

# Early prediction of coronavirus disease epidemic severity in the contiguous United States based on deep learning

I-Hsi Kao, Jau-Woei Perng<sup>\*</sup>

*Department of Mechanical and Electro-Mechanical Engineering, National Sun Yat-sen University, Kaohsiung 804, Taiwan*

## ARTICLE INFO

**Keywords:**  
 Autoencoder  
 Convolutional neural networks  
 Severe acute respiratory syndrome coronavirus 2  
 Coronavirus disease  
 Deep learning  
 Long short-term memory  
 Machine learning  
 Prediction  
 Transmission  
 Epidemic

## ABSTRACT

In November 2019, the coronavirus disease outbreak began, caused by the novel severe acute respiratory syndrome coronavirus 2. In just over two months, the unprecedented rapid spread resulted in more than 10,000 confirmed cases worldwide. This study predicted the infectious spread of coronavirus disease in the contiguous United States using a convolutional autoencoder with long short-term memory and compared its predictive performance with that of the convolutional autoencoder without long short-term memory. The epidemic data were obtained from the World Health Organization and the US Centers for Disease Control and Prevention from January 1st to April 6th, 2020. We used data from the first 366,607 confirmed cases in the United States. In this study, the data from the Centers for Disease Control and Prevention were gridded by latitude and longitude and the grids were categorized into six epidemic levels based on the number of confirmed cases. The input of the convolutional autoencoder with long short-term memory was the distribution of confirmed cases 14 days before, whereas the output was the distribution of confirmed cases 7 days after the date of testing. The mean square error in this model was 1.664, the peak signal-to-noise ratio was 55.699, and the structural similarity index was 0.99, which were better than those of the corresponding results of the convolutional autoencoder. These results showed that the convolutional autoencoder with long short-term memory effectively and reliably predicted the spread of infectious disease in the contiguous United States.

## Introduction

In November 2019, the first case of unexplained pneumonia was recorded in Wuhan, China, which soon resulted in the coronavirus disease (COVID-19) pandemic caused by the novel severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) [1]. In just over four months, its unprecedented and rapid worldwide spread affected more than 3 million people. Although the World Health Organization (WHO) declared the COVID-19 outbreak a public health emergency of international concern as early as January 2020 [2], the outbreak did not become serious in the United States until March. On March 10th, thousands of confirmed cases were recorded, and all states reported confirmed cases by March 17th. More than 10,000 confirmed cases nationwide were reported by March 19th and thousands of deaths by March 26th. Subsequently, on March 27th, the numbers surpassed the public figures of China and Italy and the US became the country with the worst outbreak. COVID-19 is deadly and highly infectious. Prediction of the number and distribution of cases can allow the government to allocate medical resources in advance or to pre-deploy personnel to areas predicted to be strongly impacted by the

pandemic. The rapid spread of the COVID-19 pandemic may be caused by a variety of factors, including the lack of information and transparency in the early stages of the pandemic. The timely and accurate release of epidemic data is extremely important for epidemic containment measures.

In a recent study, Chinazzi et al. used a model of global metapopulation disease transmission to project the impact of travel limitations on the national and international transmission of COVID-19 [3]. The authors found that even following the travel ban in Wuhan, which was initiated on January 23rd, 2020, most Chinese cities received several infected travelers. While the quarantine of Wuhan delayed the epidemic in mainland China by three to five days, the impact at the international level was more pronounced. By mid-February, the number of cases in China decreased by nearly 80%. The authors also highlighted that early detection, hand washing, self-isolation, and family isolation may have been more effective than travel restrictions in alleviating this pandemic.

Kraemer et al. used human mobility data to explain the spatial distribution of COVID-19 cases in China [4]. Following the implementation

\* Corresponding author.

E-mail addresses: [ihkao@berkeley.edu](mailto:ihkao@berkeley.edu) (I.-H. Kao), [jwperng@faculty.nsysu.edu.tw](mailto:jwperng@faculty.nsysu.edu.tw) (J.-W. Perng).

of control measures, this correlation between human mobility data and the spatial distribution of COVID-19 cases declined, with negative growth rates in most regions. Despite the demographic changes in the reported figures, they still indicate the presence of a chain of transmission outside Wuhan. The authors showed that strict control measures implemented in China greatly reduced the spread of COVID-19.

In epidemiology and medical fields, mathematical modeling is a well-established method of prediction [5–7]. The Susceptible-Infected-Removed (SIR) model is one of the most widely used model [8] and has evolved to include many complex variations. Among them, are some SIR models that consider the age of the population and the prevalence of cases to improve the accuracy of the model [9]. The deformation of the SIR model has been proven by various studies to improve the effectiveness of the prediction [9–12]. Some researchers have combined dynamic models with machine learning methods to achieve breakthroughs in infectious disease prediction [13–15].

Another recent study used the SIR model to predict the COVID-19 outbreak in mainland China [16]. The results of the study showed that in the initial research, because of the lack of large-scale reliable data, it was difficult to perform effective predictions using SIR models. In the absence of epidemiological data, it is difficult to objectively determine all parameters. Too many unknown parameters cause great uncertainty in model prediction. Therefore, the authors developed a simple SIR model and proved that it could effectively predict the infectivity rate of severe acute respiratory syndrome (SARS) and further used the model to predict the infectivity rate of COVID-19. However, the author also concluded that in the complex SIR model, the data provided to the model should be more accurate. Therefore, unreliable data were manually deleted to increase the reliability of the prediction model. The author reported that the more complex the model, the more accurate the prediction. However, unreasonable data may result in adverse results. By eliminating unreliable data via objective analysis, the author provided epidemic predictions under different scenarios with respect to different-level anti-epidemic measures and medical care represented by the two model parameters: infection rate and removal rate. The predictions were intended to guide decision-making in coping with ongoing SARS-CoV-2 transmission in China.

However, even if the SIR model is used to predict the infectivity of COVID-19, manual intervention remains an issue. It is undesirable to manually adjust the parameters or eliminate unreliable data. In particular, when researchers do not understand local epidemic prevention policies or data substitution, it is difficult to select data accurately. Deep learning can reduce the manual manipulation of data as, in this method, the importance of each dataset is automatically managed by neural networks. Several studies have shown that deep learning can solve previously challenging issues in the fields of engineering [17], manufacturing [18–20], transportation [21–23], agriculture [24–26], and medicine [27–30].

In deep learning, a variety of different architectures have been developed, such as convolutional neural networks (CNNs) [31] that are commonly used in image processing, recurrent neural networks (RNNs) [32] that are commonly used in time series, and long short-term memory (LSTM) [33] that are an improvement on RNNs.

In their study, Zhang et al. collected viral RNA sequences from the Global Initiative on Sharing All Influenza Data (GISAID) and applied deep learning methods to avoid molecular dynamics simulation [34]. The authors presented a CNN structure to identify potential drugs to inhibit SARS-CoV-2 protease by performing drug screening in four chemical compound databases, showing that the deep learning model was more accurate and suitable to ensure binding tests.

Zhu et al. also used deep learning to predict the host and infectivity rates of COVID-19 [35]. The authors introduced a virus host prediction method based on the fact that COVID-19 has a similar infectivity as other human coronaviruses. According to their prediction model, SARS-CoV-2 exhibited more similarity with the bat coronavirus than with coronaviruses infecting other vertebrates. Additionally, by comparing the

infectivity patterns of all virus hosts among vertebrates, mink viruses showed the closest host infectivity pattern to SARS-CoV-2.

Fong et al. reported the results of a study using composite Monte Carlo simulations through deep learning networks that focused on solving the problem of incomplete data available for stochastic decision making during the COVID-19 pandemic. Instead of applying the simplistic and uniform assumptions for a Monte Carlo simulation, which is a common practice, a deep learning-based composite Monte Carlo simulation was used in conjunction with fuzzy rule induction techniques [36].

