An artificial intelligence approach for the estimation of conduction heat transfer using deep neural networks

Mohammad Edalatifar
Laboratory on Convective Heat and Mass Transfer, Tomsk State University, Tomsk, Russia

Jana Shafi
Department of Computer Engineering and Information, College of Engineering in Wadi Alddawasir, Prince Sattam Bin Abdulaziz University, Wadi Alddawasir, Saudi Arabia

Majdi Khalid
Department of Computer Science and Artificial Intelligence, College of Computing, Umm Al-Qura University, Makkah, Saudi Arabia

Manuel Baro
Tecnológico Nacional de México Campus Nuevo Casas Grandes, Nuevo Casas Grandes, Mexico

Mikhail A. Sheremet
Department of Theoretical Mechanics, Tomsk State University, Tomsk, Russia, and

Mohammad Ghalambaz
Laboratory on Convective Heat and Mass Transfer, Tomsk State University, Tomsk, Russia

Received 10 November 2023
Revised 19 January 2024
18 April 2024
Accepted 11 May 2024

Abstract

Purpose – This study aims to use deep neural networks (DNNs) to learn the conduction heat transfer physics and estimate temperature distribution images in a physical domain without using any physical model or mathematical governing equation.

Design/methodology/approach – Two novel DNNs capable of learning the conduction heat transfer physics were defined. The first DNN (U-Net autoencoder residual network [UARN]) was designed to extract local and global features simultaneously. In the second DNN, a conditional generative adversarial network (CGAN) was used to enhance the accuracy of UARN, which is referred to as CGUARN. Then, novel loss functions, introduced based on outlier errors, were used to train the DNNs.

This research of Mohammad Ghalambaz and Mikhail Sheremet was supported by the Tomsk State University Development Programme (Priority-2030). This study was supported via funding from Prince Sattam bin Abdulaziz University under Project no. PSAU/2024/R/1445.

International Journal of Numerical Methods for Heat & Fluid Flow
Vol. 34 No. 8, 2024
pp. 3107-3130

0961-5539
DOI 10.1108/HFF-11-2023-0678

Findings – A UARN neural network could learn the physics of heat transfer. Within a few epochs, it reached mean and outlier errors that other DNNs could never reach after many epochs. The composite outlier-mean error as a loss function showed excellent performance in training DNNs for physical images. A UARN could excellently capture local and global features of conduction heat transfer, whereas the composite error could accurately guide DNN to extract high-level information by estimating temperature distribution images.

Originality/value – This study offers a unique approach to estimating physical information, moving from traditional mathematical and physical models to machine learning approaches. Developing novel DNNs and loss functions has shown promising results, opening up new avenues in heat transfer physics and potentially other fields.

Keywords Heat transfer, Deep convolutional neural networks, Information base learning, Physical phenome

Paper type Research paper

1. Introduction

Nowadays, there is a high capability of storing and processing raw data such as images, audio and video. However, extracting valuable information from a large data set of raw data is challenging, demanding automated approaches and robust processing means. Artificial intelligence (AI) has been one of the prominent inventions in automating the data process, extracting features and interpreting meaningful information.

State-of-the-art and prospects are well discussed (Zhang and Lu, 2021). At first, the applications of AI were limited to simple simulations until 2007, when Bengio *et al.* (2007) could train the first deep neural network (DNN). From then on, DNNs became the first candidate for classification, estimation and modeling problems. DNNs have found crucial applications in information science. Image processing (Minaee *et al.*, 2021), classification (Huertas-Tato *et al.*, 2022), neural language processing (Amirian *et al.*, 2021), digital health care (Nan *et al.*, 2022), text-to-speech (Yasuda *et al.*, 2021) and depression detection (He *et al.*, 2022) are some successful samples of DNN's applications in extracting practical and high-level information from raw data.

Physical phenomena such as heat transfer and mass transfer can be simulated by solving complex partial differential equations (PDEs). The solution results could provide image data about temperature distribution, fluid motion or mass distribution. Such images have numerous applications in engineering designs and understanding the heat and mass transfer behavior during the operation of devices. However, generally, solving a complex PDE requires expensive numerical computations. A feasible approach to reduce the computational time of numerical computations is using parallel computing. This is while the numerical solution of a PDE is an inherently weak parallel process and generally involves several serial computational stages.

Another difficulty in simulating heat and mass transfer phenome is the necessity for a PDE representing the phenome to be available. Such PDEs could be obtained by using the conservation of mass, energy and momentum laws combined with some closure models representing the behavior of materials. Then, the PDE, representing the physics of a phenome, should be validated against experimental observations. Thus, finding a PDE for a new phenomenon is a tedious and complex process. Therefore, finding new approaches to avoid the requirement of available PDE could be a significant advantage.

In some recent literature works (Zhu *et al.*, 2021; Wang *et al.*, 2021; Li *et al.*, 2019; Kim and Lee, 2020b; Lin *et al.*, 2019; Edalatifar *et al.*, 2020b, 2021), authors have shown that DNNs can learn physical phenomes from a data set and provide a solution to physical phenomes. DNNs are well compatible with parallel processing and can take full advantage of such computational resources. Therefore, applying DNNs to physical phenome could be an

alternative approach to solving PDEs (Raissi *et al.*, 2020) and taking good advantage of GPUs. Some recent research studies (Zhu *et al.*, 2022; Wang *et al.*, 2021; Li *et al.*, 2019; Lin *et al.*, 2019) showed that the heat distribution in a physical domain could be estimated using a DNN instead of the conventional numerical methods.

Löhner *et al.* (2021) thoroughly compared interpolation algorithms and DNNs in addressing inverse transfer problems with linear and nonlinear characteristics. The researchers perform multiple runs for a standard test problem, generating learning sets for both interpolation algorithms and DNNs. Additional runs are conducted to assess the predictive accuracy of each approach. The results reveal that interpolation algorithms surpass DNNs in accurately solving linear heat conduction problems, whereas DNNs excel in handling nonlinear heat conduction problems. Regarding heat convection problems, both methods yield comparable levels of accuracy. Notably, this study represents the first comprehensive endeavor of its kind.

Kim and Moon (2022) introduced a DNN to estimate the effective thermal conductivity of flat heat pipes with spreading thermal resistance. To accomplish this, they perform an extensive set of computational fluid dynamics (CFD) simulations, comprising 2,160 cases, to establish a large data set and predict a broader range of effective thermal conductivity, extending up to 10,000 W/mK. The DNN is trained and subsequently validated through reinforcement learning with an additional 8,640 CFD cases. The results demonstrate strong agreement between the DNN estimation and experimental, simulational and theoretical approaches. Furthermore, the DNN method proves to be more efficient in terms of time compared to traditional CFD simulations. The study suggests that DNN regression and estimation offer a promising avenue for securing data and making predictions across a wide range without requiring extensive experiments or simulations. These advancements in thermal and materials engineering can potentially address various challenges related to thermal obstacles encountered by researchers and practitioners.

In Edalatifar *et al.* (2020b, 2021), the authors generated a data set of temperature distribution images. Then, they designed a DNN with a convolutional structure and trained the DNN with their data set. The results showed that the convolutional DNNs could learn the physics of heat transfer. These studies adopted the mean squared error as the loss function. Although the outcomes of these works demonstrated the capability of DNN in estimating heat transfer, it is not clear if using a novel DNN or another type of loss function could further improve the ability of DNN in estimating images of temperature distribution. Unfortunately, the data sets of the previous research are not available, so a comparison between the results of the earlier works is not feasible.

In two recent publications, authors investigated the conduction heat transfer and prediction of temperature distribution in arbitrary shape geometries (Edalatifar *et al.*, 2020b) and tried to improve the loss functions (Edalatifar *et al.*, 2021). The input data was the geometry and thermal boundary conditions, whereas the temperature distribution images in a given geometry were the output data. The data set of these studies was published on the Mendeley database (Edalatifar *et al.*, 2020a), so comparison with these works is possible. The results of these studies revealed that the estimation of physical images is generally different than natural images. Physical images are those images whose pixel values are equal to a physical parameter such as temperature. These authors showed that few pixels of physical images could be estimated using a DNN with unacceptable error. They called pixels (data), which have a high estimation error, the outlier pixels. The outlier pixels are not crucial in natural images as human eyes may be unable to capture them. However, the outlier pixels in physical images could influence simulations and engineering calculation processes using the estimated images.

Edalatifar *et al.* (2021, 2020b) reported that the loss function could critically influence the accuracy and robustness of a DNN during the training process. Hence, they defined some new loss functions to reduce outlier errors during the training of DNNs. Using the new loss functions, they reduced the average and outlier estimation errors significantly. Here, there is an open question: How significantly do these new loss functions impact the estimation errors of deep networks, especially when applied to estimating physical images and heat distribution such as the deep network represented by (Sharma *et al.*, 2018) and spatially (Farimani *et al.*, 2017)?

Moreover, the structure of a DNN could affect its performance and, consequently, the accuracy of the estimated images. Considering the conduction heat transfer in a physical domain, the temperature of a point (pixel) is related to not only the temperature of its neighbor points (pixels) but also the shape of the domain and boundary conditions. Hence, the temperature of each point is dependent on local and global features. The local features are essential due to the temperature dependency of a pixel on the neighbor pixels. At the same time, the global features are important because of the influence of the shape and size of the physical domain on each pixel in the domain.

