Transactions on Machine Intelligence 7(2) (2024) 98-106

ISSN Online: 2821-1693

Transactions on Machine Intelligence

Journal Homepage: https://tmachineintelligence.ir/index.php/TMI

Phase Transition of the 2D Square Ising Model in a Homogeneous Magnetic Field Using the Metropolis Monte Carlo Algorithm and Separation of Different Phases via CNN Method

R. Rastgarpour [1], N. Majd [2,*]

[1] Faculty of Engineering Sciences, School of Engineering, University of Tehran, Tehran, Iran
[2] Assistant Professor, Algorithm and Computation Group, Faculty of Engineering Sciences, School of Engineering, University of Tehran, Tehran, Iran

ARTICLE INFO

Article History:
Received 6 March 2024
Received in revised form 15 May 2024
Accepted 30 May 2024
Available online 14 June 2024

Keywords:
Quantum Spin Networks, Hamiltonian, Deep Learning, Convolutional Neural Network (CNN), Two-Dimensional Ising Model.

ABSTRACT

Quantum spin networks represent systems in which quantum spins are arranged on a topological lattice, where the nature of spin interactions and the influence of external magnetic fields can lead to complex collective behaviors. The Hamiltonian governing such systems determines the energy landscape and phase transitions under various thermodynamic conditions. In this paper, we focus on the two-dimensional Ising spin model with periodic boundary conditions subjected to a uniform external magnetic field. Initially, we employ the Metropolis Monte Carlo (MP-MN) algorithm to simulate the system and identify its phase transitions at different magnetic field strengths. The emergence of ordered and disordered phases in response to thermal fluctuations and magnetic interactions is systematically analyzed. Subsequently, we explore the application of convolutional neural networks (CNNs), a powerful class of deep learning models, to detect and classify the phases of the Ising model based on spin configurations generated at a fixed temperature. The CNN is trained using labeled data representing different magnetic field values, and its performance in phase prediction is quantitatively evaluated. The results demonstrate that CNNs can successfully learn complex spin patterns and provide accurate classification of spin phases, highlighting their potential for analyzing phase transitions in statistical physics models.

1. INTRODUCTION

Spin networks are generally considered as fundamental classes in machine learning. This paper aims to introduce the two-dimensional Ising model in a square lattice with periodic boundary conditions under a uniform magnetic field. The goal of this study is to investigate the behavior of the magnetic field in the various phases of the model under examination [1-6].

* Corresponding Author: naymajd@ut.ac.ir
Assistant Professor, Algorithm and Computation Group, Faculty of Engineering Sciences, School of Engineering, University of Tehran, Tehran, Iran

http://dx.doi.org/10.47176/TMI.2024.98

The Ising model is a mathematical model of ferromagnetism in statistical mechanics. This model consists of discrete variables representing the magnetic dipole moments ("spins") of atomic particles, which can be in one of two states (+1 or -1). The spins are arranged on a two-dimensional lattice with periodic boundary conditions, and each spin interacts with its neighbors. The physical behavior of the spin system tends to minimize its energy. However, heat and temperature disrupt this tendency, creating the possibility of various structural phases. The two-dimensional square Ising model is one of the simplest statistical models to demonstrate phase transitions. Machine learning is considered a subset of artificial intelligence, primarily involving supervised learning, unsupervised learning, and reinforcement learning [7]. In this paper, we focus on supervised learning. Supervised learning typically involves regression models, classification models, and neural networks (NN). In this study, we utilize a deep convolutional neural network (CNN) approach to classify the Ising spin network in the presence and effect of an external magnetic field on the spins of the network.

This paper is organized as follows:

- Introduction of the Ising model with its macroscopic parameters
- Phase transition of the two-dimensional Ising model in the presence of a magnetic field
- Implementation of the MP-MN algorithm to obtain the equilibrium states of the Ising model
- MP-MN simulation results
- Presentation of the CNN neural network
- Conclusion and simulation results

2. TWO-DIMENSIONAL ISING MODEL ON A SQUARE LATTICE

Just as the Turing machine is fundamental for classical computers, the Ising model is the simplest model for quantum computers. The Hamiltonian of a system represents the total energy of the system in question.