Currently, several deep learning methods are being used to study COVID-19. Sajadi et al. used ERA5 to conduct climate analysis on areas severely infected with COVID-19 [37]. The results showed that, under limited latitudes, temperatures, and humidity, the distribution of major community outbreaks was consistent with the behavior of seasonal respiratory viruses. In addition, the authors proposed a simplified model that showed an increased risk of COVID-19 propagation in the area. Using weather models, it is possible to predict high-risk areas likely to be severely affected by community COVID-19 outbreaks in the next few weeks to allow public health authorities to focus on surveillance and containment in these areas.

Therefore, climate is believed to affect COVID-19 transmission. The present study proposes a model based on the latitude and longitude of an area to predict the severity of the epidemic. Regardless of the traditional array processing method, the latitude and longitude data processing method can retain more data and reduce data dimensionality. Feature compression via latitude and longitude can largely preserve the climate information. After the data were pre-processed, the prediction of infectivity was conducted through a deep learning architecture proposed in this study, which was a hybrid architecture including CNN, LSTM, and an autoencoder; the convolutional autoencoder with long short-term memory is termed AL-CNN. The results showed that the AL-CNN effectively predicted the infectious disease spread of COVID-19. Previous studies have generally only predicted the total number of infected individuals; relatively few have predicted the number of infected individuals in each area.

The remainder of this paper is organized as follows. In Section II, the experimental data from the WHO and US Centers for Disease Control and Prevention (CDC) are presented, along with data pre-processing. Section III presents the deep learning architectures, including AL-CNN and the convolutional autoencoder (CAE). Section IV presents the experimental results, along with a comparison of AL-CNN and CAE. Feature extraction was compared between the AL-CNN and CAE. The computational system is also discussed in this section. Section V presents the conclusions and discusses topics for future research.

## Data & pre-processing

### Data

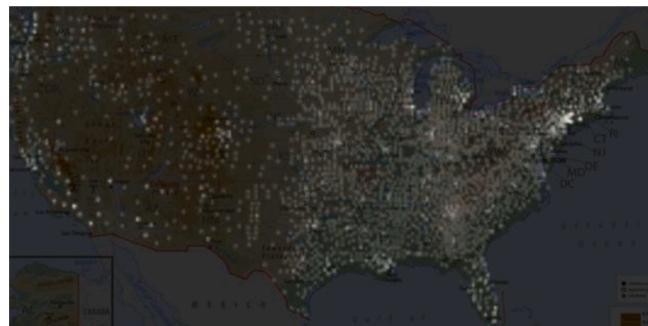
The data used in this study were collected by the Johns Hopkins University Center for Systems Science and Engineering (JHU CSSE). The data were also supported by the ESRI Living Atlas Team and the Johns Hopkins University Applied Physics Lab (JHU APL) [38]. The number of confirmed cases of COVID-19, deaths, and rehabilitations were recorded. This study processed data on the number of confirmed COVID-19 case in the contiguous US. We also used another COVID-19 dataset collected by Ustayedev et al. [39] based on the dataset published by the JHU CSSE and transformed it into a time series. The data were in comma-separated values (CSV) format and were updated daily.

The data used in this study were from 2020.01.22 to 2020.04.06. We used data from the first 366,607 confirmed cases in the US. During this time, the number of global confirmed cases, except for China, increased from 555 to 1,345,048. By the end of the time period, the US had become one of the most severely affected countries. Additionally, the dataset included records of COVID-related deaths. The number of deaths in the

**Table 1**

Area Epidemic Level.

Levels	Number of confirmed COVID-19 cases
Level 0	0
Level 1	$1 \leq N < 100$
Level 2	$100 \leq N < 1000$
Level 3	$1000 \leq N < 5000$
Level 4	$5000 \leq N < 10000$
Level 5	$10000 \leq N$

*N*, number of confirmed cases in the area**Fig. 1.** The contiguous United States was divided into 32,768 GPS regions.**Fig. 2.** Schematic diagram of epidemic level and GPS regions displayed in the contiguous United States on 2020.04.06.

US increased from 0 to 10,783 between January 22nd and April 6th, 2020. During this time, the number of deaths globally, except for China, increased from 17 to 74,565. However, we did not use data on the number of deaths in this study. Therefore, all data related to deaths were excluded.

This dataset not only included the daily numbers of COVID-19 diagnoses in the US but also the location of each diagnosed person. The position accuracy in the latitude and longitude was up to the fifth decimal place. This study aimed to predict the increase in the number of COVID-19 diagnoses in an area according to its position. Recent literature indicates that COVID-19 incidence may be related to climate [37]; thus, this study considered the latitude and longitude of the affected areas to be important. Therefore, the data extracted from the database in this study included the time, location, and number of COVID-19 cases. The experimental area did not include Hawaii or Alaska.

### Pre-processing

In this study, each region was categorized into six levels according to the number of COVID-19 cases. The area epidemic levels are shown in Table 1. An epidemic level of 0 implied 0 confirmed COVID-19 cases,

level 1 implied  $\geq 1$  and  $< 100$  confirmed cases, level 2 implied  $\geq 100$  and  $< 1000$  confirmed cases, level 3 implied  $\geq 1000$  and  $< 5000$  confirmed cases, level 4 implied  $\geq 5000$  and  $< 10,000$ , and level 5 implied  $\geq 10,000$  confirmed cases.

Our model does not predict the exact number of cases in each region; rather, it predicts epidemic levels in each region. In other words, this study focused on the prediction of the area-wise severity of the epidemic. The area considered in this study was the contiguous US (Hawaii, and Alaska were excluded), which corresponded to 25° to 49° north latitude and 70° to 130° west longitude. During pre-processing, the total area was divided into 256 equal grids along the longitude, and 128 equal grids along the latitude, after which the contiguous US was divided into 32,768 global positioning system (GPS) regions (Fig. 1). The number of patients diagnosed with COVID-19 was added to each region and the epidemic level was graded as shown in Table 1.

After estimating the epidemic level in each GPS region, a heatmap of the epidemic can be drawn (Fig. 2). Fig. 2 shows the epidemic situation on April 6th, 2020, corresponding to the end of the study period. Both the input and output of the AL-CNN and CAE were heatmaps of the GPS regions. The background map was not entered into the neural network. This background image was not completely aligned with the GPS regions. Some distance errors were noted because of the curved surface of the Earth. However, the background image was only used for display purposes, and there were no errors in the actual distances in the GPS regions. Therefore, the experimental results were not affected by the non-alignment between the background image and GPS regions.

While visualizing the features, the value of the epidemic level was scaled from 0 to 5 to 0–255. In the training process, the min–max scaling was used to compress 0–6 levels to 0–1. The formula for the min–max scaling is as follows:

$$\bar{x} = (x - x_{\min}) / (x_{\max} - x_{\min}) \quad (1)$$

where  $\bar{x}$  is the conversion feature and  $x$  is the original feature. After the neural networks, the features are scaled back to 0–255 as an image to calculate the indicators and for visualization.

### Deep learning models

We propose two architectures to predict the number of confirmed COVID-19 cases: CAE and AL-CNN. The CAE does not include the LSTM network architecture and lacks the prediction effect for the time series. However, the AL-CNN includes an LSTM to improve the prediction effect of the time series. This difference resulted in an improved prediction effect for the AL-CNN. In the following section, we introduce the network architecture and detailed design methods of the CAE and AL-CNN.

#### Convolutional autoencoder

Several studies have applied CAEs and demonstrated good performance in several research areas [40–43]. A CAE is a neural network that uses a CNN as the core neuron and has an autoencoder (AE) architecture. This allows the CAE to possess a similar ability as CNNs in processing images. Because its architecture includes an AE, the CAE can also reconstruct images.

