Journal of Information & Optimization Sciences
ISSN 0252-2667 (Print), ISSN 2169-0103 (Online)
Vol. 44 (2023), No. 4, pp. 771–793
DOI : 10.47974/JIOS-1319

Optimized deterministic multikernel extreme learning machine for classification of COVID-19 chest Xray images

Arshi Husain *
Virendra P. Vishvakarma †
University School of Information, Communication and Technology (USICT)
Guru Gobind Singh Indraprastha University
Dwarka
Delhi
India

Abstract

In this paper, a novel technique has been proposed to exploit the capability of residual network (ResNet) deep learning model to extract the features. It is utilized neither in pretrained form nor as a transfer learning model. ResNet uses shortcut connections to create shortcut blocks in order to skip blocks of convolutional layers (residual blocks). These stacked residual blocks significantly increase training effectiveness and address the degradation issue. For the purpose of classification, a multiple kernel learning based deterministic extreme learning machine (MKD-ELM) which uses a linear combination of different base kernels as target kernel function is designed to classify chest Xray images. Multiple kernels are used here to exploit their non-linear mapping capability on heterogeneous data. MKD-ELM is an enhanced classifier, which does not require iterative training of its parameters. The proposed technique has better feature extraction along with non-iterative training, thus it is having very fast training and very good generalization performance. The kernel and regularization parameters that influence how accurate MKD-ELM is at classifying data, are tuned through experimentation. So, an optimization technique called the genetic algorithm (GA) has been utilized to determine the ideal combination of these parameters for improved performance. The performance of the proposed technique is analysed for COVID-19 detection problem using chest Xray (C_hXR) images by changing the training set, types of kernels and coefficients used for combining base kernels. The proposed algorithm achieves a 97.27% recognition rate on first dataset which comprises 5,856 images and 99.06% on the second dataset which consists of 13,808 images. A higher recognition rate is attained for these C_hXR image datasets, in respect to modern techniques demonstrating the effectiveness of the proposed algorithm.

Subject Classification: *68T10.*

Keywords: *ELM, KELM, MKL, Deterministic learning.*

* *E-mail:* **arshihusaincdac@gmail.com (Corresponding Author)**
† *E-mail:* **virendravishwa@rediffmail.com**

1. Introduction

Due to the growing need for rapid diagnosis, real-time COVID-19 detection utilising radiological images has increasingly become the focus. Although Polymerase Chain Reaction (PCR) test is the primary technique for detection of COVID-19 [1] , however, this test is time-consuming and complex. On the contrary, C_hXR images are publicly available and relatively quick to scan [2]. It has become necessary to design an accurate and real-time detector. Considering the non-iterative deterministic training nature of kernel extreme learning machine (KELM), MKD-ELM is proposed for classifying chest Xray images which uses a linear combination of various base kernels as the target kernel function. There have been several research studies undertaken in an effort to develop techniques for identifying pandemic-affected patients via Deep Learning (DL) [3], [4]. DL has key features that make it capable of performing a wide range of learning tasks [5]–[7]. Instances of successful methods for training DL are gradient descent [8], conjugate gradient [9], hessian-free Optimization algorithm [10], [11] and krylov subspace descent. The aforementioned techniques take time, at least during the training phase. An approach for Xray image detection that is effective during both training and testing phases is a challenging task. Based on feature fusion of dense convolution network and VGG Network, Lingzhi Kong et. al. proposed a technique for the detection of COVID-19. They added a machine attention block and category attention block for the extraction of deep features. In order to accomplish accurate classification, they have also used residual network to segment effective image information. The average accuracy achieved was 97.3% for three category classification [12]. Using CNN, A. Saraiva et. al. described categorization of pneumonia. They used SoftMax function which is in charge of making the probabilistic distribution of the input image into each of the categories for which the network was trained. Adaptive momentum was used for optimization and k-fold cross validation was used for evaluating the model [13]. Okeke Stephen et.al. proposed a CNN model trained from scratch to identify and categorize the pneumonia category from C_hXR images[14]. Kermany et. al. suggested a transfer learning-based framework to diagnose pneumonia using C_hXR images. Inception V3 architecture was adapted in their work in which retraining was done by initializing the convolution layers with loaded pretrained weights and the final SoftMax layer was retrained for the purpose of recognizing their classes from scratch. The accuracy achieved using C_hXR images was 92.8% [15]. Sivaramakrishnan Rajaraman et. al. presented a

customized VGG16 model to detect pneumonia using chest Xray images. They have customized the VGG16 model by truncating the deepest convolution layer and adding up a dense layer and global average pooling. A randomized grid search is used to find and improve the value of several hyperparameters, such as momentum, learning rate and L2-regularization in the customized VGG16 model. The overall accuracy achieved was 96.2% [16]. Prashant Sangulagi et. al. presents a system which seeks to enhance model integration and transfer learning by classifying C_hXR images into two categories- normal, and COVID-19. In their work, ResNet101 and ResNet152 were utilized which has been trained using the ImageNet dataset. Their system could accurately reproduce 96.1% of the classes of C_hXR images on the training dataset [17]. Siyuan Lu et. al. suggested to use image feature fusion to help with the diagnosis of COVID-19 in lung window computed tomography. ResNet18 and ResNet50 deep networks were chosen initially in their approach for the purpose of extracting image feature information. Secondly, they used discriminant correlation analysis to fuse the information extracted by these two networks in order to get the refined image features. Finally, they proposed to train three randomized neural networks to classify the refined fused features, which are: schmidt neural network, extreme learning machine (ELM) and random vector functional link network. The prediction of the three randomized neural networks were then combined to produce a more efficient classification performance. 97.14% was the highest accuracy achieved in their work [18]. Murugan et. al. proposed a CNN-ResNet50 base extreme learning classifier in which ResNet50 has been used for the purpose of feature extraction followed by ELM with a sigmoid function for classifying C_hXR images into normal, pneumonia and COVID-19. The presented model achieved 94.07% accuracy [19]. Dongsheng Ji et. al. used ResNet50, ResNet152, DensNet201, Xception, Inception ResNetV2 pretrained networks for the purpose of detecting COVID-19 patients. The accuracy achieved was 96% [20]. Muhammad Umair et. al. utilized pertained models for feature extraction from C_hXR images and create a feature map. Then, the Grad-Cam technique was employed in their work which produce a class specific heat map and helps to find out the COVID-19 detection transparency. This technique is implemented by using the layers and extracted features of the trained model and actually draws attention to the areas of the input image that the model concentrates on during the classification phase, indicating that the feature maps produced in the final convolution layer have the spatial information necessary to effectively capture the visual pattern. The accuracy achieved using their technique was 96.49% [21]. It

