



Full Length Article

Experimental analysis of machine learning methods to detect Covid-19 from x-rays

Fuad M. Alkoot^{a,*}, Hussain.M. Alkhedher^b, Zahraa F. Alkoot^b

^a Higher Institute of Telecommunication & Navigation, PAAET, Kuwait

^b Ministry of Health, Kuwait

ARTICLE INFO

Keywords:

Covid-19 detection
Classifier ensembles
Machine learning
Artificial intelligence in medicine

ABSTRACT

To automate the detection of covid-19 patients most have proposed deep learning neural networks to classify patients using large databases of chest x-rays. Very few used classical machine learning methods. Machine learning methods may require less computational power and perform well if the data set is small. We experiment with classical machine learning methods on three different data sources varying in size from 55 to almost 4000 samples. We experiment with four feature extraction methods of Gabor, SURF, LBP, and HOG. Backpropagation neural networks and k-nearest neighbor classifiers are combined using one of the four combining methods of bagging, RSM, ARCx4 boosting and Ada-boosting. Results show that using the proper feature extraction and feature selection methods very high performance can be reached using simple backpropagation neural network classifiers. Regardless of combiner method used, the best classification rate achieved was 99.06% for the largest data set, and 100% for the smallest data set.

Introduction

Automating disease diagnosis is a prominent capability of machine learning and artificial intelligence systems. Machine learning and pattern recognition classifier models [1–4] can spot trends and relationships in data sets, which makes them well suited for many applications to automate decisions or applications that require classification of unknown samples, including medical applications [2,3,5]. Therefore, we aim to experimentally investigate several methods to find the best system that can detect COVID-19.

Suspected patients that suffer from COVID-19 symptoms undergo screening using chest X-Ray or chest CT Scan, both showing patterns of COVID-19 lung disease [6]. Chest X-ray exam is one of the most frequent and cost-effective medical imaging examinations. However, in contrast to CT images, clinical diagnosis of chest x-rays is found to be more difficult and challenging [6]. As repeatedly claimed by researchers on x-ray segmentation [7–9], x-ray images are more available but also difficult to detect and diagnose. Covid-19 Chest X Rays, or CXR, were assessed by [10] for the presence of a specific distribution of the disease. They found that a characteristic feature of COVID-19 pneumonia is involvement of the outermost, peripheral lung fields showing bilateral multifocal, patchy, or confluent opacities. However, other viruses such as SARS-Cov-1 and MERS-Cov also may cause pneumonia

[10]. Therefore, radiologists must screen the x-ray images to determine the cause of the lung infection. This motivated us to investigate the possibility of designing a CAD system to detect COVID-19 from x-rays. Additionally, in the pre-analytical stage, accurate molecular diagnosis of COVID-19 is not always guaranteed and is influenced by collecting the proper respiratory tract specimen at the right time from the right anatomic site [11,12]. This confirms the need for methods of diagnosis other than PCR, such as the x-ray. Also, contrary to COVID-19 test kits, X-ray machines do not require transportation of infected samples and are widely available in all hospitals. The resulting computer aided detection or computer aided diagnosis, CAD, system can be used as a first line of screening and as a parallel diagnosis method.

To build a CAD system most of the surveyed papers in the next section proposed covid-19 detection using deep learning neural networks that require more computational power and require a very large database to train. We aim to find if it is possible to achieve rates similar to deep learning neural networks using classical machine learning methods. In our experimental investigation we experiment with various machine learning methods at the different stages of the system, from the initial stage of feature extraction to the final stage of classifier combination and fusion. Our experimental results show that classical machine learning methods may yield competitive rates and outperform the sophisticated deep learning neural network models using small data

* Corresponding author.

E-mail address: fm.alkoot@paaet.edu.kw (F.M. Alkoot).

Table 1
summary of proposed methods and classification rates achieved by each method.

Authors	Method	Rate
Aytac Altan and Sicken karasu.[15]	Deep learning neural networks with chaotic salp swarm algorithm applied to the 2D transformation equation.	99%
Bejoy Abrahama and Madhu S.Nairb,[16]	Multi-CNN as a first stage for feature extraction and baysnet classifier at a final stage.	97.44%
Dalia Ezzata, Aboul EllaHassaniena and Hassan AboulEllab,[17]	Use DEnsenet121, to detect. GSA, is used to find the optimal parameters of the CNN.	98.38%
Linda wang, Zhong qiu lin and Alexander Wong[18]	Deep learning neural network which is pretrained using Image-Net dataset then covid cxr images.	93.3%
Zheng Wang et.al.[19]	Two steps with deep learning neural networks, D.L.: the first DL extracts lung features, the second DL localizes and assigns CXR to left lung, right lung or bipulmonary.	93.65%
Sergio Varela-Santos and Patricia Melin,[20]	Experiment with three classifiers; feedforward neural network with raw input images, feedforward neural network with features extracted using the methods of LBP and GLCM in addition to texture feature operators, and a convolutional neural network with raw input images.	A range of rates from 80.61% to 96.83% on different data sets.
Tej Bahadur Chandra,[21]	Two stages of classifier ensembles. Used k-NN, ANN, SVM, DT and NB classifiers in the ensemble. They extract features from the xray images using HOG and GLCM methods.	98.06% stage I and 91.32% stage II
Methods proposed in this paper	Extract features from images using LBP then reduce them using fisher score. Next data are used to build an ensemble of neural network classifiers combined using Ada-boosting, ARCx4 boosting, RSM or bagging.	99.06% 98.88% 99.03% 98.85%

sets. The contributions of our work are summarized as follows:

- We present an experimental comparison of many machine learning methods to show how each performs in this application.
- We present a machine learning method for detecting Covid-19 patients that outperforms most of the surveyed deep learning neural networks using less computational power.

