Analyse Image Classification for Simple Shapes Dataset

Emmanuel Otovwerakpo
emmanuel.otovwerakpo@stud.fra-uas.de

Gurunag Sai Udaykumar
gurunag.udaykumar@stud.fra-uas.de

Abstract—There have been several techniques to deal with image classification throughout the years, with the introduction of various technological advancements, but in this project Hierarchical temporary memory (HTM) was used. Hierarchical temporary memory is a type of artificial intelligence that evolved from neocortex neuroscience and is now one of the most widely used applications for image recognition and anomaly detection. In this paper, we conducted a series of tests to determine the optimal parameter combination for achieving good and accurate similarity between simple images (circle, square, star, and triangle), as well as an extra function for predicting a simple input image (circle, square, star and triangle). We were able to demonstrate excellent effectiveness and accuracy through several experiments and found the right optimal parameter combination

Keywords—Hierarchical Temporal Memory (HTM), Image classification, Sparse Distributed Representation (SDR), Prediction code, Local Area Density, Potential Radius, Local/Global Inhibition, Spatial Pooler (SP).

I. Introduction

Image Classification and recognition have proved to be difficult tasks in artificial intelligence for several decades. Image classification refers to the categorizing the images into one of a number of predefined classes. The Image Classification model must be able to extract critical feature information that will be learned during training phase, and store the information for further test, henceforth for the classification the trained data is used to categorize the images. In this paper Hierarchal Temporal Memory (HTM) is used as learning algorithm along with the C# as programming language.

Hierarchical temporal memory (HTM) [1], is a machine learning algorithm that was inspired by the neocortex, centre for higher brain functions of human brain, and designed to learn sequences and make predictions on spatiotemporal data. In its idealized form, it should be able to produce generalized representations for similar inputs. HTM can serve as a tool for intelligent data processing in edge computing devices.

HTM is divided into two parts, HTM Spatial Pooler (HTM SP) and HTM Temporal Memory (HTM TM). The HTM SP forms a spatial pooling on the Sparse Distributed Representation (SDR) of the input data by performing a feature encoding which is useful for visual data processing and classification problems, whereas The HTM TM is responsible for the learning and processing of temporal patterns and can be used for the prediction taking into account previous experiences. The HTM network has a tree shaped hierarchical structure, as shown in Figure. 1. Each level of HTM is made up of distinct areas with columns, each of which is made up of cells. In HTM, the columns represent neurons. The connections between the columns and the input space are made through a dendritic segment with multiple synapses. Each synapse has a certain weight called synaptic permanence.

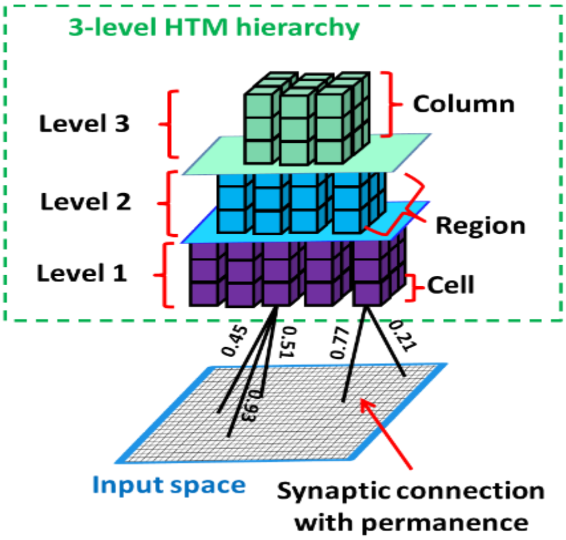


Figure 1: The three-level hierarchical structure of HTM

II. Methodology

The Image Classification project is previously implemented [2] in C# .NET Core in Microsoft visual studio 2022 Integrated Development Environment (IDE). The project uses NeoCortex API [3] for implementing the HTM. The goal of the project is to implement a program that uses the existing solution as a library to find the best parameter values for the spatial pooler which influence the training of the different images, results in a best correlation matrix. Henceforth the model should be able to predict any input image based on the training.

This section describes the reference to the methodology of the previous image classification project as the same used for the training of images. Further after the training and finding the best correlation matrix, the model will be able to predict the class of any input image as shown in the Figure 2.