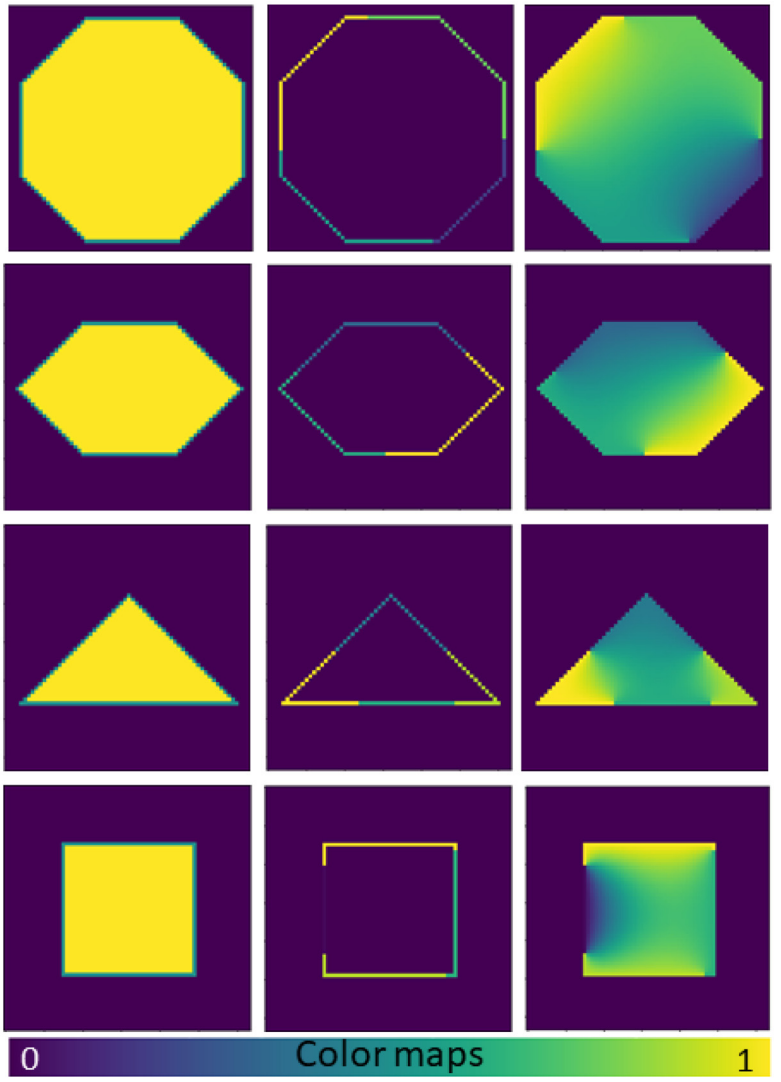
The previous works (Zhu *et al.*, 2021; Li *et al.*, 2019; Wang *et al.*, 2021; Kim and Lee, 2020b; Edalatifar *et al.*, 2020b, 2021; Lin *et al.*, 2019) constructed their DNNs using traditional convolutional layers. Unfortunately, convolutional layers only extract local features. Therefore, these convolutional neural networks (CNNs) generate and classify data using local features. The extraction of local features is adequate for processing natural images because they have almost no global features. As discussed, in addition to local features, global features could help DNNs better learn the physics of heat transfer. Hence, we believe a DNN capable of simultaneously extracting local and global features could better predict the temperature images.

As seen, the literature works used the DNN and loss functions tailored for the process and estimation of natural images. However, the process and estimation of physical images require DNNs and loss functions are adjusted to meet the demands of these images. Thus, the present study aims to introduce novel DNNs that can extract local and global features and benefit from new loss functions to improve the training process and diminish outlier errors.

2. Usage data set

Until the publishing of this paper, the data set represented in Edalatifar *et al.* (2020a) is a comprehensive heat transfer data set that is freely accessible and can be downloaded from https://data.mendeley.com/datasets/rw9yk3c559/2. Edalatifar *et al.* (2020b, 2021) used this data set to demonstrate the ability of DNNs to learn the physics of heat transfer as well as represent new methods to reduce error estimation. It involves 44,160 samples; each sample contains two input images and one 64×64 output image. Some isothermal boundary conditions and a domain geometry represent the two input images, whereas the domain's temperature distribution represents the output images. For each sample, the first input image demonstrates the geometry of the domain and the second input image represents the imposed isothermal boundary conditions. The output image consists of the temperature distribution. The data set geometries are regular octagonal, hexagonal, triangular and square. Figure 1 illustrates four samples of this data set.

There are various numerical methods that can be used to solve the heat PDE. One of these methods is the finite volume method (FVM) (Bergman *et al.*, 2011), which was used by Edalatifar *et al.* (2021, 2022, 2020b) to calculate the temperature distribution of images in the data set with the residual accuracy of 10^{-6}. They used the following equation, subject to



Notes: Left to right: Input 1, Input 2, Output
Source: Figure is original and from authors

Figure 1.
Four random samples of usage data set

$T = Ti$ for each piece of the boundary of geometry, where i is the piece number and T is the temperature of the ith piece:

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = 0 \tag{1}$$

The temperature distribution and boundary conditions are normalized between zero and one. More details about this data set are provided in Edalatifar *et al.* (2020a).

3. Deep neural network structures

In Edalatifar *et al.* (2021), the authors used a CNN to teach heat transfer physics to a DNN. A view of the used CNN is illustrated in Figure 2. The general structure of this CNN is based on an autoencoder (Baldi, 2012), which is a common structure of generator neural networks. An autoencoder has two main parts: encoder and decoder. The encoder extracts features

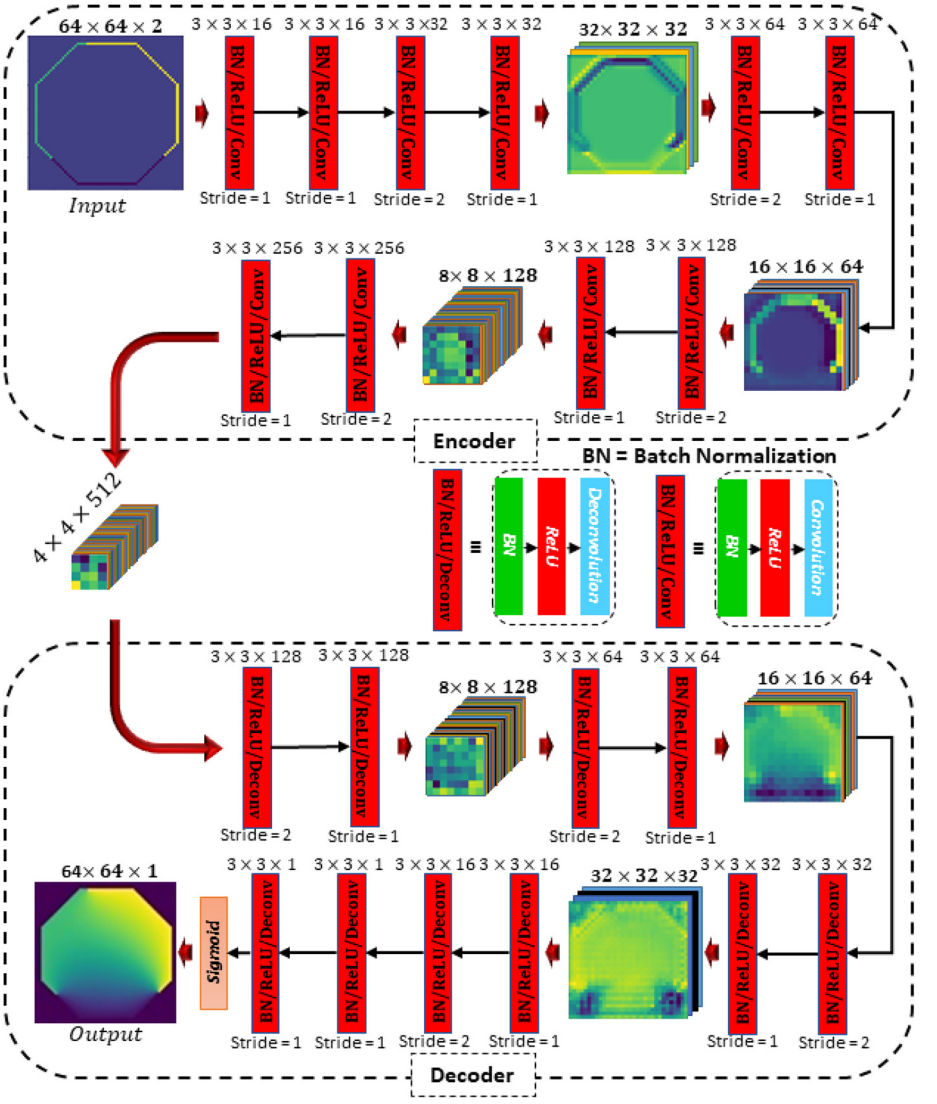


Figure 2.
Structure of CNN suggested by Edalatifar *et al.* (2021)

Source: Re-plotted with some modifications by current authors

from input data and the decoder gets the features and generates output data. The CNN in the encoder and decoder parts has 10 convolutional layers.

