Surveying Hopfeild Neural Network and its Applications

Farah Aymen Mounir
Faculty of Informatics and Computer Science-AI major
The British University in Egypt
Cairo, Egypt
farah194233@bue.edu.eg

***Abstract*—The purpose of this research is to explore the emergence of the Hopfield networks, modern Hopfield networks and their architecture. With the limitations that Hopfield networks presented new architecture has emerged which is the modern Hopfield networks. Hopfield networks are applied in many applications like Image recognition, classification, and restoration. First, the paper discusses the architecture, energy function and activation of the HNNs, then it investigates the three applications in details and the models applied on them using the HNNs. The final conclusion from this survey is that Modern Hopfield networks may be considered as the next architecture that should be explored and applied in the future since it proved its efficiency, accuracy and capacity.**

Keywords—Hopfield Network, Energy function, Activation function, pattern, weights

I. INTRODUCTION

Hopfield neural networks (HNNs) were first invented by Dr. John J. Hopfield in 1982. They were the first instance of recurrent neural network as they were invented for the purpose of having an associative memory that stores patterns and those patterns can be fetched later. The learning of the network and memorization is inspired from the brain and its neural behavior. Furthermore, the HNNs do not learn using the back propagation method unlike other networks, but they learn using the Hebbian learning method. This method is based on pattern classification. Like the brain, when a human tries to remember a certain memory, the brain fetches patterns of that memory. The classical (binary) HNNs have some limitations until in 2020, modern HNNs emerged solving these problems. The motivation behind the HNNs in general is to act as associative memory that can be used in pattern recognition, optimization, and other applications. The HNN is a set of neurons that are connected bidirectionally to each other, but each neuron is not connected to itself, where weights are symmetric between all neurons where $w_{ij} = w_{ji}$

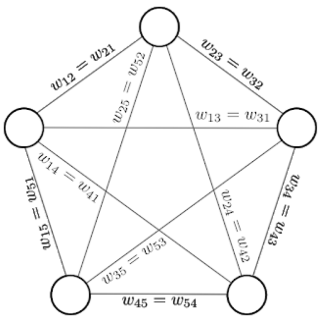

Fig 1 architecture of Hopfield neural network

Unlike other networks, HNNs learn using Hebbian learning and not back propagation. The Hebbian learning concept was first introduced by Hebb in 1949. It stated that neurons strengthen their connection. Also, the network's architecture is based on Ising model where a node can have a state either positive or negative. J. Hopfield presented the model where V_i is the neuron, V is a vector of the neurons where V = $[V_1, V_2, \dots . V_n]$, T_{ij} is the strength of the connection that is between two neurons V_i and V_j. T is a matrix that holds all the weights that are between all neurons among themselves.

$$\mathbf{T} = \begin{bmatrix} 0 & T_{12} & T_{13} & \cdots & T_{1n} \\ T_{21} & 0 & T_{23} & \cdots & T_{2n} \\ T_{31} & T_{32} & 0 & \cdots & T_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ T_{n1} & T_{n2} & T_{n3} & \cdots & 0 \end{bmatrix}$$

Typically, $T_{ii} = 0$ because there is no weight not a connection between the neuron and itself which makes it a symmetric matrix. Generally, the weight T_{ij} between the neurons is the result of the product of the two neurons V_i, V_j sharing the weight [1].

$$T_{ij} = V_i * V_j \quad (1)$$

As seen in equation (1), this means that the neurons with the same sign will attract each other while if they have different signs, they repel each other which agrees with the Hebbian postulate which states that "neurons that fire together wire together and neurons out of sync fail to link". Therefore, the network gets fed with patterns to lean and store. Once that done, it takes a random noise and try to reconstruct it according to the patterns it has learned. This happens by first initializing the neurons with values then going through the weights that connect those neurons. If the sum of the weights z connected to a neuron V_j give the same sign of that neuron, then the value of the neuron remains as it is. If not, its sign is flipped so the network changes the pattern to match what it has learned before.

$$\begin{matrix} V_i \to 1 & & & > U_i \\ V_i \to 0 & if & \sum_{i \neq j} T_{ij} V_j & < U_i \end{matrix} \quad (2)$$

