Recognition and Solution of Handwritten Partial Differential Equation Using Deep Learning

Ahmed Yehia [1], Mohamed Magdy [2], Amr Ashraf [3], Mahmoud Gamal [4], Mohanad Walid [5] Computer and AI Engineering
Military Technical College, Cairo, Egypt
[1] ahmedyehiashaat@gmail.com, [2] m7mdcena77@gmail.com, [3] Amora.z@live.com, [4] *Mahmoudgamal910@ymail.com*.
Supervisor: Maj. Gen.(Ret.) Prof. Dr. Gouda Ismail Mohamed

***Abstract*— Water pollution is one of the leading environmental issues faced especially at developing countries all over the World. However, the study of pollution movement is a necessary basis for solving water quality problems. As there is a great development in the technologies of Artificial Intelligence So it can be used for solving handwritten partial differential equation of water quality problems. The problem of handwritten mathematical expression recognition is one of the complicated issue in the area of computer vision research. Segmentation and classification of specific character makes the task more difficult. In this paper, groups of handwritten Advection Partial differential equations are considered to recognize and make a solution for those equations. Horizontal compact projection analysis and combined connected component analysis methods are used for segmentation. For classification of specific character, we apply Convolutional Neural Network. Each of the correct detection, Nangs algorithm is used for the solution of the equation. Finally, the experimental results show the great effectiveness of our proposed system.**

Keywords-Character segmentation; Convolutional neural Network; Projection analysis; Advection equation; Connected component; Partial differential equation; Nangs.

I. INTRODUCTION

Differential equations are used to describe the exponential growth or decay over time. It has the ability to predict the world around us. It is widely used in various fields such as Physics, Chemistry, Biology, and Economics and so on. From Differential equations types, there are the Partial differential equations (PDE) which are widely used in engineering and physics to model natural phenomena such as heat transfer, wave propagation, diffusion, and electrostatics. So Partial differential equations can be used as a solution for the water pollution, which is one of the leading environmental issues faced especially at developing countries all over the World.

Partial differential equations have formulated the mathematical model for soluble and insoluble water pollutants by using one-dimensional Advection diffusion equation. Now day the progress in the Computer Vision is very high as the computer is used in many applications that human to do it. Which lead scientists to the invention of a short cut approach, without discretization process, to solve the PDE problems, namely by using Deep learning (DL).

DL is a branch of Artificial Intelligence (AI) based on the idea that systems can learn from data, identify patterns and make decisions with minimal human intervention. It has many applications like Image Recognition, Sentiment Analysis, Banking Domain… etc. This is potential to make scientific DL as a new sub-field of research. After all the development that the technologies have reached today also Handwritten mathematical expression recognition is still a most challenging job to do in the area of computer vision [4] [5].

The primary task for the recognition of mathematical expression is to segment the character and then classify those characters. There are many Computer's technologies need to be identified before overall the project.

AI is now day technology to make the computer's behaviors is exactly like human. Machine Learning (ML) is a subset of AI that make the computer have the ability to learn. DL is a specified type of ML; it is working on build neural networks layers to the computer like that of the human's brain.

Convolutional neural network (CNN) is one of the mostly used classification model in computer vision area. In the last few years, deep CNN leaning has proved the outstanding performance in the field of image classification, machine learning and pattern recognition, CNN extract feature from the image by a series of operations [5].

In this research, we mainly focus on the recognition of Advection equation [6] and after successful detection of Advection equation; we apply Nangs method for solving PDE problems [2]. An operation that seeks to decompose an image of a sequence of characters into sub images of individual symbols is Character segmentation. In the case of segmenting each Advection equation from the original image a difficulty exists for applying connected component in the case of mathematical symbol such as for '=' [4], which is a combination of two connected components. We solve this problem by using combined connected component analysis method. In this research, we mainly focus on the recognition of Advection equation and after successful detection of Advection equation; we apply Nangs method for solving PDE problems.

II. Related Work

This paper as in Md Bipul Hossain paper [4] work with Self-made Dataset and not with popular one because it applies better accuracy as in Table 1.

For Classification we applied CNN only and not a combination of segmentation and classification as in Karanveer Mohan paper [5] to get better accuracy at each stage of the model and to develop each stage at its own.

Table 1. Classification comparison between papers.