Related work

Since the first appearance of the COVID-19 chest image data in March 2020 research on detection methods using CXR or CT images has received a great deal of attention in the artificial intelligence community. Most used deep learning neural networks to achieve classification rates of 90% and above. The first method that achieved an accuracy of 92% was proposed by [13]. They used a deep learning system for COVID-19 detection, however, using CT scans. For a quick review on early work done to segment and diagnose covid-19 from CT or CXR images refer to [14]. Surveying methods that only use CXR images yielded the papers listed in Table 1. This table summarizes the methods and the rates achieved by each. Papers in the first five rows [13,15–19] use deep learning neural networks, while [20] and [21] use various machine learning methods.

A detailed review of these papers shows that most use deep learning neural networks but differ in their architecture and the parameter selection method. Authors in [15] propose a 2D curvelet transform and deep neural network to detect covid-19 from CXR images. They compare the performance of three deep networks that have three different inputs; one without any transformation on the images, the second with a 2D curvelet transformation on the input images and the third with chaotic salp swarm algorithm applied to the 2D transformation equation. In a two-class problem, authors in [16] propose using multi-convolutional neural network, CNN, as a first stage for feature extraction, followed by SFSS feature selection to reduce the data dimensionality from CXR images. Finally, the images are classified using the Baysnet classifier. They experiment with different combinations of convolutional neural networks, CNNs, three different feature selection methods and many classifiers. Their data consist of more than 900 samples, approximately half of them being covid – 19 positive. They used 10-fold cross-validation to measure the different metrics of recall, specificity,

accuracy etc. Authors in [17] Use DEnsenet121, a pretrained deep learning network, to detect covid-19 from chest x-rays. They use gravitational search algorithm, GSA, to select optimal values of the hyperparameters of the network. Gravitational search algorithm, GSA, is used to find the optimal parameters of the CNN. A deep learning neural network named COVID-net is built by authors in [18], which is pretrained using Image-Net dataset then covid CXR images. Factors used by COVID-net to identify are ground glass opacities, bilateral abnormalities and inferstitial abnormalities. They compare their proposed deep learning neural network to VGG-19 and ResNet, and show it outperforms them, with a lighter design. Authors in [19] They propose a computer aided design, CAD, framework that consists of two steps with deep learning neural networks: the first deep learning neural network extracts lung features, the second deep learning neural network localizes and assigns CXR to left lung, right lung or bipulmonary.

Authors in [20] investigate different machine learning methods and one deep learning method. They experiment with three classifiers; feedforward neural network with raw input images, feedforward neural network with features extracted using the methods of local binary pattern, LBP, and gray level co-occurrence matrix, GLCM in addition to texture feature operators, and finally a convolutional neural network with raw input images. They test using Cohens databases published in March and April 2020. Authors in [21] Propose using two stages of classifier ensembles. The first classifies samples to normal or pneumonic. The second detects covid from the pneumonic images. They use k-nearest neighbor, artificial neural network, support vector machine, decision tree and naïve bayes classifiers in the ensemble then fuse the decision using vote fusion method. They extract features from the x-ray images using histogram of oriented gradients, HOG, and GLCM methods. They also use the Cohen database.

The next section presents the experimental methodology, describing machine learning methods of feature extraction, selection, classification, and combination. Additionally, the data used are described. Section 3 presents the results and discussion. The conclusion is presented in Section 4.

Research methodology

The machine learning system proposed in this paper is shown in Fig. 1. The stages of the system are described in the following

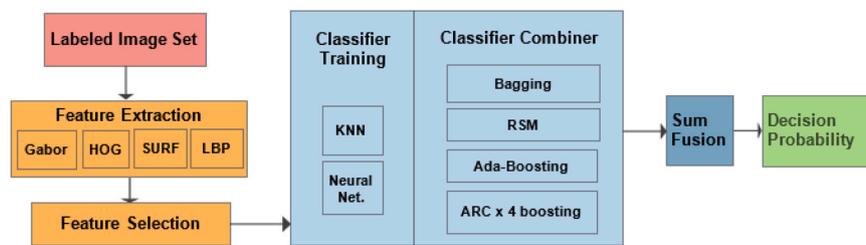


Fig. 1. Block diagram of proposed system.

Table 2
number of samples in each data set of the three sources.

x-rays:	Covid	Normal	Pneumonia	Total samples
Source 1, [22]	478	111	190	778
Source 2, [23]	35	18	2	55
Source 3, [24]	1143	1341	1345	3829

subsections. The experiments were implemented using Matlab version 2021a.

Datasets

A training set of x-ray images from both healthy and infected lungs are required to build the required classifier system. In addition to x-ray images of infected lungs due to COVID-19 there is a need for images of infected lungs due to other causes. There are several datasets of COVID-19 CXR images available. The first such data was published on GitHub repository by Dr. Josef Cohen from the University of Montreal [21]. He started collecting these COVID-19 images in March 2020 and it was expected to grow. Two other sources [23,24] were also used in our experiments. At the time of our experiments the three sources had the number of frontal x-ray pictures, as shown in Table 2. Other samples of CT or non-frontal x-rays were omitted. The data size may increase as more images are contributed and uploaded to the database. We aim at only detecting covid, therefore we have a two-class problem. We labeled the data to two classes, the first is the covid class while the second includes the non-covid samples which included normal and pneumonia due to other sources than covid-19.

Dataset preparation for training and testing

In this paper, initially as a preprocessing step, the images are resized to 512 by 512 pixels. Experiments are performed on non-cropped images then repeated for data 1 and data 2 using cropped images that include only the lungs as shown in Fig. 2 and Fig. 3. In these figures the left is the original uncropped, and the middle is the cropped and resized. Images are converted to matrices and a label is added to the

images as a covid or non-covid class. Finally, the data are partitioned to a training set and a test set using the 10-fold cross validation. This is similar to the partition used in the papers of Table 1.

Feature extraction methods

Image texture is defined as a two-dimensional phenomenon characterized by two properties: (1) spatial structure (pattern) and (2) contrast. They work because the most frequent patterns correspond to primitive microfeatures such as edges, corners, spots. We experiment with filtering or feature extraction methods that can effectively extract the changes made in a chest x-ray image by covid-19 infections. X-ray images are converted to column features using a feature extraction method. We experiment with Speeded-Up Robust Features, or SURF, Gabor, LBP, and HOG [25] methods. Look at the appendix for a detailed description of the four feature extraction methods. Here we will briefly present the parameter values of each method.