The architecture of the CAE in this study is shown in Fig. 3 and can be roughly divided into an encoder and a decoder. The encoder included five CNN layers and five max-pooling layers, whereas the decoder included six CNN layers and five up-sampling layers. The numbers of convolutional neurons in the CNN layers were as follows: 256 in the first, second, ninth, and tenth layers; 512 in the third, fourth, seventh, and eighth layers; and 1024 in the fifth and sixth layers. The last CNN layer included seven convolutional neurons. The sizes of all convolutional neurons were (3, 3).

The formula for the convolutional neurons is as follows:

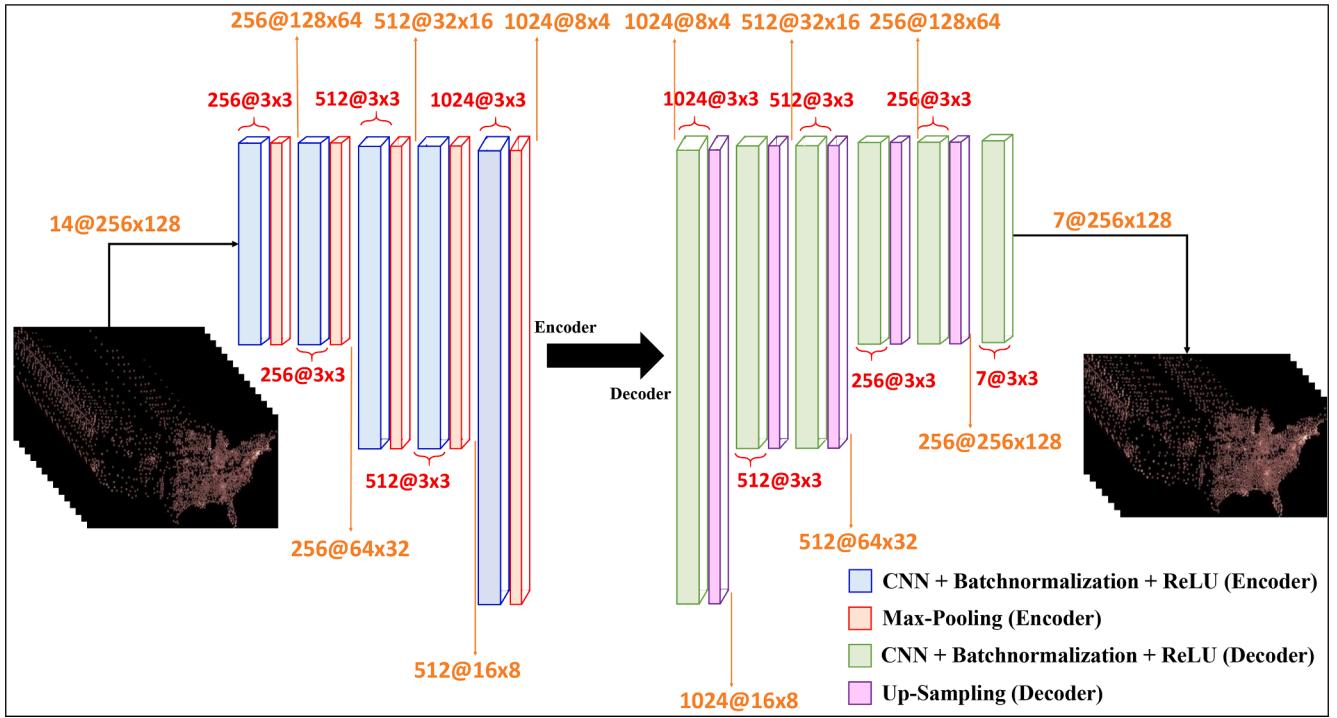


Fig. 3. The CAE architecture in this paper.

$$PO_{ij} = \sum_{m=1,n=1}^h w_{m,n} PI_{ij} + \beta \quad (2)$$

where  $PO_{ij}$  is the pixel value of the convolutional neuron output,  $h$  is the maximum size of the convolutional filter,  $m$  and  $n$  are the calculated positions in the convolutional filter,  $w_{m,n}$  is the convolutional filter of position  $(m, n)$ ,  $PI_{ij}$  is the pixel value of the input, and  $\beta$  is the bias of the convolutional filter.

In the CNN layers, batch normalization was performed to avoid the vanishing gradient problem. The formula for batch normalization is as follows:

$$\widehat{PO}_{ij}^h = \left( PO_{ij}^h - \mu^h \right) / \sqrt{(\sigma^h)^2 + \epsilon} \quad (3)$$

where  $\widehat{PO}_{ij}^h$  is the output value of batch normalization,  $PO_{ij}^h$  is the input of the batch normalization,  $\mu^h$  is the mean of the batch,  $h$  is the batch size,  $\sigma^h$  is the variance of the batch, and  $\epsilon$  is added in the denominator for numerical stability and is an arbitrarily small constant. Following normalization, a transformation step was performed as follows:

$$PO_{ij}^h = f_{act} \left( \alpha^h \widehat{PO}_{ij}^h + \beta^h \right) \quad (4)$$

where  $\alpha^h$  is the weight,  $\beta^h$  is the bias, and  $f_{act}$  is the activation function. Except for the last layer, the activation function in all of the layers in the CAE is the rectified linear unit (ReLU), as follows:

$$f(x) = \max(0, x) \quad (5)$$

The activation function of the last layer was sigmoid, as follows:

$$f(x) = \frac{1}{1 + e^x} \quad (6)$$

In the encoder process, after every CNN layer, a max-pooling layer was connected to reduce the image size by 50%. The formula for max-pooling is as follows:

$$I_{ij} = \max_{a,b=0}^1 \widehat{PO}_{2i+a,2j+b} \quad (7)$$

where  $I_{ij}$  is the pixel value of the max-pooling output and  $\widehat{PO}_{2i+a,2j+b}$  is the input of the max-pooling. In the decoder process, after every CNN layer, an up-sampling layer was connected to zoom the image to double its size. The CAE optimizer was Adam [44]. The Adam equations are as follows:

$$g_t = \nabla_\theta J(\theta_{t-1}) \quad (8)$$

where  $g_t$  is the gradient of time  $t$ ,  $\theta$  is the updated parameter, and  $J$  is the stochastic objective function. Then, the biased first moment estimate and biased second raw moment estimate can be updated as follows:

$$m_t = \beta_1(m_{t-1}) + (1 - \beta_1)g_t \quad (9)$$

$$v_t = \beta_2(v_{t-1}) + (1 - \beta_2)g_t^2 \quad (10)$$

where  $\beta_1$  and  $\beta_2$  are the exponential decay rates for the moment estimates. After updating the biased first moment estimate and biased second raw moment estimate, the bias-corrected first moment estimate and bias-corrected second raw moment estimate can be computed as follows:

$$\hat{m}_t \leftarrow \frac{m_t}{1 - \beta_1^t} \quad (11)$$

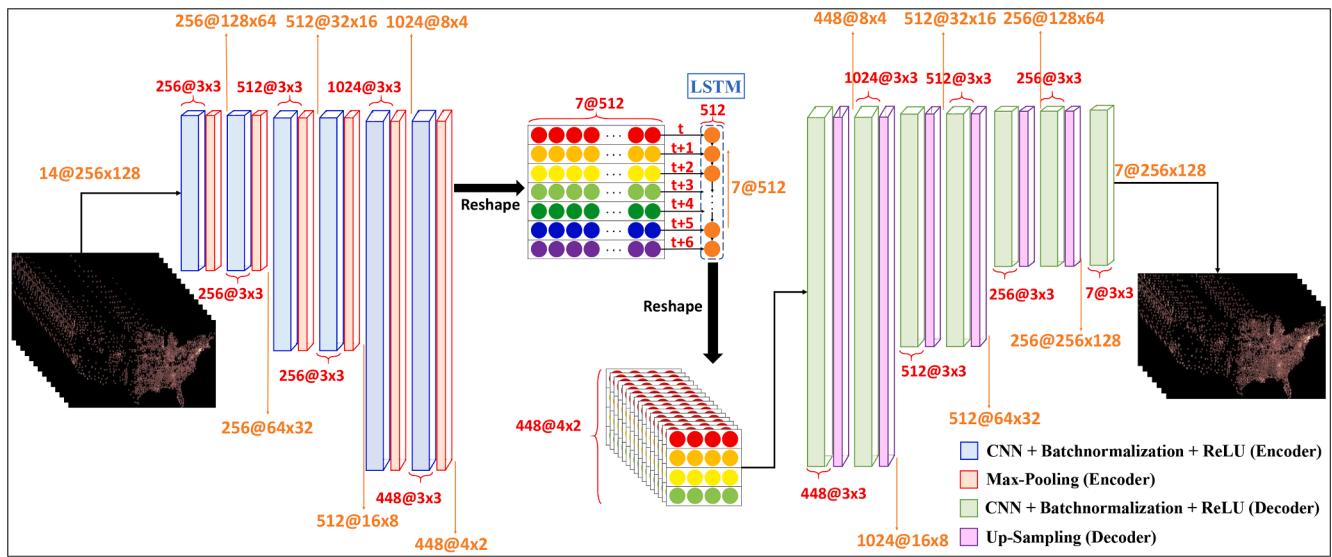
$$\hat{v}_t \leftarrow \frac{v_t}{1 - \beta_2^t} \quad (12)$$

Finally, the parameter can be updated as follows:

$$\theta_t \leftarrow \theta_{t-1} - \alpha \cdot \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \quad (13)$$

where  $\alpha$  is the learning rate and  $\epsilon$  is an extremely small value to avoid calculation error. In this study,  $\alpha$  is set to 0.001,  $\beta_1$  is set to 0.9,  $\beta_2$  is set to 0.999, and  $\epsilon$  is set to  $10^{-8}$ .

The loss is the mean squared error (MSE). The formula for MSE is as follows:



**Fig. 4.** The AL-CNN structure in this paper.

$$MSE = \frac{1}{n} \sum \left( x_{ij} - \hat{x}_{ij} \right)^2 \quad (14)$$

where  $x_{ij}$  is the label of the image and  $\hat{x}_{ij}$  is the predicted image.

The input size of the CAE was (14, 256, 128), which represented the epidemic level 14 days prior to the test. The output size of the CAE was (7, 256, 128), which showed the predicted epidemic level 7 days following the test. The encoder reduced the image size to (1024, 4, 2). There were 458,752 ( $14 \times 256 \times 128$ ) neurons in the input of the encoder, whereas there were 8192 ( $1024 \times 4 \times 2$ ) neurons in the output. Finally, the decoder rebuilt the image with 8192 neurons into an image with 229,376 ( $7 \times 256 \times 128$ ) neurons. Thus, the CAE could predict the epidemic level after seven days through a large number of dimensionality reduction procedures.

The results show that the CAE has a certain predictive ability, but the effect is not as good as that of the AL-CNN. The experimental results are discussed in the next section.

#### Convolutional autoencoder with long short-term memory

The AL-CNN is the deep-learning architecture evaluated in this study. The structure of the AL-CNN is shown in Fig. 4. The AL-CNN and CAE differ in that the AL-CNN has the structure of an LSTM between the encoder and decoder. After the encoder, the neurons were reshaped into seven time-series steps in the AL-CNN. The last layer of convolutional neurons in the encoder process comprises 1024 neurons in the CAE. However, the neuronal output of the encoder must be a multiple of seven. Therefore, the last layer of the convolutional neurons in the encoder process was 448 in the AL-CNN.

The output of the encoder was (448, 4, 2) in size, which can be reshaped to (7, 512). The next step was to design the LSTM layers. LSTM performs data operations through three cores: input, output, and forget gates. The formulas for the three gates are as follows:

$$I_t = f_{act}(w_i x_t + u_i h_{t-1} + \beta_i) \quad (15)$$

$$O_t = f_{act}(w_o x_t + u_o h_{t-1} + \beta_o) \quad (16)$$

$$F_t = f_{act}(w_f x_t + u_f h_{t-1} + \beta_f) \quad (17)$$

where  $I_t$ ,  $O_t$ , and  $F_t$  are vectors of the input, output, and forget gates, respectively;  $w$  is the weight of the input vector in time step  $t$ ;  $u$  is the weight of the previous hidden layer in time step  $t-1$ ;  $x$  is the input

vector in time step  $t$ ,  $h$  is the previous hidden layer in time step  $t-1$ ;  $\beta$  is the bias; and  $f_{act}$  is the recurrent activation function. In the AL-CNN, the recurrent activation function is a hard sigmoid function [45]. The formula for the hard-sigmoid is as follows:

$$\begin{cases} 0, & \text{if } x < -2.5 \\ 1, & \text{if } x > 2.5 \\ 0.2x + 0.5, & \text{if } -2.5 \leq x \leq 2.5 \end{cases} \quad (18)$$

The output of the hard-sigmoid function was in the range of 0 to 1, which represented the degree of massage allowed to pass through the gate. The input and forget gates determined the level of data retention and the removal of data from the new memory cell. The updating process of the memory cell was calculated as follows:

$$C_t = F_t \times C_{t-1} + I_t \times \tanh(w_c x_t + u_c h_{t-1} + \beta_c) \quad (19)$$

where  $C_t$  is the memory state in time step  $t$ , and  $C_{t-1}$  is the memory state in time step  $t-1$ . The output of the current state is as follows:

$$L_t = O_t \times C_t \quad (20)$$

which is the multiplication of the memory state and output gate

In the AL-CNN, there was one LSTM layer with 512 units. After processing in the LSTM layer, the data sizes were (7, 512). To enable the output of the LSTM layer to be used as the input of the decoder, a reshaping layer was added after all the LSTM layers. According to the reshaping layer, the output data size was (7, 512) and was reshaped into a data size of (448, 4, 2). Therefore, the input size of the decoder was (448, 4, 2), whereas the output size of the decoder was (7, 256, 128), which represented the predicted epidemic level 7 days later.

The AL-CNN optimizer was Adam with a learning rate of 0.001,  $\beta_1$  of 0.9, and  $\beta_2$  of 0.999. The loss of the AL-CNN was the MSE. Except for the last CNN layer and LSTM layers, the activation function of all layers was ReLU. The activation function of the last layer was sigmoid. The experimental results showed that the AL-CNN can predict the epidemic level of COVID-19. The results are presented in the next section.

#### Experimental

During the training process, early stopping was applied to avoid overfitting. When the training error was not updated for 10,000 epochs, the training process was stopped and the model weight and bias were frozen.

**Table 2**

Mean Results of AL-CNN and CAE for 36 Experiments.

Prediction Date	AL-CNN						CAE					
	Train			Test			Train			Test		
	MSE	PSNR	SSIM	MSE	PSNR	SSIM	MSE	PSNR	SSIM	MSE	PSNR	SSIM
Day 1	1.343	55.653	0.995	1.482	56.694	0.992	1.045	60.625	0.996	1.550	57.260	0.991
Day 2	1.406	56.028	0.995	1.362	57.292	0.992	1.120	59.995	0.996	1.656	57.108	0.991
Day 3	1.472	55.521	0.995	1.410	57.148	0.992	1.117	59.006	0.996	1.802	55.444	0.990
Day 4	1.564	54.853	0.994	1.592	56.237	0.991	1.220	57.436	0.996	1.921	55.322	0.990
Day 5	1.527	54.278	0.995	1.659	55.563	0.990	1.250	56.374	0.995	2.045	54.247	0.989
Day 6	1.560	53.533	0.995	1.720	54.309	0.990	1.242	55.403	0.995	2.331	52.944	0.986
Day 7	1.683	52.423	0.994	2.423	52.650	0.986	1.584	53.768	0.994	3.417	50.823	0.980
Average	1.508	54.613	0.995	1.664	55.699	0.990	1.225	57.515	0.995	2.103	54.735	0.988

**Table 3**

Standard Deviation Errors in the AL-CNN and CAE for 36 experiments.