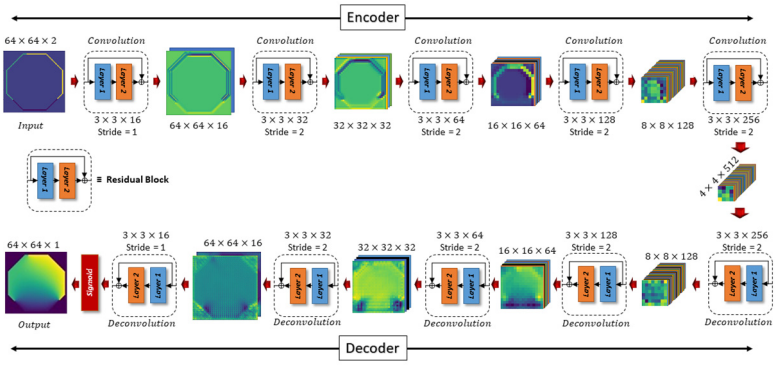
Moreover, batch normalization (Ioffe and Szegedy, 2015) and rectified linear unit (Agarap, 2018) were added before each layer. The details of the proposed CNN can be found in Edalatifar *et al.* (2021). In another study, Edalatifar *et al.* (2020b) introduced an autoencoder residual neural network (ARN) DNN, which was a combination of a CNN and a residual neural network (He *et al.*, 2016). Figure 3 depicts a view of the suggested ARN. Generally, ARN (Figure 3) shows a similar structure to CNN (Figure 2) but the main difference is the presence of residual blocks in Figure 3. A residual block, as shown in Figure 4, is a component of two layers with a feed-forward connection, skip connection, which moves input data of the first layer to an adder in the output of Layer 2. Details about the structure of the adopted residual blocks can be found in He *et al.* (2016). Comparing the structure of CNN and ARN reveals that they have almost equal trainable parameters; therefore, the difference between their results is only related to their structures.

In the next section, we will introduce two novel DNN structures that extract local and global features simultaneously. We expect they perform better than CNN and ARN in learning heat transfer physics. The performance of these novel DNNs will be demonstrated in the results section.

3.1 First new deep neural network: U-Net autoencoder residual network

As mentioned, the structure of ARN contains many residual blocks. The residual blocks not only avoid degradation phenomena (Monti *et al.*, 2018) but also increase the speed of convergence (He *et al.*, 2016) in DNNs. The degradation problem augments an estimation error instead of decreasing it when adding more layers. Edalatifar *et al.* (2020b) showed that an ARN could improve the temperature estimation capability compared to a CNN.

The convolutional layers and residual blocks extract only local features in a domain to learn the behavior of heat transfer. In natural images, features are predominantly local; hence, CNN and ARN are suitable DNNs for the process of natural images. Nevertheless, the temperature of a point is under the influence of the neighbor points and the distance between a point and its boundaries. Therefore, it seems a DNN needs not only local features (neighbor point information) but also global features (global information) to estimate temperature distribution more accurately.



Source: Re-plotted with some modifications by current authors

Figure 3.
Structure of ARN suggested by Edalatifar *et al.* (2020b)

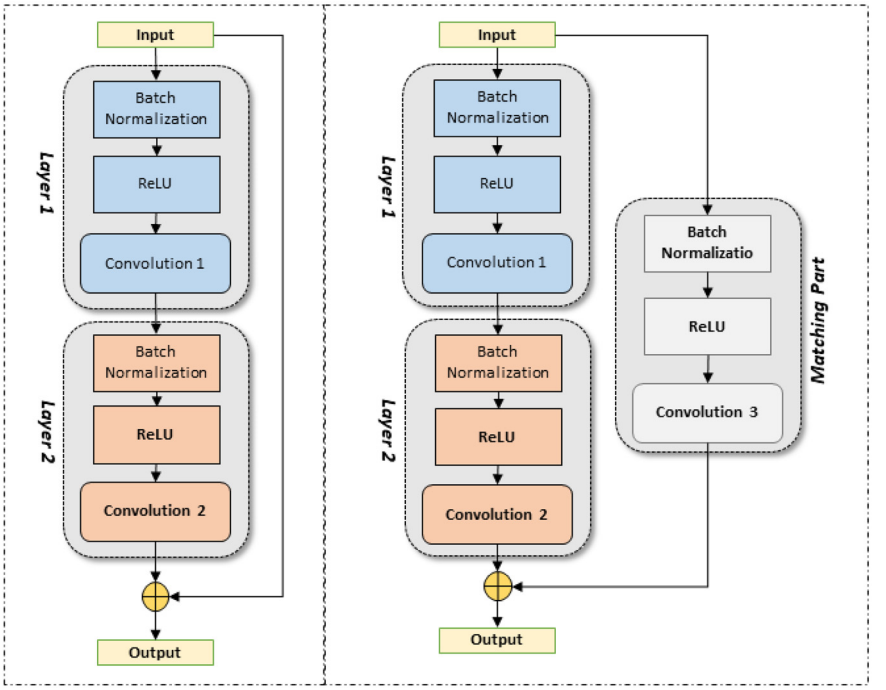


Figure 4.
Residual block structure

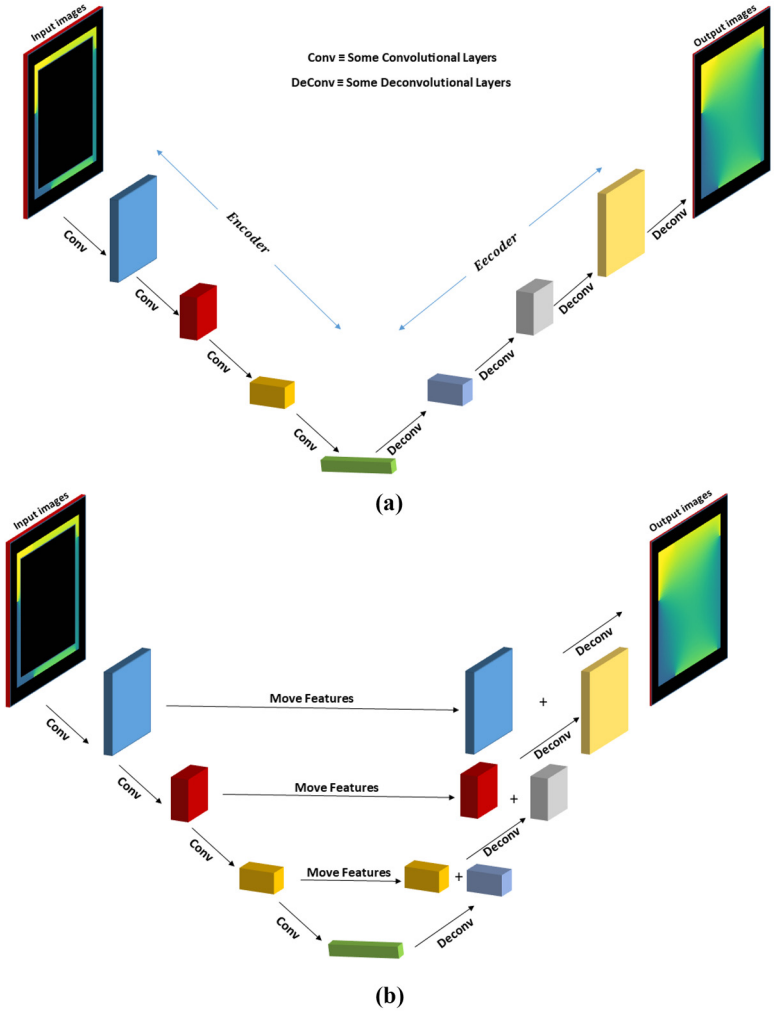
Notes: Input and output are (a) the same size; (b) different sizes
Source: Figure is original and from authors

U-Net is a DNN that was represented by Ronneberger *et al.* (2015) and has the ability to process input data globally and locally. A view of a U-Net with a symmetrical structure is depicted in Figure 5. As can be seen, U-Net has a very similar structure to that of an autoencoder but the main difference between U-Net and autoencoder is the decoder part. Unlike autoencoders, each layer of U-Net's decoder receives features from both the early layer of the decoder and the layer in the same encoder stage.

The simultaneous processing of local and global information by U-Net suggests an improved estimation of temperature distribution. Thus, in this study, the ART structure was modified from an autoencoder to U-Net, leading to the development of the new structure, UARN, as shown in Figure 6. UARN's architecture uses residual blocks instead of convolutional layers, offering enhanced accuracy and efficient training speeds compared to the ARN structure. As ARN and UARN share an equal number of layers and identical settings, their trainable parameters are nearly identical. This ensures that the accuracy of their estimations can be compared without the number of parameters significantly influencing the results.

3.2 Second new deep neural network: conditional generative adversarial network U-Net autoencoder residual network

A conditional generative adversarial network (CGAN) (Zhang *et al.*, 2019; Kim and Lee, 2020a) is a well-known deep structure that improves the training process and enhances the



Notes: (a) Autoencoder; (b) U-Net
Source: Figure is original and from authors

Figure 5.
Comparison of the structure of U-Net and autoencoder

accuracy of a deep generative network. It has two essential parts: the generator and the discriminator. The generator is a DNN that produces data and CGAN is used to enhance its accuracy. The discriminator supervises the accuracy of the produced data with the generator in the training process and navigates the training process to improve the accuracy of generated data. After the training process, the discriminator is useless; only the generator can create data. In the present study, CGAN is used to make a comparison between the influence of CGAN and the new loss function introduced by Edalatifar *et al.* (2021) called MSE+MMuMaSE on the UARN. So, according to the results, it will be decided which one is better: training UARN with MSE+MMuMaSE, CGAN or both of them.