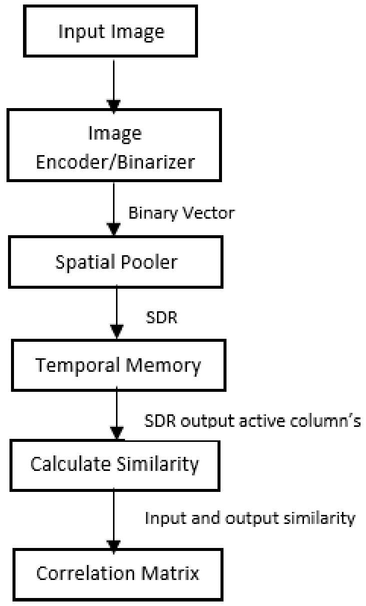


Figure 2: Methodology of the HTM image classification

A. Input Dataset:

There are a lot of factors which must be consider when choosing the dataset to yield good result, these factors are the position of the object in the images, the orientation (angle) of the object in the images and the size of the object in the images, these factors are chosen base on how deep the images are to be examine, for example if an object in an image with similar size [4] and orientation (angle) is taken, there is higher probability of getting a good result however in our work, different image with different size, position and orientation of the object was taken to examine the effect on the result and minimize such effect to get a good result. When choosing the dataset, two datasets was created with an image dimension of 64x64 and 100x100 as shown in figure 3 and figure 4, the purpose of using two different datasets is to determine how the image dimensions will influence the result i.e., if the optimal parameters combination for the small image dimension is compactable with the large image dimension.

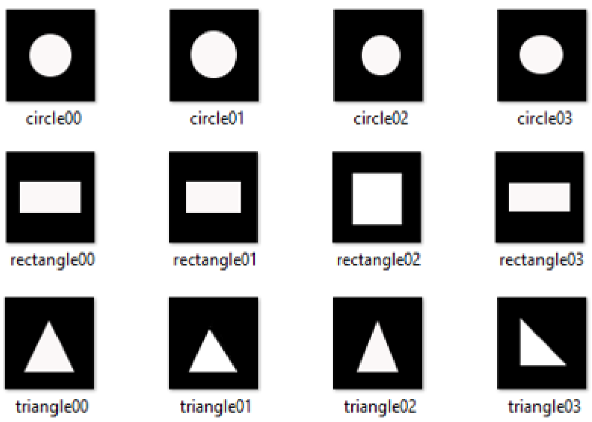


Figure 3: 64 x 64 pixels input dataset

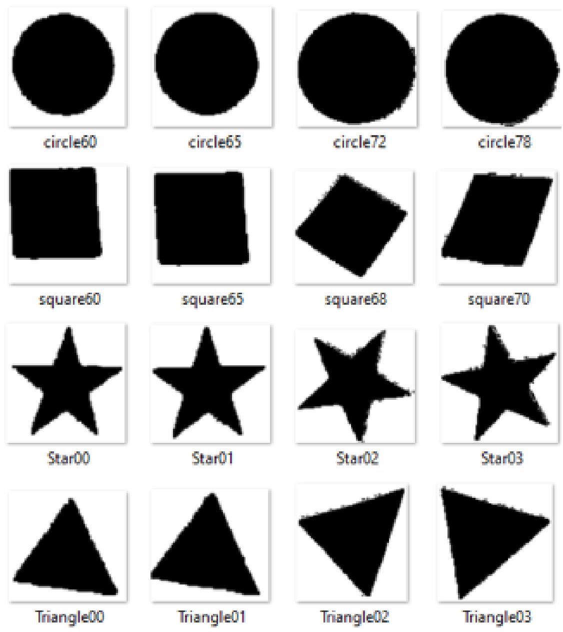


Figure 3: 100 x 100 pixels input dataset

B. Image Encoder:

Image Encoder reads the data of the images that needs to convert to binary form and coverts it to an integer array. The Image Encoder calculate the pixel values of the image and based on a threshold value it segments the image into binary. In general, for a grayscale image the white pixels are considered as one and dark pixels are considered as zero as shown in the Figure 5.

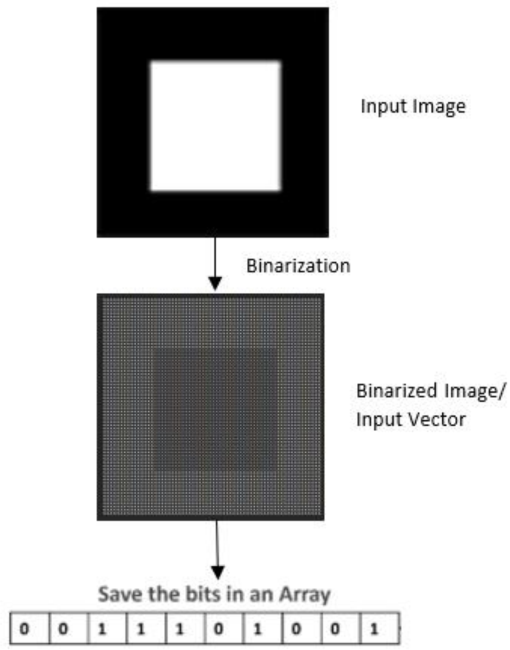


Figure 5. Image Binarization Process

C. Spatial Pooler and Sparse Distributed Representation:

To understand and explore the HTM software for running tests on image classification, the core of the program must be understood and that brings the topic spatial pooler. The Spatial Pooler Learning Method is a neocortical-inspired unsupervised machine learning algorithm for learning spatial patterns. It takes the encoder SDR input (binarized data) and generates a collection of active columns. Spatial pooler groups similar spatial patterns represented as activated neurons into highly sparse representations of cortical micro columns. Spatial distributed representation (SDR) is achieved when the number of active mini-columns in each HTM zone is limited to two to six percent of the total mini-columns [4].