Prediction Date	AL-CNN						CAE					
	Train			Test			Train			Test		
	MSE	PSNR	SSIM	MSE	PSNR	SSIM	MSE	PSNR	SSIM	MSE	PSNR	SSIM
Day 1	1.659	11.928	0.005	2.877	11.612	0.011	1.333	12.870	0.004	2.687	13.427	0.012
Day 2	1.683	12.684	0.005	2.641	11.963	0.011	1.387	12.740	0.004	2.885	13.320	0.012
Day 3	1.745	12.475	0.005	2.349	12.703	0.010	1.355	12.043	0.004	2.864	12.753	0.011
Day 4	1.790	12.267	0.005	2.422	12.905	0.010	1.427	11.506	0.004	2.739	13.356	0.012
Day 5	1.709	11.683	0.005	2.421	12.651	0.010	1.412	11.147	0.004	2.850	13.005	0.011
Day 6	1.695	10.978	0.005	2.276	11.926	0.011	1.352	10.664	0.004	2.798	12.720	0.014
Day 7	1.836	10.215	0.004	3.156	11.773	0.014	1.660	9.885	0.004	3.791	12.414	0.019
Average	1.731	11.747	0.005	2.592	12.219	0.011	1.418	11.551	0.004	2.945	12.999	0.013

### Experimental results

For comparison, the peak signal-to-noise ratio (PSNR) [46], MSE, and structural similarity index (SSIM) [47] were used as indicators for evaluating the rebuild heatmap. The formula for PSNR is as follows:

$$f_{psnr} = 20 \times \log \left\{ \frac{255}{\sqrt{\frac{1}{nm} \sum \sum [I(i,j) - \bar{I}(i,j)]^2}} \right\} \quad (21)$$

where  $I$  is the target image and  $\bar{I}$  is the reconstructed image. The formula for SSIM is as follows:

$$f_{ssim} = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \quad (22)$$

where  $\mu_x$  is the average of the signal  $x$ ,  $\mu_y$  is the average of the signal  $y$ ,  $\mu_x^2$  is the variance of the signal  $x$ ,  $\mu_y^2$  is the variance of signal  $y$ , and  $\sigma_{xy}$  is the covariance of  $x$  and  $y$ . The two variables  $c_1$  and  $c_2$  are to stabilize the division with a weak denominator, which is shown below:

$$c_1 = (k_1 \times L)^2 \quad (23)$$

$$c_2 = (k_2 \times L)^2 \quad (24)$$

where  $L$  is the dynamic range of the pixel values,  $k_1 = 0.01$  and  $k_2 = 0.03$  by default. These three indicators were used to calculate the degree of image restoration, in which the higher the degree of restoration, the lower the two values. The prediction results were divided into seven days to assess the prediction ability. The results obtained were the average values of each indicator for each forecast date. The average results, standard deviation errors, and confidence intervals are shown in Tables 2–4. More comparisons were performed to increase the reliability of the experiments. Therefore, the same hyperparameters were used to conduct an additional 36 experiments.

Table 2 shows the mean results of the indicators separated by the

prediction days. The average results for the seven days are also shown. When the prediction time was short, the prediction accuracies of the AL-CNN and CAE did not differ significantly. However, when predicting a longer time, the effect of the AL-CNN was better than that of the CAE. Comparisons of MSE, PSNR, or SSIM showed that the AL-CNN was better than the CAE. The training results of the CAE were better than those of the AL-CNN. However, the testing results of the CAE were worse than those of the AL-CNN. This is because the CAE causes more overfitting than the AL-CNN. Although the CAE can show lower results through training, overfitting leads to unsatisfactory test results. All three indicators displayed the same phenomenon, proving the reliability of the experiment.

Table 3 lists the standard deviation errors. The standard deviation of the AL-CNN was mostly smaller than that of the CAE. The stability of the AL-CNN is better than that of the CAE. The standard deviation error was not correlated with the prediction date. Regardless of whether a closer or farther date was predicted, the value of the standard deviation error was irrelevant. Only on the seventh day did the standard deviation error of the results of the two models increase. The change in the standard deviation error of the CAE at 7 days was relatively large, whereas that of the AL-CNN was relatively small. All three indicators displayed the same phenomenon, proving the reliability of the experiment. However, in the training process, the standard deviation error of the CAE was larger than that of the AL-CNN. Thus, the CAE showed more overfitting than the AL-CNN.

Table 4 presents the confidence-interval (95%) results. Evidently, the confidence interval of the AL-CNN result is smaller than that of the CAE result. Additionally, the AL-CNN was more stable than the CAE. The 95% confidence interval on the MSE gradually increased with the number of prediction days. The 95% confidence intervals for PSNR and SSIM gradually decreased with the number of prediction days. The farther the prediction date, the lower the prediction accuracy. All three indicators showed the same results, proving the reliability of this experiment.

The prediction statistics clearly showed a smaller error in the model for the prediction of the epidemic level at a date closer to the current date as, for humans or mathematical models, the prediction time is

Prediction Date	AL-CNN						CAE					
	Train			Test			Train			Test		
	MSE	PSNR	SSIM									
Day 1	1.163–1.328	55.177–56.407	0.993–0.996	1.152–1.542	55.811–57.590	0.987–0.994	1.489–3.389	59.334–60.773	0.986–0.992	1.245–3.180	55.643–57.784	0.979–0.989
Day 2	1.265–1.438	55.128–56.418	0.993–0.996	1.118–1.501	56.175–57.993	0.989–0.994	1.631–3.728	59.064–60.514	0.985–0.992	1.351–3.463	55.829–58.038	0.898–0.988
Day 3	1.296–1.472	54.779–56.049	0.993–0.996	1.213–1.564	55.296–57.180	0.987–0.994	1.764–4.022	58.125–59.522	0.984–0.991	1.550–3.816	54.354–56.515	0.976–0.986
Day 4	1.364–1.541	54.323–55.598	0.993–0.996	1.305–1.653	55.055–56.992	0.987–0.993	1.947–4.456	56.606–57.907	0.986–0.992	1.662–4.105	53.529–55.656	0.978–0.987
Day 5	1.383–1.558	53.674–54.898	0.993–0.996	1.388–1.746	54.566–56.493	0.986–0.993	2.099–4.770	55.239–56.454	0.984–0.991	1.732–4.334	52.846–54.886	0.977–0.986
Day 6	1.403–1.576	53.078–54.238	0.993–0.996	1.487–1.824	53.290–55.147	0.985–0.992	2.292–5.214	54.356–55.529	0.985–0.992	2.120–4.909	51.439–53.449	0.975–0.984
Day 7	1.512–1.704	52.041–53.118	0.992–0.995	2.190–2.672	51.634–53.490	0.981–0.988	2.559–5.460	53.163–54.268	0.973–0.984	3.270–6.025	49.426–51.402	0.956–0.972
Average	1.341–1.517	54.029–55.247	0.993–0.996	1.408–1.786	54.547–56.412	0.986–0.992	1.969–4.434	56.555–57.852	0.983–0.991	1.847–4.262	53.295–55.390	0.974–0.985

closer to the prediction process. Additionally, it was more difficult to predict the parameters farther away in time. As shown in the forecast Day 1 to Day 7, the MSE of the AL-CNN increased from 1.482 to 2.423, the PSNR decreased from 56.694 to 52.65, and the SSIM decreased from 0.992 to 0.986. In addition, the MSE of the CAE increased from 1.55 to 3.417, the PSNR decreased from 57.26 to 50.823, and the SSIM decreased from 0.991 to 0.98. Both models showed difficulty in long-term predictions.

Comparisons of the two models showed that the primary difference between the AL-CNN and CAE was the long-term prediction ability of the AL-CNN. In the short-term prediction of epidemic levels, the results of the AL-CNN and CAE did not differ significantly. However, when the forecast date was farther away, the prediction accuracy of the AL-CNN was significantly higher than that of the CAE.

