DEEP LEARNING AND SVM METHOD TO DETECT COVID-19 USING CT SCAN IMAGES

Dr L Mohana Sundari [1,] Associate Professor, Saveetha Engineering College, Chennai, India
Dr T SenthilKumar[2], Associate Professor, GRT Institute of Engineering and Technology, Tirutani, India
Bhavya Kallam Reddy[3], UG Scholar, Saveetha Engineering College, Chennai, India
Janani T[4], UG Scholar, Saveetha Engineering College, Chennai. India

Abstract—
People around the world are seriously affected by the spread of the coronavirus SARS Cov2. Testing for the presence of coronavirus in the human body is based primarily on the reverse transcription polymerase chain reaction (RTPCR) recommended by the World Health Organization. Although the results of the RTPCR test are accurate, the time required for the test is not reasonable due to the high growth rate of the corona. To speed up the virus detection process, an alternative method based on image processing using CT scans of the chest is proposed in this project. Breast images are taken from the benchmark database and tested. Since this is a binary classification problem, CT scan images are applied to a deep learning model. In particular, we use a convolutional neural network (CNN) to determine if an image is "Covid-positive" or "Covid-is-negative". In addition, the benefits of Support Vector Machines (SVMs), combined with CNNs, maximize overall detection capabilities. Training and validation performance is evaluated against a database consisting of 80% training datasets and 20% validation datasets, using a total of 736 images. By changing the image properties such as height shift, width shift, zoom level, brightness level, orientation, shear, and horizontal flip, the image is initially expanded to a larger database. This brings the total to 5888 frames, including the actual 736 raw frames. Covid detection is based on a new random image that has not been used for training and validation. Accuracy, accuracy, and recognition characteristics were determined by ensemble voting of results obtained from CNNs and SVMs. CNN has achieved and training, validation, and detection accuracy, respectively. SVM provided training, testing, and detection accuracy. Combining the capabilities of deep neural networks with traditional SVMs yields a detection accuracy. The implementation of Covid prediction is done using a Python script that leverages Keras on the TensorFlow platform.

INTRODUCTION

It was unfortunate that a deadly virus called the severe acute respiratory syndrome coronavirus (SARSCoV2) was found in Wuhan, China in December 2020 [1]. The spread of the virus has been found to be due to human-to-human contact. The spread was mainly due to the gathering of people in common places. Strict lockdowns (lockdown is the restriction that governments leave a person's place of residence) are imposed in all countries of the world to limit the spread of the virus. This ultimately disrupted people's normal lifestyles and ultimately led to unemployment and economic decline. Due to the slow results of reverse transcription-polymerase chain reaction (RTPCR), patients are more likely to be discharged in violation of self-isolation, further spreading to a larger population. It is important to run the test faster in order to identify Covid patients and isolate them more quickly. The RTPCR test is approved by the World Health Organization (WHO), but results are delayed due to the clinical protocol that follows the execution of the RTPCR test. RTPCR testing takes several hours to detect the presence of coronavirus in a cotton swab taken from the patient's respiratory system [1]. These current issues have helped the author of this report find a suitable test method that provides test results faster while maintaining accuracy. Early detection of the coronavirus is the key to reducing prevalence and mortality. Therefore, this project will perform an artificial intelligence-based Covid19 that detects Covid19 from CT scan images of the lungs using a convolutional neural network and SVM integration technique. Therefore, this project is a classification of covid and non-covid cases based on CT scan images. The database image is taken from [2] and [3].

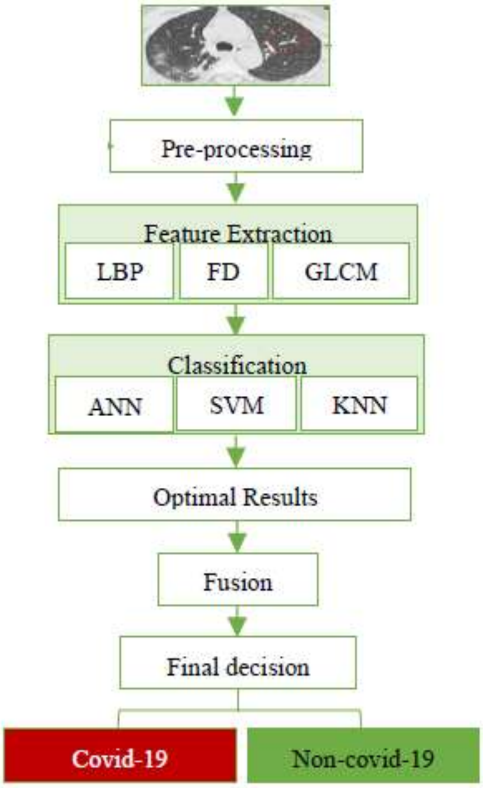
Python

Python is an open-source platform that helps researchers design and analyze artificial intelligence and machine learning algorithms that are superior to any other language. The easy-to-understand syntax provided by the Python community makes it easy for other language experts to catch up with Python. The power of programming in Python depends on the definition of the function. Python also occupies the top spot in programming languages because of the availability of separate modules to perform specific tasks. The programmer's main task is to select the appropriate modules needed for the investigation problem and install these modules using pip install or anaconda-based installation methods. This project has a pip-based installation. It is important to make sure that the module version is compatible with the Python version already installed on your computer.

