SUPERVISED HOPFIELD NEURAL NETWORKS FOR SOLVING NON-VERBAL REASONING PROBLEMS

This research paper was completed via collective efforts of

- **Chirumamilla Satya Keerthana (210290)**
- **Madhuri Macharla (210568)**
- **Shobhit Sharma (210992)**

1 Methodology

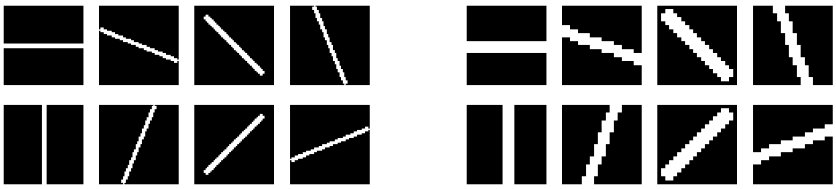
In this paper, we are dealing with two types of non-verbal reasoning - Finding odd one out from the given test images and generating the next pattern from the sequence of images given.

1.1 Odd-one-out

"Odd one out" tasks is a big domain within non-verbal reasoning which challenges individuals or systems to identify an element in a group that differs from others. This is based on specific structural properties of figures but in a confusing manner usually. In this paper, we extend this concept to associative memory models, specifically Hopfield Networks to address two main scenarios: identifying the odd one out among lines and among figures.

1.1.1 Odd One Out among LINES

As the first set, we will be solving the non-verbal reasoning problems associated with finding Odd One Out among Lines. We are going to test the capability of Hop-field networks to recover similar lines on set of test patterns for anomaly detection, see Fig 2.



(a) Training Data for 50x50 (b) Training Data for 20x20

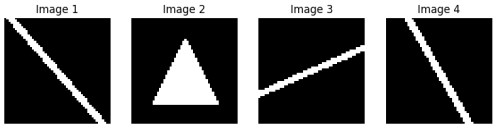
Figure 1: Training Data Overview

For the first set logic of test patterns, we are going to train the network by training it with generated lines of different angles. We use Pseudo Inverse Learning [1] for storing the training images into the network as this method has been shown to increase the capacity of the network.

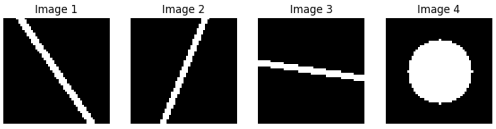
$$W = P^T(PP^T)^{-1}P, \tag{1}$$

where P represents the matrix of flattened training patterns. This approach ensures that the weight matrix W optimally stores the training patterns and facilitates their accurate recall. To avoid self-connections in neurons, the diagonal elements of W were set to zero, thereby improving the network's stability and recall performance.

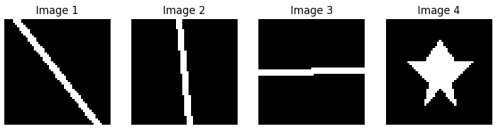
Training set consists of binary images of 8 lines which divide the 4 quadrants into 16 equal portions, shown in Fig 1. The test data consists of 1000 test sequence sets to ensure comprehensive evaluation. The test sequence is composed of 4 images which consists of 3 random lines at any angle and one odd figure which is chosen from square, star, triangle, circle, etc.



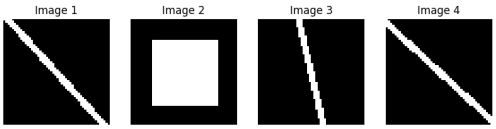
(a) Test Sequence 1



(b) Test Sequence 2



(c) Test Sequence 3



(d) Test Sequence 4

Figure 2: Test Data Sequences for Odd One Out among Lines (50x50)

1.1.2 Odd One Out among FIGURES

For the second part of this paper, we will address the Odd One Out problems with an enhanced approach by significantly expanding our training set to include a wider range of figures. This step forward aims to improve the generalization capability of the model in identifying discrepancies among diverse visual patterns shown in Fig 4.

Figure 3: Training Data

The network architecture almost remains the same as we use the same learning techniques. Training set consists of 13 binary images of varied figures. (Fig 3). The test data consists of 100 test sequence sets to ensure comprehensive evaluation. The test sequence is composed of 4 images which consists of 3 figures of 1 type and one odd figure which is any other figure. To increase the variability, we skew the images in the test patterns using OpenCV at varied angles. This helps to generalize the approach further. See (Fig 4) for random test sequences generated.

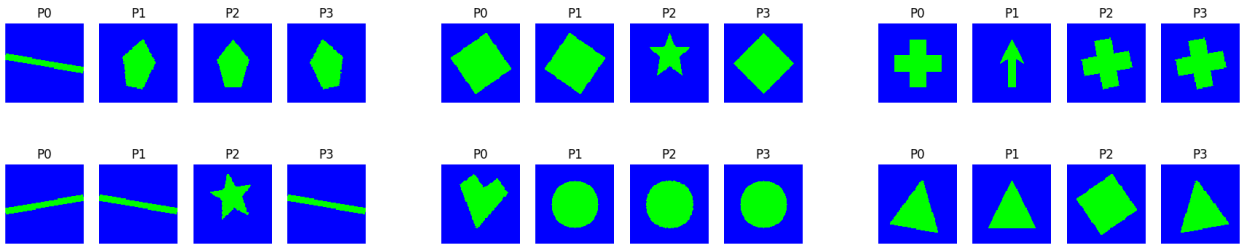


Figure 4: Overview of Test Sequences for General Anomaly detection

1.2 Finding the next pattern

As a second part of the paper, we focus on the non-verbal reasoning type of finding the next pattern when a few sample images are provided. As shown in Figure 5, we have created a dataset where each image is associated with a unique state number. These states form a cyclic sequence, i.e, after the state reaches a state 7, the next step is stage 0. We have chosen this type of dataset to have a finite state space and have a variety of patterns possible with the given type of dataset. Our aim is to build a model which recognizes the states in the test images and by understanding the pattern among the test sequence, predict the next image.

