Covid-19 Diagnosis by Gray-level Cooccurrence Matrix and Genetic Algorithm

Xiaoyan Jiang¹*, Zuojin Hu¹, Mackenzie Brown², Hei-Ran Cheong³

¹School of Mathematics Information Science, Nanjing Normal University of Special Education, Nanjing 210038, China
²School of Engineering, Edith Cowan University, Joondalup WA 6027, Australia
³Department Civil & Environment Engineering, University of Ulsan, Korea, Daehak Ro 93, Ulsan 680749, South Korea

Abstract

Currently, improving the identification of COVID-19 with the help of computer vision and artificial intelligence has received great attention from researchers. This paper proposes a novel method for automatic detection of COVID-19 based on chest CT to help radiologists improve the speed and reliability of tests for diagnosing COVID-19. Our algorithm is a hybrid approach based on the Gray-level Cooccurrence Matrix and Genetic Algorithm. The Gray-level Cooccurrence Matrix (GLCM) was used to extract CT scan image features, GA algorithm was used as an optimizer, and a feedforward neural network was used as a classifier. Finally, we use 296 chest CT scan images to evaluate the detection performance of our proposed method. To more accurately evaluate the accuracy of the algorithm, 10-run 10-fold cross-validation was introduced. Experimental results show that our proposed method outperforms state-of-the-art methods in terms of Sensitivity, Accuracy, F1, MCC, and FMI.

Keywords: Gray-level Cooccurrence Matrix, Genetic Algorithm, optimization, Feedforward Neural Network, K-fold cross-validation, COVID-19, Diagnosis.

1. Introduction

Corona Virus Disease 2019 (COVID-19) is a serious and highly contagious disease caused by the novel coronavirus discovered in 2019 [1, 2]. As of May 17, 2022, there have been 519,729,804 confirmed cases of COVID-19 in the world and 6,268,281 deaths, according to the WHO real-time statistics. The number of confirmed cases and deaths continues to rise, the virus continues to mutate, and the global epidemic is far from over [3, 4].

According to the existing case data, Covid-19 is mainly manifested by fever, dry cough, fatigue, etc., and a small number of patients have upper respiratory and gastrointestinal symptoms such as nasal congestion, runny nose, and diarrhea [5-7]. Severe cases develop dyspnea, and more severe cases rapidly progress to acute respiratory distress syndrome [8], septic shock, difficult-to-correct metabolic acidosis, coagulation dysfunction, and multiple organ failure [9,10].

It is self-evident that early detection, early diagnosis, and early treatment can significantly reduce the incidence and mortality of critical illness in patients with COVID-19 [11]. CT examination plays an important role in the diagnosis of COVID-19, and was once used as the main basis for clinical diagnosis in major epidemic areas [12]. However, conventional CT examination has some shortcomings, such as it is difficult to observe relatively hidden lesions in the early stage, and it is difficult to distinguish it from other viral pneumonia and bacterial pneumonia.

Deep learning [13-15] is a combination of machine learning methods whose algorithms specify features...
through a series of non-linear functions that are organized in combination to maximize model accuracy. Deep learning mainly focuses on automatic feature extraction and classification of images. In healthcare, deep learning is used to efficiently generate models, resulting in more accurate results when using images to predict and classify different diseases without any human intervention.

Recently, deep learning methods have been widely applied to diagnose COVID-19 based on CT images, and studies have demonstrated that these methods can reliably extract feature information from chest CT images [16-18]. For example, Ascel Qassim Abdul Ameer et al. [19] designed a CT scan detection system for Covid-19 based on Gray-level Cooccurrence Matrix, which can detect the location of infection and determine whether the lungs are infected with 94% accuracy. Lu Xu et al. [20] developed three deep learning models, CNN, LSTM, and CNN, to predict the number of COVID-19 cases, and successfully predicted the spread trend in Brazil, India, and Russia. Khabir Uddin Ahamed et al. [21] developed a deep learning-based COVID-19 case detection model trained on a dataset consisting of chest CT scans and X-ray images, helping radiologists use basic but widely available equipment to rapid diagnosis of COVID-19 cases. In [22], The authors focused on summarizing deep learning methods that have been significantly utilized in the automatic identification of COVID-19 cases from other lung diseases and normal populations while also discussing the challenges associated with current DL methods for COVID-19 diagnosis. For example, most datasets for binary or multi-class classification for COVID-19 diagnosis are highly unbalanced, medical images are often low-contrast image, etc.