$$\mathrm{E(S)} = -J \sum_{<i,j>}^{N} s_i s_j \; - h \sum_{i}^{N} s_i \tag{1}$$

where s_i and s_j represent the spins of particles ii and j, which take values of +1 or -1. h is the external magnetic field, and J is the strength of the exchange interaction between the particles. In the classical view, the Hamiltonian is understood as the sum of the kinetic and potential energies. For the simple Ising model, the Hamiltonian is defined as follows:

$$\mathrm{H(s)} = -\sum_{<i,j>}^{N} J_{i,j}\, s_i s_j \; - h \sum_{i}^{N} s_i \tag{2}$$

In this relation, s_i=±1 represents the classical spin operator of 1/2. The first summation is over the nearest neighbors <i,j>, and the second summation is over the vertices of the lattice. Additionally, $J(i,j) = 1$ represents the strength of the exchange interaction between particles i and j. Due to the simplicity of the model, the phase transition in this model serves as a suitable example for training the neural network. Neural network training is discussed in Section 6.

3. PHASE TRANSITION OF THE 2D ISING MODEL IN THE PRESENCE OF A MAGNETIC FIELD

The two-dimensional Ising model, in the absence of an external magnetic field, experiences a phase transition at a critical temperature of 2.26 Kelvin, which is known as the critical temperature. The phase transition point for this model is expressed by Equation 3.

$$T_c = \frac{2J}{\ln(1+\sqrt{2})} \tag{3}$$

To observe the phase transition in the presence of an external magnetic field, we examine the behavior of magnetization (M) as the external magnetic field changes at a constant temperature (T). Figure 1 illustrates the magnetization of ferromagnetic atoms in a spin network under an external magnetic field (H) ranging from -4 to +4. The temperature is fixed and considered to be below the critical temperature (β_c). Based on the magnitude of magnetization in the external field, the network is classified into four different categories, as indicated by four distinct colors in Figures 1 and 2.

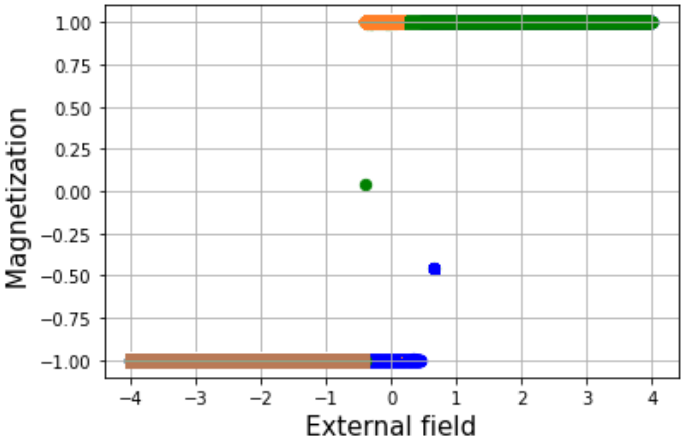

Fig. 1. Magnetization (M) of N ferromagnetic atoms as a function of the external magnetic field at a constant temperature, where $\beta < \beta_c$.

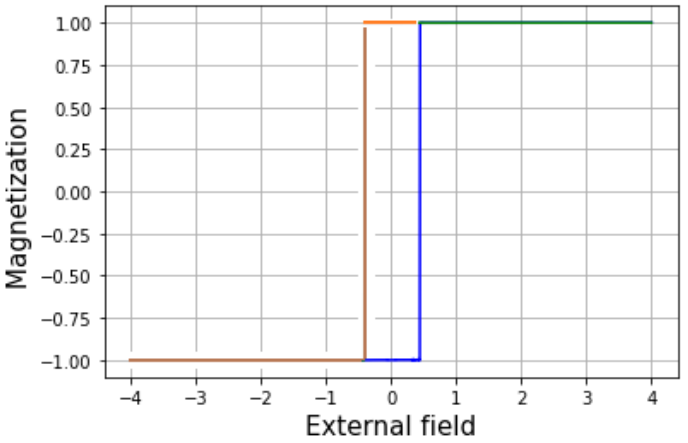

Fig. 2. Magnetization (M) of N ferromagnetic atoms as a function of the external magnetic field at a constant temperature, where $\beta < \beta c$.

The distance between the two lines in the phase transition diagram, as shown in Figure 3, is denoted as ΔH. This distance changes at different temperatures. In sections 5 and 6, the relationship between ΔH and temperature will be examined, and the ΔH versus temperature graph will be plotted for three networks with different dimensions.