Fig. 5 shows the prediction results for the last seven days. Throughout the experiment, we used COVID-19 epidemic data for a total of 75 days. The figure shows the prediction results of the last seven days of the study period; namely, Days 69 to 75. The white color represents the epidemic level, with a brighter color indicating a higher epidemic level. Visually, the results of the AL-CNN did not differ significantly from those of the CAE. Only comparisons of the various indicators shown in Tables 2–4 demonstrated the differences in the results between the models. Both are excellent prediction models, but evaluation of the indicators revealed that the AL-CNN performed better than the CAE.

#### Feature observation

In this section, we discuss the feature extraction capabilities of AL-CNN and CAE. To measure the feature extraction indexes, we extracted the encoder output of the CAE and the LSTM output of the AL-CNN. The extracted features of the CAE and AL-CNN are shown in Figs. 6 and 7, respectively.

The colors in Figs. 6 and 7 are shown using the hue, saturation, and value (HSV) color scale, as shown in Fig. 8. The features of the first and last epidemic prediction days were set at the left and right ends of the color spectrum, respectively.

Regardless of the output size of the neural network, principal component analysis (PCA) [48] was used for dimension reduction for easy display. To do so, the data were initially normalized; singular value decomposition (SVD) [49] was then used to calculate the eigenvalue and eigenvector of the covariation matrix. The SVD formula is as follows:

$$A = U\Sigma V^T \quad (25)$$

where  $A$  is the feature matrix,  $U$  is the unit eigenvector of  $AA^T$ ,  $V$  is the unit eigenvector of  $A^TA$ , and  $\Sigma$  is the square of the eigenvalue of  $AA^T$  or  $A^TA$ . The feature vectors were arranged according to their numerical values, in which feature vectors with larger feature values had more important features. We selected several feature vectors to convert and perform the following calculations:

$$\dot{X} = U_k \times X \quad (26)$$

where  $k$  is the dimension reduction level. In this experiment,  $k$  was set to three.

In Fig. 6, the extracted features of the CAE are randomly distributed. In the first 70 days, all epidemic features were clustered together. Even though the features are confusing, they still follow certain rules. The closer the feature points to the clustered position, the lower the epidemic level and the farther from the clustered position, the higher the epidemic level. The red, yellow, and green feature points represent features with low epidemic levels. These feature points were located at the center of the cluster. The Purple and blue represent features with high epidemic levels. These features are far from clustered feature points.

As shown in Fig. 7, the extracted features of the AL-CNN showed a more regular pattern than that of the CAE. The x-axis may be a criterion

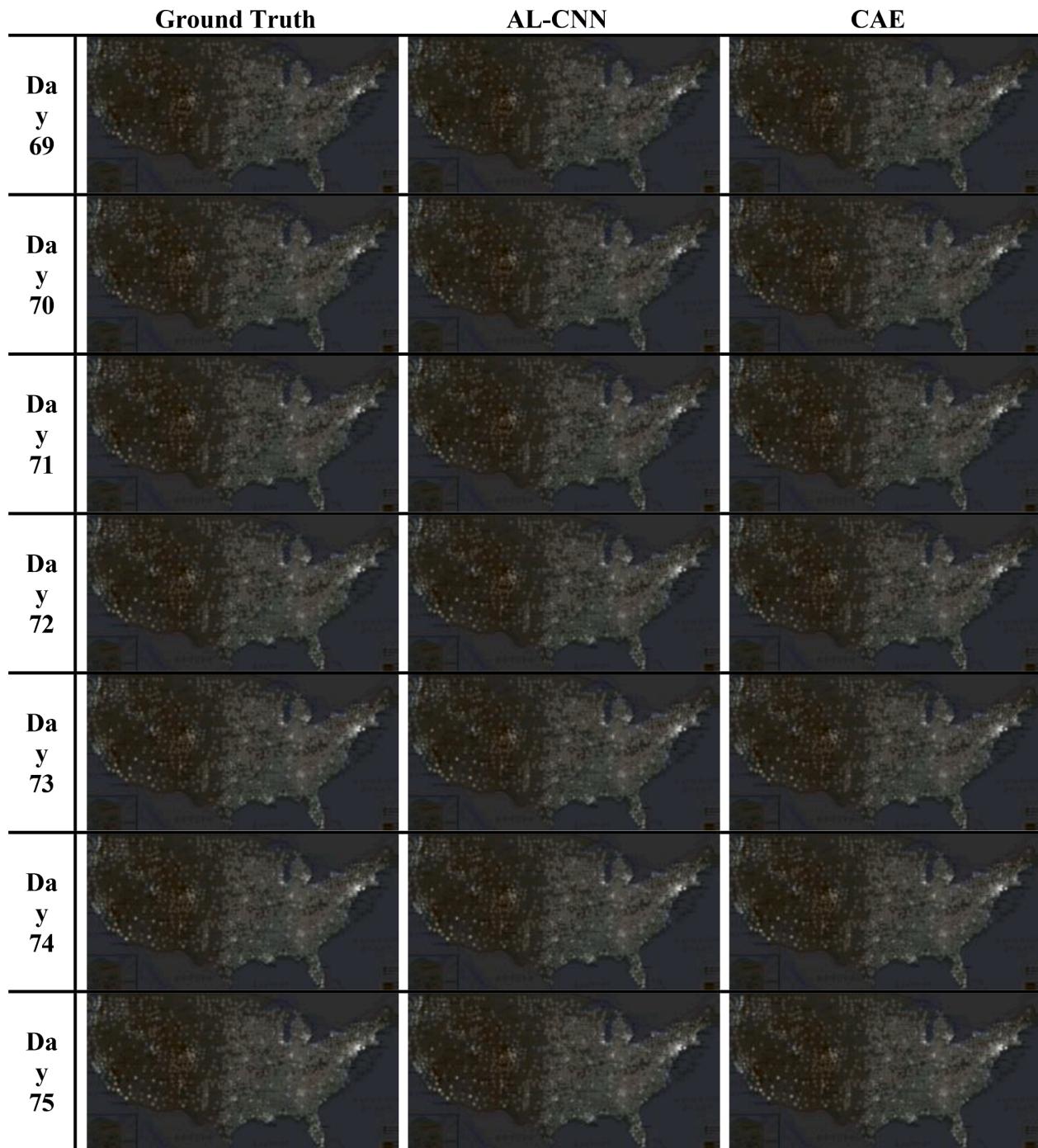
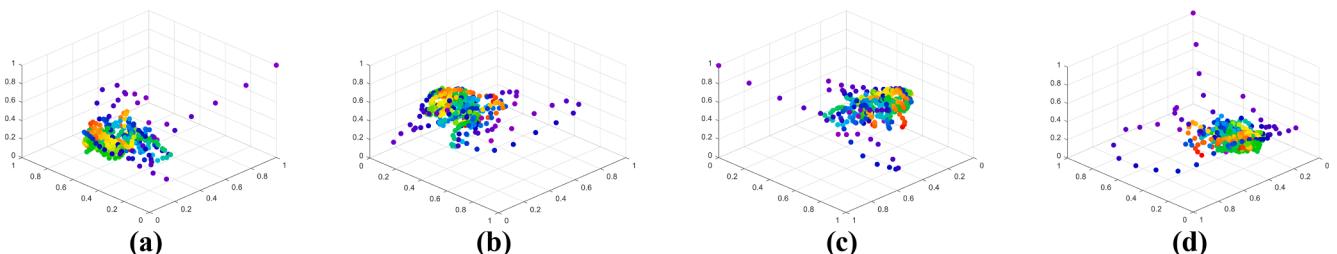
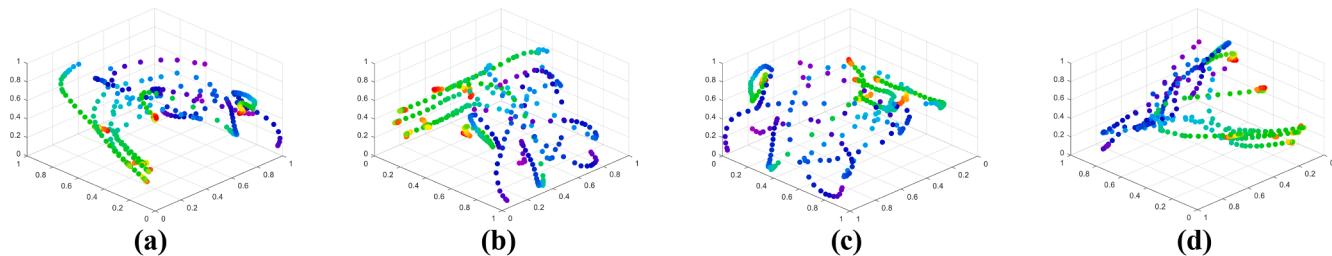


Fig. 5. The last 7 days of prediction results.