In recent years, radial basis function neural network (RBFNN) [23], kernel-based extreme learning machine (K-ELM) [24], extreme learning machine with bat algorithm (ELM-BA) [25], etc. have been applied to COVID-19 detection. However, in the process of implementing the training algorithms of the classifiers, in the process of implementing the classifier training algorithm, they all face the situation of falling into local optimization or failing to achieve global optimization [26]. To solve the above issues, we propose a method to diagnose COVID-19 by using Gray-level Cooccurrence Matrix and Genetic Algorithm. This research is expected to help clinicians achieve more precise judgments in the diagnosis of COVID-19 patients.

In this study, the gray-level co-occurrence matrix is used to extract features from CT images of COVID-19 cases [27]. We use genetic algorithms to find optimal solutions, and cross-validation is used to test the performance of the algorithms [28].

The rest of this paper is organized as follows: In section 2, we briefly introduce the data sets of this research. In section 3, we first describe the basic concepts of the Gray-level Cooccurrence Matrix (GLCM) and its four typical statistical features. Next, the Feedforward Neural Network (FNN) and Genetic Algorithm (GA) is described, and K-fold cross-validation is given. In Section 4, we describe the experimental details and a discussion of the results. The conclusion of this research is described in Section 5.

2 Dataset

In our experiments, the dataset was divided into two subsets, 148 CT images of chest scans of patients diagnosed with COVID-19 and 148 CT images of chest scans of healthy subjects, for a total of 296 images. Among them, there were 66 COVID-19 patients (41 males and 25 females) and 66 healthy subjects (31 males and 35 females). These healthy subjects were randomly selected from the healthy population. In the experiments, we focused on the identification of COVID-19 by chest CT. For COVID-19 patients, the layer with the largest lesion characteristics was selected, while for healthy subjects, the image level was chosen arbitrarily. Figure 1 shows sample CT scan images of COVID-19 patients, and healthy controls, where (a) is a CT scan image sample of a COVID-19 patient and (b) is a CT scan image sample of healthy control (HC).

![Figure 1. Comparison of CT scan image samples of COVID-19 patients and healthy controls.](https://example.com/figure1.png)

3 Methodology

3.1 GLCM

Gray-level Cooccurrence Matrix is a square matrix that can describe texture features by calculating the spatial correlation of image gray[29]. The GLCM texture considers the spatial relationship between two pixels at a specific orientation angle and distance, which reflects the two-dimensional statistical characteristics of image texture [30]. We must specify the offset value and direction as the gray-level co-occurrence matrix is constructed. When the offset value is 1, there are four directions: 0°, 45°, 90° and 135°, and there is a gray-level co-occurrence matrix in each direction [29]. Then, the element value of the gray-level co-occurrence matrix is the number of times that the two gray values
represented by rows and columns in the image appear together in the specified direction and offset value. There are 14 statistical characteristics of GLCM used to reflect the spatial correlation characteristics of images. Four of these characteristics are commonly correlated to texture, including energy, entropy, contrast, and homogeneity [31].

**Energy** (Equation (1)) [32] is the sum of squared of each element in the GLCM. Energy is a measure of the stability of the gray level change of the image texture, and reflects the uniformity of the image gray level distribution and the thickness of the texture.

\[
Energy = \sum_{p,q} (C_{a}(p,q))^2
\]  

Here, \( p \) and \( q \) are different element values of gray-level co-occurrence matrix, \( C_d(p,q) \) represents the \((p,q)\)th entry in a normalised grey-tone spatial dependence matrix which is the sum of the number of times that pixel value \( p \) was some distance and angle \( a \) from pixel intensity \( q \).