Equation (2) is considered the activation function of the network since it determines the activation of the neuron that determines its state. The neuron randomly checks its state an according to the threshold U_i which is initialized with 0 otherwise a value that is stated and decides if the neuron should fire or not [1]. The idea behind the HNN is to be asynchronous and random because that is how the human brain works and updating the neurons at the same time might cause the model to oscillate. HNNs do not use cost functions, instead they use an energy function because there is no labeled data as HNNs are a form of unsupervised learning. The energy function is simply the sum of the neurons multiplied by their weighted sum and multiplied by half due to symmetry of the weights.

$$E = -\frac{1}{2}\sum_{i\neq j}\sum T_{ij}V_iV_j \qquad (3)$$

When the neuron flips it will minimize the energy function which brings it closer to the pattern that is desired to be reached.

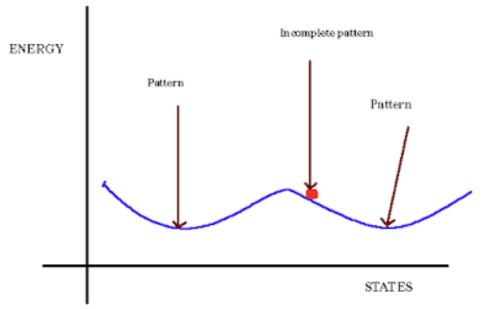

Fig 2 Graph represents the energy and the state of the neuron

The neurons keep getting flipped if needed and evolve until they reach a stable state, and the neurons stop flipping. This means that the network has reached closely to the desired pattern. The memory and the patterns are stored within the strength of the neuron connections. The limitations that the classic HNN faces is that it can only store 0.15N patterns where N is the number of neurons. For example, if the network is composed of 100 neurons, it can only store at most 15 patterns. Another problem is, if the network tries to learn additional patterns, they will become spurious minima patterns.

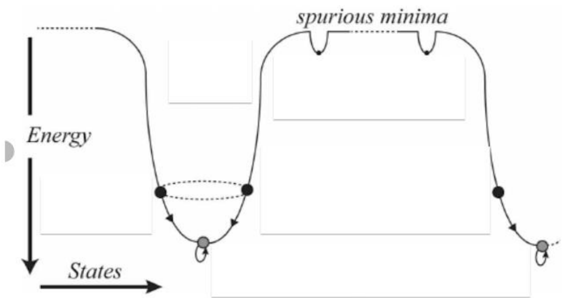

Fig 3 spurious minima

This means that those patterns will be memorized unintentionally and create local minima that may cause the model to be trapped in them. Due to the limitations of the storage of the neural networks, it was not applicable anymore. In 2020, Hubert Ramsauer et al introduced modern HNNs. Before the modern HNNs, the alternatives for networks that act as a memory were Recurrent neural networks (RNNs), linear memory networks, etc. The idea behind the modern HNN is that it stores more patterns exponentially [2]. It introduces a new energy function and a new update rule that minimizes the energy function.

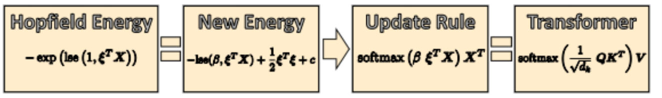

Fig 4 Modern Hopfield's new energy function and update rule

The modern Hopfield networks are continuous rather than the classical binary Hopfield. This makes it differentiable, and it stores vectors of continuous values rather than binary values. In section 2, applications using the HNNs will be explored in different research papers like image restoration, image recognition and image classification. In section 3, the research papers will be compared and analyzed on each application and discuss the results of the analysis.

II. Literature Review

A. Image Restoration

Two papers were found that apply the HNNs for image restoration. In the first paper, Zhimin Zhang discusses using the HNN by modifying in its activation function so it can enhance the restoration of images taken by UAV (unmanned aerial vehicle) [3]. UAVs take images from high altitude and the possibility of getting a noisy blurred image is high. This paper proposes to use continuous Hopfield networks (CHNNs) with an improved activation function to accelerate convergence and improve noise immunity. The new activation function is similar to the sigmoid that it converges faster and more immune to noise.

$$\sigma(x) = C \times arc\tan(\mu x) \qquad (5)$$

Where C and μ set the trend of the function. The paper also compared between several activation functions and Paik model and the energy to see which converges faster at which number of iterations.