For the LBP we set the number of divisions of an image to 10. We set the number of neighbors to 20. The radius of circular pattern used to select neighbors is set at 1.

The parameters for the Gabor are set as follows. The wavelength is set in increasing powers of 2 starting from $\frac{4}{\sqrt{2}}$ up to the hypotenuse length of the input image. We initially find the minimum and maximum wavelengths. Minimum is $\frac{4}{\sqrt{2}}$ which is 2.83, while maximum is the square root of sum of square of number of columns and number of rows in an image. Next, assume n is the integer result from the base 2 logarithm of the maximum wavelength value divided by the minimum wavelength value. The wavelength is the set of values resulting from the following equation:

$$\text{Wavelength} = \text{minimum wavelength value} \times 2^{(0 \text{ to } n-2)}$$

Gabor orientation is set at 0 and 90 for data 2 and at 0 and 126 for data 1 and 3. Based on the given set of orientation and wavelength values we create a set of Gabor filters. Then, we extract Gabor magnitude features from source images using the filters. We use simple Gaussian low-pass filtering to smooth the Gabor magnitude information. We choose a sigma that is matched to the Gabor filter that extracted each feature. We use a smoothing term K= 3 that controls how much smoothing is applied to the Gabor magnitude responses.



Fig. 2. Images from source 1, for uncropped covid (L), cropped & resized covid (middle) and non-covid (R).



Fig. 3. Images from source 2, for uncropped covid (L), cropped & resized covid and non-covid (R).

Speeded-Up Robust Features, SURF, feature descriptor is based on the sum of the Haar wavelet response around the point of interest. So, the region is split up regularly into smaller 4×4 square sub-regions. For each sub-region, we compute a few simple features at 5×5 regularly spaced sample points. The Euclidean distance is used to calculate the distance between the descriptor vectors of two images.

For the HOG we group 16×16 pixels and replace their gradients with one that represents the arrow directions that were the strongest. The result is an image of directed arrows that captures the structure of an image.

Feature selection methods

After extracting the features using any of the four methods above, we reduce their dimensionality using feature selection. The full feature set shown as columns of data obtained from the feature extraction method is reduced using the feature selection method proposed in [4]. It is regularly used as a clustering tool; however, we do not aim to find classes or clusters because the classes are known. We use this clustering tool as a feature selection method to find the most distinguishing features. Therefore, we use this clustering tool to find features that yield the largest distance between the means of the two clusters and yield clusters with smallest standard deviation. This can be found using the following equation, known as fisher score [1].

$$f_{1,2} = \frac{(\mu_1 - \mu_2)^2}{(\sigma_1 - \sigma_2)} \quad (1)$$

Based on their fisher scores we sort the features in a descending order. We take a maximum of top 500 features with the highest fisher score. Next, we find and remove the redundant features if they exist. We keep all features if the data has a dimensionality less than 500. Initially extracted features in this project exceeded 500.

Classifier methods

Following the transformation of images to columns of data and reducing the number of columns to a minimum we use the data to build or train the classifiers. Using the reduced feature set we create an ensemble of neural network classifiers [1]. We also experiment with the k-nearest neighbor classifier which does not require training [1]. For the nearest neighbor classifier k is set at 3 and the distance metric is the Euclidean metric. The neural network classifier used is a back-propagation network with Levenberg-Marquardt training function. It consists of three layers, where the number of neurons in the first layer is equal to the number of features, while that for the hidden (second) layer is set at 5. The number of neurons at the output layer is equal to

the number of classes. The number of epochs is set at 100, while the training error goal is set at 0.001.

Classifier combiner methods

A combination of classifiers is used to achieve the best performance possible. We experiment with four prominent classifier combiner methods: Bagging [29], Random subspace or RSM [30], Adaboosting [31,33] and ARCX4 boosting [33].

Bagging predictors proposed by Breiman, is a method of generating multiple versions of a predictor or classifier, via bootstrapping and then using those to get an aggregated classifier. We set the number of multiple versions of classifiers to 25, as recommended by Breiman. The total number of samples in each bootstrap set is equal to those of the original training set.

The second combiner 'RSM' aims at creating diverse classifiers by assigning different features to each classifier. The number of features is set at a fixed value, m, less than the total number of features. Each classifier is assigned a subset of features that are randomly selected without replacement from the full feature set. This results in classifiers having different views of the data space. We set m to equal 75% of the total number of available features. The number of combined classifiers is set equal to bagging at 25.

Boosting is an effective combiner method proposed by Freund in 1995 and improved by Freund & Schapire in 1997 [31,32]. It works by focusing on the difficult samples. It creates many classifiers and assigns samples to classifiers randomly. However, each sample has a weight that is related to the probability of a sample being selected for inclusion in the classifier. Therefore, it iteratively builds a new classifier by changing the number of some samples in the classifiers training set. This is done by changing the weight of each sample based on the performance of the last built classifier. Weights of misclassified samples are increased, and weights of correctly classified samples are decreased. Therefore, for the Ada-boosting and ARCx4 boosting combiners the number of built classifiers depends on the data. Following is a detailed description of a version of the boosting method called ARCx4 which was proposed by Brieman [33]:

- Initialize the weights of samples to be $1/N$, where N is the number of samples in the original training set.
- Create the first classifier using a training subset created from the original training set by randomly selecting samples given the initial weights of samples.
- Check the performance of the classifier to find misclassified samples.
- For each sample we count the number of classifiers that misclassified the sample, then divide by number of classifiers in the combiner to get a value p.
- Calculate $h = 1 + p^4$ for each sample

Table 3

number of features generated by each feature extraction method and number of features remaining after reducing using the fisher score feature selection.

	HOG	HOG reduced	SURF	SURF reduced	Gabor	Gabor reduced	LBP	LBP reduced
Data 1	142885	500	35713	182	262145	439	2201	491
Data 2	142885	500	13825	20	262145	280	2201	479
Data 3	142885	500	31233	427	262145	469	2201	439

- f. Find the new weights for each sample as h of a sample divided by sum of all h for all samples
- g. If maximum number of classifiers of a combiner is not reached create a new classifier using the new weights. Otherwise stop.
- h. Repeat steps c to g.