is essential to mention that conventional ELM suffers from uncertainty and poor conditioning which leads to proposing our novel non-iterative, MKD-ELM approach to produce an output solution that is deterministic and immutable. The remaining sections of this paper are structured as follows structured as follows. Section 2 reviews background materials. The proposed methodology and analysis are discussed in Section 3. Section 4 presents experimental results and finally, Section 5 concludes this paper.

2. Preliminaries

Residual Network

The main principle of residual networks, which are deep convolutional networks, is to use shortcut connections to skip entire blocks of convolutional layers. ResNet use shortcut connections to create shortcut blocks in order to skip blocks of convolutional layers (residual blocks). These stacked residual blocks significantly increase training effectiveness and address the degradation issue in major part [22]. The residual block represented by Figure 1 illustrates the skip connection. All algorithms train on the output 'Y' but, ResNet trains on F(X). In simpler terms, ResNet tries to make F(X) = 0 so that Y=X. Due to this skip link, the output is different. Input X is multiplied by the layer weights in the absence of the skip connection, and then a bias term is added. Therefore, the resulting output in the absence of the skip connection is as follows: F (w*x + b) (=F(X)). But with skip connection, the output obtained is: F(X) + x. But with skip connection, the output we get is: F(X) + x. The formulation of F(X) + x can be realized by feedforward neural networks with "shortcut connections" (Figure 1).

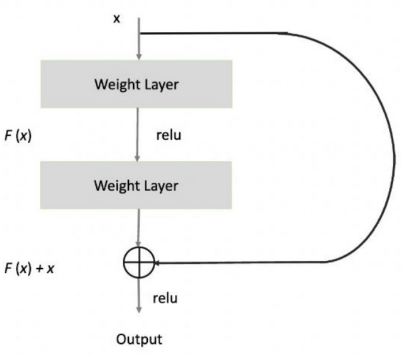

Figure 1

Residual Block

Extreme Learning Machine

ELM, developed by G. Huang in 2006 are feedforward neural networks and are based on the idea of inverse matrix approximation. In an ELM, Weights and biases are randomly assigned, unlike other iterative algorithms, and do not need to be learned iteratively making it faster with better generalization performance as compared to networks trained using the backpropagation approach [23], [24]. It is due to this randomness that ELMs have been shown to have universal approximation theorem powers with relatively small nodes in the hidden layer. ELM was introduced to minimize the training error, the computational cost, but because of its high space and time complexity and memory-residency, the traditional ELM is not able to train big data quickly and efficiently and it also raised the possibility of local minima issue and over-fitting. Kernel matrix with ELM can be used to overcome these issues, which was suggested in 2010 by Huang et. al. by using the kernel functions to ELM approach. Further improving the operation of ELM, KELM does not need activation function as a user's input. Afterward, the kernel function determines the output [25]. The abbreviations used in this explanation are as follows:

IP: Input
OP: Output
HN: Hidden Node
N: Neurons
V: Vector
HL: Hidden Layer
Mx: Matrix

The OP function f or ELM can be expressed as:

$$g_o = \sum_{j=1}^{\eta} \theta_j d_j(x_i) = \sum_{j=1}^{\eta} \theta_j d(iw_j . x_i + p_j) = d(x)\theta = F, i = 1, \ldots, M \quad (1)$$

where,

g_o : ELM OP

$iw_{j:}$ Synaptic weights linking i^{th} IP node to j^{th} HN

$p_{j:}$ bias for j^{th} node

θj : weights related to OP layer, linking to j^{th} HN to OP node

η: number of N in the HL

M: number of training observations

V Mx multiplication representation of Equation (1) is:

$$T\theta = F \tag{2}$$

where, T represents OP Mx corresponding to HL. It is expressed as follows:

$$T = \begin{pmatrix} d(x_1) \\ \vdots \\ d(x_M) \end{pmatrix} = \begin{pmatrix} d(iw_1, p_1, x_1) \cdots d(iw_\eta, p_\eta, x_1) \\ \vdots \\ d(iw_1, p_N, x_1) \cdots d(iw_\eta, p_\eta, x_N) \end{pmatrix}_{M \times \eta} \tag{3}$$

The q^{th} column of T is the OP V of q^{th} HN and q^{th} row of T is OP of HL similar to IP x_q

$x = [x1, \ldots x_M]^T$: IP V

$Y = [y1, \ldots y_M]^T$: Target OP V for M number of IP occurrences.

$\theta = [\theta 1, \theta 2, \ldots, \theta \eta]^T$: OP V for η number of hidden N and every element of θ is a V of dimension equivalent to OP nodes.

The IP weight Mx can be expressed in the following manner:

$$RW = \begin{pmatrix} iw_{11} iw_{12} \cdots iw_{1s} \\ \vdots \\ iw_{\eta,1} iw_{\eta,2} \cdots iw_{\eta,s} \end{pmatrix}_{\eta \times s} \tag{4}$$

For s dimensional IP data, the size of IP weight vector that was randomly assigned is $\eta \times s$. Since the number of concealed nodes is estimated dynamically rather than statically, RW has a dynamic size.