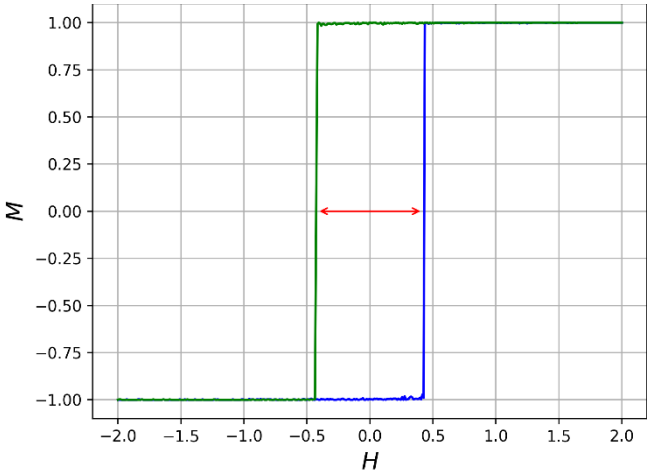

Fig. 3. The distance ΔH in the phase transition diagram, highlighted by the red arrow.

4. IMPLEMENTATION OF THE MP-MN ALGORITHM FOR OBTAINING THE GROUND STATE OF THE ISING MODEL

We are searching for a state of the Ising network that has the lowest energy and has reached equilibrium. Therefore, we use the MP-MN algorithm. In this algorithm, at each iteration, a particle is randomly selected, and the system's energy is calculated. For the calculation, it is sufficient to compute the energy of that particle with its neighboring particles. The spin of the selected particle is flipped, and the energy is recalculated. The state with lower energy is preferred. This process is continued until the network reaches equilibrium. The flowchart of the MP-MN algorithm is shown in Figure 4.

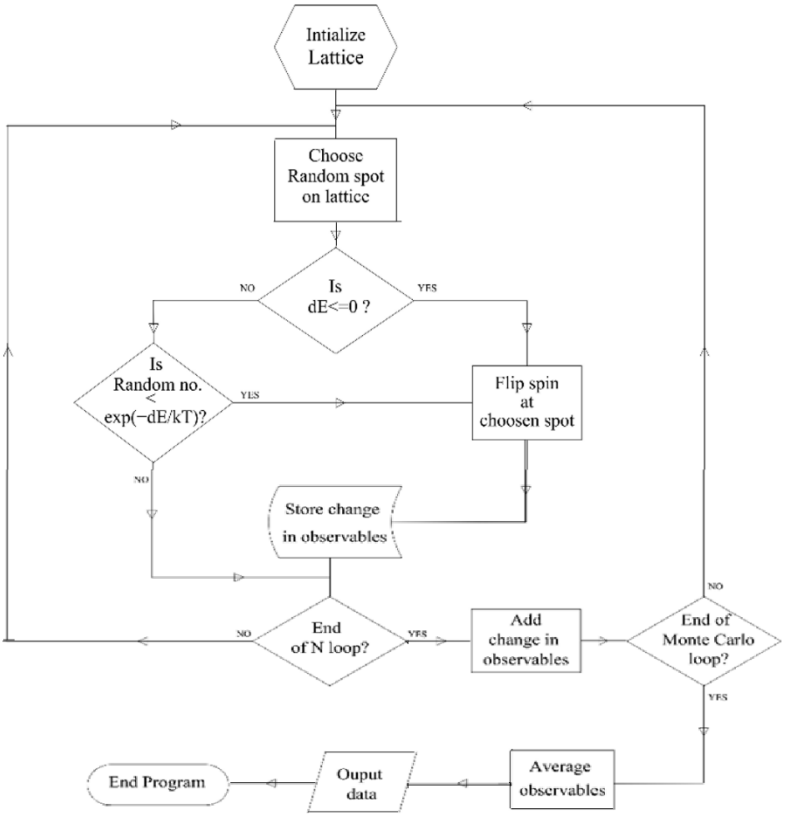

Fig. 4. Flowchart of the MP-MN Algorithm.

After the network reaches equilibrium, we calculate the magnetization of the N×N network. This is done for three networks with dimensions of 8×8, 10×10, and 15×15 at a temperature β and external magnetic field β_c and H_c represent the critical temperature and critical field, respectively. The number of steps in the algorithm is shown in Table 1, with 4000 steps to reach equilibrium and 3000 steps during the execution of MP-MN. For each network, the minimum value of ΔH is also calculated.

