\_triangle\_32

1. We took frame no 21 which is Pentagon\_pentagon\_21.png and found the result below. In the case of Pentagon, it predicted the next frame successfully. (Multiple times)

**Predicted nextFrame: Pentagon\_Pentagon\_22**

**Predicted nextFrame: Pentagon\_Pentagon\_23**

Predicted nextFrame: Pentagon\_Pentagon\_23

Predicted nextFrame: Pentagon\_Pentagon\_21

Predicted nextFrame: Pentagon\_Pentagon\_22

Predicted nextFrame: Pentagon\_Pentagon\_23

Predicted nextFrame: Pentagon\_Pentagon\_23

Predicted nextFrame: Pentagon\_Pentagon\_21

Predicted nextFrame: Pentagon\_Pentagon\_22

Predicted nextFrame: Pentagon\_Pentagon\_23

**Predicted nextFrame: Pentagon\_Pentagon\_23**

**Predicted nextFrame: Pentagon\_Pentagon\_24**