Paper	Dataset	Methodology	Results
Md Bipul Hossain [4], 2018	Modified version of NIST for digits, Self-made Dataset	Divided segmentation into equation line and character segmentation	Equation recognition rate=39.11 % Character recognition rate=91.08 %
Karanveer Mohan [5]	Data comes from CHROME competition	Combination of segmentation and classification used as it use heuristic+CNN	Segmentation rate=39% Classification rate=90% Heuristic+CNN=88%
Jonathan J.HULL [12]	20,000 isolated characters extracted from handwritten postal addresses.	Three independent character recognition algorithms are and the results are combined.	When 3 models are combined it get accuracy of 97.1%

There are three methods for solving PDEs based on ANN include PyDEns, NeuroDiffEq, and Nangs. Those methods differ by the way generating the data points, setting up the boundary conditions, as well as the loss functions as shown on Table 2.

Table 2.ANN methods for solving PDEs.

PyDEns	NeuroDiffEq	Nangs
Built under of DeepGalerkin-Method as it was introduced in Alexander Koryagin paper [14]	The key idea of solving PDEs using NeuroDiffEq is by casting the trial approximate solution (TAS).	similar to the PyDEns. It is not necessarily, to build any trial solution in order to minimize the loss functions. The only difference is that to define a set of points inside the domain, it is generated without any distributions, instead, we build a mesh points as done in the traditional methods.

When the three algorithms are used to solve Heat equation we noticed that Nangs method is the best one since it consumed only 14.13 minutes to solve 200 dimensional problems, compared to 27.35 minutes and 1.30 hours for PyDEns and NeuroDiffEq respectively solving the same dimensions as shown in Table 3. Form Danang A. Pratama paper [2] so we have chosen Nangs for solving Advection differential equation.

Dim	PyDEns			NeuroDiffEq			Nangs		
	Iter	Time	MSE	Iter	Time	MSE	Iter	Time	MSE
50	10000	03:57	3.9×10^{-4}		03:03	0.0153		05:29	0.0471
75	10000	06:54	2.2×10^{-4}		08:37	0.0031		05:55	0.0081
100	5000	06:53	0.0056	50	21:52	0.0017	100	06:22	0.0045
150	5000	14:25	0.0055		48:34	4.5×10^{-4}		09:25	5×10^{-4}
200	5000	27:35	0.0635		01:30:55	1.5×10^{-5}		14:13	1.9×10^{-5}

Table 3. Comparison result for solving heat equation.

III. PROPOSED MODEL

Fig 1. Shows the functional block diagram of the proposed model. The Recognition and Solution of Handwritten Partial Differential Equation model composed of three stages, firstly there is a Segmentation stage, then a Detection stage and finally there is a solving stage.

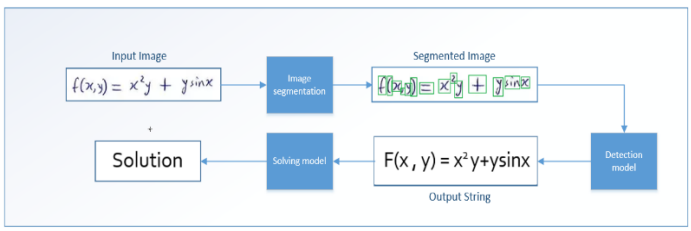

Fig 1. Workflow diagram of proposed method

A. Dataset preparation

Preparation of the dataset is the fundamental concern for this work. Therefore, we built a Python algorithm for data division into training data with 70% of Dataset, testing data of 20% and Validation data of 10%. Characters such as English digit, alphabet and mathematical symbol all can be well defined by their edges. For this reason, we first prepare dataset as the most precedence given to its edges that is enlighten the edges. We prepare some dataset by ourselves. For each category, we use 12,000 data item for the training of the network. And most of the case our network training we gained more than 95.91% training accuracy and 95.93% testing accuracy. Image size we used in dataset is 45×45 gray level image.

B. Pre-Processing

Pre-processing of the input image is the procedure, which performs changes and modifications to the image to make it fit for recognition.

C. Segmentation

Which is the process of dividing an image into multiple parts.

The Segmentation stage workflow diagram is shown in Fig 2. It firstly removes noise from the original input image by applying Binarization to it. After that, we find specific characters in the form of connected component. Each segmented character is then providing as input to the convolutional neural network model for classification of the character. The resulting character that is the output of CNN is then used for making a character string, which is similar to the original equation.