Table 1. Magnetization Calculation for Three N×N Networks at Temperature and External Magnetic Field and the Minimum ΔH Value

N	Steps in MP-MN Execution	Steps to Reach Equilibrium	Min ΔH	$\beta < \beta_c$ $H < H_c$	$\beta < \beta_c$ $H > H_c$	$\beta > \beta_c$ $H < H_c$	$\beta > \beta_c$ $H > H_c$
8	3000	4000	0.02706766917	-1	1	-0.95416666	0.975
10	3000	4000	0.08120300752	-1	1	-0.96266666	0.96266666
15	3000	4000	0.08320802005	-1	1	-0.96503703	0.96385185

5. RESULTS OF MP-MN SIMULATION

When plotting the phase transition diagram for a network at different temperatures, we observe that as we approach the critical temperature, ΔH becomes smaller. For each of the three different network sizes, 10 networks are randomly generated at a specific temperature, and ΔH during the phase transition is calculated. This process is repeated for temperatures of 1.2, 1.4, 1.6, 1.8, 2.0, 2.2, and 2.4. As a result, for each temperature β, ΔH is calculated 10 times. The average ΔH values for each temperature are then plotted, as shown in Figure 5. Specifically, the blue graph in Figure 5 represents the average of the 10 ΔH plots versus temperature for the 8×8 network. Similarly, for the 10×10 and 15×15 networks, the orange and green plots are provided, respectively. As expected, ΔH decreases with increasing temperature until it reaches the critical temperature.

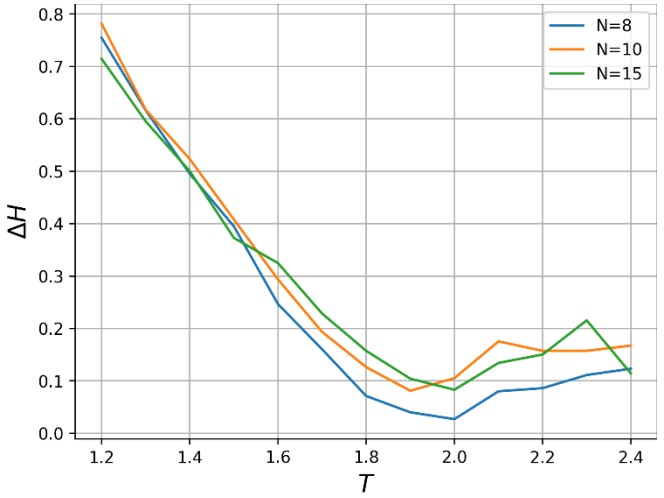

Fig. 5. The average ΔH as a function of temperature for three networks with dimensions N×N

6. CNN METHOD

In the domain of deep learning, Convolutional Neural Networks (CNNs) have emerged as one of the most prominent and widely adopted architectures. Their primary strength lies in their ability to automatically extract relevant features from data, thereby eliminating the need for manual feature engineering. CNNs have been successfully deployed across numerous fields such as computer vision, speech recognition, and facial identification. The architecture of CNNs is inspired by the neural connectivity patterns observed in biological brains, and it resembles the structure of traditional artificial neural networks.

A typical CNN is composed of multiple layers, each serving a distinct purpose:

- **Convolutional Layer:** This is the core component of a CNN. It comprises a set of filters, also known as kernels, that slide over the input—often an image represented as an N-dimensional matrix—to produce feature maps that capture spatial hierarchies in the data.
- **Pooling Layer:** Pooling is a form of downsampling that reduces the spatial dimensions of the feature maps, while preserving the most salient features. Predefined kernel sizes and stride values are used during this operation. Common pooling techniques include max pooling, average pooling, minimum pooling, global average pooling (GAP), and global max pooling, with max pooling and GAP being the most widely used.
- **Activation Function:** These functions introduce non-linearity into the model. The activation function processes the weighted sum of inputs and bias to determine the neuron's output. It effectively governs whether a neuron should be activated, which is crucial for the network's learning capacity.
- **Fully Connected Layer:** Positioned near the end of the CNN, the fully connected (FC) layer links every neuron to all neurons in the previous layer. It typically acts as the classifier, receiving input in the form of a flattened vector from the preceding convolutional or pooling layer. The output of the FC layer represents the final prediction of the network.
- **Loss Function:** The loss function quantifies the discrepancy between the predicted output and the ground truth labels in the training data. This error signal is then minimized during training using optimization algorithms, ensuring the CNN learns to improve its predictions over time.