Least-square solution of equation (2) is:

$$\Theta = T^{\dagger}F \tag{5}$$

Where, T† represents oore-enrose inverse [26] OP Mx T of HL. The orthogonal rojection method has been utilized to get oore- enrose generic inverse of Mx T: $T\dagger = T^T(T\,T^T)^{-1}$.

In order to find the OP that is invariant and to achieve strong generalization performance solution, in the TT^T diagonal's solution, a positive number W is added [27].

$$g_o = d\theta = d(x)T^T\left(\frac{I}{W} + TT^T\right)^{-1} F \tag{6}$$

Where, W: regularization coefficient

d(x) : ELM feature mapping

The goal of ELM, in contrary to iterative learning algorithms, is to reduce both training error and OP weight. For classification-related tasks, the purpose of ELM is:

$$\text{Min}: \frac{1}{2}\| \theta \|^2 + \frac{W}{2}\sum_{k=1}^{M}\| \delta \|^2$$

$$\text{Such that}: \quad d(x)\theta = y_k^T - \delta_k^T \text{ for } k = 1,\ldots,M \tag{7}$$

Where, $\delta_k = [\delta_k, 1, \ldots, \delta_{k,u}]^T$ is a training error vector of u OP N corresponding to IP data x_k.

Multiple kernel learning (MKL)

It is well known that the efficacy of kernel-based classifiers depends on selecting the best kernel for a given set of data [28]–[30]. A lot of work has gone into selecting the optimal kernel for a particular application. When classifying data that is heterogeneous, a single kernel-based classifier does not perform well empirically [31], [32]. In order to solve this issue, multiple kernel learning is used which offers an elegant solution to such a problem by optimizing a data-dependent kernel. MKL's goal is to merge various feature sets by figuring out the best combination of various kernels [33]. Using MKL, features from many sources are concatenated and supplied to a single learning approach. The MKL formulation utilizing linear combination of [29], [34] set of μ pre-defined kernels is expressed as:

$$K(x_i, x_k) = \sum_{l=1}^{\mu} \gamma_l K_l(x_i, x_k) \tag{8}$$

With $\gamma_l > 0$ and $\Sigma_{l=1} = 1$ $K(x_i, x_k)$ is resultant optimal kernel. $K_l(x_i, x_k)$ denotes the l^{th} sub-kernel and γ_l is the weight of l_{th} kernel.

For each instance, each sub-kernel employs a unique set of feature vectors. To accomplish accurate classification, weights are provided to the various kernels according on the relevance of the features [33], [35]. The Eq. (8) can be equivalently expressed as:

$$K(\cdot,\cdot;\gamma) = \sum_{l=1}^{\mu} \gamma_l K_l(\cdot,\cdot) \tag{9}$$

The perceptivity of the classification problem determines the selection of kernel $K(\cdot,\cdot)$ and its combination coefficient. In terms of feature mapping, the MKL can be stated as follows:

$$\theta(\cdot;\gamma) = [\gamma_1\theta_1(\cdot), \gamma_2\theta_2(\cdot), \cdots, \gamma_l\theta_l(\cdot), \cdots \gamma_\mu\theta_\mu(\cdot)] \tag{10}$$

Where $\theta(\cdot;\gamma)$ and $\theta_1(\cdot)$ depicts the mapping of feature representing $K(x_i, x_k)$ and $K_1(x_i, x_k)$ respectively.

3. Proposed Methodology

In this paper, a novel technique has been proposed for COVID-19 detection using the C_hXR images. The block diagram of the architecture that has been proposed in this paper is presented in Figure 2. The images of the original dataset are preprocessed. First, *Minmax* has been utilized for the purpose of normalization of images. Then, in view of the fact that the required IP size varies for each CNN, we resize all the IP images to 224 × 224 × 3 pixels as needed by ResNet used in this study and compatible with the ImageNet. We use a 50-layer pretrained ResNet, for our first piece of model because training a deep CNN model from scratch takes a lot of time. The proposed novel technique exploits the capability of ResNet deep learning model to extract the features neither in pretrained form nor as a transfer learning model. ResNet50, has been used for the process of feature information extraction of COVID-19 C_hXR images and the options set for training the network are presented in Table 1. The *MiniBatchSize* is set 32 to make sure that the CNN and image data fit into GPU memory and the training time decreases significantly by setting the appropriate batch size. The activations OP is organized into columns which helps in speeding up the process of binary classification that follows. Split is set to randomize as it randomizes the distribution of the specified number of images from each label among the new datastores. There are three fully connected layers in the pretrained network, with fully connected for classification being the last layer. In our experiment, we are using the pretrained network only for extracting the features maps and not for classification because, for classification fine tuning is required. In fine tuning, weights of the network are updated iteratively using the back propagation algorithm which is a very time-consuming process. Therefore, we utilize the layers of pretrained network till the global average pooling layer only. After that, for classification purpose, a novel approach, MKD-ELM has been proposed which uses a linear combination of different base kernels as the target kernel function to classify C_hXR images. By employing multi kernel approach, the proposed method seeks to improve classification performance, takes much less time while overcoming the arbitrariness of the traditional ELM by using evaluation of IP weights and biases rather

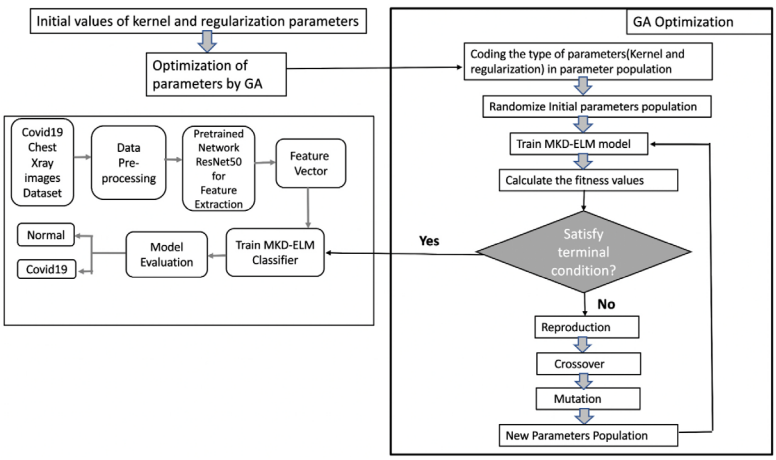

Figure 2

Block diagram of MKD-ELM based proposed methodology.