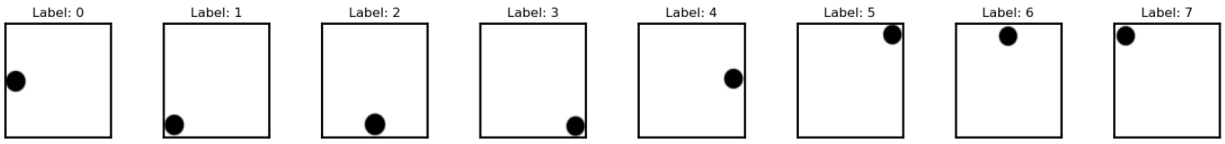


Figure 5: Training Data, shown with a bounding box for the images

As a part of the model building, we use a Hopfield Network. We train the Hopfield network with our dataset to recognize and recall states of the input sequence images. The Hopfield network uses associate memory to retrieve the patterns effectively. Once the states of the test images are retrieved, next we predict the next state by assuming that the sequence follows a logical progression cyclically. We assume that the states lie in an Arithmetic Progression (AP) or that the difference of states lies in an AP, both having cyclic nature. This assumption is valid for our dataset because with the limited number of states, it assumes to cover all possible type of patterns. Hence, we assume that the logical progression either follows an AP or the difference of states follows an AP.

The two progression types are :

- **Arithmetic Progression (AP) :** The difference between successive states is a constant difference.
  Example: The states 7,0,1 are in AP. We arrive at each step by adding +1 to the state in a cyclic fashion (here taking modulo by 8)
- **Difference of numbers in AP :** If the given sequence does not lie in AP, we assume that their difference lies in AP. Example: The states 1,2,4 have their numbers in AP. The difference between 2nd and 1st is 1 and between 3rd and 2nd is 2, hence the difference between 4th and 3rd is 3. Hence, the next state predicted would be $(4 + 3)\%8 = 7$.

By examining the pattern in the states, the next pattern can be predicted using this logic. After this prediction of states, by mapping the state to the Hopfield network to recall the image gives the next image in the pattern. This particular idea of prediction of the next pattern assuming that it follows a logical progression solves a variety of problems - such as extending to multiple states, having rotation of a group of objects and the case when the number of states increases and generalizing this among other shapes. All these problems have been dealt specifically in the upcoming subsections.

1.2.1 Rotation of Multiple States

In this case, we introduce multiple states in a single image which rotate in coordination. The Hopfield network identifies this combined arrangement and predicts the positions of all the states. These states are considered as a set and using the progression rules in a supervised fashion, the network predicts the pattern of this coordinated rotation. The method is based on identifying a constant common difference between the sets. After recognizing the common difference, the network applies it to the third pattern in the sequence which generates a prediction. Using this prediction, the network retrieves the predicted image. Table 1 shows few sample sequences for the given case.

Sequence	Common Difference	Pattern to be Predicted
$\{4,5\} \rightarrow \{3,4\} \rightarrow \{2,3\}$	-1	$\{1,2\}$
$\{1,2,4\} \rightarrow \{3,4,6\} \rightarrow \{5,6,8\}$	2	$\{8,2,7\}$
$\{3,4,5,7,8\} \rightarrow \{2,3,4,6,7\} \rightarrow \{1,2,3,5,6\}$	-1	$\{1,2,4,5,8\}$
$\{1,2,3,4,5,6\} \rightarrow \{2,3,4,5,6,7\} \rightarrow \{3,4,5,6,7,8\}$	1	$\{1,4,5,6,7,8\}$

Table 1: Example Sequences

1.2.2 Addition of New States

In this sequence, the number of states increases at each step, with new states added to the existing pattern. Here, we have considered the case where at each step only one state is being introduced as there will be an overlap in states when the number of states is increased by more than one. The sequence begins with n circles, where n ranges from 1 to 8. The hopfield network identifies the number of states in each of these three input images which are treated as a set. The pattern is predicted based on adding a new state to the set according to the progression rules applied to the most recently

added state in the previous pattern. When presented with the updated states, the model retrieves the predicted image. Table 2 shows few sample sequences for the given case.