Finally, the decisions or probability values of each of the classifiers in a combiner are fused using the sum fusion method to obtain an average probability. We compare the performance of the different combiners and classifier types by calculating the classification rate, specificity and sensitivity. Experiments are repeated using 10-fold cross validation after randomly shuffling the samples. The results of the ten trials are averaged to achieve statistical reliability.

Deep learning neural network methodology

In addition to deep learning neural networks results provided by others we also run experiments on deep learning neural networks to compare the performance of our machine learning method to a deep learning neural networks network, on the same data. We run experiments of deep learning neural networks for all three datasets. Initially we load the image data into an image datastore database readable by the deep learning neural networks. Then, we randomly split the data to 90% training and 10% test sets using 10-fold cross validation. Next, we define the convolutional deep learning network architecture as follows, progressing sequentially from first layer to last:

1. An input layer that is the same size as the input image, which are 512 by 512. The channel size is set to 1 because our data consists of grayscale images.
2. Next layer is the convolution layer with 20 Filters which corresponds to the number of neurons that connect to the same area in the image and filter size 10.
3. Next a batch normalization layer is used to normalize the activations propagating through a network. This layer is used between convolutional layers and ReLU layers.
4. Next a ReLU layer is used as a nonlinear activation function.
5. A max pooling layer is used as a down sampling layer to increase the number of filters in deeper convolutional layers without increasing amount of computation per layer. Maximum value of rectangular regions is set at 2 with step size of training function set at 2.
6. Layer at steps 2-5 are repeated with 30 filter in the convolutional layer and filter size 8.
7. Again, layers at steps 2-5 are repeated with 40 filters in the convolutional layer and filter size 5.
8. Next, we have a fully connected layer where neurons fully connect to neurons in the preceding layer. The output of this layer is equal to the number of classes, i.e. 2.
9. Next we have a softmax layer which normalizes the output of the last fully connected layer, i.e. the output consists of positive numbers that sum to 1.
10. Finally the last layer is the classification layer which uses the output of the preceding layer to assign the input to one of the classes.

Results and discussion

We have tables of results for each of the two classifier types and four combiner methods. The experiments were also repeated for each of the three data sets extracted using four feature extraction methods. [Table 3](#)

Table 4

Classification rates using 10-fold cross validation partition of non-cropped CXR image data.

Classifier	Combiner	HOG	SURF	LBP	Gabor	Gabor & LBP	SURF & LBP	Gabor & SURF	Gabor, SURF & LBP		
k-NN	Data 1	Boosting	67.69	61.54	72.05	61.67	61.67	68.97	61.03	61.41	
		Arcx4	69.62	62.82	73.33	60.38	61.03	69.62	61.15	60.13	
		RSM	70.51	65.00	73.97	60.90	60.90	68.59	61.41	61.28	
	Data 2	Boosting	68.33	63.21	72.82	62.05	62.31	69.23	61.67	62.05	
		Arcx4	100	63.33	81.67	68.33	61.67	76.67	63.33	63.33	
		RSM	100	70.00	81.67	71.67	70.00	78.33	73.33	73.33	
	Data 3	Boosting	100	66.67	85.00	68.33	68.33	81.67	71.67	68.33	
		Arcx4	100	71.67	83.33	71.67	73.33	80.00	73.33	73.33	
		RSM	100	93.26	67.70	98.25	84.73	88.49	95.07	85.54	87.68
	Neural networks	Data 1	boosting	92.38	70.86	98.04	85.30	88.30	95.01	86.58	88.02
			Arcx4	92.85	74.33	98.28	85.35	88.30	95.48	86.11	87.73
			RSM	92.01	72.09	98.20	85.69	88.36	94.88	86.40	87.99
Data 2		boosting	69.87	63.85	72.56	62.31	72.44	69.10	65.38	70.77	
		Arcx4	70.51	57.95	72.31	48.59	74.10	70.90	61.92	69.49	
		RSM	69.87	63.85	72.56	62.31	72.44	69.10	65.38	70.77	
Data 3		boosting	71.03	61.28	72.18	63.59	73.97	70.13	64.10	71.79	
		Arcx4	100	71.67	88.33	56.67	76.67	86.67	75.00	83.33	
		RSM	100	70.00	85.00	71.67	76.67	88.33	75.00	81.67	
Data 3		boosting	100	71.67	85.00	68.33	75.00	88.33	78.33	83.33	
		Arcx4	93.76	76.11	99.06	87.96	98.15	97.42	85.12	97.55	
		RSM	94.18	76.06	98.88	87.91	98.17	97.36	85.01	97.55	
Data 3	boosting	94.44	76.08	99.03	87.52	98.07	97.73	85.56	97.86		
	Arcx4	94.20	76.61	98.85	87.73	98.04	97.31	85.20	97.70		
	RSM	94.20	76.61	98.85	87.73	98.04	97.31	85.20	97.70		

Table 5

Metric results of all combiners, averaged over 10 cross validations, for largest data source 3 using LBP features and neural network classifier.

	Ada-boost	ARCx4 boost	RSM	bagging
Sensitivity	0.99	0.98	0.99	0.98
Specifity	0.99	0.99	0.99	0.99
Precession	0.98	0.98	0.98	0.98
Accuracy	0.99	0.99	0.99	0.99
F1 score	0.98	0.98	0.98	0.98
True positive	112.30	111.80	112.20	111.90
False positive	2.10	2.60	2.20	2.50
False negative	1.50	1.70	1.50	1.90
True negative	267.10	266.90	267.10	266.70
All negative	269.20	269.50	269.30	269.20

displays the number of features generated by each feature extraction method, for each data set. It also shows the number of features after reducing them using our proposed feature selection method.

Results in Table 4 show that our proposed system yield very high rates. Previous works surveyed in the introduction and summarised in Table 1 use a large data set. Therefore, we focus more on results of the largest data set, number 3.