Table 1: Default values of Spatial Pooler parameters

Parameters	**Default value**
InputDimensions	(100,100)
ColumnDimensions	(64,64)
InhibitionRadius	1
PotentialRadius	30
PotentialPct	0.5
GlobalInhibition	False
LocalAreaDesity	0.1
NumActiveColoumnPerInhArea	-1
StimulusThreshold	0.0
SynPermInactiveDec	0.01
SynPermActiveInc	0.1
SynPermConnected	0.1
DutyCyclePeriod	10
MaxBoost	10

D. *Calculation of Similarities*

After the training of each image in learning phase, the next task is to calculate the similarity between the SDR's of each trained image. The similarity is calculated between the active columns of the image's SDR. This similarity is defined in two ways, one the similarity between the images of same shape called Micro similarity and the other is the similarity between the images of different shapes called Macro similarity. For example, for all images of the shape circle is compared to itself gives the micro similarity and circle to rectangle give the macro similarity. Together these micro and macro similarity values form a similarity matrix, which is used to analyse the learning accuracy and further image class prediction.

III. EXPERIMENT

In this section we will discuss the series of experiments, as the goal is to investigate how the similarity of the images varies with the changing of the parameters, determine the best parameters that provide satisfactory results, and forecast whether the image is a circle, square, rectangle, star, or triangle.

In this project, tests are performed on a simple shape dataset: square, rectangle, circle, triangle, and star, with four images of varying size, position, and orientation in each category. The similarity of the images as the parameter changes is represented by a graph.

We investigate on two datasets with an input dimension of 64x64 and 100x100, the purpose of examining two datasets with different input dimensions is to gain a thorough understanding of whether the size of the dataset has a significant impact on the optimal parameters, i.e., if one optimal parameter is the same for both datasets. For this reason, we will discuss both datasets separately (100x100 intensively and 64x64 outline the important factor).

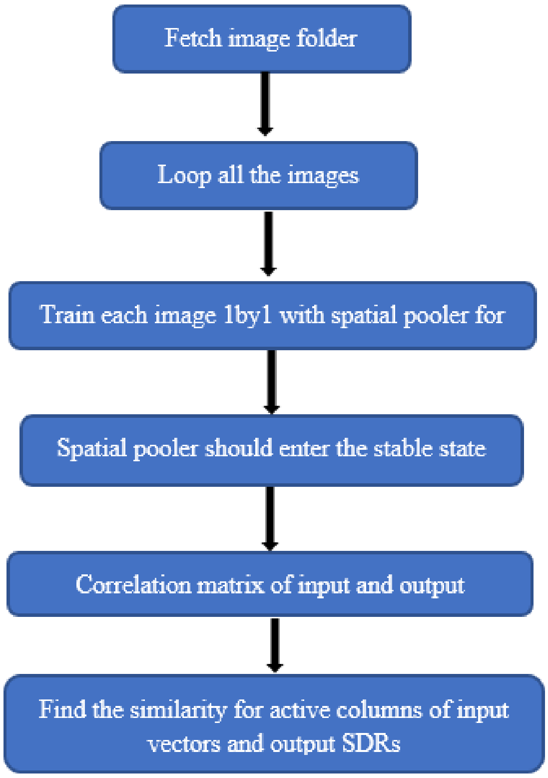


Figure 6. Learning Phase Flow Chart

Learning Phase:

An image (circle, square, triangle, and star) dimensions of 100x100 was taken, then these images where binarized (0 for the object in the image and one for the background), these binarized images are now the input. In the initial phase of the experiment, the same instance of spatial pooler was used with the default parameters [fig default parameters table] from the *htmconfig.json* file.

The argument *iterationsteps* specifies how many iteration steps are used to train each image. The images are trained until the spatial pooler achieves a stable state, which is managed by the *HomeostaticPlasticityController* class. When the learning process begins, the purpose of the *HomeostaticPlasticityController* is to place the spatial pooler in a new-born state. At this moment, the boosting is highly active, but the spatial pooler is unstable. The HPC will emit an event that alerts the code that the spatial pooler is stable after the SDR generated for each input becomes stable. *trainingImagePathLength* x newBornStageIteration automatically calculates the minimum number of cycles.

Local area density: density of active columns inside of local inhibition. It should be more than zero and less one if it is less than 0 the *numActiveColumnsPerInhArea* will be used.

Potential radius: this defines the radius in number of input cells visible to columns cells. It is important to choose this value, so every input neuron is connected to at least a single column.