Sequence	Common Difference	Pattern to be Predicted
$\{2\} \rightarrow \{2, 8\} \rightarrow \{2, 6, 8\}$	-2	$\{2, 4, 6, 8\}$
$\{6, 7\} \rightarrow \{1, 6, 7\} \rightarrow \{1, 3, 6, 7\}$	2	$\{1, 3, 5, 6, 7\}$
$\{5, 7, 8\} \rightarrow \{1, 5, 7, 8\} \rightarrow \{1, 2, 5, 7, 8\}$	1	$\{1, 2, 3, 5, 7, 8\}$
$\{3, 5, 6, 7\} \rightarrow \{3, 5, 6, 7, 8\} \rightarrow \{1, 3, 5, 6, 7, 8\}$	1	$\{1, 2, 3, 5, 6, 7, 8\}$

Table 2: Example Sequences

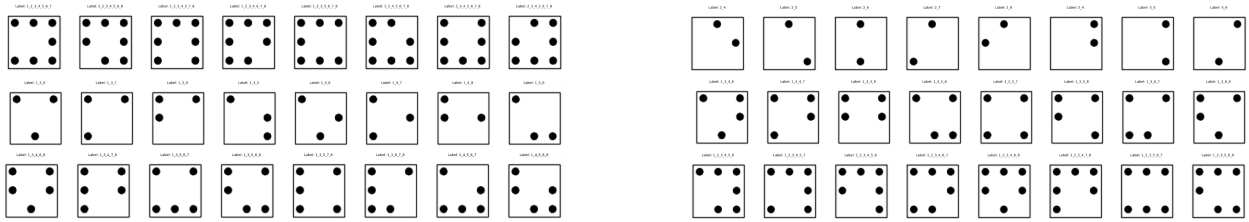
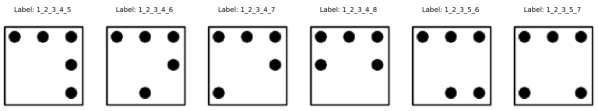


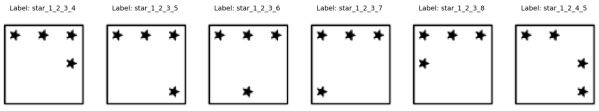
Figure 6: Training dataset for Rotation of multiple states and Addition of new states

1.2.3 Generalized Prediction Across Various Shapes

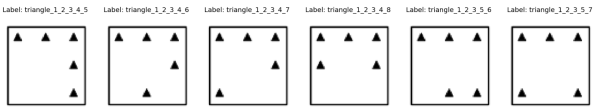
In this case, we train the model on four different shapes (circle, square, triangle and star) and test it on all possible sequences for each shape. The network identifies the current shape and its position in each pattern. Using the progression rule - which incorporates all previous cases, the network predicts the expected pattern accurately by considering both the shape and the state. This method effectively manages multiple attributes within a single framework.



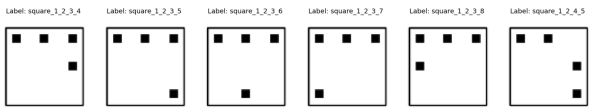
(a) Training dataset of circles



(b) Training dataset of stars



(c) Training dataset of triangles



(d) Training dataset of squares

Figure 7: Training datasets for different shapes

2 Results

2.1 Odd-one-out

2.1.1 Among lines

Results are generated on the basis of overlapping of the train-sequences with recalled pattern. The number of iterations of the network evolution are fixed to 10.

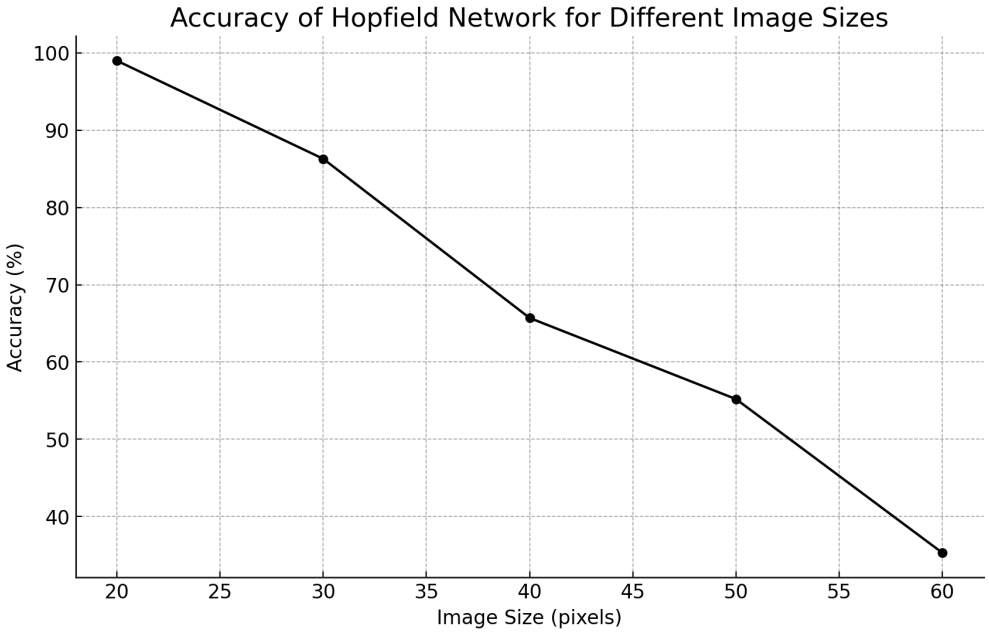
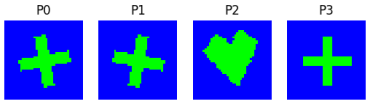


Figure 8: Accuracy v/s Image Size for same Thickness of Lines

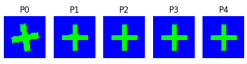
We get a maxima of 99.2% accuracy in test sequences which are of size 20x20. Observe in Fig 8 that as the size of images increases, the number of neurons increase but as the thickness of lines remain the same, the visual features in larger images with thin lines may become less distinguishable, as the network now has to process more background pixels relative to the line pixels. Thus the accuracy of our Hop-field model to classify the odd one out among lines dilutes as the size of image increases. Hence, Hop-field Networks perform nicely at finding odd one out among lines.