**Entropy** (Equation (2)) is a measure of the randomness of the amount of information contained in an image. When all values in the co-occurrence matrix are equal, or the pixel values show the greatest randomness, the entropy is the largest. Therefore, entropy indicates the complexity of image gray distribution. The greater the entropy, the more complex the image is.

\[
Entropy = \sum_{p,q} C_{a}(p,q) \cdot \log(C_{a}(p,q))
\]  

**Contrast** (Equation (3)) [30] reflects the contrast of the brightness between one pixel and its neighbor in relative location \( a \).

\[
Contrast = \sum_{p,q} |p - q|^2 C_{a}(p,q)
\]

**Homogeneity** (Equation (4)) [32] compares the distribution of the values on the diagonal of the GLCM to the distribution of the values off the diagonal.

\[
Homogeneity = \sum_{p,q} \frac{C_{a}(p,q)}{1 + |p - q|}
\]

### 3.2 Feedforward Neural Network

Feedforward neural network (FNN) is a directed acyclic graph that consists of input, hidden, and output neuron layers [33]. Layer 0, the layer that provides input Figure 2[41]. In the figure, \( t \) refers to the evolutionary algebra counter.

![Figure 2. Diagram of a genetic algorithm.](image)

variables to the network is called the input layer, the last layer is called the output layer, and other intermediate layers are considered hidden layers. In this neural network [34-36], each layer contains several neurons that can receive the signal of the previous layer of neurons and produce output to the next layer [26]. Any two nodes of two adjacent layers are connected. In other words, layers are fully connected.

In a feedforward neural network, the output of each neuron in the upper layer is used as input, and the output of this node is obtained through linear transformation (combined with bias) and non-linear function activation and is passed to the node in the next layer. The activation function gives the neural network the ability to “piecewise function” and can approximate any function. Without an activation function, a neural network is just a linear function and cannot solve any non-linear problems. ReLU is an activation function commonly used in artificial neural networks (Equation (5)) [33].

\[
ReLU(m) = \begin{cases} 
m & m \geq 0 \\
0 & m < 0 
\end{cases}
\]

Here, \( m \) is the input variable. When FNN is trained, it is necessary to find the most appropriate parameter weights and biases to best approximate the data. Usually, a loss function is designed to measure the approximation effect, and the optimal parameters should minimize the loss function. The cross-entropy loss function is the most commonly used in classification problems [37-39].

### 3.3 Genetic Algorithm

A genetic Algorithm (GA) is a computational model that simulates the natural selection and genetic mechanism of Darwin’s theory of biological evolution, and it is a method to search for optimal solutions by simulating the natural evolution process [40]. The genetic algorithm does not directly deal with the decision variables of the solution space, but converts them into chromosomes composed of genes according to a certain structure through coding. The typical steps of genetic algorithm are: initialization, selection operation, crossover operation, mutation operation and termination condition judgment. The basic operation process of a typical genetic algorithm is shown in
3.3.1. Initial population
The process starts with a group of individuals called a population, each of which is a proposed solution to a defined problem. Individuals are also called chromosomes. A chromosome consists of a certain number of genes. The smallest taxonomic unit in the genetic algorithm is a gene. The generation of the initial population is random [42, 43]. Before the initial population is assigned, try to make a rough interval estimation to avoid the initial population being distributed in the coding space far from the optimal solution, which will limit the search range of the genetic algorithm. If the size of the group is set too small [44, 45], it may lead to premature convergence, and eventually the global optimal solution cannot be obtained; and if the size of the group is set too large, it will affect the calculation time. Therefore, the size of the population should be determined according to the actual problem. In the initial stage, the evolutionary algebra counter $t$ is set to 0, and the maximum evolutionary algebra is set.