We also experiment with merging data from two or three feature types. The classification rates for merged feature sets are displayed in the columns following the single feature columns in Table 4 and Table 6.

Results show that the classical machine learning system yields very good results. For some feature extraction methods, the performance reaches close to the optimum rate of 100%. This was achieved using neural networks and LBP features. Using HOG at smallest data set size,

for both classifier types, the classification rate reaches the highest rate of 100%. Table 5 shows the resulting metric values for the LBP features using neural network classifiers. These achieved rates outperform most surveyed methods in Table 1. Only the first deep learning neural networks method in Table 1 is close to our method, however, significantly lower than our best rate. To compare using the same data set we also run experiments on deep learning neural networks. Our deep learning neural networks rates achieved on data 1, 2 and 3 are 71.79%, 40% and 98.43%, respectively. Comparing to our method we find that our method outperforms our runs of deep learning neural networks. Note that deep learning yields good results only if a large data set exists. However, for small data as in source 1 or very small as in source 2 the deep learning neural networks method yields very low rates, while our method yields higher rates.

Fig. 4 and Fig. 5 show the performance and time needed for each training and validation process of the deep learning neural networks for the smallest source 2 and the largest source 3 data sets.

One reason for the superiority of the classical system used in our experiments over all other methods is the feature extraction method used. The filtering method of Gabor and LBP catch the changes caused by covid-19 in the lungs. Another reason for the superiority of our method over surveyed deep learning neural networks methods is our feature selection method, which reduces the dimensionality very effectively without losing the discriminating information.

We manually cropped images of data sets 1 and 2, to exclude areas beyond the ribs. Comparing Table 4 of non-cropped images to Table 6 of cropped images we find that LBP always benefits from cropping. However, HOG of Data 1 benefits from cropping and degrades on Data 2. Surf always degrades when we crop images and Gabor mostly degrades. Therefore, the best performance achieved

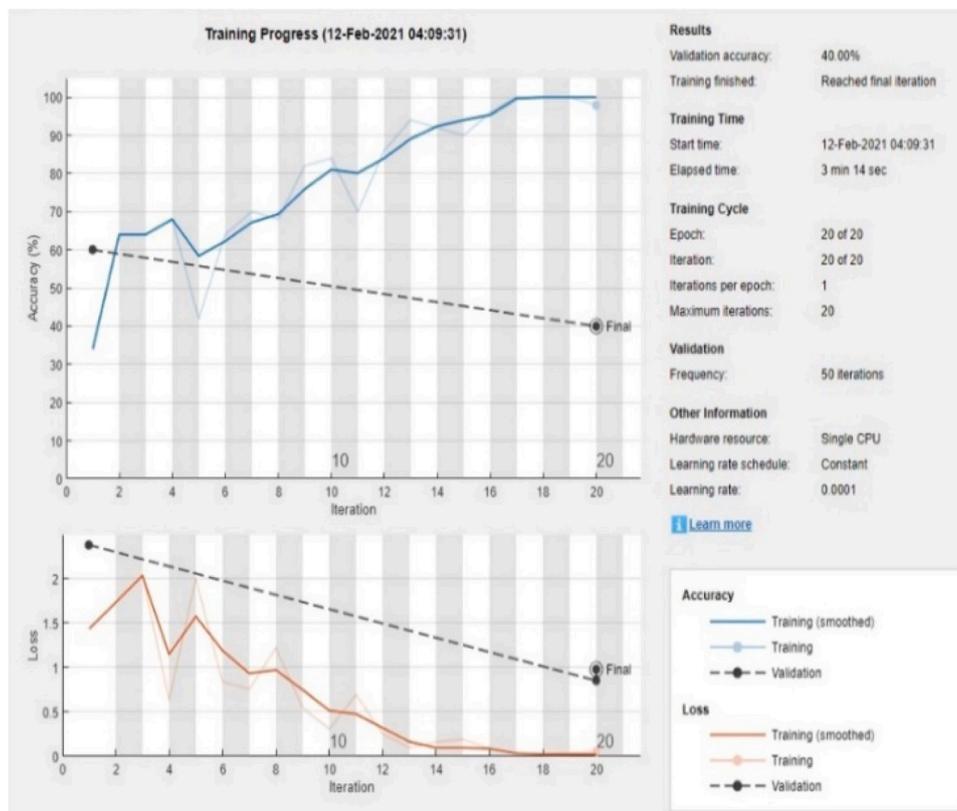


Fig. 4. Deep Learning results for the source 2 data.