2.1.2 Among Shapes

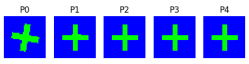
Results are generated on the basis of overlapping of the train-sequences with recalled pattern. The number of iterations of the network evolution are fixed to 5. The pattern size is fixed to 30x30, and the skew is varied at random between -10, 0, +10 for applying to test data.



(a) Test Sequence 1

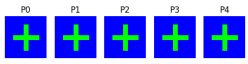


(b) Network Evolution for Pattern 1



(c) Network Evolution for Pattern 2

(d) Network Evolution for Pattern 3



(e) Network Evolution for Pattern 4

Figure 9: Test Data Sequences for Odd One Out among Lines (30x30)

A staggering 100% accuracy is achieved when testing with random skew from [-10,0,+10]. It was observed that as the skew angle is increased, the accuracy of the network in correct anomaly detection decreased significantly over 40% (Plot 10). As the skew angle increases from 10° to 20° a sharp decline indicates that even a moderate increase in skew can impact the network's ability to correctly identify the odd one out. As angle increases, patterns such as star, plus, square, triangle reach initial shape again due to uniformity which leads to stagnation in accuracy.

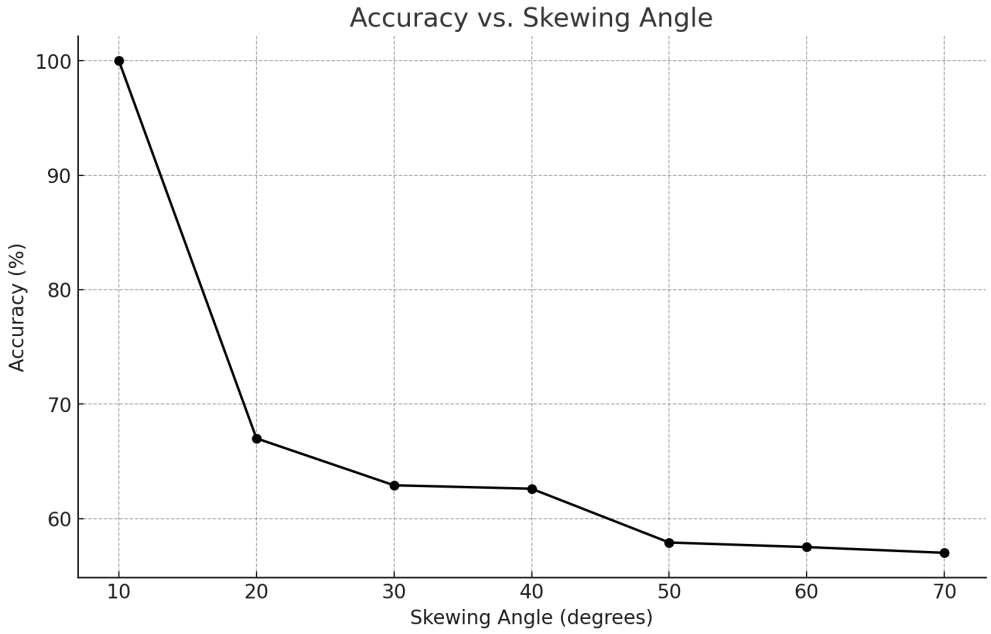


Figure 10: Test Accuracy v/s Skew Angle for Test Patterns

2.2 Sequence generation

After training the dataset to detect the states and follow the supervised logic of either AP or difference of numbers in AP cyclically, we have evaluated the model's performance on a set of 3 test images. This achieved an accuracy of 100% which confirms with the network's robust pattern recognition ability. Figure 11 shows a few sample test images and their predictions from the dataset generated. To further test the network's ability to recall, we have introduced the noise by flipping the images in each test image. The size of each image is 100 x 100 and we varied the number of flips from 200 to 5000 flips in steps of 200 as shown in Figure 12.