3.3.2. Selection operation
In this step, individuals who are more adapted to the environment need to be selected from the population. The purpose is to directly inherit the optimized individuals to the next generation or to generate new individuals through pairing and crossover to the next generation, so this operation is also called regeneration. The selection operation is based on the fitness evaluation of individuals in the group, that is, the probability of an individual being selected is related to the fitness value. The higher the individual fitness value, the greater the probability of being selected.

In genetic algorithm, roulette wheel selection and elitism selection are two typical selection methods. Taking the roulette wheel selection as an example, assuming that the size of the current population is $M$, and the fitness of individual $i$ is $f_i$, the probability $p_i$ (Equation (6)) of individual $i$ being selected is:

$$p_i = \frac{f_i}{\sum_{n=1}^{M} f_n}$$  \hspace{1cm} (6)

Here, the denominator refers to the total fitness of the population.

3.3.3. Crossover operation
In the crossover operation, two individuals are randomly selected from the population, and through the exchange and combination of two chromosomes, the excellent characteristics of the parent string are inherited to the substring, thereby generating new excellent individuals.

3.3.4. Mutation operation
In the search process, in order to maintain the diversity of the population and prevent the genetic algorithm from prematurely converging in the optimization process and falling into a local optimal solution, it is necessary to mutate the individual, thereby changing the initial value of one or more genes in the chromosome [46, 47]. In practical applications, single-point mutation, also known as bit mutation, is mainly used, that is, only a certain bit in the gene sequence needs to be mutated. Take binary coding as an example, that is, 0 becomes 1, and 1 becomes 0.

3.3.5. Termination operation
When the fitness of the optimal individual reaches a given threshold, or the fitness of the optimal individual and the fitness of the group no longer rise, or the number of iterations reaches a preset number of generations, the algorithm terminates. Otherwise, replace the previous generation population with the new generation population obtained through selection, crossover, and mutation, and return to the selection operation to continue the loop execution.

3.4 K-fold cross validation
In the case of insufficient sample size, in order to make full use of the data set to test the effect of the algorithm, cross validation is introduced. Cross validation is a commonly used method when building models and validating model parameters in machine learning. In K-fold cross validation, a given data set is chunked into $K$ parts in equal proportions, and one of them is used as test data, and the other $K-1$ data is used as training data. Afterward, $K$ experiments were operated. In each experiment, a different data part is selected from the $K$ parts as the test data, and the remaining $K-1$ parts are used as the training data, and the corresponding error rate is obtained. Finally, the error rate of the obtained $K$ experimental results is averaged, so as to estimate the accuracy of the algorithm. Figure 3 below shows the sequence of these operations. In $K$-fold cross validation, it should be noted that the data of $K$ parts must be tested separately.
Covid-19 Diagnosis by Gray-level Cooccurrence Matrix and Genetic Algorithm

\[ E = \frac{1}{K} \sum_{i=1}^{K} E_i \]  

(7)

In Equation (7), \( i \) is the number of folds (\( i \) loops from 1 to \( K \)), \( E_i \) is the \( i \)th misclassification rate, and \( E \) is the average of \( K \) misclassification rates.

There may be bias or variance in the cross-validation. If \( K \) is increased, the variance will go up but the bias may go down [48-50]. Conversely, \( K \) is lowered, the bias may rise but the variance will fall. In general, the value of \( K \) depends on the situation of different projects. Of course, there must be \( K \) is less than the number of data pieces whose the training set [51].