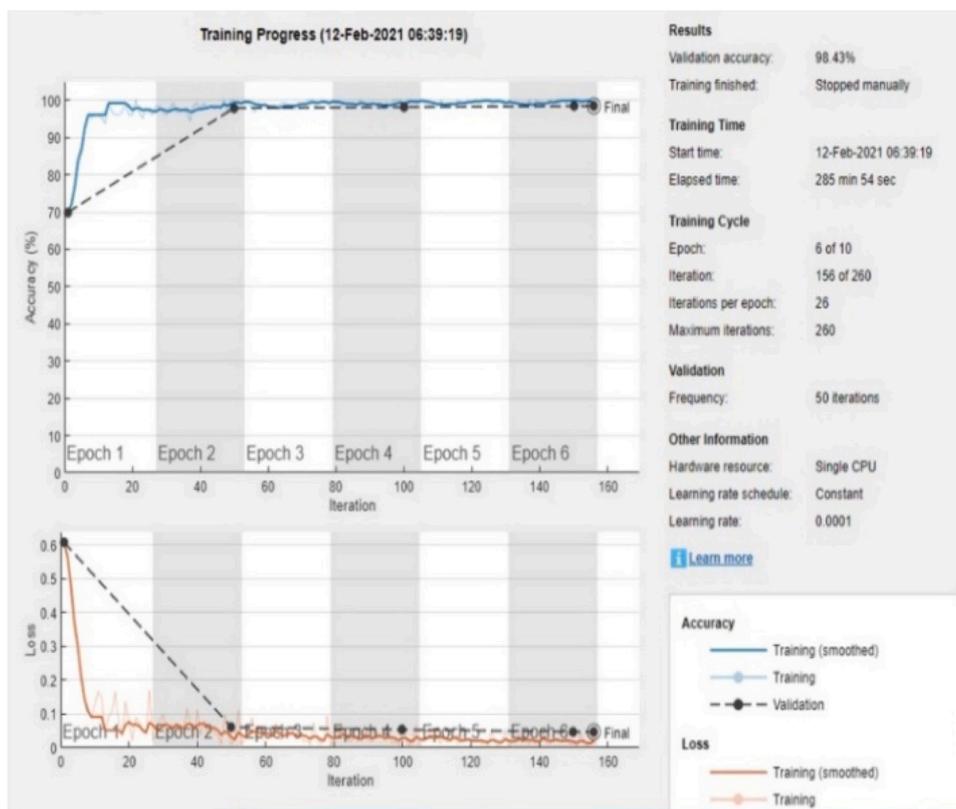


Fig. 5. Deep learning results for the source 3 data.

Table 6
Classification rates using 10-fold cross validation partition of cropped CXR image data.

Classifier		Combiner	HOG	SURF	LBP	Gabor	Gabor & LBP	SURF & LBP	Gabor & SURF	Gabor, SURF & LBP
k-NN	Data 1	Boosting	70.77	63.85	70.90	61.28	59.23	65.26	60.39	62.05
		Arcx4	72.18	64.10	70.26	59.49	59.36	64.62	62.56	61.03
		RSM	73.59	66.28	72.18	60.26	60.13	65.13	60.77	60.90
	Data 2	Bagging	72.31	66.03	69.62	61.28	60.90	65.51	61.15	61.15
		boosting	98.33	70.00	76.67	63.33	70.00	76.67	65.00	63.33
		RSM	98.33	78.33	75.00	73.33	73.33	81.67	75.00	73.33
Neural networks	Data 1	RSM	98.33	76.67	71.67	73.33	73.33	81.67	73.33	73.33
		Bagging	98.33	80.00	70.00	75.00	75.00	83.33	75.00	75.00
		boosting	73.72	58.72	67.56	57.31	69.62	63.33	61.80	65.77
	Data 2	Arcx4	72.95	61.92	67.95	63.85	69.87	63.08	66.41	66.80
		RSM	74.62	67.31	70.00	65.13	69.36	66.92	69.23	68.08
		Bagging	72.05	63.46	66.67	65.26	69.36	63.97	67.82	66.03
	Data 1	boosting	100.	85.00	76.67	51.67	76.67	91.67	80.00	91.67
		Arcx4	100.	88.33	81.67	68.33	73.33	95.00	80.00	90.00
		RSM	100.	81.67	76.67	73.33	76.67	93.33	81.67	88.33
	Data 2	Bagging	100.	85.00	83.33	68.33	75.00	96.67	81.67	91.67

by LBP and neural networks on the largest data set may be improved if cropped images are used. However, due to the large size of the data, manual cropping is not feasible, and an automatic segmentation method must be used.

Conclusion

To find the best automated COVID-19 detection system, we experiment with classical machine learning methods in contrast to deep learning methods. The four stage system consists of a feature extraction stage followed by a feature selection then classification and combiner stages. We experiment with four feature extraction methods of Gabor, SURF, LBP, and HOG. Then, we reduce the number of features using a

fisher-score based clustering measure that is used as a feature selection method. The resulting features are used to build a multiple classifier system of backpropagation neural networks or k-nearest neighbour. Classifiers are combined using one of the four methods of bagging, RSM, ARCx4 boosting and Ada-boosting.

We found that using a suitable feature extraction method we can build an effective classical machine learning method that competes well with the more expensive deep learning neural networks and outperforms most of them. LBP and HOG features yield best results using neural network classifiers, regardless of combiner method used, reaching 99.06% for the largest data set, and 100% for the smallest data set. However, bagging and RSM are more favourable as they are faster to train than boosting based combiners.

Funding

This work was funded by Kuwait Foundation for the Advancement of Science, KFAS. [grant number PN20-13MX-01].

Data Availability

In our experiments three different data sets were used in the form of chest x-ray (CXR) images of covid-19 and non-covid-19 infected lungs in addition to healthy lungs. These images are freely available at github [22-24] and kaggle [24]. In our experiments only frontal CXR images were used.

Appendix A

Histogram of Oriented Gradients, HOG

Histogram of Oriented Gradients was widely recognized after its introduction in 2005 [25,26] for face detection but was introduced earlier as a patent by Robert K. McConnell in 1986. It works by replacing pixels with gradients that show in which direction the grey image is getting darker, by comparing intensity of current pixel with the pixels directly surrounding it. Then we draw an arrow showing in which direction the image is getting darker. This is repeated for all pixels in an image. By considering only the direction that brightness changes we can compare images of different intensities. We can group 16×16 pixels, to reduce the amount of information, and replace their gradients with one that represents the arrow directions that were the strongest. The result is an image that captures the structure of an image.

Local Binary Pattern (LBP)

Local Binary Pattern (LBP) is a simple yet very efficient texture operator which labels the pixels of an image by thresholding the neighborhood of each pixel and considers the result as a binary number. The original LBP operator [25,28] forms labels for the image pixels by thresholding the 3×3 neighborhood of each pixel with the center value and considering the result as a binary number. The histogram of these $2^8 = 256$ different labels can then be used as a texture descriptor. We divide the image into blocks and apply LBP to each block separately. We set the number of divisions of an image to 10. Each application of the LBP to an image, or a block part of an image, will yield a histogram of LBP values of all pixels. We can increase the number of neighbors, between 4 and 24, to encode greater detail around a pixel. We set the number of neighbors to 20. The radius of circular pattern used to select neighbors captures more detail if increased. In our experiments we set the radius at 1.

Speeded-Up Robust Features, SURF

An effective feature extraction algorithm is the speed up robust features method [25,27]. SURF is composed of two steps: Feature Extraction and Feature Description. The SURF detector focuses its attention on blob-like structures in the image. Blobs or interest points are detected by convolving the source image with the determinant of the Hessian (DoH) matrix, which contains different 2-D Gaussian second order derivatives. The first step is to find interest points at different scales to find blobs. Next, we find the orientation of the blob by finding its strongest direction in a circular window. A square region is constructed on the interest point with an orientation as found in the previous step to find and extract the SURF descriptor which indicates the nature of the pixels around the interest point.