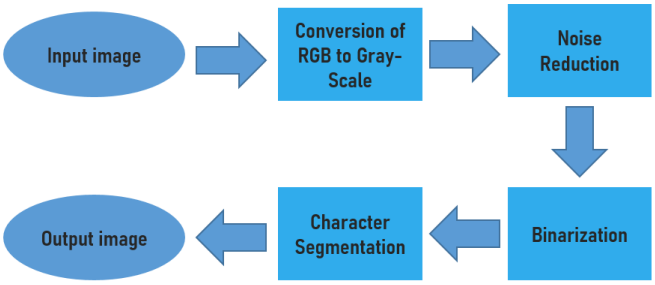

Fig 2. Workflow diagram of Segmentation stage

1) Conversion of RGB to Gray-Scale

First, this colored image is transformed into a typical gray-scale image [1] because the detection of characters on a colored image is more challenging than on a gray-scale image, formula we used is [8] [4]:

$$Y = 0.299R + 0.587G + 0.114B \quad (1)$$

2) Noise Reduction

As shown in Fig 3. Low pass filtering is used to remove the Gaussian noise from the image. Where all component, which is less than five pixels, is removed for simplicity of small-unwanted pixel noise.

Fig 3. Noise Reduction using Gaussian Blur

3) Binarization

As shown in Fig 4. The procedure of choosing a threshold value for adaptation of pixel values into 0's and 1's [4].

Fig 4. Image Binarization

4) Segmentation

The procedure that look for to decompose an image of a series of characters into sub images of individual symbols as shown in Fig 5.

A Problem arises at the point of extracting of math symbol '=' [4] [5] which is a combination of two connected components, so we used combined connected component

analysis method for the segmentation of specific character from the image.

Fig 5. Image Segmentation

D. Detection model

In this research, we used CNN [5] as a Detection model; specifically, we used a modified version of VGG16 model. So our model contains 11 Convolution Layers each with Kernel 5×5.

A convolutional layer contains of a set of filters to extract fundamental features whose parameters need to be learned. The height and weight of the filters are smaller than those of the input volume. Each filter is convolved with the input volume to compute an activation map made of neurons.

It is important to apply a nonlinear layer (Activation Layer) [4], Relu activation function is applied to avoid vanishing gradient descent problem instantly after Convolution layer.

Also 4 Max Pooling layers with pool size 2×2 The persistence of this layer is to introduce nonlinearity to a system. Pooling layers are used to reduce the dimensions of the feature maps. Thus, it reduces the number of parameters to learn and the amount of computation performed in the network. The pooling layer summaries the features present in a region of the feature map generated by a convolution layer [7].

Flatten layer is used to flatten the input, without affecting the batch size. A Flatten layer flattens each batch in the inputs to 1-dimension.

Finally, a 3 Fully Connected Layers is used which is a regular neural network layer that takes input from the previous layer and computes the class scores and outputs the 1-D array of size equal to the number of classes.

E. Solving model

Our proposed solving model is Nangs, which is a Python library, built on top of Pytorch to solve Partial Differential Equations, Nangs is based on ANN.

ANN is a computational model that mimics the way nerve cells work in the human brain. Artificial neural networks (ANNs) use learning algorithms that can independently make adjustments - or learn, in a sense - as they receive new input. This makes them a very effective tool for non-linear statistical data modeling.

Deep learning ANNs play an important role in machine learning (ML) and support the broader field of artificial intelligence (AI) technology.

The scenario of Nangs method for solving PDE problems is based on defining a set of points inside the domain, it is generated without using any distributions, instead, we build a mesh points as done in the traditional methods [2]. Here we will also overview the general methodology of neural networks to solve PDE. Consider the PDE of the form:

$$\frac{\partial u}{\partial t}(x,y) + \mathcal{L}u(x,t) = f(x,t), \quad (x,t) \in [0,T] \times \Omega,\ \Omega \subset \mathcal{R}^d \qquad (2)$$

With initial condition

$$u(x,t=0) = u_0(x) \qquad (3)$$

And boundary conditions

$$u(x,t) = g(x,t),\ x \in \sigma\Omega, \qquad (4)$$

The algorithms of solving PDE using Nangs are described as follows [2]:

1) Define a set of mesh points of P = (xm, tn) ∈ [0, T] × Ω inside the domain. These mesh points are the combination of the internal, boundary and initial points (see Fig. 5). In addition, they are used as the input values of the ANNs feed