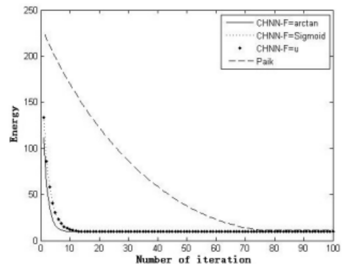

Fig 5 convergence of HNNs using different activation functions

It is seen that the CHNN with activation function arctan reached the lowest energy first at almost 10 iterations. The second paper proposes a Memristive continuous Hopfield neural network (MHNN) circuit that can be more efficient than a digital neural network in image restoration. It compares both approaches and proves that both average processing time and error is reduced using the circuit [4]. A crossbar array is used to represent the weights. The network consists of 25 neurons total. The circuit's purpose remains the same which is minimizing the energy function. Finally, the paper compares between the performance of the digital circuit and the simulation.

Table 1: Processing time between software and hardware

Size of image	Operation time	
	MATLAB simulation in software (s)	Analog computation in hardware (ms)
3 × 3	0.35	0.38
4 × 4	0.56	0.39
5 × 5	0.94	0.41

It is seen that the circuit was faster than the digital network as the size of the image increases. This means that the analog circuit can implement massively parallel calculations that can only be performed by cyclic iteration on the software. Finally, using the error metric, it is found that the average error decreased from 1.11 x 10^-3 to 1.27 x 10^-5 after using the circuit.

B. *Image Classification*

There are two papers that were found to have applied image the HNNs into classifying images. The first paper uses the HNN in classifying oranges according to their size and quality [5]. It proposes that with the property of the HNN of storing patterns, the network will classify according to the closest pattern. The average of the correct answers given by this model was 85% for both criteria of classification. The activation function used for the model is as follows:

$$y = g\left(\sum_{i=1}^{n} w_i\, x_i - \theta\right) \tag{4}$$

Where y is the output of the classification, g is the activation function that calculates the sum of all differences between the weight w_i of input x_i subtracted to the threshold θ. The hyper parameter considered here is the threshold. The dataset that the paper used was manually gathered as they took pictures of the oranges and processed it in uniform sizes and format. The dataset was of total 59 images of different sizes and quality.

Table 2 ANN confusion matrix for classification according to quality.

Condition	→	Predicted		
↓	Class	Good	Spoiled	Total
Real	Good	37	12	49
	Spoiled	3	46	49
	Total	40	58	98

Table 3 ANN confusion matrix for classification according to size.

Condition	→	Predicted		
↓	Class	Large	Small	Total
Real	Large	35	14	49
	Small	1	48	49
	Total	36	62	98

The tables show that in the quality criteria the model gave results of the correctly detecting the spoiled oranges in high while the healthy ones are acceptable, and for the size criteria the small oranges attain almost perfect classification while the large ones are somehow acceptable. Regarding the evaluation, Roc curves were used to evaluate the quality criteria which resulted in average accuracy of 64.4% and variance of 1.5%. While for the size criteria, it reached accuracy of 68.4% and variance 0.5%. The second paper trains the HNN for classification. It proposes to make the network asymmetric to imitate the brain biologically more. The weight change is proportional to the firing rate of the neuron and the state change of the neuron which approximates a classical spike-timing-dependent-plasticity (STDP) rule. The model proposed is made hierarchal structure using asymmetric weights with two hidden layers and an output layer and only the hidden neurons can change their state.

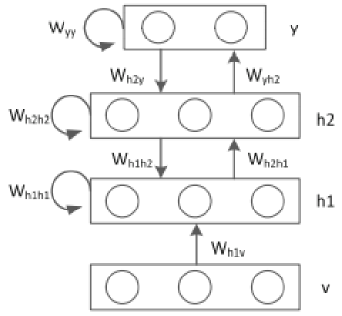

Fig 6 Hierarchical HNN

Then, to assure of the stability of the HNN, the weights are first initialized with chosen weights to make it stable. The activation function that was used was linear rectifier activation function and small learning rates. Finally, the model was applied on MNIST dataset using 60,000 training images and 10,000 test images. The weights were first randomly initialized to normal distribution and to standard deviation of 0.01, and the weight decay coefficient was set to 0.005. Finally, the paper compared between the results between the baseline model, recurrent model, and model with symmetric weights.