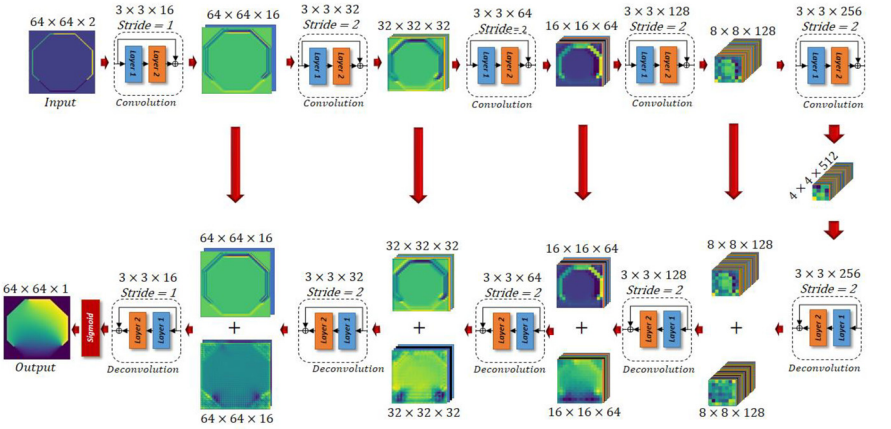


Figure 6.
Structure of the utsed UARN

Source: Figure is original and from authors

Our suggested CGAN is shown in Figure 7 and is referred to as CGUARN. As illustrated in Figure 7, the UARN is added for the generator part, and the discriminator, shown in Figure 8, is called ResNet2. Figure 7 denotes that the generator gets input data from the data set and estimates output data in the training process. Next, the discriminator receives estimated data (with generator), analyzes them and returns an error to enhance the accuracy of the generator (Error 2). After that, the optimizer gets two errors: an error of the generator estimation (Error1) and an error of the discriminator's analysis (Error 2). Finally, the optimizer modifies the trainable parameters of the generator (weights of the generator) to enhance the accuracy of the generator estimation.

ResNet2, which is used as a discriminator in CGUARN, is a classifier. It must detect whether the input image is a real or fake image. If the image belongs to the data set, it is real; otherwise, if it is made with the generator, it is fake. In the training process, the generator and discriminator are always in competition. The generator tries to become stronger and generate images with minimum error to mislead the discriminator and identify the generated images as

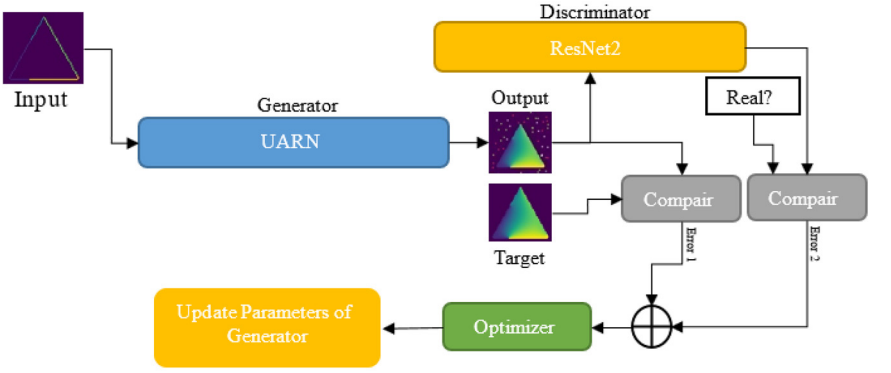


Figure 7.
Structure of the used CGUARN

Source: Figure is original and from authors

real. In contrast, the discriminator attempts to become stronger and correctly identify the fake images. This competition causes both the generator and discriminator to become stronger during the training process. As a result, the more powerful the detector, the more powerful the generator. The steps of the training process for CGAN are as follows:

- Select a sample of the data set and call it Sample1.
- Create a subsample data set and call it Sub_Sample_Real. Its input and output are the output image of Sample 1 and the "Real" label, respectively.
- Give input images of Sample1 to the generator and estimate output as Fake_Image.
- Create another subsample data set and call it Sub_Sample_Fake. Its input and output are Fake_Image and "Fake" labels, respectively.
- Train the discriminator with Sub_Sample_Real and Sub_Sample_Fake.
- Train CGAN (Figure 8) with Sample1 while the discriminator weights are fixed. Therefore, in this step, only the generator weights are adapted.

In the structure of ResNet2, the residual blocks are added. They improve the speed of convergence in the training process and increase accuracy. For more details, see Edalatifar *et al.* (2020b).

4. Loss functions

Introducing a proper loss function is one of the key parts of the DNN training and its outputs are a criterion for error estimation. The optimizer is responsible for tuning the DNN's weights in each step of an iterated process. It checks the estimation error using the loss function and modifies weights according to loss values. Therefore, loss function directly influences the accuracy of DNNs, and thus, a loss function must be defined carefully. Research by Edalatifar *et al.* (2021, 2020b) showed that when a DNN estimates an image, a small portion of pixels could be estimated with a substantial error, called an outlier error. The error of these pixels, i.e. outlier errors, could not be detected with eyes. Hence, they are not important for estimating natural images but estimating physical images affects later processes and design decisions that may use these images.

Edalatifar *et al.* (2021, 2020b) suggested four loss functions that could be used in the training process of DNNs. They showed the influence of the suggested loss functions on the training speed, mean estimation errors and outlier errors. The best of those loss functions was called MSE+MMuMaSE. It could reduce mean and outlier errors and speed up the

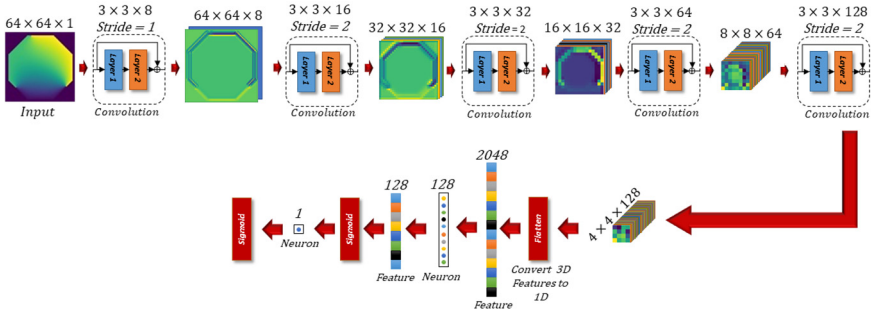


Figure 8.
Structure of the used ResNet2

Source: Figure is original and from authors

training process. In the current study, the mean square error (MSE), a popular loss function and MSE+MMuMaSE are used to train DNNs and their results will be compared.

MSE is defined as the mean of square errors (SEs). Suppose *Trg* is a 3D matrix that represents a collection of target images of a data set. Here, *Est* is a matrix with a size similar to *Trg*, estimated with a DNN. We expect *Est* to be as similar as *Trg* for a trained DNN. Hence, different criteria could be defined to calculate unsimilarity between *Est* and *Trg*. One of the common loss functions that have been used in literature for natural images is the MSE, which can be calculated as:

$$MSE = mean((Trg - Est) \odot (Trg - Est)) \tag{2}$$

in which, $\odot$ is the element-wise product. The value of a pixel on *Trg* and *Est* is accessible with $Trg_{n,i,j}$ and $Est_{n,i,j}$ that n is the image number index, as well as i and j denote the row and column index, respectively. MSE is the sole number because it is the average of all SEs. The SE of estimation for a pixel is calculated as:

$$SE_{n,r,c} = \left(Trg_{n,r,c} - Est_{n,r,c}\right)^2 \tag{3}$$

The SE for all pixels is computed using the matrix calculations as:

$$SE = (Trg - Est) \odot (Trg - Est) \tag{4}$$

As a result, equation (1) could be written as:

$$MSE = mean(SE) \tag{5}$$

MMuMaSE is the mean of M bigger SEs, where M is a finite number, so MMuMaSe can be considered as a maximum pixel error index. To calculate MMuMaSE, first, M biggest SEs must be determined and then the mean of them should be calculated. So, MMuMaSE for an image can be computed as follows:

- Calculate SE.
- Sort the SE.
- Select the M biggest elements of SE.
- Calculate the mean of M chosen SEs.

For a collection of estimated images, for instance, a batch of images, MMuMaSE is the mean of MMuMaSE of each image. MSE+MMuMaSE is the sum of MMuMaSE and MSE, and thus, it represents a composite outlier-mean error index. Therefore, to calculate MSE+MMuMASE, MSE and MMuMaSE are computed separately and added together. More details about these loss functions can be found in Edalatifar *et al.* (2021). Following Edalatifar *et al.* (2021), M is set to 82 for all computations of the current study.