Next, the creation of SURF descriptor takes place in two steps. The first step consists of fixing a reproducible orientation based on information from a circular region around the key-point or point of interest. Then, we construct a square region aligned to the selected orientation and extract the SURF descriptor from it.

SURF feature descriptor is based on the sum of the Haar wavelet response around the point of interest. So, the region is split up regularly into smaller 4×4 square sub-regions. For each sub-region, we compute a few simple features at 5×5 regularly spaced sample points. By comparing the descriptors obtained from different images, matching pairs can be found. The matching is carried out as follows. An interest point in the test image is compared to an interest point in the reference image by calculating the Euclidean distance between their descriptor vectors.

GABOR: Gabor features

The Gabor filter [25] is a linear filter used in a lot of image processing applications for edge and object detection, texture analysis and feature extraction. These filters have been shown to possess optimal localization properties in both spatial and frequency domain and thus are well suited for texture segmentation problems. Gabor filters are special classes of band pass filters, i.e., they allow a certain 'band' of frequencies and reject the others. A Gabor filter can be viewed as a sinusoidal signal of particular frequency and orientation, modulated by a Gaussian wave. In practice to analyze texture or obtain features from an image, a bank of Gabor filters with several orientations are used. The patterns in an image are highlighted when the input image is convolved with all the Gabor filters. The result is a high response at edges and at points where texture changes. Therefore, certain features respond to a certain filter. There are many parameters that control how a Gabor filter will be, and which features will it respond to. These parameters are described next.

λ — Wavelength of the sinusoidal component. The wavelength governs the width of the strips of Gabor function. Increasing the wavelength produces thicker stripes and decreasing the wavelength produces thinner stripes.

θ — The orientation of the normal to the parallel stripes of Gabor function. The theta controls the orientation of the Gabor function. The zero-degree theta corresponds to the vertical position of the Gabor function.

γ — The spatial aspect ratio or gamma specifies the ellipticity of the support of Gabor function, or height of the Gabor function. For very high aspect ratio the height becomes very small and for very small gamma value the height becomes quite large.

Declaration of Competing Interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: There is nothing to declare.

Acknowledgement

Authors would like to thank the Kuwait Foundation for the Advancement of Science, KFAS, for their financial support of this project under grant number PN20-13MX-01.

σ — The bandwidth or sigma/standard deviation of the Gaussian envelope. It controls the overall size of the Gabor envelope. For larger bandwidth, the envelope increase allowing more stripes and with small bandwidth the envelope tightens.

In our experiments we set the wavelength in increasing powers of two starting from $\frac{4}{\sqrt{2}}$ up to the hypotenuse length of the input image. We initially find the minimum and maximum wavelengths. Minimum is $\frac{4}{\sqrt{2}}$ which is 2.83, while maximum is the square root of sum of square of number of columns and number of rows in an image. Next, assume n is the integer result from the base 2 logarithm of the maximum wavelength value divided by the minimum wavelength value. The wavelength is the set of values resulting from the following equation:

$$\text{Wavelength} = \text{minimum wavelength value} \times 2^{(0 \text{ to } n-2)}$$

Orientation is a set of values between 0 and 180 with increments of delta theta. We found best results when using delta theta increment of 126 for data 1 and 3 and 90 for data 2. Therefore, gabor orientation will be 0 and 90 for data 2 and will be 0 and 126 for data 1 and 3. Based on the given set of orientation and wavelength values we create a set of gabor filters. Then, we extract Gabor magnitude features from source images using the filters. To use Gabor magnitude responses as features for use in classification, some post-processing is required. This post processing includes Gaussian smoothing, adding additional spatial information to the feature set, reshaping our feature set to the form expected by the pca and kmeans functions, and normalizing the feature information to a common variance and mean. We use simple Gaussian low-pass filtering to smooth the Gabor magnitude information. We choose a sigma that is matched to the Gabor filter that extracted each feature. We use a smoothing term $K=3$ that controls how much smoothing is applied to the Gabor magnitude responses.