GlobalInhibition: If true, then during inhibition phase the winning columns are selected as the most active columns from the region. Otherwise, the winning columns are selected with respect to their local neighbourhoods.

numActiveColumnsPerInhArea; An alternate way to control the density of the active columns. If numActiveColumnsPerInhArea is specified then localAreaDensity must be less than 0, and vice versa.

Micro: this is the similarity between two images in the same category e.g., circle to circle or rectangle to rectangle

Macro: this is the similarity between two images from different category e.g., circle to rectangle or rectangle to triangle

Prediction Phase:

The prediction process follows the same as the learning phase where the learn flag is set to false state. After the learning phase is completed and similarity matrix with Micro and Macro correlation for all the classes are generated, the image which needs to be predicted is converted to binary, computed with the spatial pooler results in an SDR, which is compared with each image of the trained image SDR's by using the *CalculateSimilarity* Method. Further the best match is searched in the correlation matrix and displays the prediction image similarity between all the trained shapes with maximum, average and minimum percentage of similarity as shown in the flowchart Figure 7.

In the first step, we calculate the percentage similarity between the SDR of the image to be predicted with SDRs of training images.

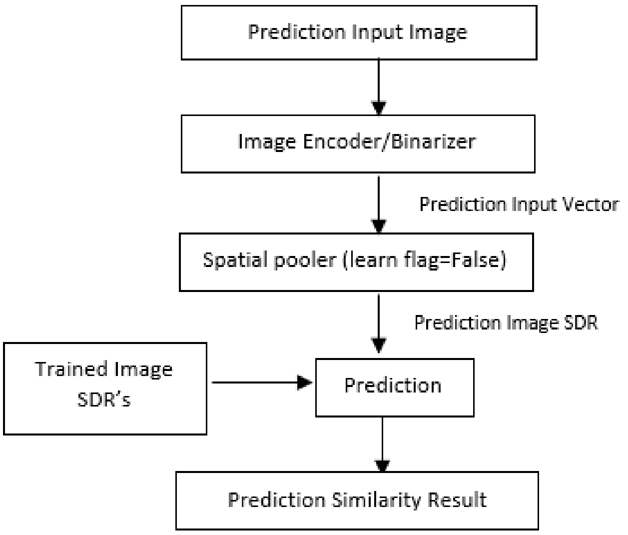


Figure 7. Prediction Phase Flow Chart

IV. RESULT AND DISCUSSION

In this section, the results of the above experiments conducted will be discussed for both the datasets worked on

Dataset input dimensions 100 x 100

In this experiment, an image dimensions of 100x100 and 64x64 column is used. the whole learning process took a minimum of ninety-four cycles and maximum of 214 cycles depending on the configured parameters in the *htmconfig.json* file. The goal of the experiment is to get the optimal local area density and potential radius that give optimal similarity between the images.