5. Distinction between the present method and physics informed neural networks (PINNs)

It is important to note that the key difference between our current approach and the PINNs method developed by Raissi *et al.* (2019, 2020) is centered on the training strategy and how the loss function is defined. In the PINN method, the DNN is tailored to solve a particular problem with set parameters like a specific PDE, geometry and boundary conditions. This

specialization means that a DNN trained with PINNs may not fully grasp the broader implications of how changes in PDE coefficients, geometry and boundary conditions affect the solution. On the other hand, the present approach involves training the DNN on a wide range of solved problems, enabling the network to understand the links between various input problems and their solutions, thereby enhancing its ability to generalize.

When it comes to the loss function, PINNs use a two-component system: one for the PDE error and another for boundary conditions error, merged using weighted coefficients. Finding the perfect balance for these weights can be challenging, especially as the error scales for PDE and boundary conditions can vary greatly. In contrast, the present method simplifies this by not needing to separate the PDE and boundary conditions errors. The loss function is straightforwardly defined using solutions obtained from numerical methods. As a result, the DNN trained with the current approach is capable of generating solutions for new problems through generalization, independent of specific numerical methods. The main limitation of the current method is the necessity of having a data set comprising solutions from numerical methods to train the DNN.

6. Evaluation parameters

MSE is an appropriate parameter for exploring the average estimation error. However, here, we need a parameter to examine the outlier errors. Edalatifar *et al.* (2020b) introduced MaSE as an appropriate parameter to check outlier errors for the first time. It is defined as the mean of the maximum outlier error of each image in a group of images. The calculation of MMaSE is strict as follows:

- Determine the SEs of each image.
- For each image, select the maximum SE.
- Compute the average of selected SEs in the last step.

The comparison of MMuMaSE and MMaSE reveals that they are equal for $M = 1$. As the square is a nonlinear operator, it alters the actual error scale. Thus, an appropriate operator to have a good view of errors may be the absolute error (AE). Therefore, similar to SE, AE, which is a 3D matrix with a similar size as SE, is defined as:

$$AE = \left| Trg - Est \right| \tag{6}$$

and mean absolute error (MAE), similar to MSE, is calculated as:

$$MAE = mean\left(\left| Trg - Est \right| \right) = mean(AE) \tag{7}$$

Likewise, mean of maximum absolute error (MMaAE) is similar to MMaSE, but instead of using SEs, it uses absolute errors. These error functions were used to analyze the new DNNs introduced in previous sections.

7. Results and discussion

As mentioned, the number of trainable parameters of UARN, ARN and CNN are almost equal. The main difference is their structures, which is the main reason for the differences in accuracy. Three tests were performed in this section: the details are discussed in the next section. In the first test, CNN, ARN and UARN were trained with the MSE loss function. The results of this part aim to study the ability of UARN compared to CNN and ARN to learn the

physical phenomena. In the second test, UARN was trained with a CGAN structure called CGUARN. It was trained twice, once with MSE and once with the MSE+MMuMaSE loss function. The results of this test are very important and reveal information about how much MSE+MMuMaSE improves the accuracy of CGAN. In the third test, UARN and CGUARN are compared, and the best of them was selected as the ultimate DNN for estimating the temperature distribution images.

The conditions of training and comparison for the three tests are similar to each other and are detailed in Table 1. Moreover, the main settings of the training process are also illustrated in Table 1. Each DNN was trained for 2,000 epochs, and an epoch with less loss function for validation data, i.e. validation loss error, was selected as the best training point. DNN was saved in this epoch for later usage. The estimation error of training and validation data was preserved throughout the training process and later plotted in diagrams. Then, the estimated results were examined by the adopted mean of errors (MSE) and the outlier errors (MMuMaSE) parameters.

7.1 Test 1

In this test, CNN, ARN and UARN are trained with the MSE loss function, which is the most common loss function for training generator DNNs. The evaluation parameters for this test are represented in Table 2. As seen, the results of UARN are fantastic; the mean (MSE, MAE) and

No. of epochs	Tests 1 and 2: 2,000 Test 2: 1,000
Optimizer	Adam (Kingma and Ba, 2014)
Learning rate	0.001
$\beta 1$ of Adam	0.9
$\beta 2$ of Adam	0.999
Programing language	Python 3.7.2
Library of machine learning	Tensorflow
Tensorflow version	1.12.0
Batch size	32
Method of initializing weights	Glorot Uniform (Glorot and Bengio, 2010)
Loss function	MSE or MMuMaSE (based on test)
M (parameters of MMuMaSE)	82

Source: Table is original and from authors

Table 1. Main setting of training DNNs

Data type	DDN type	MSE	MMaSE	MAE	MMaAE	MaAE
Train	CNN	1.010e-5	0.00441	0.00115	0.0611	0.2091
	ARN	2.2509e-6	0.000582	0.00061	0.0219	0.972
	UARN	2.0598e-6	0.000374	0.000582	0.0170	0.1186
Validation	CNN	1.455e-5	0.00748	0.00133	0.07619	0.801
	ARN	5.450e-6	0.0022	0.000796	0.0326	0.9964
	UARN	2.462e-6	0.000415	0.00064	0.0174	0.1513
Test	CNN	1.438e-5	0.00736	0.00132	0.076	0.6224
	ARN	4.728e-6	0.00178	0.00079	0.0322	0.9953
	UARN	2.399e-6	0.000397	0.000633	0.0172	0.1866

Source: Table is original and from authors

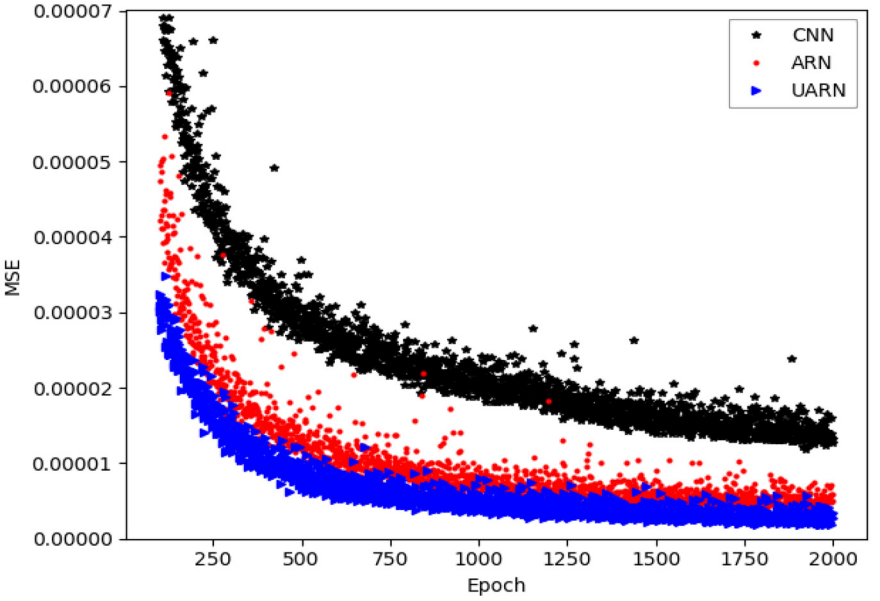
Table 2. Validation parameters for CNN, ARN and UARN were trained with the MSE loss function

outlier errors (MMaSE, MMaAE, MaAE) were reduced incredibly. As the sole difference between UARN and ARN is the U-Net structure of UARN, the decrease of UARN errors in comparison to ARN is related to the U-Net structure. Hence, it could be concluded that UARN, in addition to local features, could extract global features and better learn the physics of heat transfer. It should be noted that the estimation improvement of ARN in comparison to CNN depends on residual blocks because it is the only difference between them.

Figures 9 and 10 illustrate the variation of MSE and MMaSE during the training procedure, respectively. In these figures, some of the first epochs have been ignored for the sake of a graphical representation of data. Figures 9 and 10 show that a UARN network could decrease the mean and outlier errors faster than other DNNs. In addition, at the early epochs, MSE and MMaSE parameters for UARN are notably smaller than other examined neural network structures. Thus, it can be seen that UARN could provide very low MSE and MMaSE error parameters and keep its superiority during later epochs of the training process. Another interesting result is that CNN and ARN never could achieve the value of MSE and MMaSE of UARN at the final epoch, i.e. 500. It can be seen that UARN, during the first few epochs, could reach an accuracy that it seems CNN and ARN can never obtain. Finally, a comparison between Figures 9 and 10 show that the fluctuations (variance) of MSE and MMaSE for UARN are pretty small compared to CNN and ARN. This observation is more pronounced for MMaSE compared to MSE.