In this research, a CNN was employed to classify the phases of a two-dimensional Ising spin model under varying magnetic field strengths. Notably, the network does not require prior knowledge of the system's Hamiltonian. To train and evaluate the model, data samples were generated for a lattice of size L×L=20×20L \times L = 20 \times 20L×L=20×20, comprising 400 spins. For each network configuration, simulations were conducted at a fixed temperature, with the external magnetic field swept from +4 to -4 (yielding 5000 samples), and then back from -4 to +4 (producing another 5000 samples), resulting in 10,000 data points in total. These configurations were grouped into four distinct classes based on their spin arrangements. A CNN, implemented in Python, was trained to learn and accurately classify these phases. The architecture of the implemented CNN model is depicted in Figure 6.

```
#----Creating CNN model-----
from keras.models import Model
import keras
from keras import layers
from keras.optimizers import adam_v2
from keras.optimizers import gradient_descent_v2
from keras.losses import categorical_crossentropy
sgd = gradient_descent_v2.SGD()
#--------------------Architecture of model----------------
model = keras.Sequential(
   [
      keras.Input(shape=(20,20,1)),
      layers.Conv2D(64, kernel_size=(3, 3), activation="relu", padding='same'),
      layers.MaxPooling2D(pool_size=(2, 2)),
      layers.Conv2D(128, kernel_size=(3, 3), activation="relu", padding='same'),
      layers.MaxPooling2D(pool_size=(2, 2)),
      layers.Flatten(),
      layers.Dense(100, activation='relu'),
      layers.Dropout(0.1),
      layers.Dense(4, activation='sigmoid')
   ]
)
```

```
#-------------------------------------------------
model.summary()
batch_size = 128
epochs = 200
model.compile(loss="categorical_crossentropy", optimizer="adam", metrics=["accuracy"])
# Train CNN model
network_history =model.fit(train_x, train_y, batch_size=batch_size, epochs=epochs, validation_split=0.1)
```

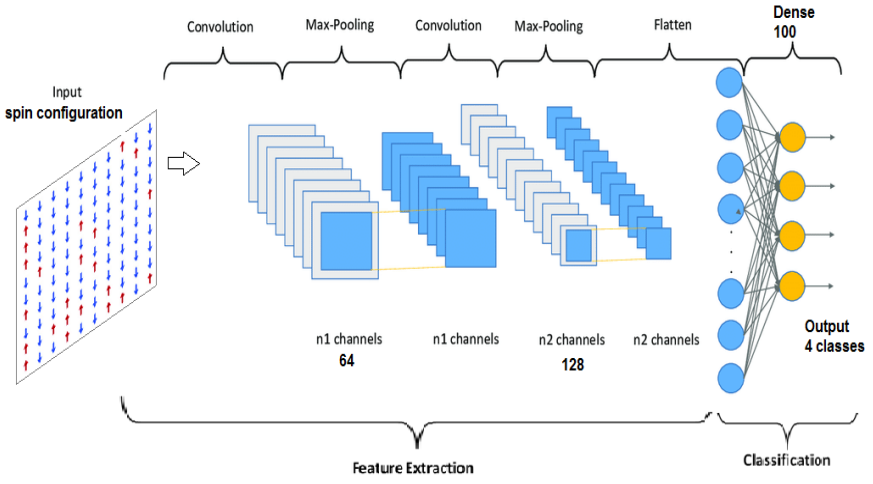

Fig. 6. Architecture of the CNN model

7. SIMULATION RESULTS OF CNN

To evaluate the CNN model, we plot the cost and accuracy over 500 epochs. As shown in Figures 3 and 4, the cost or error on the test data has decreased compared to the training data. Additionally, the accuracy on the test data is higher than the accuracy on the training data. As a result, the trained model, without any knowledge of the system's Hamiltonian, has demonstrated good classification capabilities.

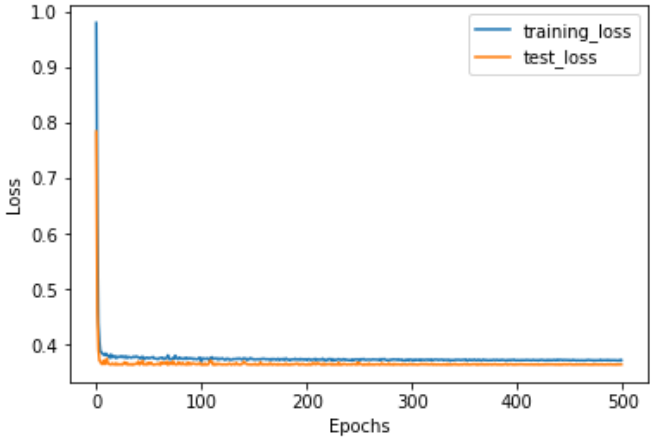

Fig. 7. Error plot versus the number of Epochs for training and test data.