than random allocation. The kernel and structure parameters, which are tuned through experimentation, are what influence how accurate MKD-ELM is at classifying data. The Brute force method makes it challenging to evaluate the best selection of these parameters from a given set of values. In order to determine the ideal combination of these parameters for improved performance, an optimal algorithm is needed. The optimal regularization coefficient and kernel parameters have been balanced using genetic algorithm to get the highest recognition possible. Images from C_hXR images of COVID-19 cases and healthy people are included in the dataset. For the purposes of training and assessing the model, we set the label of healthy cases images to 1 and the label of COVID-19 images to 2. The options selected for MKD-ELM architecture are presented in Table 2. This non-iterative learning-based approach MKD-ELM speed up the detection of covid-19.

Table 1

Options set for the Network training

Property	Option selected
Batch Size	32
Split	Randomized
OP Type	Columns

Table 2
Options set in MKD-ELM classifier

Property	Option selected
Elm Type	Classification
Regularization coefficient	30000
Kernel type	RBF
Kernel parameter 1	750
Kernel parameter 2	1

Multi kernel deterministic extreme learning machine

A quick and deterministic learning algorithm is required by the real-world pattern classification application to categorize complex data. The ELM produces varying and non-deterministic OP solutions as a result of the random practice of structural parameters [21], [22]. By varying the number of HNs, various classification results for the same dataset can be achieved. The statistics of the IP and hidden must be evaluated with the aid of invariant parameters in order to achieve the immutable and deterministic OP solution. In this paper, we present a deterministic ELM fused with multi kernel technique. Simple and complex data can be classified using the proposed technique by selecting the most appropriate choices for kernels and combination coefficients. The IP and HL parameters in MKD-ELM are determined with the use of IP data samples in order to generate the deterministic OP. The feature vector in MKD-ELM that is used to map IP data to OP classes is a concatenation of feature vectors that were generated with the support of predefined sub-kernels [30], [31]. The IP and HL parameters in MKD-ELM are determined with the use of IP data samples in order to generate the deterministic OP. Due to the analytical evaluation of the structural parameters in the proposed methodology, the OP solution is stable. Formulation of MKD-ELM is:

The IP weight IW = [iw_1,…, iw_η] can be defined with aid of IP data x_i of dimension r:

The following is an overview of MKD-ELM:

$$\frac{1}{m}\left[\frac{x_1}{Z_1}, \frac{x_2}{Z_2}, \ldots, \frac{x_\eta}{Z_\eta}\right] \tag{11}$$

Where, Z is computed by analyzing the IP data's norm in the form shown below:

$$Z_i = \left[\sum_{j=1}^{r} (x_{ij})^2\right]^{1/2} \tag{12}$$

m is calculated by finding norm of all Zi:

$$m = \left[\sum_{i=1}^{\eta} (Z_i^2)\right]^{1/2} \tag{13}$$

$$= \frac{1}{m}\left[\frac{x_1}{Z_1}, \frac{x_2}{Z_2}, \ldots, \frac{x_N}{Z_N}\right] \tag{14}$$

The mean of the IP weight V of size N X r determines the bias of the k^{th} HN for r-dimensional IP data.

$$a_k = \frac{1}{r}\sum_{i=1}^{r} iw_{ki} \tag{15}$$

For a multi-class classification problem, the MKD-objective ELM's function is:

$$\text{Min}: \frac{1}{2}\| \theta^2 \| + \frac{W}{2}\sum_{i=1}^{M} \| \delta_i \|^2$$

$$\text{Such that,} \quad \theta^T d(x_i, iw_i : \gamma) = y_i - \delta_i, \forall i \sum_{l=1}^{\mu} \gamma_l = 1, \gamma_l \geq 0 \tag{16}$$

With respect to each of the kernels $1, 2, \ldots, \mu$, $T(.;\gamma) = [\gamma_1 T_1(.), \ldots, \gamma_u T_u(.)]$ is the feature vector.

Algorithm: Deterministic learning with multiple kernel learning

IP: A set of C_hXR image training samples (xk, yk), k=1, 2, .. N and specified r kernels $K_r(x_k, x_i)$

OP: $g_{0,}$ (an OP function) estimating class y.

1. Calculate the mean and norm of the IP data x_i to extract the IP weight V from equation 11.
2. By computing the mean of the IP V using equation 16, determine the biases of HNs.
3. Achieve the optimal kernel using $K(\cdot, \cdot; \gamma) = \sum_{l=1}^{\mu} \gamma_l K_l(\cdot, \cdot)$
4. By using the feature vectors obtained in step 3, calculate the final solution for MKD-ELM that is deterministic.