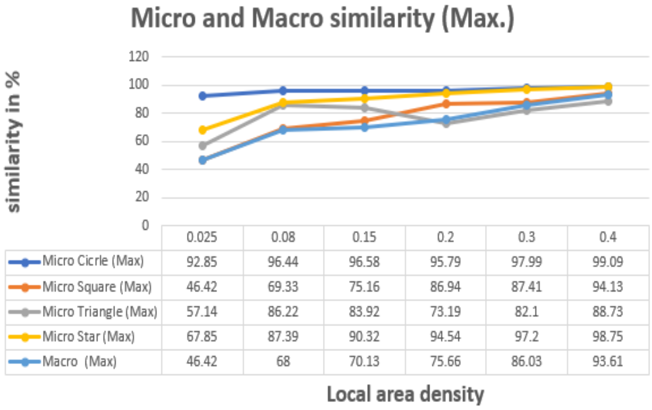


Figure 8. Micro vs Macro similarity for Potential Radius=30

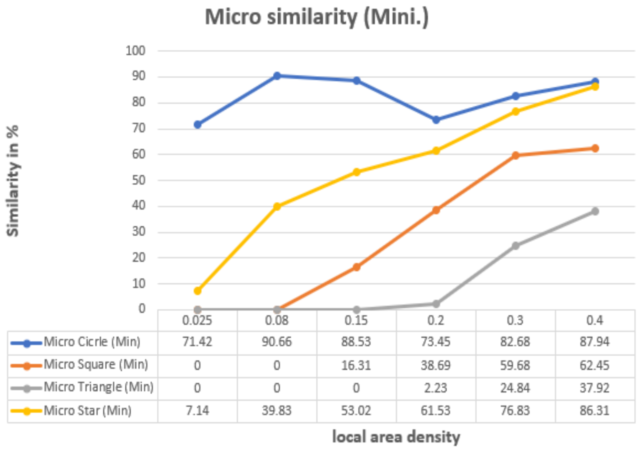


Figure 9. Micro similarity for all shapes at Potential Radius=30

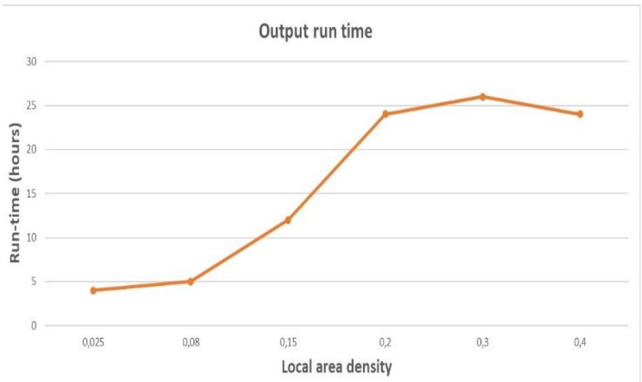


Figure 10. Output run times of the experiment for Potential Radius=30

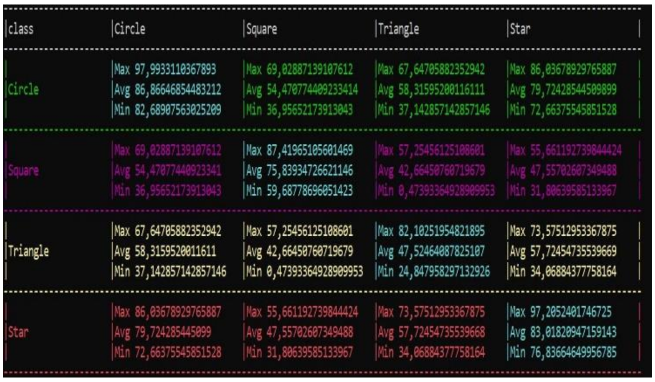
Figure 8 show the graphical result we get on the similarity between micro and macro (max.) images, while Figure 9 show the graphical result we get on the micro similarity (mini), when we run different experiment on various local area density and potential radius values. Figure 10 shows the run time of the experiments with increasing local area density for the potential radius at 30.

From figure 8 and figure 9 it can be understood that similarity in micro images and macro images increases with respect to the increase in local area density, but it is important to find an optimal local area density value that matches both the macro and micro similarity. After series of test, it is understood that potential radius 30 is the standardized value for this dataset

Table 2: Optimal Parameters for 100x100 input dataset

LocalAreaDensity	0.3
PotentialRadius	30
NumActiveColomnsPerInhArea	-1
GlobalInhibition	FALSE

Table 1 shows the optimal value for local area density and potential radius that give a balance result between the micro and macro similarity.



class	Circle	Square	Triangle	Star
Circle	Max 97,9933110367893 Avg 86,86646854483212 Min 82,68907563025209	Max 69,02887139107612 Avg 54,470774409233414 Min 36,95652173913043	Max 67,64705882352942 Avg 58,31595200116111 Min 37,142857142857146	Max 86,03678929765887 Avg 79,72428544509899 Min 72,66375545851528
Square	Max 69,02887139107612 Avg 54,47077440923341 Min 36,95652173913043	Max 87,41965105601469 Avg 75,83934726621146 Min 59,68778696051423	Max 57,25456125108601 Avg 42,66450760719679 Min 0,47393364928909953	Max 55,661192739844424 Avg 47,55702607349488 Min 31,80639585133967
Triangle	Max 67,64705882352942 Avg 58,3159520011611 Min 37,142857142857146	Max 57,25456125108601 Avg 42,66450760719679 Min 0,47393364928909953	Max 82,10251954821895 Avg 47,52464087825107 Min 24,847958297132926	Max 73,57512953367875 Avg 57,7245473553966 Min 34,0688437775816
Star	Max 86,03678929765887 Avg 79,724285445099 Min 72,66375545851528	Max 55,661192739844424 Avg 47,55702607349488 Min 31,80639585133967	Max 73,57512953367875 Avg 57,72454735539668 Min 34,06884377758164	Max 97,2052401746725 Avg 83,01820947159143 Min 76,83664649956785

Figure 11. Similarity Matrix for 100x100 input

Figure 11 shows the output gotten when the parameters in table 2 are used, this output yields an excellent result both for micro and macro similarity. After conducting several tests, the general range of similarity that determine if it is a good result. Micro similarity should be at least 80% and 30-90% in the minimal range, while macro similarity should be between 0% and 88%. (max., avg., and mini.).

Figure 10 shows that the micro similarity is between 82% and 97% for maximum and between 24% and 82% for minimum, while the macro similarity is between 55% and 86% for maximum (macro average and minimum do not matter because the lower the macro, the better the result), this indicate that the required output is met.

Dataset input dimensions 64 x 64

In this experiment, an image dimensions of 64x64 as mentioned in Figure 3 and 32x32 column is used. The training process is repeated as demonstrated in dataset 100x100. The learning process took a maximum cycle of 180.