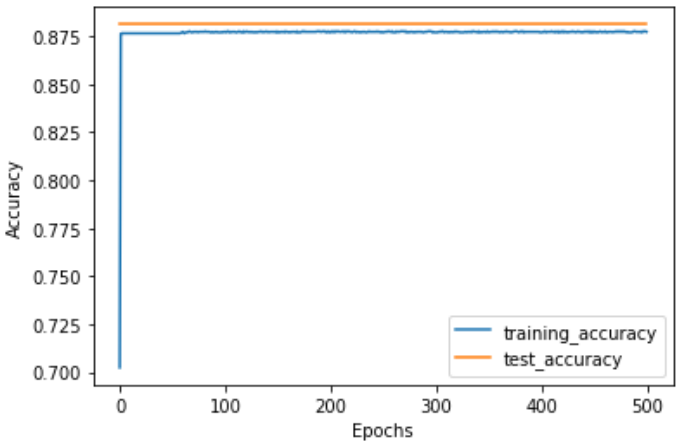

Fig. 8. Accuracy plot versus the number of Epochs for training and test data.

8. CONCLUSION

In this study, the phase behavior of the two-dimensional Ising spin network with periodic boundary conditions under a uniform external magnetic field was investigated. The results obtained through the Metropolis Monte Carlo (MP-MN) algorithm revealed that variations in the magnetic field lead to distinct phase transitions in the spin configurations. Subsequently, a Convolutional Neural Network (CNN) was employed to analyze and predict the different phases of the system. After being trained on simulation data, the CNN demonstrated a high capability in accurately identifying various phases of the spin network at a fixed temperature. These findings indicate that deep neural networks are highly effective in recognizing complex patterns and predicting phase transitions, making them a powerful tool for the analysis of statistical and physical systems. Overall, this research highlights the potential of integrating traditional statistical methods with modern machine learning techniques to advance the modeling and understanding of complex physical phenomena.

Declaration

We acknowledge that we used ChatGPT to enhance the academic writing of our manuscript while ensuring the originality and integrity of our work.

Transparency Statement

The data supporting this study are available upon reasonable request to the corresponding author, subject to ethical and confidentiality considerations.

Acknowledgments

We would like to express our gratitude to all individuals who contributed to this project.

Declaration of Interest

The authors declare that they have no competing interests.

Funding

This research received no specific grant from any funding agency, commercial, or not-for-profit sectors.

REFERENCES

[1] Huang, K. (1987). Statistical mechanics (2nd ed.). Wiley.

[2] Reichl, L. E. (2016). A modern course in statistical physics (4th ed.). Wiley.

[3] Landau, L. D., & Lifshitz, E. M. (1980). Statistical physics (Vol. 5, 3rd ed.). Butterworth-Heinemann.

[4] Pathria, R. K., & Beale, P. D. (2011). Statistical mechanics (3rd ed.). Elsevier.

[5] Binder, K., & Heermann, D. W. (1997). Monte Carlo simulation in statistical physics (2nd ed.). Springer. https://doi.org/10.1007/978-3-662-03336-4

[6] Hu, W., Singh, R. R. P., & Scalettar, R. T. (2017). Discovering phases, phase transitions, and crossovers through unsupervised machine learning: A critical examination. Physical Review E, 95(6), 062122. https://doi.org/10.1103/PhysRevE.95.062122

[7] Świderski, B., Osowski, S., Gwardys, G., Kurek, J., Słowińska, M., & Lugowska, I. (2022). Random CNN structure: Tool to increase generalization ability in deep learning. EURASIP Journal on Image and Video Processing, 2022(1). https://doi.org/10.1186/s13640-022-00580-y

[8] Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al Amidie, M., & Farhan, L. (2021). Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. Journal of Big Data, 8, Article 53. https://doi.org/10.1186/s40537-021-00444-8

[9] Jadon, S. (2020). A survey of loss functions for semantic segmentation. In 2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB). https://doi.org/10.1109/CIBCB48159.2020.9277638