7.2 Test 2

In Test 1, the superior ability of UARN was demonstrated in learning heat transfer physics. Using UARN could dramatically reduce mean and outlier errors. Now, in this test, UARN is trained with CGAN structure. As mentioned, CGAN has been used in the literature (Mirza and Osindero, 2014; Al-Shargabi *et al.*, 2021) as a powerful structure for



Source: Figure is original and from authors

Figure 9. Variation of MSE during the training process for CNN, ARN and UARN when they train with MSE loss function

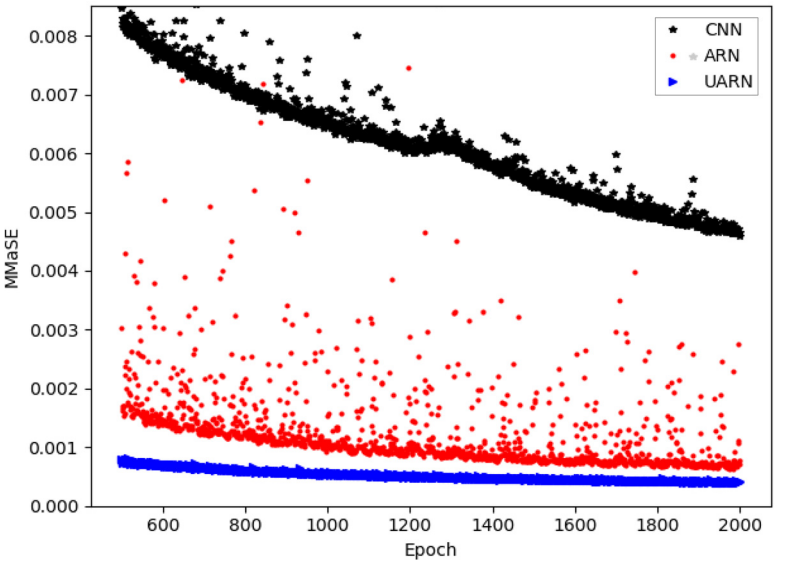


Figure 10. Variation of MMaSE during the training process for CNN, ARN and UARN when they train with MSE loss function

Source: Figure is original and from authors

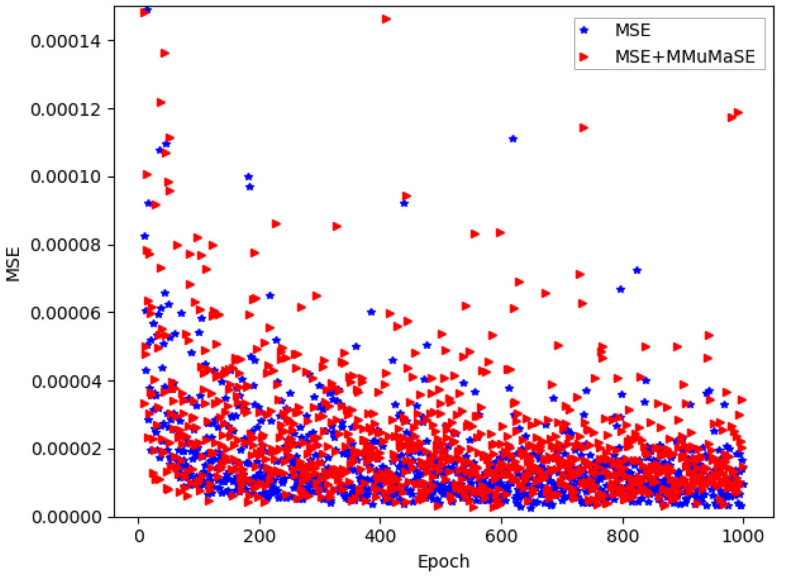
training a deep generator network. The new structure was discussed earlier as a CGUARN. Here, CGUARN was trained using the MSE loss function. Then, the same CGUARN was also trained using the MSE+MMuMaSE loss function for comparison. The results are represented in Table 3. Unlike other tests that are trained for 2000 epochs, Test 2 was specially trained for only 1,000 epochs as it was adequate for the purpose of the investigation of the current study. The variation of MSE and MMaSE for each epoch is plotted in Figures 11 and 12.

The results show that training the DNN using MSE+MMuMaSE and MSE as the loss functions has a minor influence on MSE (mean error) parameter. Using MSE+MMuMaSE as the loss function decreases not only the MSE parameter slightly but also the outlier errors (MMaSE, MMaAE, MaAE) effectively. As another result, in Figure 12, the variance (scattering) of outlier errors during the training process for MSE+MMuMaSE is less than MSE. In summary, Test 2 demonstrated the ability of MSE+MMuMaSE to reduce mean and outlier errors. While the CGAN, which is known as a powerful structure for training generator deep networks, could not outperform MSE+MMuMaSE loss function.

Table 3. Validation parameters for CGUARN, trained with MSE and MSE+MMuMaSE loss functions

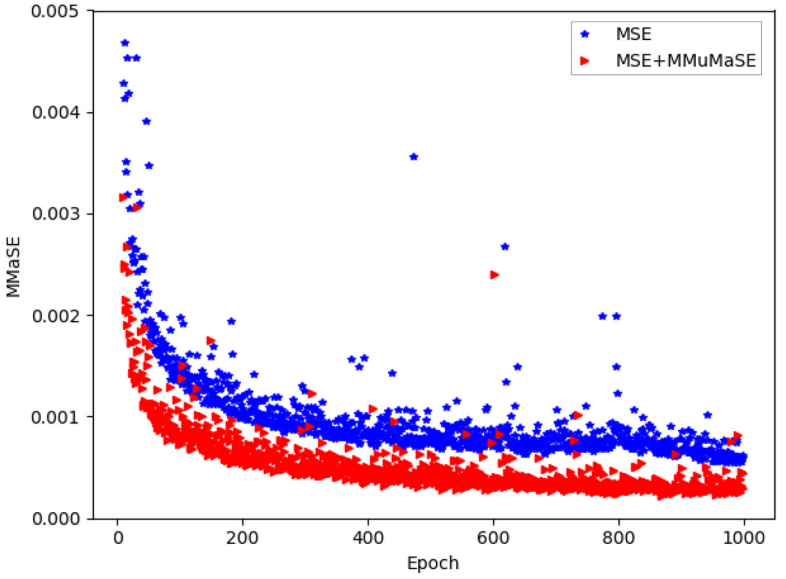
Data type	Loss function	MSE	MMaSE	MAE	MMaAE	MaAE
Train	MSE	3.238e-6	0.000556	0.00100	0.02096	0.1529
	MSE+MMuMaSE	2.7848e-6	0.000297	0.000839	0.01554	0.0904
Validation	CGUARN	3.8646e-6	0.0007197	0.00105	0.02128	0.9637
	MSE+MMuMaSE	3.027e-6	0.000318	0.000864	0.0158	0.09314
Test	CGUARN	3.614e-6	0.000569	0.00105	0.0211	0.1310
	MSE+MMuMaSE	3.0051e-6	0.000308	0.000863	0.0157	0.0741

Source: Table is original and from authors



Source: Figure is original and from authors

Figure 11. Variation of MSE during the training process for CGUARN when it trains with MSE and MSE+MMuMaSE loss functions



Source: Figure is original and from authors

Figure 12. Variation of MMaSE during the training process for CGUARN when it trains with MSE and MSE+MMuMaSE loss functions

7.3 Test 3

In the first test, the power of UARN to learn heat transfer physics was demonstrated. In the second test, it was shown that MSE+MMuMaSE could outperform CGUARN loss functions. In the present test, UARN, which was the best DNN structure, is trained with MSE and MSE+MMuMaSE loss functions. The aim is to see how the accuracy of UARN could be improved using the superior MSE+MMuMaSE loss function. Moreover, the results of training CGUARN with MSE+MMuMaSE were also added to the graphs for further comparison. In this test, DNNs were trained for 2,000 epochs. Table 4 represents the evaluation parameters and Figures 13 and 14 illustrate the variation of MSE and MMaSE during the training process.

The results show an impact of using MSE+MMuMaSE on average and outlier errors. The outlier errors are also significantly influenced by the MSE+MMuMaSE loss function.

Table 4. Validation parameters for UARN and CGUARN when they are training with MSE+MMuMaSE loss function

Data type	DDN type	Loss func.	MSE	MMaSE	MAE	MMaAE	MaAE
Train	CGUARN	MSE+MMuMaSE	2.6559e-6	0.00020	0.00078	0.01295	0.0769
	UARN	MSE+MMuMaSE	1.6440e-6	0.00015	0.00063	0.01098	0.0797
	UARN	MSE	2.0598e-6	0.000374	0.000582	0.0170	0.1186
Validation	CGUARN	MSE+MMuMaSE	2.887e-6	0.00023	0.00081	0.0133	0.099
	UARN	MSE+MMuMaSE	1.9657e-6	0.00022	0.00067	0.0117	0.365
	UARN	MSE	2.462e-6	0.000415	0.00064	0.0174	0.1513
Test	CGUARN	MSE+MMuMaSE	2.892e-6	0.00023	0.00080	0.0133	0.3189
	UARN	MSE+MMuMaSE	1.9189e-6	0.00020	0.000665	0.01158	0.322
	UARN	MSE	2.399e-6	0.000397	0.000633	0.0172	0.1866