Micro: this is the similarity between two images in the same category e.g., circle to circle or rectangle to rectangle

Macro: this is the similarity between two images from different category e.g., circle to rectangle or rectangle to triangle.

Case 1: Micro Similarity between images

From the input dataset mentioned in the Figure 3, Considering the two rectangle shape images as shown in figure 12 and comparing the similarity between them for various values of parameters.

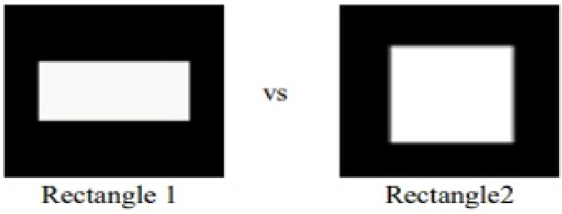


Figure 12. Input images for micro similarity calculation

Figure 13 and Table 3 shows the result we get on the similarity between rectangle and rectangle, when we run different experiments on various local area density and potential radius values.

Table 3: Micro Similarity for Rectangle1 and Rectangle2

Local Area Density	Output Similarity between rectangle1 and rectangle2					
	Potential Radius = 1	Potential Radius = 5	Potential Radius = 10	Potential Radius = 15	Potential Radius = 20	Potential Radius = 30
0.1	48.6	18.12	29.41	73.24	71.08	24.5
0.2	70.35	26.19	38.82	82.71	79.6	25.94
0.3	79.76	39.45	46.05	75.89	80.44	39.8
0.4	85.83	46.77	58.57	79.29	83.49	43.99
0.5	93.54	78.8	76.9	88.36	85.13	63.55
0.6	91.68	89.14	79.18	83.51	84.45	70.62
0.7	95.49	95.24	84.73	87.14	90.14	84.66
0.8	98.07	98.04	90.88	88.28	90.96	87.59
0.9	99.6	98.69	97.8	96.49	93.91	99.89
1	100	100	100	100	100	100

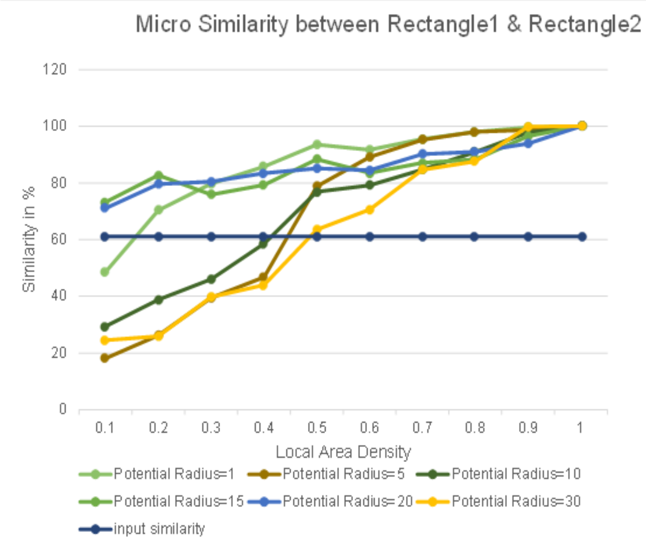


Figure 13. Graph Depicting the change in micro similarity for change in PotentialRadius and LocalAreaDensity

From figure 12, it can be understood that similarity between rectangle and rectangle increases with respect to the increase in local area density, but it is important to find an optimal value that matches with the macro similarity.

Case 2: Macro Similarity between images

From the input dataset mentioned in the Figure 3, macro similarity is achieved by comparing the similarity values between two different class images. In the following case, considering the similarity between a rectangle image and triangle class image as shown in figure 14 has resulted the following.

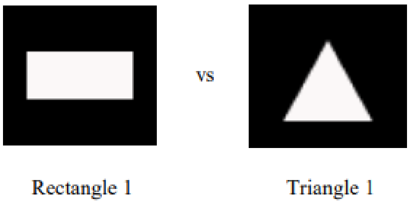


Figure 14. Input images for macro similarity calculation

Table 4: Macro Similarity for Rectangle1 and Triangle1

Local Area Density	Output Similarity between rectangle1 and Triangle1					
	Potential Radius = 1	Potential Radius = 5	Potential Radius = 10	Potential Radius = 15	Potential Radius = 20	Potential Radius = 30
0.1	59.21	20.3	22.1	43.29	46.27	21.7
0.2	64.96	29.04	24.55	47.42	56.75	20.28
0.3	72.85	39.75	27.32	52.23	62.13	28.3
0.4	84.93	45.98	37.75	65.04	66.9	44.95
0.5	92.7	79.14	77.77	77.6	68.16	48.46
0.6	92.45	89.48	80.29	73.68	70.82	61.12
0.7	95.56	95.46	85.72	80.36	75.29	80.48
0.8	98.08	97.32	91.82	87.38	82.33	81.9
0.9	99.01	99.09	97.08	94.58	90.72	99.78
1	100	100	100	100	100	100