References

- [1] M. Christopher, Bishop, Pattern Recognition and Machine Learning, sixth ed. Springer, 2011.
- [2] Kim Kyung-joong, Cho Sung-Bae, An evolutionary Algorithm Approach to optimal ensemble classifiers for DNA microarray data analysis, IEEE Trans. Evolut. Comput. vol 12, (3) (2008).
- [3] M. Dettling, Bagboosting for tumor classification with gene expression data, Bioinformatics vol. 20, (18) (2004) 3583–3593.
- [4] Fuad M. Alkoot, Multimodal biometric authentication using adaptive decision boundaries, Kuwait J. Sci. vol. 41, (3) (2014) 103–128.
- [5] A. Fuad Alkoot, Alqallaf, Investigating machine learning techniques for the detection of autism, Int. J. Data Min. Bioinforma. 2 (2016) 141–169.
- [6] Adam Jacobi, Michael Chung, Adam Bernheim, and Corey Eber. Portable Chest X-Ray in Coronavirus disease 19 (COVID-19): A pictorial review. Elsevier Inc COVID-19 Resource Center, online April 8, 2020.
- [7] S. Candemir, S. Jaeger, K. Palaniappan, J.P. Musco, R.K. Singh, Z. Xue, A. Karargyris, S. Antani, G. Thoma, C.J. McDonald, Lung segmentation in chest radiographs using anatomical atlases with nonrigid registration, IEEE Trans. Med. Imaging 33 (2) (2014) 577–590.
- [8] Souza J.C.1, Bandeira Diniz J.O.2, Ferreira J.L.2, França da Silva G.L.2, Corrêa Silva A.2, de Paiva A.C. An automatic method for lung segmentation and reconstruction in chest X-ray using deep neural networks. Comput Methods Programs Biomed. 2019 Aug;177:285–296. doi: 10.1016/j.cmpb.2019.06.005. Epub 2019 Jun 6.
- [9] Sema Candemir, Sameer Antani, A review on lung boundary detection in chest X-rays, Int. J. Comput. Assist. Radiol. Surg. 14 (2019) 563–576.
- [10] Diletta Cozzi, Marco Albanesi, Edoardo Cavigli, Chiara Moroni, Alessandra Bindi, Silvia Luvàrà, Silvia Lucarini, Simone Busoni, Lorenzo Nicola Mazzoni, Vittorio Miele, Chest X-ray in new Coronavirus Disease 2019 (COVID-19) infection: findings and correlation with clinical outcome, Radio. Med. 125 (2020) 730–737, <https://doi.org/10.1007/s11547-020-01232-9>
- [11] Michael J. Loeffelholz, Yi-Wei Tang, Laboratory diagnosis of emerging human coronavirus infections – the state of the art, Emerg. Microbes Infect. VOL. 9 (2020), <https://doi.org/10.1080/22221751.2020.1745095>
- [12] Yi-Wei Tang, Jonathan E. Schmitz, David H. Persing, Charles W. Stratton, The laboratory diagnosis of COVID-19 infection: current issues and challenges, e00512-20, J. Clin. Microbiol. 58 (6) (2020), <https://doi.org/10.1128/JCM.00512-20>
- [13] Shuai Wang, Bo Kang, Jinlu Ma, Xianjun Zeng, Mingming Xiao, Jia Guo, Mengjiao Cai, Jingyi Yang, Yaodong Li, Xiangfei Meng, Bo Xu, A deep learning algorithm using CT images to screen for Corona Virus Disease (COVID-19), Eur. Radiol. (2021) 1–9, <https://doi.org/10.1007/s00330-021-07715-1>. [published online ahead of print, 2021 Feb 24].
- [14] F. Shi, J. Wang, J. Shi, Z. Wu, Q. Wang, Z. Tang, K. He, Y. Shi, D. Shen, Review of artificial intelligence techniques in imaging data acquisition, segmentation and diagnosis for COVID-19, IEEE Rev. Biomed. Eng. (2020), <https://doi.org/10.1109/RBME.2020.2987975>
- [15] Aytac Altan, Sicken karasu, Recognition of COVID-19 disease from X-ray images by hybrid model consisting of 2D curvelet transform, chaotic salp swarm algorithm and deep learning technique Volume 140 Chaos, Solitons & Fractals, Elsevier Ltd, 2020110071.
- [16] Bejoy Abrahama, Madhu S. Nairb, Computer-aided detection of COVID-19 from X-ray images using multi-CNN and Bayesnet classifier (October–December), Biocybernetics Biomed. Eng. Volume 40 (4) (2020) 1436–1445.
- [17] Dalia Ezzata, Aboul EllaHassaniena, Hassan Aboul Ellab, An optimized deep learning architecture for the diagnosis of COVID-19 disease based on gravitational search optimization, Appl. Soft Comput. Volume 98 (2021) 106742.
- [18] Linda wang, Zhong qiu lin, Alexander Wong, COVID-Net: a tailored deep convolutional neural network design for detection of COVID-19 cases from chest X-ray images, Nat. Res. 10 (2020) 19549.
- [19] Zheng Wang, Ying Xiao, Yong Li, Jie Zhang, Fanggen Lu, Muzhou Hou, Xiaowei Liu, Automatically discriminating and localizing COVID-19 from community-acquired pneumonia on chest X-rays, Pattern Recognit. Volume 110 (2021) 107613ISSN 0031-3203.
- [20] Sergio Varela-Santos, Patricia Melin, A new approach for classifying coronavirus COVID-19 based on its manifestation on chest X-rays using texture features and neural networks, Inf. Sci. Volume 545 (2021) 403–414.
- [21] Tej Bahadur Chandra, Kesari Verma, Bikesh Kumar Singh, Deepak Jain, Satyabhuwan Singh Netam, Coronavirus disease (COVID-19) detection in Chest X-Ray images using majority voting based classifier ensemble, Expert Syst. Appl. Volume 165 (2021) 113909ISSN 0957-4174.
- [22] Cohen COVID-19 images at GitHub, <https://github.com/ieee8023/covid-chestxray-database>. <<https://github.com/ieee8023/covid-chestxray-database>>. Accessed 2/2/2021.
- [23] x-ray images of covid, non-covid pneumonia and normal samples. <<https://github.com/agchung/Figure1-COVID-chestxray-dataset>>. Accessed 2/2/2021.
- [24] Covid-19 Radiography database 1st update of 1st release, Covid-19 chest x-ray database, <<https://www.kaggle.com/tawsifurrahman/covid19-radiography-database>>. Accessed 2/2/2021.
- [25] Simon Prince, Computer Vision: Models, Learning, and Inference, first ed Cambridge University Press, 2012.
- [26] Navneet Dalal and Bill Triggs. Histograms of Oriented Gradients for Human Detection. CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), June 2005, Volume 1, Pages 886–.
- [27] Herbert Bay, Tinne Tuytelaars, Luc Van Gool, SURF: speeded up robust features, Eur. Conf. Comput. Vis. ECCV (2006) 404–417.
- [28] T. Ojala, M. Pietikäinen, D. Harwood, A comparative study of texture measures with classification based on feature distributions, Pattern Recognit. 19 (3) (1996) 51–59.
- [29] L. Breiman, Bagging predictors, Mach. Learn. 24 (1996) 123–140.
- [30] T. Ho, The random subspace method for constructing decision forests, IEEE Trans. Pattern Anal. Mach. Intell. 20 (8) (1998) 832–844.
- [31] Yoav Freund, Robert E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, J. Comput. Syst. Sci. 55 (1) (1997) 119–139.
- [32] Yoav Freund, Robert E. Schapire, A short introduction to boosting, J. Jpn. Soc. Artif. Intell. 14 (5) (1999) 771–780.
- [33] Leo Breiman, Arcing classifiers, Ann. Stat. vol. 26, (3) (1998) 801–824 (JSTOR).