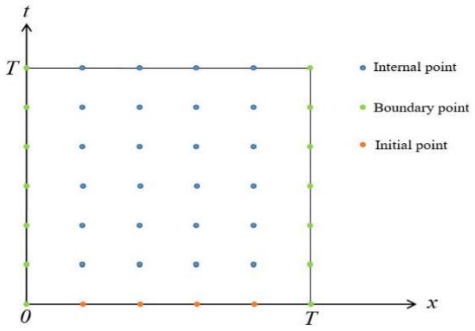

Fig 5. Internal, Boundary and Initial points

2) The internal points of ANN are compared with the right hand side of the PDE itself by using the loss function

$$\mathcal{J}(\theta)_{DE} = \left[\frac{\partial u_{net}}{\partial t}(x,t;\theta) + \mathcal{L}u_n(x,t;\theta) - f(x,t)\right]^2_{(x,t)\in P_{internal}} \qquad (5)$$

3) The initial and boundary points of the PDE are compared with the outputs of the initial and boundary conditions of ANN 3 and 4 respectively by using the loss function:

$$\mathcal{J}(\theta)_{BC} = [u_{net}(x,t;\theta) - g(x,t)]^2 \quad (x,t) \in P_{boundary} \qquad (6)$$

$$\mathcal{J}(\theta)_{IC} = [u_{net}(x,0;\theta) - u_0(x)]^2. \quad x \in P_{initial} \qquad (7)$$

IV. EXPERIMENTAL RESULTS

After training of the network of the Detection model, we use 20% of our Dataset as test images. We face some problem and we modify those parts in our system for getting correct solution.

For the segmentation, a problem occurs when we apply connected component analysis algorithm to the segmented quadratics. The symbol '= 'is a single character but composed with two distinct part as shown in the Fig 6 (a). In this research we solve this problem by considering two consecutive components with smallest width and also considering two components with same horizontal direction [4] as shown in Fig 6 (b).

(a) (b)

Fig 6. Combined connected components

For Detection Firstly we trained the model with 100 epochs and we get high accuracy of 90% but while testing we got a very low testing accuracy so to overcome this problem we used early stopping mechanism as shown in Fig 7. to avoid overfitting and making better generalization results with training accuracy of 95.91% and testing accuracy of 95.93%, thus we obtained less number of epochs and made model less complex.

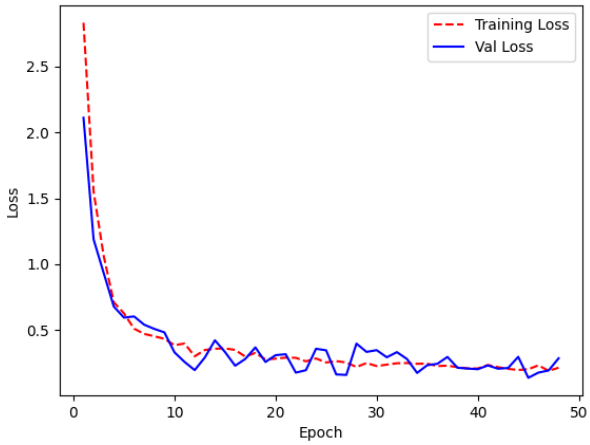

Fig 7. Training loss & Validation loss Vs. Epoch no.

The detected string is then given to the Solving model as the initial condition of the Advection differential equation, the loss is then calculated for initial condition, boundary condition and internal condition as in Fig 8.

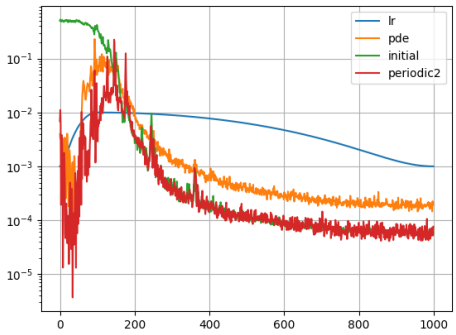

Fig 8. Loss of initial condition, boundary condition and internal condition

After training the result of the Advection, differential equation is obtained based on the initial condition of such function and then plotted as in Fig 9.