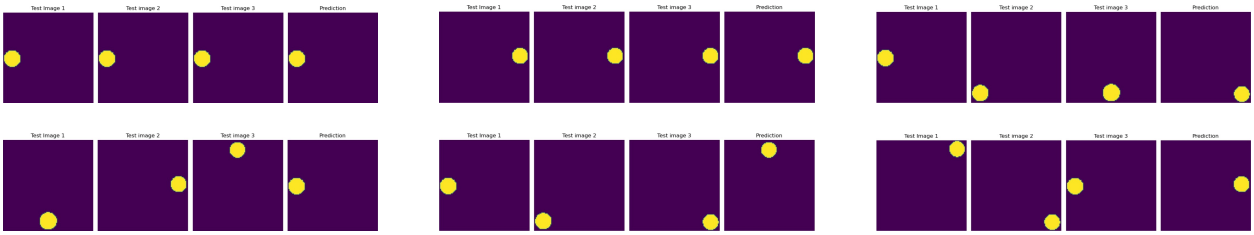


Figure 11: Prediction for a few sample test cases

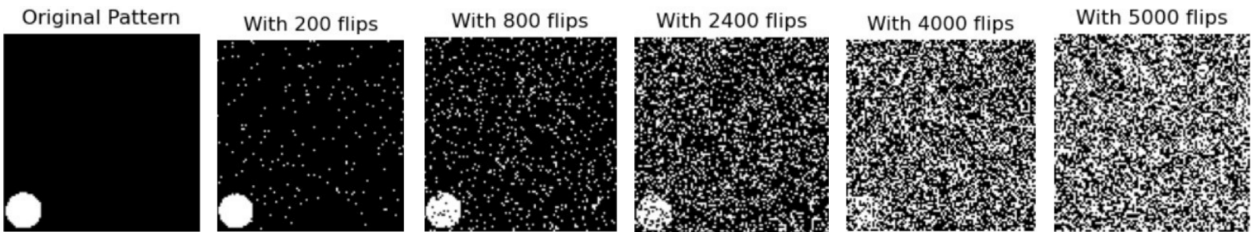


Figure 12: Test Images with varying noise level

The results reveal that the Hopfield network can successfully recall and predict the next patterns with supervised logic even with a substantial noise. At 4000 pixels (40% noise), the network still achieves a 100% accuracy, proving the recall capacity. But after this point, the model decreases it's performance and at 5000 flips (50 % noise), the accuracy drops to zero. Figure 13 shows the plot between Number of flips (noise) and Accuracy in %. The results suggest that this supervised Hopfield model is well-suited for high noise environments in our problem.

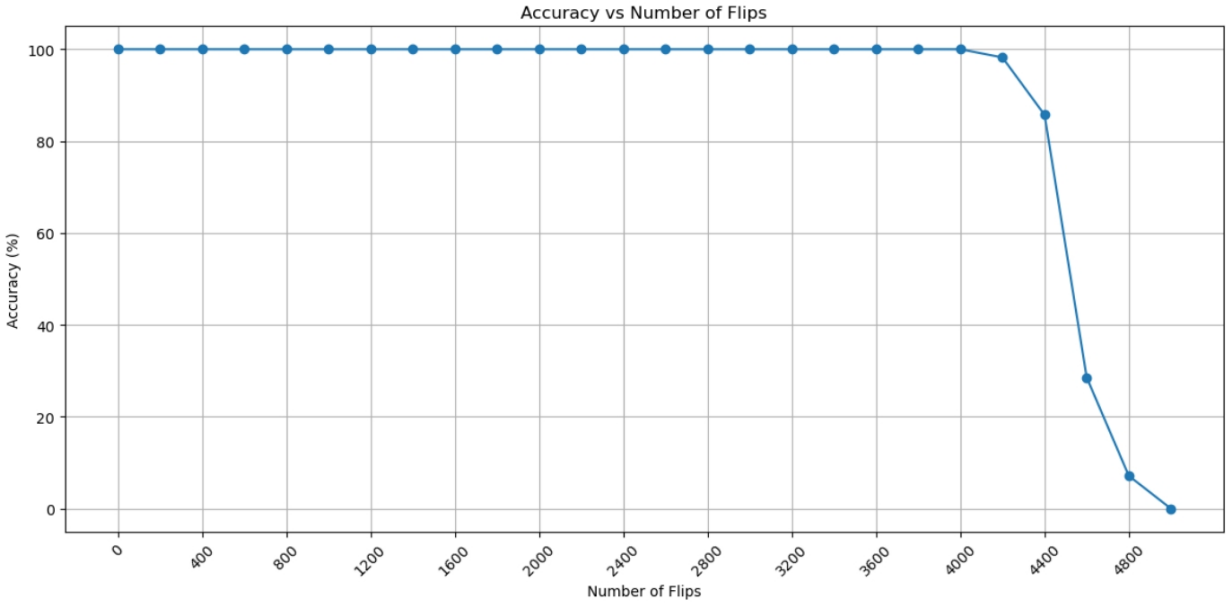
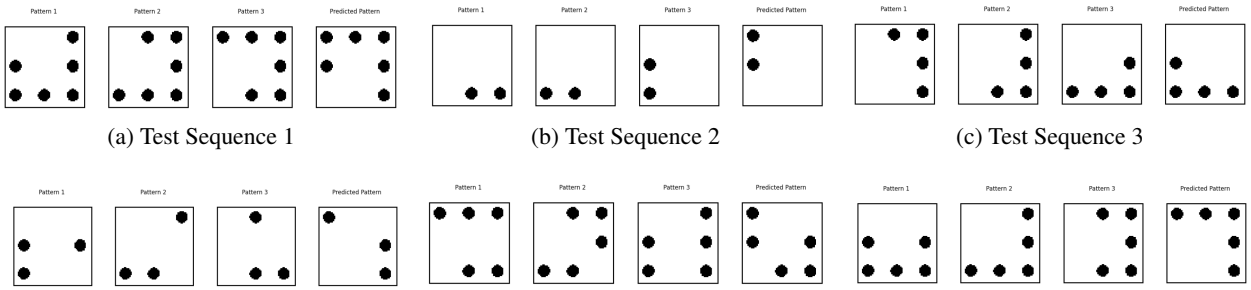


Figure 13: Accuracy Vs Number of flips

2.2.1 Rotation of multiple states

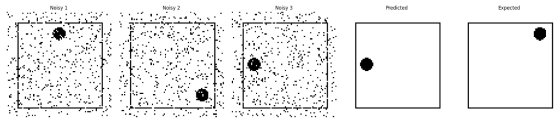
In this case, three test images of a particular sequence were given as input, in which each image contains multiple states. The model predicted the next pattern of the sequence. We tested this on all possible test sequences and got 100 percent accuracy. Figure 14 shows the results when tested with few sequences without noise. We then introduced noise and monitored the performance of hopfield. As depicted in figure 15, the model was predicting correct pattern until 2000 flips. From the graph in figure 16, we observe that until 2500 flips, we achieve 100 percent accuracy. As the number of flips increase, the accuracy drops gradually and reaches 0 when the flips are 4000. This demonstrates the robustness of network in noisy environment.