Genetic Algorithm

An optimization method known as the genetic algorithm (GA) is based on Darwin's idea of biological evolution which was introduced by Holland and Goldberg [36], [37]. Since the last four decades, it has been a well-known algorithm that has solved numerous inherent problems by applying parameter optimization. The foundation of this approach is the idea that initial populations are generated randomly. Chromosomes are the term used to characterize each individual in the problem. Crossover, mutation and selection are the operators based on which this algorithm is performed. The steps using which the parameter optimization is performed by GA to minimize the value of cost fitness function by is described as follows: The best chromosome is chosen from the randomly produced initial population in the first phase, which is the selection operator (if the initial population is not specified). The parent chromosomes used to create the new chromosomes for the following generation are those with the highest values for fitness. In the next step, which is crossover operator, the new child chromosomes created from two parent chromosomes are used to check the values of the fitness function. Then, comparisons are made between the fitness function values obtained from newly formed child chromosomes and those obtained from their parent chromosomes. This random chromosome combination is chosen so that new child chromosomes can have higher fitness function values than their parent chromosomes. The final phase in GA involves searching for the best fit among the new potential chromosomes using a mutation operator [38]. To accomplish this, some of the chromosomes' genes are changed at random. These three phases result in offspring chromosomes, which are subsequently treated as the parent in the next generation. The next iteration of determining the best fitness function value makes use of these new generations. Until the termination criteria are not met, these steps are repeated. Termination occurs when the maximum number of iterations has been reached, or the number of validated chromosomes reaches the population size, or the best fitness function value and the mean fitness function for each iteration are identical [39].

4. Experimental evaluation

The experiments performed in this study were carried out using Matlab2022a. The experimental results obtained at different split ratios on Dataset1 and Dataset2 using ResNet50, are presented in Table 3 and Table 4 respectively. The results using only KELM on dataset 1 and dataset 2 are shown in Table 5 and Table 6 respectively The results obtained by applying our proposed novel approach to dataset 1 are presented in Table 7 and

Table 8 for random and sequential split respectively. The results achieved by applying our proposed novel approach to dataset 2 are presented in Table 9 and Table 10 for random and sequential split respectively. With our proposed novel technique, we are able to reduce the testing error up to 0.94 with dataset 2 using 90% of the total images for training and the remaining for testing.

4.1 *Comparison of results using ResNet50, KELM and proposed novel technique*

We evaluated the recognition rate using ResNet50, KELM and our proposed novel technique at nine different split ratios. It can clearly be seen that among all the three approaches used, MKD-ELM stand out in terms of accuracy. This accuracy comparison is based on the results presented in Tables 3 to 10.

Table 3

Results using ResNet50 on dataset1

ResNet_only: Epoch=1, Batch Size=10		
Split Ratio (Training-Testing)	**Accuracy (%)**	**Test Error (%)**
10-90	72.80	27.20
20-80	75.62	24.38
30-70	76.34	23.66
40-60	82.72	17.28
50-50	86.62	13.38
60-40	84.59	15.41
70-30	87.04	12.96
80-20	85.76	14.24
90-10	87.28	12.72

Table 4

Results using ResNet50 on dataset2

ResNet_only: Epoch=1, BatchSize=10		
Split Ratio (Training-Testing)	**Accuracy (%)**	**Test Error (%)**
10-90	73.82	26.18
20-80	79.20	20.80
30-70	76.23	23.77

Contd...

40-60	79.09	20.91
50-50	81.72	18.28
60-40	80.09	19.91
70-30	82.16	17.84
80-20	81.69	18.31
90-10	81.51	18.49

Table 5

Results using KELM on dataset1

Split Ratio (Training-Testing)	Test Error (%)	Accuracy (%)
10-90	8.24	91.76
20-80	7.59	92.41
30-70	5.78	94.22
40-60	6.01	93.99
50-50	5.82	94.18
60-40	6.06	93.94
70-30	5.63	94.37
80-20	5.14	94.86
90-10	4.64	95.36

Table 6

Results using KELM on dataset2

Split Ratio (Training-Testing)	Test Error (%)	Accuracy (%)
10-90	14.19	85.81
20-80	14.50	85.5
30-70	13.84	86.16
40-60	12.84	87.16
50-50	8.35	91.65
60-40	7.42	92.58
70-30	6.34	93.66
80-20	6.41	93.59
90-10	6.80	93.2

Table 7

Results using MKD-ELM (proposed technique) on dataset1 (random Split)

Split Ratio (Training-Testing) / Iteration	10-90	20-80	30-70	40-60	50-50	60-40	70-30	80-20	90-10
1st	4.55	3.91	3.68	3.39	2.77	2.99	3.59	2.82	1.88
2nd	4.70	4.27	3.93	2.85	2.49	2.52	2.68	2.73	3.59
3rd	4.80	3.76	3.39	3.27	2.53	3.07	2.56	2.39	2.91
4th	4.89	4.14	3.64	3.98	3.28	2.65	2.50	2.56	2.39
5th	4.25	3.65	3.27	3.36	3.11	3.12	2.73	2.73	4.44
6th	4.02	4.33	3.56	3.04	3.01	3.25	2.96	3.65	2.56
7th	4.44	4.06	3.34	3.04	2.63	2.99	2.39	2.47	2.74
8th	4.84	3.86	3.39	3.13	3.21	2.86	2.79	3.75	4.27
9th	4.57	4.38	3.78	3.36	3.04	2.82	3.07	2.82	2.91
10th	4.59	3.80	3.61	2.96	2.77	3.33	2.85	1.62	1.03
11th	4.36	3.86	3.51	2.96	3.01	3.12	3.53	2.65	3.08
12th	4.53	3.71	3.49	3.53	3.25	3.12	2.85	3.58	3.08
Average Test Error	4.55	3.98	3.55	3.24	2.92	2.99	2.87	2.73	2.91
Accuracy	95.45	96.02	96.45	96.76	97.08	97.01	97.13	97.27	97.09
SD of the Test Error	0.25	0.25	0.19	0.31	0.28	0.24	0.37	0.54	0.94

Table 8

Results using MKD-ELM on dataset1 (sequential split)

Split Ratio (Training-Testing)	10-90	20-80	30-70	40-60	50-50	60-40	70-30	80-20	90-10
Test Error	5.42	4.65	4.80	5.63	4.40	4.52	3.01	3.92	4.78

Table 9

Results using MKD-ELM (proposed technique) on dataset2 (random split)