Source: Table is original and from authors

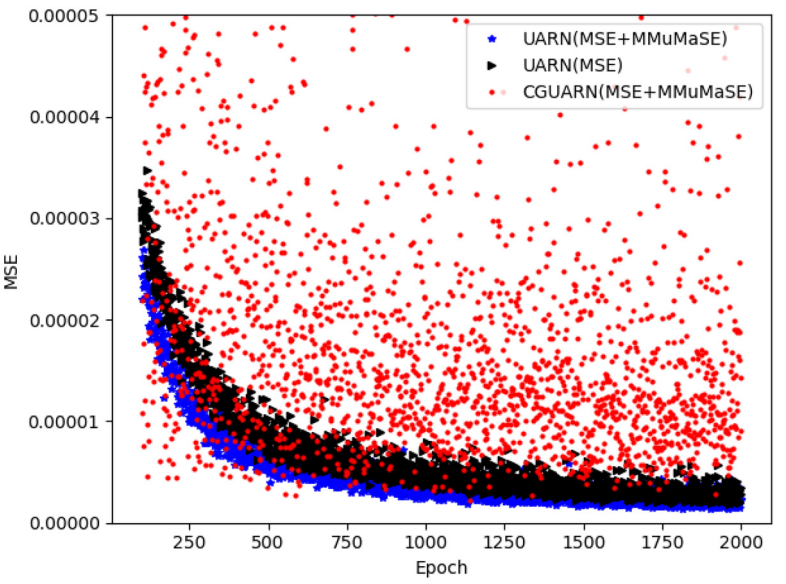
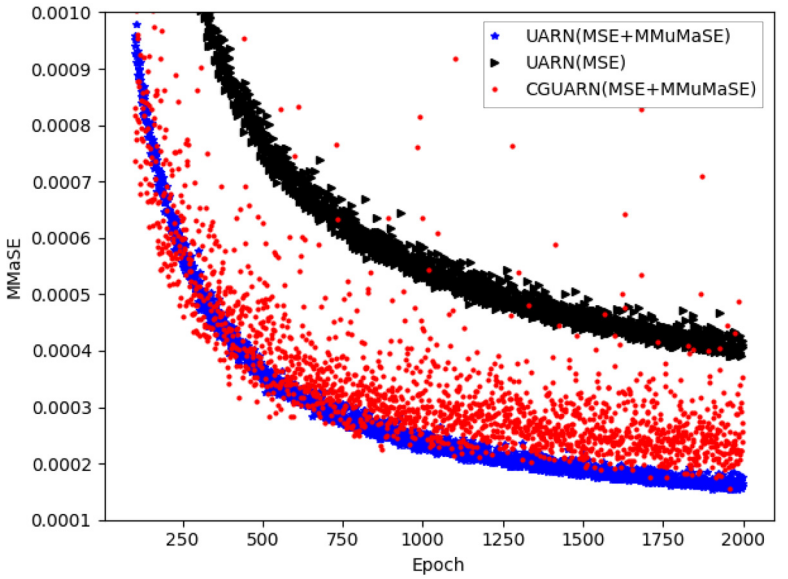


Figure 13. Variation of MSE during the training process for UARN and CGUARN when training with MSE and MSE+MMuMaSE loss functions

Source: Figure is original and from authors



Source: Figure is original and from authors

Figure 14. Variation of MMaSE during the training process for UARN and CGUARN when they are training with MSE and MSE+MMuMaSE loss functions

Using this loss function can notably reduce outlier errors. Interestingly, a combination of MSE+MMuMaSE and UARN could better estimate thermal images than CGUARN and MSE+MMuMaSE. Thus, the combination of UARN and MSE+MMuMaSE is the best means to simulate and estimate physical images.

Two more important points could be concluded from Figures 13 and 14 during the training process. First, CGUARN plots and UARN (trained with MSE+MMuMaSE) plots correspond to the most and least variances. The data for CGUARN are extremely scattered. As mentioned, we expect a uniform decrease in errors when the variances are small. Second, the MMaSE values for URAN(MSE) are notably larger than URAN (MSE+MMuMaSE) and CGUARN (MSE+MMuMaSE). Thus, it can be concluded that using MSE+MMuMaSE can significantly reduce the outlier errors, regardless of the adopted DNN structure.

Figure 15 shows some test data samples estimated with UARN when trained with MSE+MMuMaSE. The columns from left to right are input images to UARN, the target image of the data set and the estimated image with UARN. Under each row, the range of pixel values is written. The values MSE, MAE and MaAE for the estimated image are added on the right side of the rows.

7.4 Time of training and prediction process

An essential consideration when using numerical solutions in mathematics is the computational expense associated with these calculations. DNNs possess the remarkable ability to forecast data swiftly. Although the training phase in the current approach may be time-consuming, the approach's generalization capability ultimately yields a rapid and precise solution. To investigate this matter, Table 5 illustrates the time required for the training and prediction processes of the UARN. These times are calculated with a system

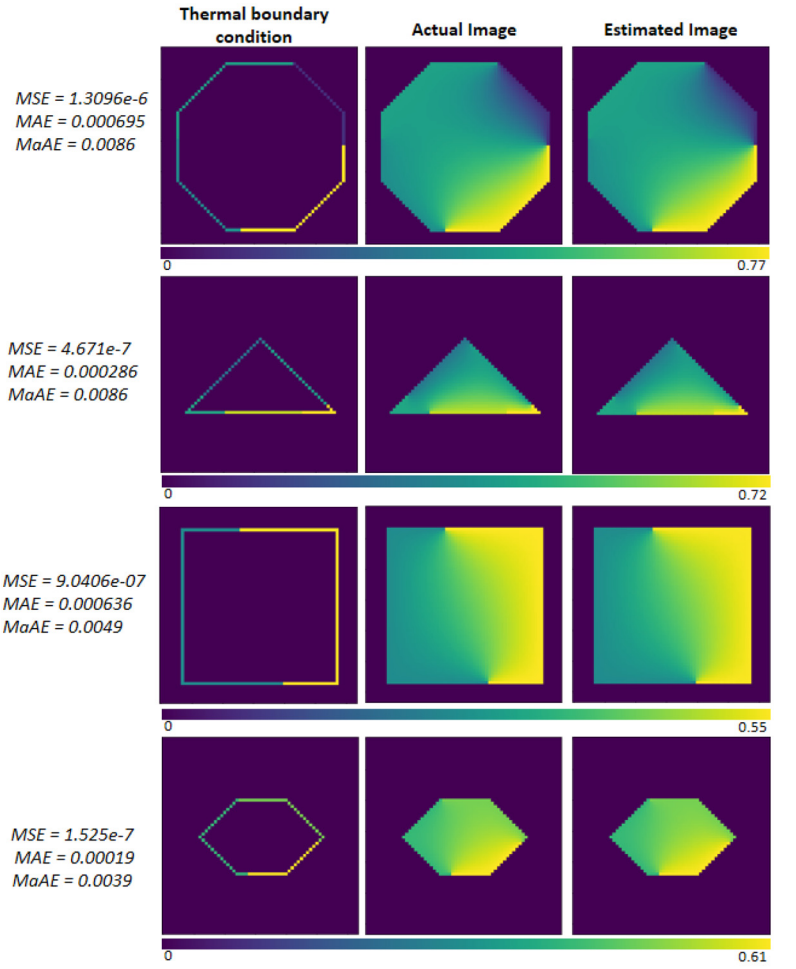


Figure 15.
Some samples of test data of the data set that are estimated with UARN after it was trained with MSE+MMuMaSE

Source: Figure is original and from authors

	Training time for one epoch (s)	Time of prediction of all of the testing data (s) Batch = 1	Batch = 32	Batch = 256
GPU				
Training data	52	−−−	−−−	−−−
Testing data	−−−	45	4	3
CPU				
Training data	421	−−−	−−−	−−−
Testing data	−−−−	123	24	21

Table 5.
Training and prediction process time with UARN while using GPU or CPU

Notes: The training process is done with MSE+MMuMaSE loss function
Source: Table is original and from authors

with an AMD Ryzen 7 5800H CPU, 16 GB RAM and an NVIDIA GeForce GTX 1650 graphic card.

A DNN is conventionally trained with training data and evaluated with a collection of data that is never seen in the training process, called testing data. Validation data is sometimes used in the training process to validate training during the process and stop the process before being trapped in overfitting. Therefore, in Table 3, the time of training is presented only for training data and the prediction time is only for testing data.

The time of the training process in Table 3 is for an epoch, which means the time of one optimization UART with all training data in the form of batches (groups) of 32 images. The prediction time is also presented to predict all testing data, which consists of 6,624 output images. Predictions can be made simultaneously for a single image or a group of images called batches. For more comparison, testing data are predicted three times with different batch sizes.

8. Conclusions

Extracting meaningful information from complex data is always a challenging task. Heat transfer physical images contain a complex pattern of temperature distribution. Thus, for DNN to learn heat transfer physics, it demands a well-designed structure compatible with the physical nature of the phenomenon and a robust loss function to guide adequately. The present study focused on designing a DNN structure capable of capturing local and global views of the temperature distribution and developing new loss functions to train the DNN better. The investigations of thermal images showed that, unlike natural images, the temperature of each point depends on neighbor points as well as the distance of the point to the heat bound and the characteristic of the bound. Therefore, a DNN called UARN was suggested in the present study. A UARN, in addition to having a high convergence speed, could simultaneously extract local and global features. As, unlike natural images, physical images contain local and global features, UARN had better estimation in comparison to DNNs without the ability to extract global features. Considering the loss functions handling the error estimation of physical images, two loss functions, MSE and MSE+MMuMaSE, were introduced. Moreover, a CGAN structure, which is a famous method to train generator deep networks, was also introduced. The combined structure of UARN and CGAN was called CGUARN. In this structure, CGAN was performing as an intelligence loss function. The impact of using different DNN structures and loss functions on the accuracy and performance of DNNs in learning heat transfer physics and predicting temperature images was examined in three cases. The key findings of the current research can be listed as follows:

- In Test 1, UARN was trained with the MSE loss function. It showed that UARN has a stronger ability to learn the heat transfer physics phenomena in comparison to similar DNNs. Moreover, UARN could rapidly reduce mean and outlier errors. In about epoch 500 of the training process, it reached a level of mean and outlier errors that CNN and ARN could never achieve. The fluctuations of outlier errors during the UARN training process were tiny and error reduction was relatively stable.
- In Test 2, CGUARN was trained using MSE and MSE+MMuMaSE as loss functions. It was revealed that MSE+MMuMaSE could slightly reduce MSE errors and significantly reduce the outlier errors. Thus, using MSE+MMuMaSE not only did not reduce the performance of CGUARN in dealing with MSE errors but also increased its performance in dealing with the outlier errors.

