# Test Case 2:

1. The existing Video Learning project was trained with three small video data set know as Circle, Triangle and Rectangle. But now we have generated new video dataset named **Pentagon**. Parameters were specified in the code and was used to run the **Run2()** function. By keeping all the parameters stagnant we have trained the HTM model with the SP + TM (using 4 Videosets) and documented the accuracy after 1000 cycles. After the stable pattern was reached after each video set being trained, we documented the following result.
2. Video Set of Label: **Circle reaches accuracy: 62.06896551724138%**

Cycle**: 1000**  Matches= **18 of 30 Stable pattern reached**

1. Video Set of Label: **Rectangle reaches accuracy: 79.028976345272%**

Cycle: **89** Matches= **22 of 30 Stable pattern reached**

1. Video Set of Label: **Triangle reaches accuracy: 93.10344827586206%**

Cycle**: 57**  Matches= **27 of 30 Stable pattern reached**

1. Video Set of Label: **Pentagon reaches accuracy: 66.0377358490566%**

Cycle**: 1000** Matches= **35 of 45 Stable pattern reached**

**Average Accuracy: 75.0597815%**

1. Afterwards we have changed the parameter **maxCycles = 1200** and kept every other parameter same.
2. Video Set of Label: **Circle reaches accuracy: 79.3103448275862%**

Cycle**: 1200**  Matches= **23 of 30**  **Stable pattern reached**

1. Video Set of Label: **Rectangle reaches accuracy: 82.7586206896551%**

Cycle**: 163** Matches= **24 of 30**  **Stable pattern reached**

1. Video Set of Label: **Triangle reaches accuracy: 93.10344827586206%**

Cycle**: 57**  Matches= **27 of 30 Stable pattern reached**

1. Video Set of Label: **Pentagon reaches accuracy: 66.0377358490566%**

Cycle**: 1000** Matches= **35 of 45 Stable pattern reached**

**Average Accuracy: 79.80253741%**

1. So, we have seen that when we have increased the maxCycle parameter to **1200** the average training accuracy of the model is optimum for the given 4 videosets. Now we have manipulated the frameWeight and frameHeight parameters. We set these parameters to **20** and generated the following results.
2. Video Set of Label: **Circle reaches accuracy: 65.51724137931035%**

Cycle**: 1200**  Matches= **19 of 30**  **Stable pattern reached**

1. Video Set of Label: **Rectangle reaches accuracy: 68.96551724137932%**

Cycle**: 1200** Matches= **20 of 30**  **Stable pattern reached**

1. Video Set of Label: **Triangle reaches accuracy: 86.20689655172413%**

Cycle**: 58**  Matches= **25 of 30 Stable pattern reached**

1. Video Set of Label: **Pentagon reaches accuracy: 62.264150943396224%**

Cycle**: 1200** Matches= **33 of 54 Stable pattern reached**

**Average Accuracy: 70.73845153%**

1. Again, we have increased the frameWeight and frameHeight parameters to **22** and generated the following result.
2. Video Set of Label: **Circle reaches accuracy: 79.3103448275862%**

Cycle**: 1200**  Matches= **23 of 30**  **Stable pattern reached**

1. Video Set of Label: **Rectangle reaches accuracy: 89.6551721379311%**

Cycle**: 115** Matches= **26 of 30**  **Stable pattern reached**

1. Video Set of Label: **Triangle reaches accuracy: 93.10344827586206%**

Cycle**: 57**  Matches= **27 of 30 Stable pattern reached**

1. Video Set of Label: **Pentagon reaches accuracy: 67.9245283018868%**

Cycle**: 1200** Matches= **36 of 54 Stable pattern reached**

**Average Accuracy: 82.49837339%**

1. So, we have seen that **1200** max cycles and when the frameHeight, frameWeight is **22** gives us the highest training average accuracy. We have taken one random **frame** from each of the **Converted** video set folders and experimented the prediction section of the code when the **maxCycles = 1200.**
2. We took frame no 13 which is Circle\_circle\_13.png and found the result below. The model has successfully predicted the series of next frame sequences.