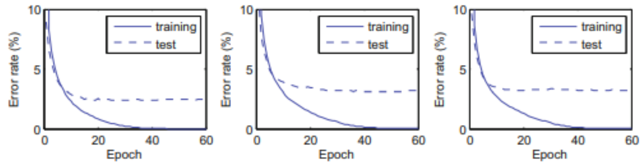

Fig 7 Left: Baseline model. Middle: Network with recurrent connections. Right: Network with symmetric weights.

The results showed that baseline setting reached 0 error rate after 45 epochs while the recurrent model between the hidden layers reached error rate of 0.02% after 60 epochs, and the symmetric model reached error rate of 0.01% after 60 epochs. This shows that the model with the hierarchical structure with no recurrent connections performed best [6].

C. *Image Recognition*

HNNs were applied for image recognition and their success were discussed in three papers. The first paper compares between the performance of the traditional asynchronous HNN and cloud HNN. It discusses how cloud HNN performs better with higher success rate [7]. The research's purpose is to use low resolution images and based on the patterns learned in the network, it can recognize and retrieve the original image. It uses a dataset of 7 images each 60 × 60 and translate it into a 10 × 10 with some distortions and occlusions applied on them. The traditional HNN gives accuracy of 75.4% with distortion of 35% while the cloud HNN gives accuracy of 75.4% with same distortion level. The traditional Hopfield has the same basic architecture where it is non-layered and uses the sign function as activation function. While the cloud HNN modifies the basic architecture where it updates a "cloud" of neurons simultaneously as stated by Singh and Kapoor when they introduced the model [8]. The results of the successfully retrieved images for the cloud Hopfield network is 3172 images out of 3500 distorted images with percentage 45% distortion, while the traditional HNN retrieved only 2716 successful images.

Table 4 Number of successful retrievals for AHNN and CHNN for 60 X 60 images with distortion level 30-45%

Number of successful retrievals								
Faces	Distortion percentage							
	30%		35%		40%		45%	
	H	C	H	C	H	C	H	C
1	500	500	499	500	472	492	361	414
2	500	500	500	500	499	499	444	479
3	500	500	500	500	497	499	431	483
4	500	500	500	500	500	500	426	469
5	500	500	500	500	480	495	345	452
6	500	500	488	498	455	487	315	418
7	500	500	497	499	480	497	391	457

The paper then stated that with trial of different distortion levels to conclude that the cloud HNN achieves successful retrieval with 99% with distortion up until 20%.

The second paper developed by the same team elaborates on the same application but experiments on the two models: one works with 60 × 60 (3600 neurons) and a model works with 10 × 10 (100 neurons) investigates the ability of the Hopfield network into face recognition but on a limited basis. It transforms the image into a low resolution, then greyscale and use the Hebb rule to store the image in a weight matrix [9]. It experiments with different distortion levels on the dataset to see the level of accuracy that the network will achieve. It showed detailing the accuracy of both models with various distortion levels.

Table 5 Number of successful for 60 X 60 images with distortion level 5-45%

Results for 60×60 (pixel size) facial images									
Facial images	Distortion Percentage								
	5%	10%	15%	20%	25%	30%	35%	40%	45%
1	100	100	100	100	100	100	99.8	94.4	72.2
2	100	100	100	100	100	100	100	99.8	88.8
3	100	100	100	100	100	100	100	99.4	86.2
4	100	100	100	100	100	100	100	100	85.2
5	100	100	100	100	100	100	100	96	69
6	100	100	100	100	100	100	97.6	91	63
7	100	100	100	100	100	100	99.4	96	78.2

Table 6 Number of successful retrievals for 10 X 10 images with distortion level 1-17%