- In Test 3, UARN was trained with MSE and MSE+MMuMaSE. The results showed that using MSE+MMuMaSE could reduce MSE slightly and significantly reduce the outlier errors. Thus, MSE+MMuMaSE leads to better results compared to MSE. Using MSE+MMuMaSE for training UARN could outperform the powerful CGUARN (tainted with MSE+MMuMaSE).
- The UARN was able to predict solutions for all 6,624 test data cases within a matter of seconds. This highlights that the current approach cannot only generalize solutions to test cases that were never encountered during the training process but also can swiftly predict solutions in a matter of seconds.

References

Agarap, A.F. (2018), "Deep learning using rectified linear units (relu)", arXiv preprint arXiv:1803.08375.

Al-Shargabi, A.A., Alshobaili, J.F., Alabdulatif, A. and Alrobah, N. (2021), "COVID-CGAN: efficient deep learning approach for COVID-19 detection based on CXR images using conditional GANs", *Applied Sciences*, Vol. 11 No. 16, p. 11.

Amirian, S., Rasheed, K., Taha, T.R. and Arabnia, H.R. (2021), "Automatic generation of descriptive titles for video clips using deep learning", arXiv preprint arXiv:2104.03337.

Baldi, P. (2012), "Autoencoders, unsupervised learning, and deep architectures", Proceedings of ICML workshop on unsupervised and transfer learning, pp. 37-49.

Bengio, Y., Lamblin, P., Popovici, D. and Larochelle, H. (2007), "Greedy layer-wise training of deep networks", *Advances in Neural Information Processing Systems*, pp. 153-160.

Bergman, T.L., Incropera, F.P., Lavine, A.S. and Dewitt, D.P. (2011), *Introduction to Heat Transfer*, John Wiley and Sons.

Edalatifar, M., Ghalambaz, M., Tavakoli, M.B. and Setoudeh, F. (2021), "New approaches to improve deep learning estimation of heat transfer", *Neural Computing and Applications (NCAA)*, Vol. 34 No. 18.

Edalatifar, M., Ghalambaz, M., Tavakoli, M.B. and Setoudeh, F. (2022), "New loss functions to improve deep learning estimation of heat transfer", *Neural Computing and Applications*, Vol. 34 No. 18.

Edalatifar, M., Tavakoli, M.B., Ghalambaz, M. and Setoudeh, F. (2020a), "A dataset for conduction heat transfer and deep learning", *Mendeley Data*, Vol. v1.

Edalatifar, M., Tavakoli, M.B., Ghalambaz, M. and Setoudeh, F. (2020b), "Using deep learning to learn physics of conduction heat transfer", *Journal of Thermal Analysis and Calorimetry*, Vol. 146 No. 3.

Farimani, A.B., Gomes, J. and Pande, V.S. (2017), "Deep learning the physics of transport phenomena", arXiv preprint arXiv:1709.02432.

Glorot, X. and Bengio, Y. (2010), "Understanding the difficulty of training deep feedforward neural networks", Proceedings of the thirteenth international conference on artificial intelligence and statistics, pp. 249-256.

He, K., Zhang, X., Ren, S. and Sun, J. (2016), "Deep residual learning for image recognition", Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770-778.

He, L., Niu, M., Tiwari, P., Marttinen, P., Su, R., Jiang, J., Guo, C., Wang, H., Ding, S. and Wang, Z. (2022), "Deep learning for depression recognition with audiovisual cues: a review", *Information Fusion*, Vol. 80, pp. 56-86.

Huertas-Tato, J., Martín, A., Fierrez, J. and Camacho, D. (2022), "Fusing CNNs and statistical indicators to improve image classification", *Information Fusion*, Vol. 79, pp. 174-187.

Ioffe, S. and Szegedy, C. (2015), "Batch normalization: accelerating deep network training by reducing internal covariate shift", arXiv preprint arXiv:1502.03167.

Kim, H.-J. and Lee, D. (2020a), "Image denoising with conditional generative adversarial networks (CGAN) in low dose chest images", *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, Vol. 954, p. 161914.

Kim, J. and Lee, C. (2020b), "Prediction of turbulent heat transfer using convolutional neural networks", *Journal of Fluid Mechanics*, Vol. 882, p. A18.

Kim, M. and Moon, J.H. (2022), "Deep neural network prediction for effective thermal conductivity and spreading thermal resistance for flat heat pipe", *International Journal of Numerical Methods for Heat and Fluid Flow*, Vol. 33 No. 2, pp. 437-455.

Kingma, D.P. and Ba, J. (2014), "Adam: a method for stochastic optimization", arXiv preprint arXiv:1412.6980.

Li, Y., Wang, H. and Deng, X. (2019), "Image-based reconstruction for a 3D-PFHS heat transfer problem by ReConNN", *International Journal of Heat and Mass Transfer*, Vol. 134, pp. 656-667.

Lin, Q., Liu, Z. and Hong, J. (2019), "Method for directly and instantaneously predicting conductive heat transfer topologies by using supervised deep learning", *International Communications in Heat and Mass Transfer*, Vol. 109, p. 104368.

Löhner, R., Antil, H., Tamaddon-Jahromi, H., Chakshu, N.K. and Nithiarasu, P. (2021), "Deep learning or interpolation for inverse modelling of heat and fluid flow problems?" *International Journal of Numerical Methods for Heat and Fluid Flow*, Vol. 31 No. 9, pp. 3036-3046.

Minaee, S., Boykov, Y.Y., Porikli, F., Plaza, A.J., Kehtarnavaz, N. and Terzopoulos, D. (2021), "Image segmentation using deep learning: a survey", *IEEE Transactions on Pattern Analysis and Machine Intelligence,* pp. 1-1.

Mirza, M. and Osindero, S. (2014), "Conditional generative adversarial nets", arXiv preprint arXiv:1411.1784.

Monti, R.P., Tootoonian, S. and Cao, R. (2018), "Avoiding degradation in deep feed-forward networks by phasing out skip-connections", International Conference on Artificial Neural Networks, Springer, pp. 447-456.

Nan, Y., DEL Ser, J., Walsh, S., Schönlieb, C., Roberts, M., Selby, I., Howard, K., Owen, J., Neville, J. and Guiot, J. (2022), "Data harmonisation for information fusion in digital healthcare: a state-of-the-art systematic review, meta-analysis and future research directions", *Information Fusion*, Vol. 82.

Raissi, M., Perdikaris, P. and Karniadakis, G.E. (2019), "Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations", *Journal of Computational Physics*, Vol. 378, pp. 686-707.

Raissi, M., Yazdani, A. and Karniadakis, G.E. (2020), "Hidden fluid mechanics: learning velocity and pressure fields from flow visualizations", *Science*, Vol. 367 No. 6481, pp. 1026-1030.

Ronneberger, O., Fischer, P. and Brox, T. (2015), "U-net: convolutional networks for biomedical image segmentation", International Conference on Medical image computing and computer-assisted intervention, Springer, pp. 234-241.

Sharma, R., Farimani, A.B., Gomes, J., Eastman, P. and Pande, V. (2018), "Weakly-supervised deep learning of heat transport via physics informed loss", arXiv preprint arXiv:1807.11374.

Wang, Y., Zhou, J., Ren, Q., Li, Y. and Su, D. (2021), "3-D steady heat conduction solver via deep learning", *IEEE Journal on Multiscale and Multiphysics Computational Techniques*, Vol. 6, pp. 100-108.

Yasuda, Y., Wang, X. and Yamagishi, J. (2021), "Investigation of learning abilities on linguistic features in sequence-to-sequence text-to-speech synthesis", *Computer Speech and Language*, Vol. 67, p. 101183.

Zhang, C. and Lu, Y. (2021), "Study on artificial intelligence: the state of the art and future prospects", *Journal of Industrial Information Integration*, Vol. 23.

Zhang, H., Sindagi, V. and Patel, V.M. (2019), "Image de-raining using a conditional generative adversarial network", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 30 No. 11, pp. 3943-3956.

Zhu, F., Chen, J. and Han, Y. (2021), "A multiple regression convolutional neural network for estimating multi-parameters based on overall data in the inverse heat transfer problem", *Journal of Thermal Science and Engineering Applications*, Vol. 14 No. 5.

Zhu, F., Chen, J. and Han, Y. (2022), "A multiple regression convolutional neural network for estimating multi-parameters based on overall data in the inverse heat transfer problem", *Journal of Thermal Science and Engineering Applications*, Vol. 14 No. 5, p. 051003.

Corresponding author
Mohammad Ghalambaz can be contacted at: m.ghalambaz@gmail.com