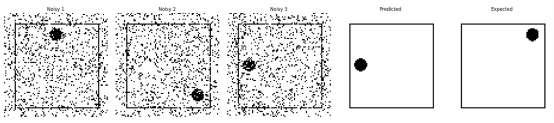
(a) Test Sequence 1 (b) Test Sequence 2 (c) Test Sequence 3

(d) Test Sequence 4 (e) Test Sequence 5 (f) Test Sequence 6

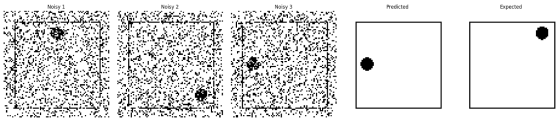
Figure 14: Test Sequences with no noise



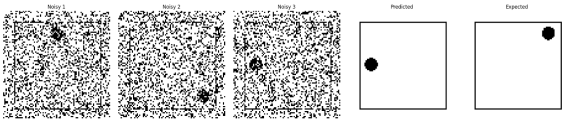
(a) Test Sequence with 500 flips



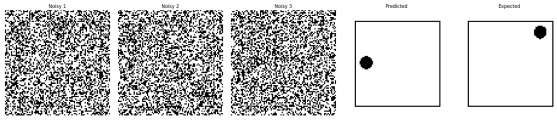
(b) Test Sequence with 1000 flips



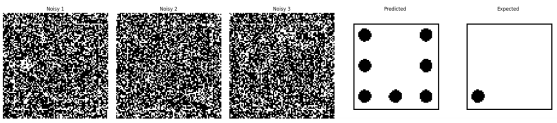
(c) Test Sequence with 1500 flips



(d) Test Sequence with 2000 flips



(e) Test Sequence with 3000 flips



(f) Test Sequence with 5000 flips

Figure 15: Test sequences when noise is introduced

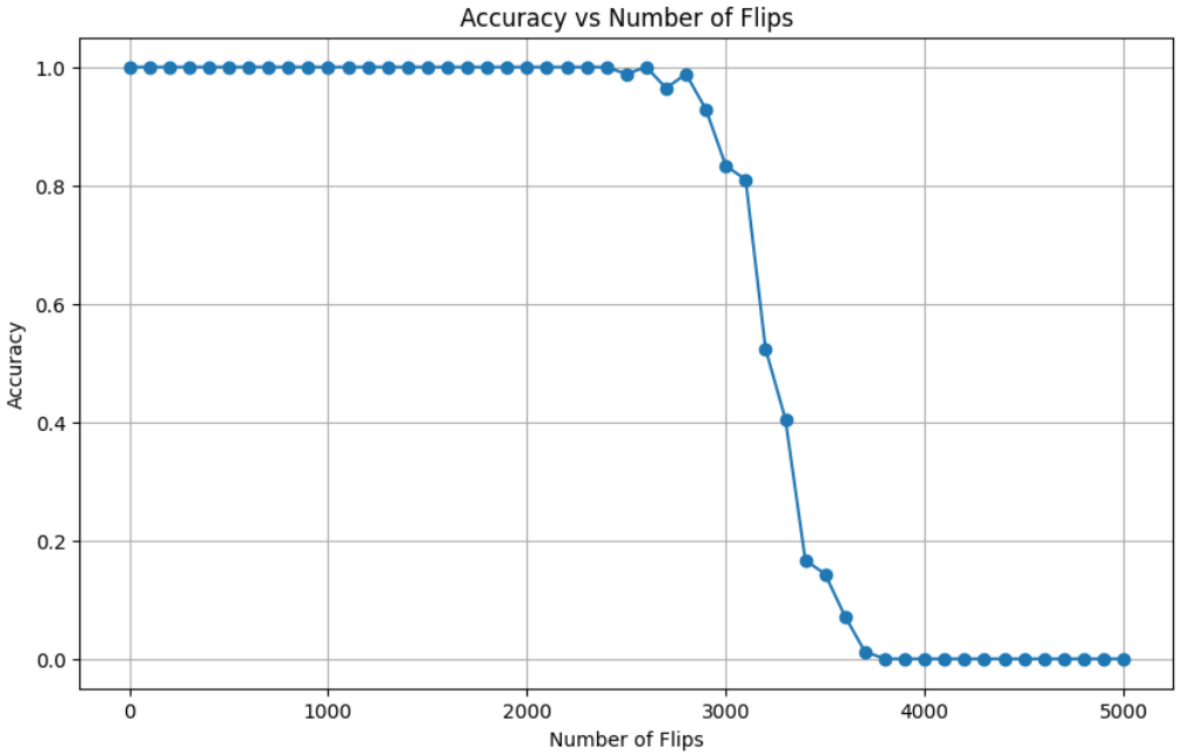
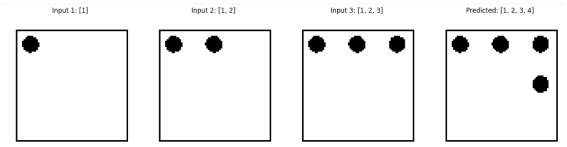