Results for 10×10 (pixel size) facial images									
Facial images	Distortion Percentage								
	1%	3%	5%	7%	9%	11%	13%	15%	17%
1	100	100	100	100	100	100	100	100	100
2	100	100	100	100	100	100	100	100	99.8
3	100	100	100	100	100	100	100	99.4	99.8
4	100	100	100	100	100	100	100	99.8	99
5	100	100	100	100	100	100	99.8	100	99.6

The results were that the 60 × 60 model achieves 100% until 30% distortion while the 10 × 10 model achieves 100% at only 15%. So, it is a trade of although the smaller model is less computationally expensive, but it may produce less accurate results at high distortion. The third paper embedded HNNs into a convolutional neural network (CNN) to solve the complexity of the fully connected layers and save the features created by the networks in the HNNs, then optimize the pattern storage using the knapsack problem formulation. Finally, they tried the final model on MNIST dataset with added noise to it [10]. The CNN first is trained for feature selection then convert the trained patterns into binary patterns so it can distribute it over the parallel HNNs. Then, the HNN with the closes result (converges closest to the desired state) is chosen as the final result.

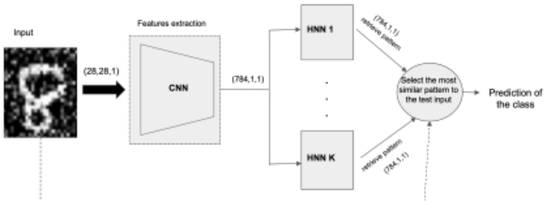

Fig 8 Architecture for Image recognition using CNN and parallel HNNs

The activation function for the CNN was Relu function. The size of each HNN was 784 neurons.

Table 6: Accuracy rate comparison

Model	# of Parameters	Accuracy AWGN	Accuracy Motion	Accuracy Contrast
Lenet5-Like (with FC Layers)	324,858	97.12%	96.50%	93.82 %
Our approach (parallel networks)	**124,344**	**97.52%**	**97.72%**	**94.84%**

In conclusion, the paper showed that it achieved high accuracy of detection with significantly less parameters.

III. COMPARITIVE ANALYSIS

Comparing between the papers and applications using the size of the dataset as criteria will not be taken into consideration as a critical factor since the HNN does not depend critically on the size of the dataset; because the HNNs can learn a few patterns from a little data so it can store the patterns and evaluate based on them. Therefore, it does not need a huge quantity of data to learn and store these patterns. Especially when they have the limitation of storing too few patterns.

A. Image Restoration

Putting the two papers against each other, The first paper that restores the UAV motion blurred images It did not state the accuracy of its approach, but it stated that it converges faster than traditional HNNs. It takes only 8 times to convergence while the traditional activation function can take 17 iterations [3]. So, it is seen that the critical parameter that affects the improvement of that model is the new activation function as seen in equation (5). In the second paper, the model was evaluated using the error rate which turned out to be 1.27 x 10^-5 [4]. Since the first paper did not provide any evaluation metric or numerical results, the first paper's digital model can be a reference to the same error rate, so it is seen that the hardware circuit was faster and more efficient. Also, the hardware circuit can perform parallel computations in a much less time than any digital network regardless of the architecture because the digital model will perform it iteratively.

B. *Image Classification*

The first paper's classification model resulted in accuracy of 85% which is a reasonable result.

The Second paper was discussing a method of supervised learning to help the HNNs in classification. In this paper it has used MNIST dataset and used the 60,000 images set which makes this model remarkably the only model that a HNN was trained on a relatively large dataset. The hidden layers were 256 neurons while the input layer was 784 neuron (28 x 28), and the output layer was 10 neurons which are the 10 digits. The network's structure is generally different than any different HNN. It has a hierarchical structure, and it is not recurrent like the traditional HNN. Also, it has asymmetric weights [6], While the first paper, its structure was a typical HNN. The following table compares between the accuracy percentage of both models. The second paper did not state the accuracy rate, but it stated the error rate. So, it can be converted by: accuracy rate = 1 - error rate.

low-cost digital image processing system for classification	Hopfield Neural Network for Classification Using a STDP-Like Rule
85%	100% after 45 epochs

It is seen that the classification model of the second paper mostly reaches a perfect accuracy. Looking at the reasons that might have been the factors that contributes to this accuracy, the architecture is considerably a factor. The activation function is also different than a HNN where a HNN uses a sign function as activation function, but this model uses a rectifier activation function.