4 Experiment Results and Discussions

4.1 GLCM Results

In the study, GLCM extracts the texture features of CT images from four directions (0°, 45°, 90° and 135°), and generates four matrices for each image. When the distance between the current pixel and its neighbor is set to 1, the angle can be set:

- Offset=[0 1]: GLCM at 0°.
- Offset=[-1 0]: GLCM at 90°.
- Offset=[-1 -1]: GLCM at 135°.

\[ \text{Figure 3. The main process of K-fold cross-validation.} \]

\[ \text{Figure 4. GLCM values in four different directions.} \]
In the experiment, four $6 \times 6$ gray-level co-occurrence matrices in different directions are generated according to Figure 1(a), as shown in Figure 4. We use the obtained matrix to calculate features. Each gray-level co-occurrence matrix can be described by 4 features: energy, entropy, contrast, and homogeneity, and a total of 16 features are obtained and sent to the feedforward neural network.

### 4.2 Statistical Results

In the experiments, 10-Fold CV is introduced, meanwhile to more accurately evaluate the accuracy of the algorithm, we run 10-Fold CV 10 times. We evaluated the algorithm proposed in this study through 7 evaluation indicators: Sensitivity, Specificity, Precision, Accuracy, F1, Matthew’s Correlation Coefficient (MCC), and Fowlkes–Mallows index (FMI). Here, F1 refers to F1-score, which is used as an evaluation standard to measure the comprehensive performance of the classifier [26]. The evaluation results are shown in Table 1. It can be seen from the evaluation results that the performance of these metrics is significantly better than that of traditional methods.

**Table 1. Statistics of experimental results.**

<table>
<thead>
<tr>
<th>Run</th>
<th>Sensitivity</th>
<th>Specificity</th>
<th>Precision</th>
<th>Accuracy</th>
<th>F1</th>
<th>MCC</th>
<th>FMI</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>75.68</td>
<td>72.30</td>
<td>73.20</td>
<td>73.99</td>
<td>74.42</td>
<td>48.00</td>
<td>74.43</td>
</tr>
<tr>
<td>2</td>
<td>68.24</td>
<td>76.35</td>
<td>74.26</td>
<td>72.30</td>
<td>71.13</td>
<td>44.74</td>
<td>71.19</td>
</tr>
<tr>
<td>3</td>
<td>71.62</td>
<td>71.62</td>
<td>71.62</td>
<td>71.62</td>
<td>71.62</td>
<td>43.24</td>
<td>71.62</td>
</tr>
<tr>
<td>4</td>
<td>75.68</td>
<td>71.62</td>
<td>72.73</td>
<td>73.65</td>
<td>74.17</td>
<td>47.34</td>
<td>74.19</td>
</tr>
<tr>
<td>5</td>
<td>78.38</td>
<td>70.27</td>
<td>72.50</td>
<td>74.32</td>
<td>75.32</td>
<td>48.81</td>
<td>75.38</td>
</tr>
<tr>
<td>6</td>
<td>74.32</td>
<td>74.32</td>
<td>74.32</td>
<td>74.32</td>
<td>74.32</td>
<td>48.65</td>
<td>74.32</td>
</tr>
<tr>
<td>7</td>
<td>77.70</td>
<td>70.27</td>
<td>72.33</td>
<td>73.99</td>
<td>74.92</td>
<td>48.11</td>
<td>74.97</td>
</tr>
<tr>
<td>8</td>
<td>77.70</td>
<td>68.24</td>
<td>70.99</td>
<td>72.97</td>
<td>74.19</td>
<td>46.15</td>
<td>74.27</td>
</tr>
<tr>
<td>9</td>
<td>74.32</td>
<td>73.65</td>
<td>73.83</td>
<td>73.99</td>
<td>74.07</td>
<td>47.97</td>
<td>74.07</td>
</tr>
<tr>
<td>10</td>
<td>68.92</td>
<td>74.32</td>
<td>72.86</td>
<td>71.62</td>
<td>70.83</td>
<td>43.31</td>
<td>70.86</td>
</tr>
<tr>
<td>MSD</td>
<td>74.26</td>
<td>72.30</td>
<td>72.86</td>
<td>73.28</td>
<td>73.50</td>
<td>46.63</td>
<td>73.53</td>
</tr>
<tr>
<td></td>
<td>±3.60</td>
<td>±2.40</td>
<td>±1.09</td>
<td>±1.07</td>
<td>±1.65</td>
<td>±2.15</td>
<td>±1.65</td>
</tr>
</tbody>
</table>