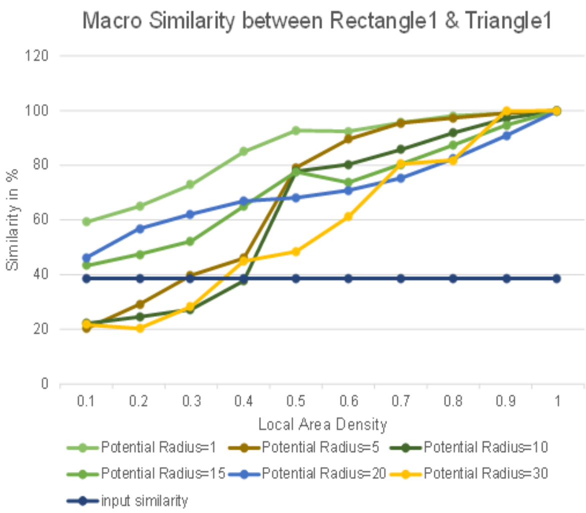


Figure 15. Graph Depicting the change in macro similarity for change in PotentialRadius and LocalAreaDensity

Figure 15 show the output result graphically on the similarity between rectangle and triangle, when we run different experiment on various local area density and potential radius values.

From figure 15 it can be understood that similarity between rectangle and triangle increase with respect to the increase in local area density, which will eventually give bad result.

Similarity Matrix:

The output similarity of the micro and macro photos is shown in Figure 16 which is the best fit similarity matrix obtained from the values depicted in table 5

Table 5: Optimal Parameters for 64x64 input dataset

LocalAreaDensity	0.3
PotentialRadius	10
NumActiveColomnsPerInhArea	-1
GlobalInhibition	False

For all categories, the micro similarity is 87 percent maximum, 78 percent -82 percent average, and 67 percent -74 percent. These are satisfactory results for micro similarity since the higher the similarity, the better the outcome; for micro similarity, the maximum similarity should be 80% or higher, the average similarity should be between 70% and 85%, and the minimum similarity should be 40-80%.

All categories have a macro similarity of 45 percent -66 percent maximum, 40 percent -58 percent average, and 36 percent -51 percent. Because the lower the similarity, the better the outcome, these numbers provide a favourable result for macro similarity. For these reasons, the maximum similarity should be considered alone, and the maximum similarity should be between 20% and 70%.

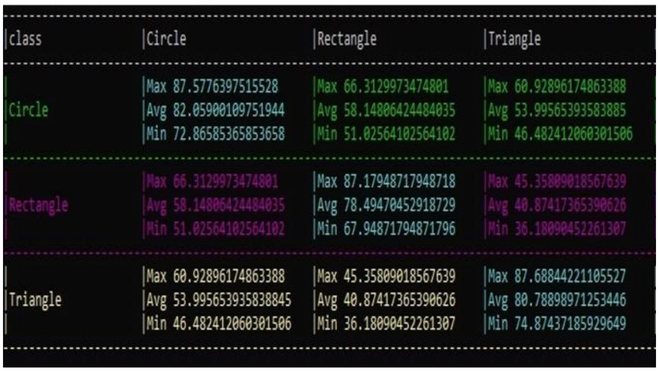


Figure 16. Similarity matrix for 64x 64 image dataset for the parameters shown in table 5

Prediction Testing

After the training phase, the model will be able to predict the distinct set of images. As discussed in the Experiment, Prediction phase section, by creating the SDR of the image to be predicted, the active columns of the same will be compared to the active columns of training images SDR, which provides a set of similarity values in percentage between the different classes.

The Figure 17. Consisting of the images of Circle, rectangle, and star classes, are subjected for testing the prediction of the image classification model which used the input simple shapes dataset which consists of classes Circle, Rectangle and Triangle with images 64x64 input image pixels as shown in the Figure 3. From the figure 16, theoretically, the circle and rectangle belong to the dataset of trained image model and should be able to achieve good prediction result for it, where as the star class should optimistically have a less percentage similarity for the trained dataset.

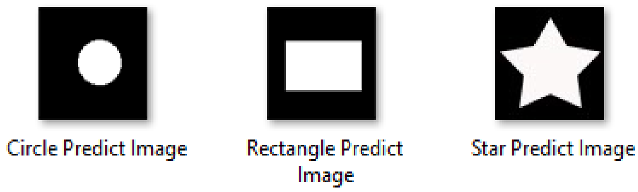


Figure 17: Images used for prediction

For the prediction testing, the following cases are considered for an optimal test of the learning model.

Case 1: Predict image belongs to same class used in training

For the following testcase, Consider the images 'Circle Predict Input' and 'Rectangle predict Input' form the Figure 16 for the input dataset consisting of circle, rectangle, and triangle as the dataset.