C. *Image Recognition*

The first two papers discuss the same scope using the same dataset of the seven facial images. Comparing those papers with the third paper, first it is noticed that both models used noise. The first two papers use distortion with multiple percentages to see the limits of their model [7] [9]. Referring to table 5 and 6, The accuracy rate reaches 100% in both 60 x 60 images at distortion up until 30% and 10 x 10 images up until 17% distortion. The third paper uses multiple kinds of noise like white Gaussian noise both contrast and Gaussian noise, and motion blur. The accuracy rate of the third paper was evaluated for each kind of noise. The paper was comparing with another approach, but the rate of the paper's approach will be more focused on.

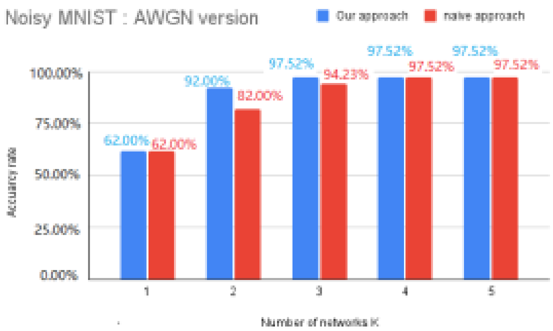

Fig 9 Classification's accuracy on the MNIST dataset with added white Gaussian noise

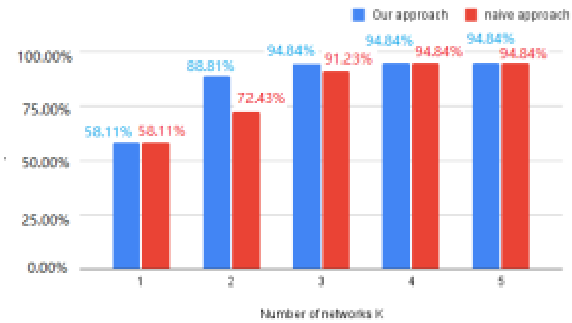

Fig 10 Classification's accuracy on the MNIST dataset with reduced contrast + AWGN

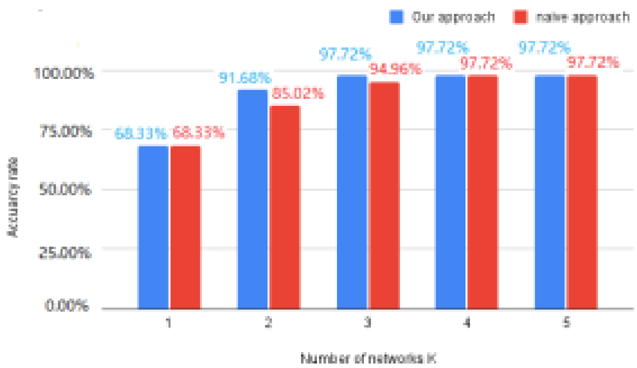

Fig 11 Classification's accuracy on the MNIST dataset with motion-blurred noise.

Each figure states the accuracy rate after embedding certain number of HNNs into the CNN with different kinds of noise. Using the gaussian noise, the highest accuracy achieved was 97.52% while having 4 or 5 HNNs. While using both contrast and Gaussian noise, the highest accuracy reached was 94.84% while having 4 or 5 HNNs. Finally, the highest accuracy reached was 97.72% when motion blur was applied using 4-5 HNNs. This paper's network proved to have achieved higher accuracy than the first and second paper's model. The reason behind this can be the use of CNNs for feature selection in the second model so it helped in selecting the most important features for training. Also using multiple HNNs to find the best result is more accurate than using just one HNN and consider its result the best. Comparing the results of both models, the accuracy is close, but the second model tries on many kinds of noises and ensure high accuracy in all of them. The first model also has its advantages where the model that is applied on 10 x 10 images is small and efficient and proves of good use on extremely low-resolution images.