Fig. 6. The feature extraction results of CAE encoder: (a)  $0^\circ$  viewing angle, (b)  $45^\circ$  viewing angle, (c)  $90^\circ$  viewing angle, and (d)  $135^\circ$  viewing angle.



**Fig. 7.** The feature extraction results of AL-CNN LSTM: (a) 0° viewing angle, (b) 45° viewing angle, (c) 90° viewing angle, and (d) 135° viewing angle.



**Fig. 8.** HSV color scale used in Figs. 6 and 7.

for determining the epidemic level. The red, yellow, and green feature points represent features of low epidemic levels, with their position on the x-axis below 0.5. Purple and blue represent features of a high-level epidemic, with the position of their x-axis being greater than 0.5.

The ability of feature extraction was also positively correlated with the prediction ability of the mode such that the stronger the feature extraction ability, the stronger the prediction ability. The addition of LSTM resulted in a significant change in the CAE. Thus, the AL-CNN showed good performance in both feature extraction and epidemic-level prediction.

#### Computation system

The computer hardware used in this study had an Intel I7-8700k CPU, Nvidia GeForce 1080Ti, and 32 GB of RAM. In the AL-CNN model, the training and testing processes required 53 ms and 25 ms to calculate the batch size, respectively. In the CAE model, the training and testing processes required 39 ms and 18 ms to calculate the batch size, respectively. In both processes, the AL-CNN was slightly more time-consuming than the CAE. However, as mentioned earlier, AL-CNN has improved feature extraction and prediction ability compared to CAE.

This study used Python as a basic programming tool and makes extensive use of Python expansion packages, including TensorFlow, Scikit-Learn, Scikit-Image, NumPy, Pandas, and OpenCV. TensorFlow was used to design and build the neural networks. Scikit-Learn and Scikit-Image were used to organize the big data and calculate various indicators. NumPy and Pandas were used for array reading and the conversion of various data. OpenCV is a technology related to image processing. All images in this study were drawn using MATLAB R2018a because of its convenient drawing function and high image quality.

#### Conclusion

This study attempted to predict the epidemic levels of COVID-19 in the contiguous US by using CAE and AL-CNN models. The CAE is a deep learning neural network connected by a series of CNNs, as shown in Fig. 3. Although CAE has a good predictive ability, there remains a lack of a design framework for time-series data in neural networks. The AL-CNN is a modified version of the CAE that adds an LSTM between the encoder and decoder to increase the processing power of the model for time series data, as shown in Fig. 4.

The mean square error of the AL-CNN was 1.664, the PSNR was 55.699, and the structural similarity index was 0.99, which were better than the corresponding results of the CAE, as shown in Table 2. Moreover, our study proved the predictive performance of AL-CNN. Observation of the feature extraction capabilities showed that the features extracted by AL-CNN were more regular and not cluttered as in CAE, as demonstrated in Figs. 6 and 7.

The model structure used in this study showed very high accuracy in the prediction of epidemic levels. Hence, this forecast can effectively assist the country in adjusting its epidemic prevention policy. This model can provide more information for observing the regional diffusion responses. Previous studies have only predicted the total number of infected individuals; relatively few have predicted the number of infected individuals in each area.

The levels of the epidemic can be refined as needed. For example, when the government needs to predict whether a region will reach 1000 confirmed cases in the near future, an epidemic level should be settled with a threshold of 1000. The epidemic level can be adjusted according to the needs of the government or by the number of confirmed cases. The design of the GPS area is more complicated because it involves the design of a neural network. The present model should be practical for predicting epidemic levels in other countries or regions. Some necessary data include the confirmed case numbers for each clinic and the GPS location of each clinic. When designing a neural network, the designer must determine the appropriate size of the GPS grid. The design of the GPS grid affects the input and output of the neural network. According to the design of the GPS region, which is 32,768 in this study, the input and output size of the neural network is  $14 \times 128 \times 256$  and  $7 \times 128 \times 256$ , respectively. In typical cases, the scalar of the GPS grid is expected to have a power of 2. Scalars with a power of two make the max-pooling layer easier to affect. For example, when the scalar is  $16 \times 16$ ,  $32 \times 32$ , and  $128 \times 128$ , the input can be max-pooled for four, five, and seven times, respectively. By using a scalar with a power of two, the designer can change the regional scale according to their needs.

In the future, we plan to organize more regional data and expand the coverage of the model. It is expected that this model can be applied not only to the contiguous US but also globally. Furthermore, additional features should be included, such as the humidity and the number of deaths. We believe that including more features can increase the prediction accuracy. Our future research will create complex deep-learning architectures that are adapted to more features.

#### Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

#### References

- [1] (2020). Wuhan Municipal Health Commission Infection Data. Available: <http://wjw.wuhan.gov.cn/front/web/list2nd/no/710>.
- [2] (2020). Coronavirus Disease (COVID-19) Outbreak. Available: <https://www.who.int/westernpacific/emergencies/covid-19>.
- [3] M. Chinazzi, J. T. Davis, M. Ajelli, C. Gioannini, M. Litvinova, S. Merler, A. P. Piontti, K. Mu, L. Rossi, K. Sun, C. Viboud, X. Xiong, H. Yu, M. E. Halloran, I. M. Jr, and A. Vespignani, "The Effect of Travel Restrictions on the Spread of the 2019 Novel Coronavirus (COVID-19) Outbreak," *Science*, vol. 368, no. 6489, pp. 395, 2020.
- [4] M. U. G. Kraemer, C. H. Yang, B. Gutierrez, C. H. Wu, B. Klein, D. M. Pigott, Open COVID-19 Data Working Group†, L. d. Plessis, N. R. Faria, R. Li, W. P. Hanage, J. S. Brownstein, M. Layman, A. Vespignani, H. Tian, C. Dye, O. G. Pybus, and S. V. Scarpino, "The Effect of Human Mobility and Control Measures on the COVID-19 Epidemic in China," *Science*, vol. 368, no. 6490, pp. 493, 2020.