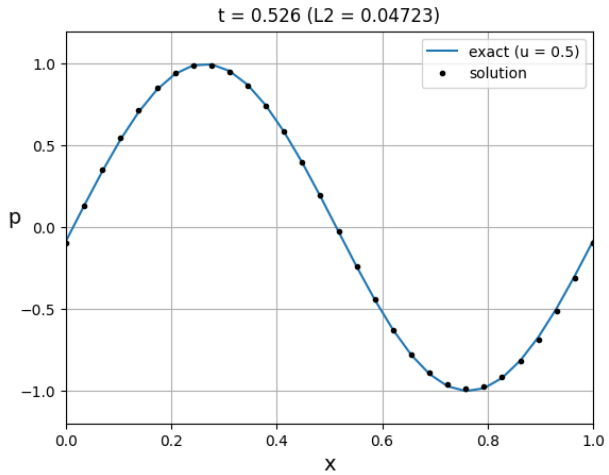

Fig 9. Solution of Advection equation

Finally, we managed to create a GUI windows program that combines the whole system to make it easy for the user to estimate his Advection differential equation problem based on his initial condition as shown in Fig 10.

Fig 10. Graphical User Interface

V. CONCLUSION

In this paper, we mainly focused on recognizing and solving of handwritten PDE. Connected component, which has very high success rate, is used for character segmentation. Improved version of connected component is used for the symbol like '=' detection which is a single symbol combined with two distinct connected components. Feature extraction the most complicated part of classification. Moreover, with some predefined feature about handwritten it is difficult to recognize handwritten. Convolutional Neural Network the most powerful classification model is used in the classification part. Once successful recognition of the Advection equation in any combination we further process the detected equation for finding the solution of the Advection equation. Artificial Neural Network the most convenient Neural Network for solving Partial Differential Equations, especially Nangs library. QT5 is a simple way for designing a GUI.

VI. REFERENCES

[1] H. El Bahi, Z. Mahani, A. Zatni and S. Saoud 'A robust system for printed and handwritten character recognition of images obtained by camera phone', Université Ibn Zohr, ESTA, Laboratoire Matériaux, Systèmes et Technologies de l'information.

[2] Danang A. Pratama, Maharani A. Bakar, Mustafa Man, and Mashuri M. 'ANNs-BASED METHOD FOR SOLVING PARTIAL DIFFERENTIAL EQUATIONS' Faculty of Ocean Engineering Technology and Informatics, Universiti Malaysia Terengganu, Malaysia.

[3] Fisher Yu, Vladlen Koltun, Thomas Funkhouser.' Dilated Residual Networks'

[4] Md Bipul Hossain 1 , Feroza Naznin 2 , Y.A. Joarder 3 , Md Zahidul Islam 4 , Md Jashim Uddin 'Recognition and Solution for Handwritten Equation Using Convolutional Neural Network' 2018 Joint 7th International Conference on Informatics.

[5] Catherine Lu , Karanveer Mohan, 'Recognition of Online Handwritten Mathematical Expressions Using Convolutional Neural Networks' Computer Science, Stanford University.

[6] Juan B. Pedro 1 Juan Maronas ˜ 2 Roberto Paredes 2.' Solving Partial Differential Equations with Neural Network.

[7] Chen-Yu Lee, Patrick W. Gallagher, Zhuowen Tu.' Generalizing Pooling Functions in Convolutional Neural Networks: Mixed, Gated, and Tree'

[8] C. Saravanan,. 'Color Image to Grayscale Image Conversion' 2010 Second International Conference on Computer Engineering and Applications.

[9] Nicholas E. Matsakis 'Recognition of Handwritten Mathematical Expressions' massachusetts institute of technology may 1999.

[10] Vinod Nair, Geoffrey E. Hinton" Rectified Linear Units Improve Restricted Boltzmann Machines"2010.

[11] Ankita Kawade1, Prof. Vrushali Dhanokar2 'HANDWRITTEN EQUATION SOLVER USING CNN' International Research Journal of Engineering and Technology (IRJET) 2021.

[12] Jonathan J. HULL, Alan COMMIKE and Tin-Kam HO'Multiple Algorithms for Handwritten Character Recognition' State University of New York at Buffalo 226 Bell Hall Buffalo, New York 14260

[13] Paul Gader, Magdi Mohamed and Jung-Hsien Chiang 'Segmentaion-Based Handwritten Word Recognition' Department of Electrical and Computer Engineering.

[14] Alexander Koryagin, Roman Khudorozkov, Sergey Tsimfer ‘PYDENS: A PYTHON-FRAMEWORK FOR SOLVING DIFFERENTIAL EQUATIONS WITH NEURAL NETWORKS’ 2019