Comparing the two papers that both used the same dataset MNIST [7] [10]. Although Image recognition and classification can be considered close, but each paper was grouped with other papers to fit the comparison criteria. But since they are both using the same dataset for the same purpose, a comparison can take place. The paper that uses hierarchal model does not apply noise to the dataset while the other model does. Applying noise can prepare the network to detect images better in case the incoming images are already distorted. The limitations of the first model are that HNNs can be unstable [7]. The second model could be facing the same problem since it uses many parallel HNNs, but it was no stated explicitly in the paper. The average accuracy reached in the first paper designed a model that reached 100% accuracy (0% error rate) after 5 epochs where the model consisted of two hidden layers with feedforward connections with learning rates 0.002, 0.001 for each layer respectively. Then, they proposed another model that reaches accuracy of 99.98% accuracy (0.02% error) after 60 epochs where they connected the hidden neurons of the first model to make it recurrent. The third and final model that they introduced where it uses symmetric weights had an accuracy rate of 99.09% accuracy (0.01% error rate) after 200 epochs. While in the model of the other paper that uses CNN and HNNs had was 97% accuracy on average with all kinds of noise. Although the fit paper did not augment or apply noise to its data, it still reached higher accuracy than the second paper. Therefore, the data augmentation might not be a clear factor in this case of increasing the model's accuracy. The second paper used Relu function as activation function which might explain that a linear activation function may not have fit the model and application as much as the first paper's models did.

IV. CONCLUSION

The conclusion of this paper is that Hopfield networks were considered obsolete due to their limitations and weaknesses although they are closer to imitating the brain than the other artificial neural networks that use back propagation. With the emergence of modern Hopfield, there is now a chance of applying the modern Hopfield networks into many applications and make use of its big capacity and ability to store patterns, and the more efficient energy function. Also, it is concluded that Hopfield networks act as a great layer that can be embedded into other networks so it can store the patterns and features like in the paper that used the CNN [10]. For future research, it is recommended to apply modern Hopfield Networks more since they are new. Also, to explore the idea of making the Hopfield Network asymmetric as in the application discussed above, it proved its efficiency and accuracy.

V. REFERENCES

[1] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proceedings of the National Academy of Sciences,* vol. 79, no. 8, 1982.

[2] H. S. B. L. J. S. P. W. M. A. T. G. L. H. M. P. M. S. G. K. G. V. K. D. K. M. K. G. B. J. &. H. S. Ramsauer, "Hopfield Networks is All You Need," 2020.

[3] Z. Zhang, "UAV Motion-Blurred Image Restoration Using Improved Continuous Hopfield Network Image Restoration Algorithm," *Journal of Information Hiding and Multimedia Signal Processing* , vol. 8, no. 4, 2017.

[4] Y. L. X. W. Qinghui Hong, "Memristive continuous Hopfield neural network circuit for image restoration," *Neural Computing and Applications,* 2020.

[5] T. S. C. L. B.-R. C. L. S. F. C. F. C. M. A. Igor R. Fermo, "Development of a low-cost digital image processing system for oranges selection using hopfield networks," *Euruopean Federation of Chemical Engineering: Part C,* vol. 125, pp. 181-192, 2021.

[6] T. W. Xiaolin Hu, "Training the Hopfield Neural Network for Classification Using a STDP-Like Rule," in *Neural Information Processing,* Guangzhou, 2017.

[7] N. S. A. K. E. K. S. Neha Soni, "Low-Resolution Image Recognition Using Cloud Hopfield Neural Network," *Advances in Intelligent Systems and Computing,* vol. 563, pp. 39-46, 2018.

[8] A. K. Narotam Singh, "Cloud Hopfield Neural Network: Analysis And Simulation," in *International Conference on Advances in Computing, Communications and Informatics* , 2015.

[9] E. K. S. A. K. Neha Soni, "Application of Hopfield Neural Network for Facial Image Recognition," *International Journal of Recent Technology and Engineering,* vol. 8, no. 1, 2019.

[10] A. N. Fekhr Eddine Keddous, "Optimal CNN–Hopfield Network for Pattern Recognition Based on a Genetic Algorithm," *Algorithms,* 2021.