- [5] Anderson Roy M. The pandemic of antibiotic resistance. *Nat Med* 1999;5(2):147–9.
- [6] Koopman Jim. Modeling infection transmission. *Annu Rev Public Health* 2004;25(1):303–26.
- [7] Levin SA, Grenfell B, Hastings A, Perelson AS. Mathematical and computational challenges in population biology and ecosystems science. *Science* 1997;275(5298):334.
- [8] Ng TW, Turinici G, Danchin A. A double epidemic model for the SARS propagation. *BMC Infect Dis* 2003;3(1):19.
- [9] Iannelli Mimmo, Martcheva Maia, Li Xue-Zhi. Strain replacement in an epidemic model with super-infection and perfect vaccination. *Math Biosci* 2005;195(1):23–46.
- [10] Kwok Kin On, Tang Arthur, Wei Vivian WI, Park Woo Hyun, Yeoh Eng Kiong, Riley Steven. Epidemic models of contact tracing: systematic review of transmission studies of severe acute respiratory syndrome and middle east respiratory syndrome. *Comput Struct Biotechnol J* 2019;17:186–94.
- [11] Dye C, Gay N. Modeling the SARS epidemic. *Science* 2003;300(5627):1884.
- [12] Lipsitch M, Cohen T, Cooper B, Robins JM, Ma S, James L, et al. Transmission dynamics and control of severe acute respiratory syndrome. *Science* 2003;300(5627):1966.
- [13] Han XN, De Vlas SJ, Fang LQ, Feng D, Cao W-C, Habbema JDF. Mathematical modelling of SARS and other infectious diseases in China: a review. *Trop Med Int Health* 2009;14(s1):92–100.
- [14] Watts DJ, Strogatz SH. Collective dynamics of ‘small-world’ networks. *Nature* 1998;393(6684):440–2.
- [15] Pastor-Satorras R, Vespignani A. Epidemic spreading in scale-free networks. *Phys Rev Lett* 2001;86(14):3200–3.
- [16] Zhong L, Mu L, Li J, Wang J, Yin Z, Liu D. Early prediction of the 2019 Novel Coronavirus Outbreak in the Mainland China based on simple mathematical model. *IEEE Access* 2020;8:51761–9.
- [17] Yuan J, Tian Y. A multiscale feature learning scheme based on deep learning for industrial process monitoring and fault diagnosis. *IEEE Access* 2019;7:151189–202.
- [18] Kao IH, Hsu YW, Lai YH, Perng JW. Laser cladding quality monitoring using coaxial image based on machine learning. *IEEE Trans Instrum Meas* 2020;69(6):2868–80.
- [19] Kao I-Hsi, Wang Wei-Jen, Lai Yi-Horng, Perng Jau-Woei. Analysis of permanent magnet synchronous motor fault diagnosis based on learning. *IEEE Trans Instrum Meas* 2019;68(2):310–24.
- [20] Luo L, Yang Z, Tang L, Zhang K. An ELM-embedded deep learning based intelligent recognition system for computer numeric control machine tools. *IEEE Access* 2020;8:24616–29.
- [21] Qin Y, Luo H, Zhao F, Wang C, Wang J, Zhang Y. Toward transportation mode recognition using deep convolutional and long short-term memory recurrent neural networks. *IEEE Access* 2019;7:142353–67.
- [22] Wang H, Hou J, Chen N. A survey of vehicle re-identification based on deep learning. *IEEE Access* 2019;7:172443–69.
- [23] Wang P, Hao W, Sun Z, Wang S, Tan E, Li L, et al. Regional detection of traffic congestion using in a large-scale surveillance system via deep residual TrafficNet. *IEEE Access* 2018;6:68910–9.
- [24] García-Pedroso A, Lillo-Saavedra M, Rodríguez-Esparragón D, Gonzalo-Martín C. Deep learning for automatic outlining agricultural parcels: exploiting the land parcel identification system. *IEEE Access* 2019;7:158223–36.
- [25] Altaheri H, Alsulaiman M, Muhammad G. Date fruit classification for robotic harvesting in a natural environment using deep learning. *IEEE Access* 2019;7:117115–33.
- [26] Kao IH, Hsu YW, Yang YZ, Chen YL, Lai YH, Perng JW. Determination of lycopersicon maturity using convolutional autoencoders. *Sci Hortic* 2019;256:108538.
- [27] Perng JW, Kao IH, Chen YW, Lai YH, Su CM, Hung SC, et al. Analysis of the 72-h mortality of emergency room septic patients based on a deep belief network. *IEEE Access* 2018;6:76820–30.
- [28] Perng JW, Kao IH, Kung CT, Hung SC, Lai YH, Su CM. Mortality prediction of septic patients in the emergency department based on machine learning. *J Clin Med* 2019;8(11):1906.
- [29] Tang Z, Chen K, Pan M, Wang M, Song Z. An augmentation strategy for medical image processing based on statistical shape model and 3D thin plate spline for deep learning. *IEEE Access* 2019;7:133111–21.
- [30] Jiang Dong, Hao Mengmeng, Ding Fangyu, Fu Jingying, Li Meng. Mapping the transmission risk of zika virus using machine learning models. *Acta Trop* 2018;185:391–9.
- [31] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” presented at the Advances in Neural Information Processing Systems 25, 2012.
- [32] Mikolov T, Karafiat M, Burget L, Černocký J, Khudanpur S. Recurrent neural network based language model. Eleventh annual conference of the international speech communication association. 2010.
- [33] Hochreiter S, Schmidhuber J. Long short-term memory. *Neural Comput* 1997;9(8):1735–80.
- [34] Zhang H, Saravanan KM, Yang Y, Hossain MT, Li J, Ren X, Wei Y. Deep learning based drug screening for novel coronavirus 2019-nCov. *Preprints* 2020.
- [35] Q. Guo, M. Li, C. Wang, P. Wang, Z. Fang, J. tan, S. Wu, Y. Xiao, and H. Zhu, “Host and Infectivity Prediction of Wuhan 2019 Novel Coronavirus Using Deep Elarning Algorithm,” *bioRxiv*, pp. 2020.01.21.914044, 2020.
- [36] Fong SJ, Li G, Dey N, Crespo RG, Herrera-Viedma E. Composite Monte Carlo decision making under high uncertainty of novel coronavirus epidemic using hybridized deep learning and fuzzy rule induction. *Appl Soft Comput* 2020;106282.
- [37] M. Sajadi, P. Habibzadeh, A. Vintzileos, S. Shokouhi, F. Miralles-Wilhelm, and A. Amoroso, “Temperature and Latitude Analysis to Predict Potential Spread and Seasonality for COVID-19,” *SSRN Electronic Journal*, 2020.
- [38] (2020). Johns Hopkins University CSSE COVID-19 Dashboard. Available: <https://github.com/CSSEGISandData/COVID-19>.
- [39] (2020). Novel Coronavirus 2019 Time Series Data on Cases. Available: <https://github.com/datasets/covid-19>.
- [40] Chen M, Shi X, Zhang Y, Wu D, Guizani M. Deep features learning for medical image analysis with convolutional autoencoder neural network. *IEEE Trans Big Data Early Access* 2017.
- [41] Dizaji KG, Herandi A, Deng C, Cai W, Huang H. Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization. In: *IEEE International Conference on Computer Vision (ICCV)*; 2017. p. 5747–56.
- [42] Leng Biao, Guo Shuang, Zhang Xiangyang, Xiong Zhang. 3D object retrieval with stacked local convolutional autoencoder. *Signal Process* 2015;112:119–28.
- [43] Huang H, Hu X, Zhao Y, Makkie M, Dong Q, Zhao S, et al. Modeling task fMRI data via deep convolutional autoencoder. *IEEE Trans Med Imaging* 2018;37(7):1551–61.
- [44] D. P. Kingma and J. J. a. p. a. Ba, “Adam: A Method for Stochastic Optimization,” 2014.
- [45] M. Courbariaux, Y. Bengio, and J. P. David, “Binaryconnect: Training Deep Neural Networks with Binary Weights During Propagation,” in *Advances in neural information processing systems*, pp. 3123–3131, 2015.
- [46] Huynh-Thu Q, Ghanbari M. Scope of validity of PSNR in image/video quality assessment. *Electron Lett* 2008;44(13):800–1.
- [47] Horé A, Ziou D. Image Quality Metrics: PSNR vs. SSIM. In: *2010 20th International Conference on Pattern Recognition*; 2010. p. 2366–9.
- [48] Wold Svante, Eshbensen Kim, Geladi Paul. Principal component analysis. *Chem Intelligent Lab Syst* 1987;2(1):37–52.
- [49] G. H. Golub and C. Reinsch, “Singular Value Decomposition and Least Squares Solutions,” In *Linear Algebra*, pp. 134–151, 1971.