Split Ratio (Training-Testing) iteration	10-90	20-80	30-70	40-60	50-50	60-40	70-30	80-20	90-10
1st	4.13	2.63	2.36	1.96	1.85	1.47	1.23	1.34	1.30
2nd	3.85	2.44	2.52	1.81	1.71	1.36	1.45	0.45	1.09
3rd	4.43	2.85	2.10	1.75	1.59	1.36	1.54	0.54	0.87
4th	4.13	3.01	2.32	1.80	1.48	1.58	1.13	1.12	1.38
5th	4.15	2.83	2.15	1.94	1.40	1.54	1.13	1.45	1.01
6th	4.10	2.76	2.16	1.88	1.51	1.48	1.33	1.23	1.81
7th	4.06	2.89	2.15	2.05	1.71	1.59	1.33	1.38	0.87
8th	4.09	2.90	2.16	1.89	1.72	1.56	1.26	1.38	1.23
9th	4.51	3.19	2.15	2.03	1.55	1.56	1.28	1.20	1.38
10th	4.17	2.73	2.30	1.80	1.69	1.52	1.21	1.30	0.94
11th	3.88	2.71	2.05	1.97	1.55	1.38	1.28	1.41	1.45
12th	4.05	2.77	2.16	1.86	1.85	1.52	1.62	1.30	1.52
Average Test Error	4.13	2.81	2.23	1.89	1.64	1.49	1.32	1.26	1.24
Accuracy	95.87	97.19	97.77	98.11	98.36	98.51	98.68	98.74	98.76
SD of the Test Error	0.19	0.19	0.13	0.10	0.14	0.09	0.15	0.25	0.29

Table 10

Results using MKD-ELM on dataset2 (sequential split)

Split Ratio (Training-Testing)	10-90	20-80	30-70	40-60	50-50	60-40	70-30	80-20	90-10
Test Error	6.81	6.84	6.45	5.58	3.11	2.19	1.59	1.66	0.94

4.2 *Comparison with the state-of-the-art methods*

We carried out a comparative analysis of our proposed MKD-ELM with other contemporary methods. Table 11. presents the comprehensive information. Our novel approach turned out to be the best among the listed state-of-the-art methods as we are able to reduce the error up to 0.94.

Table 11
Comparison with existing state of the art methods

Reference	Technique	Dataset	Result (Accuracy)
Dongsheng Ji et. al. [20]	ResNet152, Xception and InceptionResNetV2	4099 C_hXR images	96%
Abdul Waheed et. al. [40]	VGG16	1124 C_hXR images	95%
Mohd Zulfaezal Che Azemin et. al. [41]	ResNet-101	5,828 C_hXR images	71.9%
Muhammad Umair et. al. [21]	VGG16, ResNet-50, DenseNet-121, and MobileNet, Grad Cam Technique	Dataset1:7232 C_hXR images Dataset2:450 C_hXR images	96.49%
Arpan Mangal et. al. [42]	CheXNet	5856 C_hXR images	90 .5%
R. Murugan et. al. [19]	CNN Resnnet50 based ELM Classifier	2700 images	94.07%
Shuai Wang et. al. [43]	GoogleNet, Inception v3	1,065 CT images	89.5%
Proposed technique (ours)	Resnet50 and MKD-ELM	Dataset1:5856 C_hXR images Dataset2:13,808 C_hXR images	99.06% 97.27%

Dataset

Dataset1: The originally available dataset, C_hXR images (Pneumonia) comprises a total of 5,856 C_hXR which are organized in three main folders namely: training, testing and validation. Each of these three folders includes two subfolders for each image category, i.e., normal Vs pneumonia. We reorganized the entire data into two sets only as normal and pneumonia and the label of folder comprising images of healthy (normal) people is set to 1 which consists a total of 1583 images and set the label of the other folder is set to 2 which consists a total of 4,273 pneumonia images. This dataset was randomly split into two independent data sets with 10% and 90% for training, and testing respectively by adding *splitEachLabel* function

in our matlab code. The similar experiment was repeated eight more times for different split ratios as 20:80, 30:70, 40:60, 50:50, 60:40, 70:30, 8;20 and 90:10 and test error were recorded for 12 iterations and then average testing error was computed. Then the same dataset was split sequentially and test error was recorded for nine different ratios.

Dataset2: This COVID-19 radiography dataset available publicly consists a total of 3,616 COVID-19 positive cases images along with 10,192 normal, 6,012 lung opacity (non-covid lung infection), and 1,345 viral pneumonia images and corresponding lung masks. But as we are performing the binary classification in this study, we reorganized the entire data into two folders namely Normal (consisting of 10,192 images) whose label is set to 1 and covid (consisting of 3,616 images) whose label is set to 2. Subsequently, splitting process for training and testing datasets has been repeated as done for dataset1. This dataset was also split both randomly and sequentially in nine different split ratios.

5. Conclusions

A novel approach based on non-iterative MKD-ELM, has been proposed for a classification of COVID-19 using C_hXR images in this paper. First, a novel technique has been used for the purpose of feature extraction which exploits the capability of the ResNet deep learning model neither in pretrained form nor as a transfer learning model. The ELM classifier, whose structural parameters have been thoroughly analyzed serves as the framework for the proposed method. Despite its efficiency, the ELM classifier cannot be used to address classification problems that call for deterministic and immutable results. So, to improve its effectiveness and make it deterministic, a data related multi kernel approach has been proposed whereby the problem's perceptivity is examined in order to experimentally determine the kernel and their combination coefficients. Also, the presented experimentation with the optimization of parameters of MKD-ELM using GA is a novel and reliable attempt for the purpose of parameter optimization and performs far better than the usual ELM approach, which initializes the weights randomly. We have demonstrated the performance of our proposed approach using two datasets, and we are able to minimize the testing error up to 2.73 on dataset 1 and up to 0.94 on dataset 2. However, the present investigation has been carried out for binary classification (to categorize COVID-19 patients and healthy people). Multi-class classification (COVID-19, bacterial pneumonia, and viral pneumonia) will be investigated in the future study. We will explore larger

datasets and use data augmentation to get better classification results in further studies. In summary, it can be said that the results reported in this study using our proposed novel approach based on MKD-ELM was the best performing technique achieves the best classification accuracy. At the same time, it requires less time for training the model.