Figure 16: Accuracy Vs Number of flips

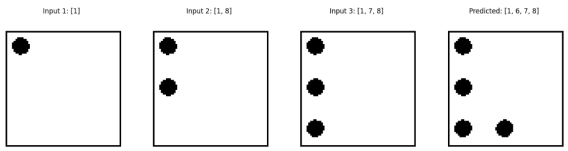
2.2.2 Addition of new states

We gave three test images of a particular sequence as input. The number of states will be incremented in each step of the sequence. The model predicted the fourth pattern of the sequence. We've tested the model on all possible test sequences and got 100% accuracy. Figure 17 shows results for some sequences without noise. Then noise was introduced to check how the Hopfield network performs in a noisy environment. As shown in Figure 18, the model predicted correctly up to 2000 flips. From Figure 19, we see that the accuracy stayed at 100% up to 2500 flips. However, as the number of

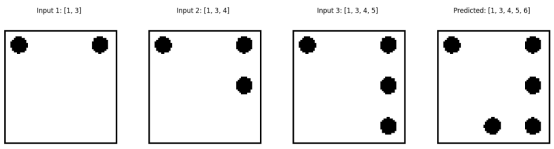
flips increased, the accuracy slowly dropped to 0 at 4000 flips. This shows that the network works well even in noisy conditions.



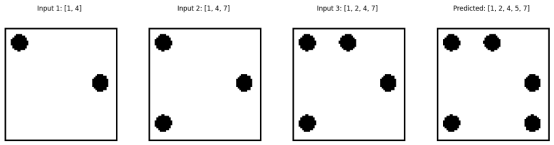
(a) Test Sequence 1



(b) Test Sequence 2

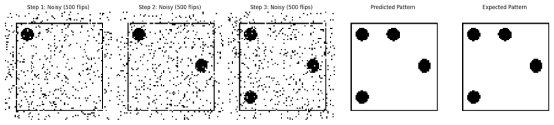


(c) Test Sequence 3

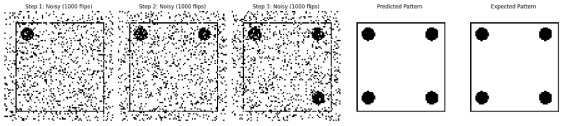


(d) Test Sequence 4

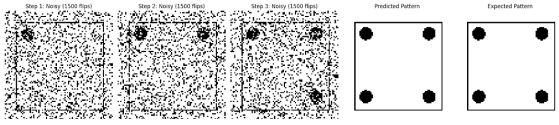
Figure 17: Test sequences when noise is not introduced



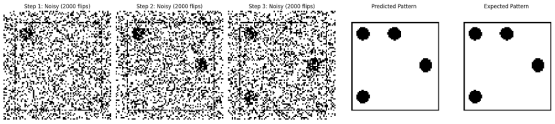
(a) Test Sequence with 500 flips



(b) Test Sequence with 1000 flips



(c) Test Sequence with 1500 flips



(d) Test Sequence with 2000 flips

Figure 18: Test sequences when noise is introduced

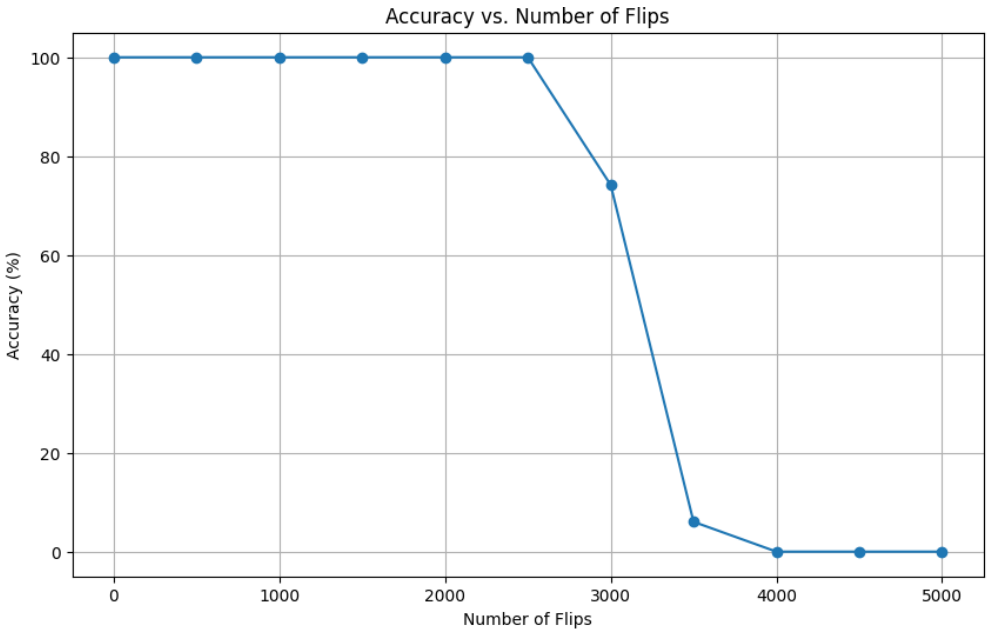


Figure 19: Accuracy Vs Number of flips

2.2.3 Generalized prediction across shapes

The model was trained to predict the shape as well as the next pattern in the sequence. We have observed similar results as of that with the training with single shape case. Despite the close resemblance between the shapes, the model still predicts with 100% accuracy in the baseline model.