The Figure 18 shows the prediction percentages for the 'Circle Predict Image'. The prediction percentage for the circle class has the maximum of 82.72% and a minimum of 67.44%. For the Rectangle class, the 'Circle Predict Input' has the maximum of 55.02% and a minimum of 53.16% and for the triangle class it has a maximum similarity percentage of 42.94% and minimum of 37.54%. By looking into the overall average percentages of all the three class, the circle class has the highest average percentage with 73.26% which gives that the image 'Circle Predict Image' belongs to the Circle class.

```
The predicted results for the input image "CirclePredictImage" is
Circle:        Max:82.72%     Avg:73.26%     Min:67.44%
Rectangle:     Max:55.02%     Avg:53.95%     Min:53.16%
Triangle:      Max:42.94%     Avg:40.08%     Min:37.54%
```

Figure 18: Predicted results for image 'Circle Predict Image'

Consider the second image for the prediction, 'Rectangle Predict Image'. The following figure 19 shows the prediction similarity results for all the three trained classes. The prediction percentage for the circle class has the maximum of 59.16% and a minimum of 51.29%. For the Rectangle class, the 'Rectangle Predict Input' has the maximum of 84.83% and a minimum of 62.54% and for the triangle class it has a maximum similarity percentage of 25.83% and minimum of 23.96%. From the average similarity percentages, the rectangle class has the highest of 71.07% which depicts that the image for prediction used does belong to the rectangle dataset.

```
The predicted results for the input image "Rectangle Predict Image" is
Circle:        Max:59.16%     Avg:54.6%      Min:51.29%
Rectangle:     Max:84.83%     Avg:71.07%     Min:62.54%
Triangle:      Max:25.83%     Avg:25.09%     Min:23.96%
```

Figure 19: Predicted results for image 'Rectangle Predict Image'

Case 2: Predict image belongs to different class

For the following testcase, Consider the image 'Star Predict Input' form the Figure 17, for the input dataset consisting of circle, rectangle, and triangle as the dataset, theoretically the training model should provide the low similarity percentages when compared to the trained images.

The Figure 20 shows the prediction percentages for the 'Star Predict Image'. The prediction percentage for the circle class has the maximum of 53.51% and a minimum of 33.51%. For the Rectangle class, the 'Star Predict Input' has the maximum of 38.95% and a minimum of 33.51% and for the triangle class it has a maximum similarity percentage of 51.35% and minimum of 46.49%. From the average similarity values, the triangle has 43.78%, rectangle of 36.62% and triangle of 49.05%, comparatively which is a low set of similarity values for the trained image classes. Therefore, it can be proved that the features of the 'star predict image' has less similarity between the trained input classes Circle, Rectangle and Triangle

```
The predicted results for the input image "Star Predict Image" is
Circle:        Max:53.51%     Avg:43.78%     Min:33.51%
Rectangle:     Max:38.95%     Avg:36.62%     Min:33.51%
Triangle:      Max:51.35%     Avg:49.05%     Min:46.49%
```

Figure 20: Predicted results for image 'Star Predict Image'

V. CONCLUSION

The goal of this project is to experiment with several image categories to find the optimal value for classifying these images and predicting which category an untrained input image belongs to base on a trained image. After conducting several experiments with different image dimensions (100x100 and 64x64), we can conclude that micro similarity of maximum 80% and higher, as well as minimum 40-80 percent, and macro similarity between 0%-70% (max. avg. and mini.) indicate an optimistic result. Another finding was that the local area density for small and large image dimensions is the same (LocalAreaDensity= 0.3), but the potential radius is different (potential Radius for large image dimensions= 30 and potential Radius for small image dimensions =10).

For future work, it would beneficiary to add an additional function that indicates which potential radius is suitable for an image depending on their dimensions, since the potential radius differ based on the image dimension. We can see that for local area density value of 1, the image matches 100% irrespective of whatever the image is taken.

VI. REFERENCES

[1] "Numenta," [Online]. Available: https://numenta.com/resources/biological-and-machine-intelligence/spatial-pooling-algorithm/.

[2] D. S. D. Sundari, "Hierarchical Temporal Memory Image Classification".

[3] D. Dobric, "Github," [Online]. Available: https://github.com/ddobric/neocortexapi-classification.

[4] O. Rukundo, "Effects of image size on deep learning," [Online]. Available: https://arxiv.org/ftp/arxiv/papers/2101/2101.11508.pdf.

[5] Z. C. Y. Q. Z. Y. Y. X. Wen Zhuo, "Image classification using HTM cortical learning algorithms," [Online]. Available: https://ieeexplore.ieee.org/document/6460663.

[6] H. J. a. G. D., "Hierarchical Temporal Memory: Concepts, Theory, and Terminology.," [Online]. Available: http://www.numenta.com/Numenta_HTM_Concepts.pdf.