Data Availability Statement

Dataset 1: https://www.kaggle.com/datasets/paultimothymooney/chest-xray-pneumonia.

Dataset 2: https://www.kaggle.com/datasets/tawsifurrahman/covid19-radiography-database.

References

[1] P. Kalane, S. Patil, B. P. Patil, and D. P. Sharma, "Automatic detection of COVID-19 disease using U-Net architecture based fully convolutional network," *Biomed Signal Process Control*, Vol. 67 (May 2021), doi: 10.1016/j.bspc.2021.102518.

[2] M. Canayaz, "MH-COVIDNet: Diagnosis of COVID-19 using deep neural networks and meta-heuristic-based feature selection on X-ray images," *Biomed Signal Process Control*, Vol. 64 (Feb. 2021), doi: 10.1016/j.bspc.2020.102257.

[3] M. Otoom, N. Otoum, M. A. Alzubaidi, Y. Etoom, and R. Banihani, "An IoT-based framework for early identification and monitoring of COVID-19 cases," *Biomed Signal Process Control*, Vol. 62 (Sep. 2020), doi: 10.1016/j.bspc.2020.102149.

[4] Q. Zou, P. Xing, L. Wei, and B. Liu, "Gene2vec: gene subsequence embedding for prediction of mammalian N 6-methyladenosine sites from mRNA" (2019), doi: 10.1261/rna.

[5] A. Amirkhani, E. I. Papageorgiou, A. Mohseni, and M. R. Mosavi, "A review of fuzzy cognitive maps in medicine: Taxonomy, methods, and applications," *Computer Methods and Programs in Biomedicine*, vol. 142. Elsevier Ireland Ltd, pp. 129–145 (Apr. 01, 2017), doi: 10.1016/j.cmpb.2017.02.021.

[6] A. Li *et. al.* "A Tutorial on Interference Exploitation via Symbol-Level Precoding: Overview, State-of-the-Art and Future Directions."

[7] S. Yang, T. Gao, J. Wang, B. Deng, B. Lansdell, and B. Linares-Barranco, "Efficient Spike-Driven Learning With Dendritic Event-Based Processing," *Front Neurosci*, Vol. 15 (Feb. 2021), doi: 10.3389/fnins.2021.601109.

[8] W. Rawat and Z. Wang, "Deep convolutional neural networks for image classification: A comprehensive review," *Neural Computation*, *MIT Press Journals*, Vol. 29, No. 9. pp. 2352–2449 (Sep. 01, 2017), doi: 10.1162/NECO_a_00990.

[9] Q. v Le, J. Ngiam, A. Coates, A. Lahiri, B. Prochnow, and A. Y. Ng, "On Optimization Methods for Deep Learning" (2011).

[10] J. Martens, "Deep learning via Hessian-free optimization" (2010).

[11] M. W. Klein, C. Enkrich, M. Wegener, and S. Linden, "Second-harmonic generation from magnetic metamaterials," *Science (1979)*, Vol. 313, No. 5786, pp. 502–504 (Jul. 2006), doi: 10.1126/science.1129198.

[12] L. Kong and J. Cheng, "Classification and detection of COVID-19 X-Ray images based on DenseNet and VGG16 feature fusion," *Biomed Signal Process Control*, Vol. 77 (Aug. 2022), doi: 10.1016/j.bspc.2022.103772.

[13] A. A. Saraiva *et. al.* "Classification of images of childhood pneumonia using convolutional neural networks," in *BIOIMAGING 2019 - 6th International Conference on Bioimaging, Proceedings; Part of 12th International Joint Conference on Biomedical Engineering Systems and Technologies, BIOSTEC 2019*, pp. 112–119 (2019), doi: 10.5220/0007404301120119.

[14] O. Stephen, M. Sain, U. J. Maduh, and D. U. Jeong, "An Efficient Deep Learning Approach to Pneumonia Classification in Healthcare," *J Healthc Eng*, Vol. 2019 (2019), doi: 10.1155/2019/4180949.

[15] D. S. Kermany *et. al.* "Identifying Medical Diagnoses and Treatable Diseases by Image-Based Deep Learning," *Cell*, Vol. 172, No. 5, pp. 1122-1131.e9 (Feb. 2018), doi: 10.1016/j.cell.2018.02.010.

[16] S. Rajaraman, S. Candemir, I. Kim, G. Thoma, and S. Antani, "Visualization and interpretation of convolutional neural network predictions in detecting pneumonia in pediatric chest radiographs," *Applied Sciences (Switzerland)*, Vol. 8, No. 10 (Sep. 2018), doi: 10.3390/app8101715.

[17] P. Sangulagi and A. Kumar, "Detection of Covid-19 from Chest X-Ray Images," *Journal of Scientific Research*, Vol. 66, No. 02, pp. 172–178 (2022), doi: 10.37398/jsr.2022.660223.

[18] S. Lu, D. Wu, Z. Zhang, and S. H. Wang, "An explainable framework for diagnosis of COVID-19 pneumonia via transfer learning and discriminant correlation analysis," *ACM Transactions on Multimedia Computing, Communications and Applications*, Vol. 17, No. 3s (Oct. 2021), doi: 10.1145/3449785.

[19] R. Murugan and T. Goel, "E-DiCoNet: Extreme learning machine based classifier for diagnosis of COVID-19 using deep convolutional network," *J Ambient Intell Humaniz Comput*, Vol. 12, No. 9, pp. 8887–8898 (Sep. 2021), doi: 10.1007/s12652-020-02688-3.