**Predicted Series:**

**Circle\_circle\_13-Circle\_circle\_14-Circle\_circle\_15-Circle\_circle\_16-Circle\_circle\_17-Circle\_circle\_18-Circle\_circle\_19-Circle\_circle\_20-Circle\_circle\_21-Circle\_circle\_22-Circle\_circle\_23-Circle\_circle\_24-Circle\_circle\_25-Circle\_circle\_26-Circle\_circle\_27-Circle\_circle\_28-Circle\_circle\_29-Circle\_circle\_0-Circle\_circle\_1-Circle\_circle\_2-Circle\_circle\_3-Circle\_circle\_4-Circle\_circle\_5-Circle\_circle\_6-Circle\_circle\_7-Circle\_circle\_8-Circle\_circle\_9-Circle\_circle\_10-Circle\_circle\_11-Circle\_circle\_12**

1. We took frame no 17 which is Rectangle\_rectangle\_17.png and found the result below. The model has initially predicted that the frame belongs to the rectangle video set and then formed the sequences of the frames from rectangle videoset. Afterwards it has predicted all the frames to be wrong.

**Predicted Series:**

**Rectangle\_rectangle\_12-Rectangle\_rectangle\_13-Rectangle\_rectangle\_14-Rectangle\_rectangle\_15-Rectangle\_rectangle\_16-Rectangle\_rectangle\_17-Rectangle\_rectangle\_18-Rectangle\_rectangle\_19-Rectangle\_rectangle\_20-Rectangle\_rectangle\_21-Rectangle\_rectangle\_22-Rectangle\_rectangle\_23-Rectangle\_rectangle\_24-Rectangle\_rectangle\_25-Rectangle\_rectangle\_26-Rectangle\_rectangle\_27-Rectangle\_rectangle\_28-Rectangle\_rectangle\_29-Rectangle\_rectangle\_0-Rectangle\_rectangle\_1-Rectangle\_rectangle\_2-Rectangle\_rectangle\_3-Rectangle\_rectangle\_4-Rectangle\_rectangle\_5-Rectangle\_rectangle\_6-Rectangle\_rectangle\_7-Rectangle\_rectangle\_8-Rectangle\_rectangle\_9-Rectangle\_rectangle\_10-Rectangle\_rectangle\_11**

Predicted Series:

Circle\_circle\_0-Circle\_circle\_1-Circle\_circle\_2-Circle\_circle\_3-Circle\_circle\_4-Circle\_circle\_5-Circle\_circle\_6-Circle\_circle\_7-Circle\_circle\_8-Circle\_circle\_9-Circle\_circle\_10-Circle\_circle\_11-Circle\_circle\_12-Circle\_circle\_13-Circle\_circle\_14-Circle\_circle\_15-Circle\_circle\_16-Circle\_circle\_17-Circle\_circle\_18-Circle\_circle\_19-Circle\_circle\_20-Circle\_circle\_21-Circle\_circle\_22-Circle\_circle\_23-Circle\_circle\_24-Circle\_circle\_25-Circle\_circle\_26-Circle\_circle\_27-Circle\_circle\_28

1. We took frame no 4 which is Triangle\_triangle\_4.png and found the result below. The model has initially predicted that the frame belongs to the triangle video set and then formed the sequences of the frames from triangle videoset. Afterwards it has predicted all the frames to be wrong.

**Predicted Series:**

**Triangle\_triangle\_6-Triangle\_triangle\_7-Triangle\_triangle\_8-Triangle\_triangle\_9-Triangle\_triangle\_10-Triangle\_triangle\_11-Triangle\_triangle\_12-Triangle\_triangle\_13-Triangle\_triangle\_14-Triangle\_triangle\_15-Triangle\_triangle\_16-Triangle\_triangle\_17-Triangle\_triangle\_18-Triangle\_triangle\_19-Triangle\_triangle\_20-Triangle\_triangle\_21-Triangle\_triangle\_22-Triangle\_triangle\_23-Triangle\_triangle\_24-Triangle\_triangle\_25-Triangle\_triangle\_26-Triangle\_triangle\_27-Triangle\_triangle\_28-Triangle\_triangle\_29-Triangle\_triangle\_0-Triangle\_triangle\_1-Triangle\_triangle\_2-Triangle\_triangle\_3-Triangle\_triangle\_4-Triangle\_triangle\_5**

Predicted Series:

Pentagon\_Pentagon\_0-Pentagon\_Pentagon\_1-Pentagon\_Pentagon\_2-Pentagon\_Pentagon\_3-Pentagon\_Pentagon\_4-Pentagon\_Pentagon\_5-Pentagon\_Pentagon\_6-Pentagon\_Pentagon\_7-Pentagon\_Pentagon\_8-Pentagon\_Pentagon\_9-Pentagon\_Pentagon\_10-Pentagon\_Pentagon\_11-Pentagon\_Pentagon\_12-Pentagon\_Pentagon\_13-Pentagon\_Pentagon\_14-Pentagon\_Pentagon\_15-Pentagon\_Pentagon\_16-Pentagon\_Pentagon\_17-Pentagon\_Pentagon\_18-Pentagon\_Pentagon\_19-Pentagon\_Pentagon\_20-Pentagon\_Pentagon\_21-Pentagon\_Pentagon\_22-Pentagon\_Pentagon\_23-Pentagon\_Pentagon\_24-Pentagon\_Pentagon\_25-Pentagon\_Pentagon\_26-Pentagon\_Pentagon\_27-Pentagon\_Pentagon\_28-Pentagon\_Pentagon\_29-Pentagon\_Pentagon\_30-Pentagon\_Pentagon\_31-Pentagon\_Pentagon\_32-Pentagon\_Pentagon\_33-Pentagon\_Pentagon\_34-Pentagon\_Pentagon\_35-Pentagon\_Pentagon\_36-Pentagon\_Pentagon\_37-Pentagon\_Pentagon\_38-Pentagon\_Pentagon\_39-Pentagon\_Pentagon\_40-Pentagon\_Pentagon\_41-Pentagon\_Pentagon\_42-Pentagon\_Pentagon\_43-Pentagon\_Pentagon\_44-Pentagon\_Pentagon\_45-Pentagon\_Pentagon\_46-Pentagon\_Pentagon\_47-Pentagon\_Pentagon\_48-Pentagon\_Pentagon\_49-Pentagon\_Pentagon\_50-Pentagon\_Pentagon\_51-Pentagon\_Pentagon\_52

1. We took frame no 4 which is Pentagon\_pentagon\_4.png and found the result below. The model has successfully predicted the sequences of next frames.

**Predicted Series:**

**Pentagon\_Pentagon\_5-Pentagon\_Pentagon\_6-Pentagon\_Pentagon\_7-Pentagon\_Pentagon\_8-Pentagon\_Pentagon\_9-Pentagon\_Pentagon\_10-Pentagon\_Pentagon\_11-Pentagon\_Pentagon\_12-Pentagon\_Pentagon\_13-Pentagon\_Pentagon\_14-Pentagon\_Pentagon\_15-Pentagon\_Pentagon\_16-Pentagon\_Pentagon\_17-Pentagon\_Pentagon\_18-Pentagon\_Pentagon\_19-Pentagon\_Pentagon\_20-Pentagon\_Pentagon\_21-Pentagon\_Pentagon\_22-Pentagon\_Pentagon\_23-Pentagon\_Pentagon\_24-Pentagon\_Pentagon\_25-Pentagon\_Pentagon\_26-Pentagon\_Pentagon\_27-Pentagon\_Pentagon\_28-Pentagon\_Pentagon\_29-Pentagon\_Pentagon\_30-Pentagon\_Pentagon\_31-Pentagon\_Pentagon\_32-Pentagon\_Pentagon\_33-Pentagon\_Pentagon\_34-Pentagon\_Pentagon\_35-Pentagon\_Pentagon\_36-Pentagon\_Pentagon\_37-Pentagon\_Pentagon\_38-Pentagon\_Pentagon\_39-Pentagon\_Pentagon\_40-Pentagon\_Pentagon\_41-Pentagon\_Pentagon\_42-Pentagon\_Pentagon\_43-Pentagon\_Pentagon\_44-Pentagon\_Pentagon\_45-Pentagon\_Pentagon\_46-Pentagon\_Pentagon\_47-Pentagon\_Pentagon\_48-Pentagon\_Pentagon\_49-Pentagon\_Pentagon\_50-Pentagon\_Pentagon\_51-Pentagon\_Pentagon\_52-Pentagon\_Pentagon\_53-Pentagon\_Pentagon\_0-Pentagon\_Pentagon\_1-Pentagon\_Pentagon\_2-Pentagon\_Pentagon\_3-Pentagon\_Pentagon\_4**