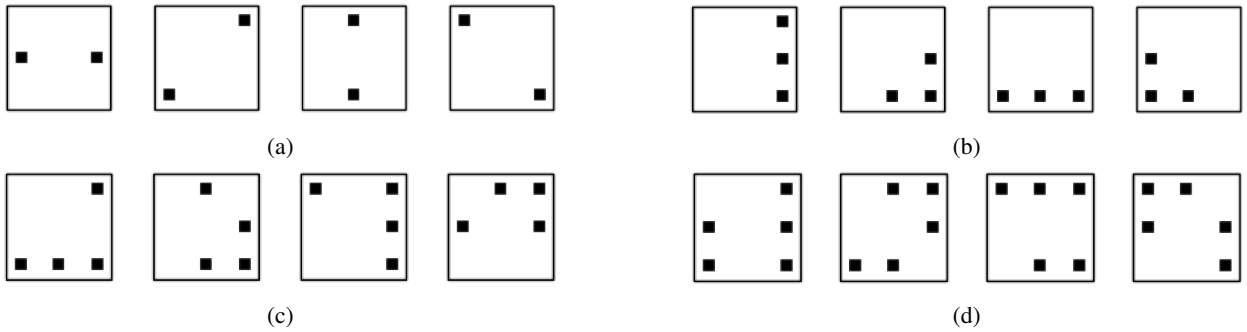


Figure 20: Test sequences of squares in the pattern

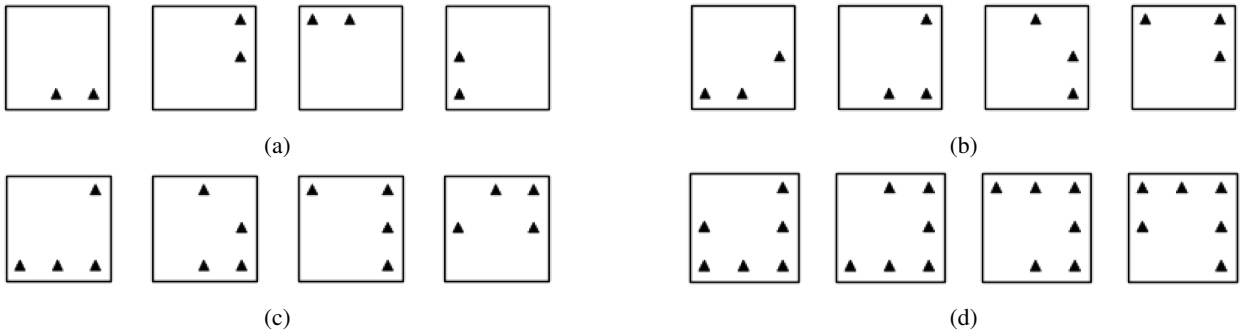


Figure 21: Test sequences of triangles in the pattern

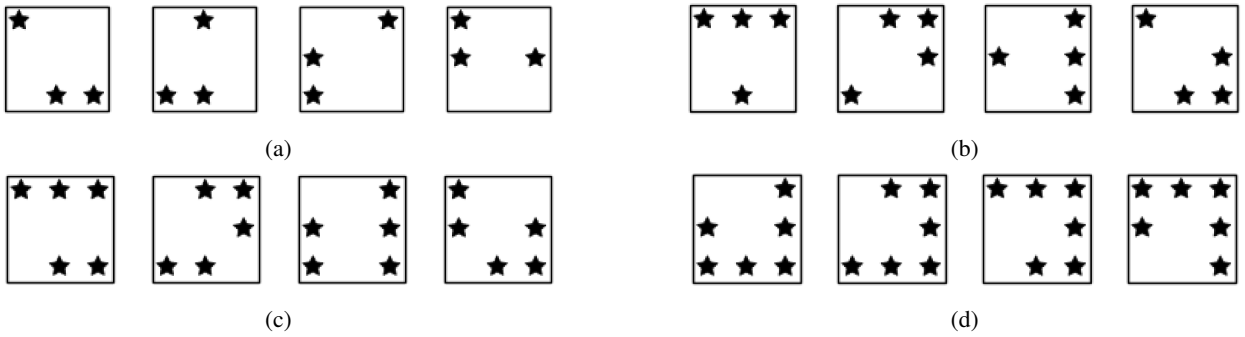


Figure 22: Test sequences of stars in the pattern

3 Conclusion

In this particular research paper, we dealt with two types of non-verbal reasoning - finding the odd one out and predicting the next image in the sequence. In the first problem, we were able to obtain a accuracy of almost 99.2% accuracy for finding odd one out among lines and almost 100% accuracy for finding odd one out among set of skewed figures. The accuracy decreased to almost 56% when the skewing angle was increased to 70 degrees. In the second problem of pattern generation, we were able to achieve a 100% accuracy on the baseline model. By adding noise in the form of

flipping pixels, the model gave 100% accuracy till the noise ratios of 40%, 25%, 25% – in the single state, multiple states with rotation and addition of new states type of problems. These problems extended to different shapes such as squares, triangles and stars also performed similarly giving a 100 % accuracy. In this paper, using Hopfield fields and supervised logic, we were able to satisfactorily solve two types of problems.

Acknowledgments

This research was conducted as a part of the Computational Cognitive Science course project. We thank Prof. Nisheeth Srivasatava for this opportunity and guidance through out the work.

References

[1] Amos Storkey. Increasing the capacity of hopfield network without sacrificing the functionality. Technical report, Imperial College, 2016.