[20] D. Ji, Z. Zhang, Y. Zhao, and Q. Zhao, "Research on Classification of COVID-19 Chest X-Ray Image Modal Feature Fusion Based on Deep Learning," *J Healthc Eng*, Vol. 2021 (2021), doi: 10.1155/2021/6799202.

[21] M. Umair *et. al.* "Detection of COVID-19 using transfer learning and grad-cam visualization on indigenously collected X-ray dataset," *Sensors*, Vol. 21, No. 17 (Sep. 2021), doi: 10.3390/s21175813.

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 2016-December, pp. 770–778 (Dec. 2016), doi: 10.1109/CVPR.2016.90.

[23] G. bin Huang, Q. Y. Zhu, and C. K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, Vol. 70, No. 1–3, pp. 489–501 (Dec. 2006), doi: 10.1016/j.neucom.2005.12.126.

[24] G. bin Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, Vol. 42, No. 2, pp. 513–529 (Apr. 2012), doi: 10.1109/TSMCB.2011.2168604.

[25] I. Arora, A. Saha, B. Ahuja, and V. P. Vishwakarma, "ELM and KELM based software defect prediction using feature selection techniques," *Journal of Information and Optimization Sciences*, Vol. 24, No. 3, pp. 843–858 (2019), doi: 10.1080/09720529.2021.1877409.

[26] Rao and Mitra, "*Generalized Inverse of Matrices and Its Applications*".

[27] A. E. Hoerl and R. W. Kennard, "Ridge Regression: Biased Estimation for Nonorthogonal Problems," *Technometrics*, Vol. 12, No. 1, pp. 55–67 (1970), doi: 10.1080/00401706.1970.10488634.

[28] C. A. Micchelli and P. Bartlett, "Learning the Kernel Function via Regularization Massimiliano Pontil," (2005).

[29] T. Evgeniou, "Learning Multiple Tasks with Kernel Methods Charles A. Micchelli Massimiliano Pontil," (2005).

[30] B. Ahuja and V. P. Vishwakarma, "Optimised multikernels based extreme learning machine for face recognition Optimised multikernels-based extreme learning machine," (2018).

[31] F. Aiolli and M. Donini, "EasyMKL: A scalable multiple kernel learning algorithm," *Neurocomputing*, Vol. 169, pp. 215–224 (Dec. 2015), doi: 10.1016/j.neucom.2014.11.078.

[32] Z. Xu, R. Jin, H. Yang, I. King, and M. R. Lyu, "Simple and Efficient Multiple Kernel Learning by Group Lasso," (2010), [Online]. Available: www.shogun-toolbox.org/

[33] S. S. Bucak, R. Jin, and A. K. Jain, "Multiple kernel learning for visual object recognition: A review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 36, No. 7. IEEE Computer Society, pp. 1354–1369 (2014), doi: 10.1109/TPAMI.2013.212.

[34] J. Zhuang, I. W. Tsang, S. C. H. Hoi, J. ; Zhuang, and I. W. ; Tsang, "Two-Layer Multiple Kernel Learning Two-Layer Multiple Kernel Learning Citation Citation Two-Layer Multiple Kernel Learning," (2011), [Online]. Available: https://ink.library.smu.edu.sg/sis_research

[35] X. Xu, I. W. Tsang, and D. Xu, "Soft margin multiple kernel learning," *IEEE Trans Neural Netw Learn Syst*, Vol. 24, No. 5, pp. 749–761 (2013), doi: 10.1109/TNNLS.2012.2237183.

[36] D. E. Goldberg and J. H. Holland, "Genetic Algorithms and Machine Learning _ Enhanced Reader".

[37] W. Chen, M. Panahi, and H. R. Pourghasemi, "Performance evaluation of GIS-based new ensemble data mining techniques of adaptive neuro-fuzzy inference system (ANFIS) with genetic algorithm (GA), differential evolution (DE), and particle swarm optimization (PSO) for landslide spatial modelling," *Catena (Amst)*, Vol. 157, pp. 310–324 (Oct. 2017), doi: 10.1016/j.catena.2017.05.034.

[38] B. Ahuja and V. P. Vishwakarma, "Optimization of regularization coefficient and kernel parameters of KELM in face recognition using genetic algorithm," *Journal of Discrete Mathematical*

Sciences and Cryptography, Vol. 24, No. 3, pp. 843–858 (2021), doi: 10.1080/09720529.2021.1877409.

[39] N. Metawa, M. K. Hassan, and M. Elhoseny, "Genetic algorithm based model for optimizing bank lending decisions," *Expert Syst Appl*, Vol. 80, pp. 75–82 (Sep. 2017), doi: 10.1016/j.eswa.2017.03.021.

[40] A. Waheed, M. Goyal, D. Gupta, A. Khanna, F. Al-Turjman, and P. R. Pinheiro, "CovidGAN: Data Augmentation Using Auxiliary Classifier GAN for Improved Covid-19 Detection," *IEEE Access*, Vol. 8, pp. 91916–91923 (2020), doi: 10.1109/ACCESS.2020.2994762.

[41] M. Z. Che Azemin, R. Hassan, M. I. Mohd Tamrin, and M. A. Md Ali, "COVID-19 Deep Learning Prediction Model Using Publicly Available Radiologist-Adjudicated Chest X-Ray Images as Training Data: Preliminary Findings," *Int J Biomed Imaging*, Vol. 2020 (2020), doi: 10.1155/2020/8828855.

[42] A. Mangal *et. al.* "CovidAID: COVID-19 Detection Using Chest X-Ray" (Apr. 2020), [Online]. Available: http://arxiv.org/abs/2004.09803

[43] S. Wang *et. al.* "A deep learning algorithm using CT images to screen for Corona virus disease (COVID-19)," *Eur Radiol*, Vol. 31, pp. 6096–6104 (2021), doi: 10.1007/s00330-021-07715-1/Published.

Received October, 2022

Revised November, 2022