Table 2 and

**Figure 5.** In terms of Sensitivity, Accuracy, F1, MCC, and FMI for detecting COVID-19, the best is GLCM-GA (Ours), followed by RBFNN, the third is ELM-BA, and the last is K-ELM. In the detection of Sensitivity for COVID-19, the best is GLCM-GA (Ours),
followed by RBFNN, the third is K-ELM, and the last is ELM-BA. In the Precision tests for COVID-19, the best is RBFNN, followed by GLCM-GA (Ours), the third is ELM-BA, and the last is K-ELM. In the Specificity tests for COVID-19, the best is RBFNN, followed by ELM-BA, the third is GLCM-GA (Ours), and the last is K-ELM.

Through the above analysis, it is found that the method of GLCM-GA (Ours) proposed by us had the best performance among the five detection indicators of COVID-19, although it is slightly lower than other methods in terms of specificity and precision.

Table 2. Comparison with state-of-the-art methods for detecting COVID-19.

<table>
<thead>
<tr>
<th>Method</th>
<th>Sensitivity</th>
<th>Specificity</th>
<th>Precision</th>
<th>Accuracy</th>
<th>F1</th>
<th>MCC</th>
<th>FMI</th>
</tr>
</thead>
<tbody>
<tr>
<td>RBFNN [23]</td>
<td>66.89 ±2.43</td>
<td>75.47 ±2.53</td>
<td>73.23 ±1.48</td>
<td>71.18 ±1.48</td>
<td>69.88 ±0.80</td>
<td>42.56 ±1.08</td>
<td>69.97 ±1.04</td>
</tr>
<tr>
<td>K-ELM [24]</td>
<td>58.78 ±3.36</td>
<td>60.27 ±3.89</td>
<td>59.73 ±1.68</td>
<td>59.53 ±1.42</td>
<td>59.19 ±1.86</td>
<td>33.46 ±1.79</td>
<td>59.22 ±1.79</td>
</tr>
<tr>
<td>ELM-BA [25]</td>
<td>56.55 ±2.99</td>
<td>76.22 ±2.56</td>
<td>70.45 ±1.61</td>
<td>66.39 ±1.05</td>
<td>62.69 ±2.63</td>
<td>33.46 ±1.60</td>
<td>63.09 ±1.60</td>
</tr>
<tr>
<td>GLCM-GA (Ours)</td>
<td>74.26 ±3.60</td>
<td>72.30 ±2.40</td>
<td>72.86 ±1.09</td>
<td>73.28 ±1.07</td>
<td>73.50 ±1.65</td>
<td>46.63 ±1.65</td>
<td>73.53 ±1.65</td>
</tr>
</tbody>
</table>

Figure 5. Bar plot of comparison results.

5 Conclusions

In this study, we propose a hybrid diagnosis method based on the Gray-level Cooccurrence Matrix and Genetic Algorithm for COVID-19. The method consists of three parts: GLCM as feature extractor, FNN as classifier, and GA algorithm as optimizer. In the experiment, 10-run 10-fold cross validation was introduced. Compared with the existing global optimization methods (See Section 4.3), the GLCM-GA method proposed in this paper has better performance than other methods in terms of Sensitivity, Accuracy, F1, MCC and FMI.

The proposed method still has some shortcomings that need to be improved. For example, our method is not optimal in terms of specificity and accuracy, and we will improve the GLCM-GA method. Moreover, the dataset used in the experiment is small, containing only 296 images, so we will expand the dataset to test our method by acquiring multiple CT scan images.

In future work, we will attempt to combine other deep learning algorithms to improve the reliability of automatic diagnosis of COVID-19. In the future, we will also differentiate between COVID-19 and other viral diseases, such as common pneumonia, based on computer vision and deep learning.
References