LITERATURE SURVEY:

In all the conventional methods such as artificial neural network, SVM, fuzzy logic systems, K means clustering, random forest, decision trees, etc., it is the responsibility of the researcher to extract the suitable features. Later, these features are used to train the network while knowing the target classes. After successful training, the testing features with unknown labels are classified using the model with same features as done in training. Few literatures were identified that could produce better accuracy in detecting the covid using basic image features such as; edges, intensity, etc and morphological operations In [4], a combination of laplacian edge features and morphological operations was used to identify the virus infected portions in the images. Histogram equalisation was performed on images to increase the visibility of the blood vessels followed by the edge detection. In addition, if the value of the adjacent pixel is "1", an image extension was performed to enter the value "1" in the output binary image. The dilation made the lesions more prominent. As a final step, the entropy was calculated according to the equation given in equation (2.1) and compared to the threshold.

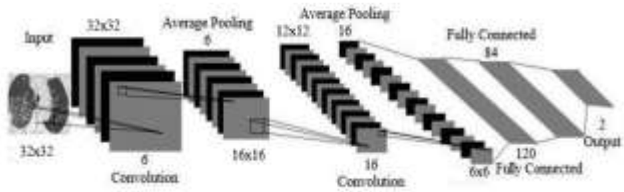
$$H = -\sum_{i=0}^{255} p_i log_2 p_i$$



Deep learning methods

In Ref. [10], Covid and non-Covid images were ranked using the ensemble's hard voting method. Three deep learning models have been built. Model 1 and Model 3 consisted of three CNNs. Model 2 consisted of four CNNs. We combined the outputs of the three models to create an ensemble prediction. The accuracy obtained was 0.954, 0.936 and 0.944 for Model 1, Model 2 and Model 3, respectively. Surprisingly, when the outputs of the three models were combined, the ensemble mode gave an accuracy of 0.960. Reference [11] mentions the need for faster detection of Covid. Such rapid detection saves many lives and ensures that governments and medical authorities are prepared for the worst. In the above literature, the disease-free parts of the image were discarded to improve the accuracy of the classification. Therefore, segmentation was performed on the input lung CT scan image. Segmentation performance was evaluated using the Sorensen Cube Coefficient (DSC) and was obtained as 0.83 when 50 images were considered in the experiment. In this study, images of covid and pneumonia were classified using a random input hostile generation network (RANDGAN) consisting of a generator and a discriminator. This network is similar to a game between two players. The generator always tries to minimize the discriminator's ability to distinguish between real images from the database and fake images from the generator. At the same time, during training, the discriminator gradually knows how to distinguish between real and fake images. Considering the 573 images in each class, the ROC curve bottom area (AUC) 0.77 was obtained from RANDGAN. Before actually processing the image, the input image was converted from RGB to gray and the image size was changed to 128x128.

A simple CNN model was proposed in [13], inspired by the LeNet5 algorithm originally developed in [14] to recognize handwriting. Due to the different size of the images, the size was standardized to 32 x 32 and normalized in the range 0 to 1. As shown in Figure 2.2, the first convolution layer has six filters with a size of 5x5 and a one-step value. Layer 3 of the CNN used 16 filters that measure 5x5 with one stride value. The second and fourth layers of CNN bundled 2x2 size layers in an averaging method instead of maximal bundling. Finally, three fully connected layers of 120, 84, and 2 units were built in sequence in a size of 5x5. The last layer may provide covid or non-covid results in the form of binary data. The integrated values for the model, accuracy, and loss function are shown in Table A.1 and Figure A.1 in the Appendix for space reasons in this chapter.



Integrated methods

Methods of finding covid have also been using combination of algorithms that include a separate feature extraction and automatic learning using deep neural network models. Such a work was done in [19] where the problem of outliers was addressed. Outliers are the observations that do not fit in any model. Therefore, Kim et al in [19] used edge features to differentiate the local regions in the chest xrays using proper discrimination of edge features. Since the diagnosis using medical images need high resolution images, the energy consumed for computation is huge and hence a dimension reduction method is introduced in this literature. An outlier detection model was designed by extracting the edges from the difference in the adjacent segments. These edges in binary format were obtained using AND operation. A detection filter was used to obtain the series of lines that were classified into 16 different types. Top ranking Eigen values extracted from the covariance matrix of the datasets could yield the important features but with reduced dimensions could help the researchers in computation time and energy consumption. The features extracted from Laplacian, sobel and canny operators were applied to recurrent neural network (RNN) to identify the covid images. The proposed fusion method achieved 98.9% accuracy, while the Laplace, Sobel, and Canny methods achieved 95.2%, 97.6%, and 97.9%.

System architecture:

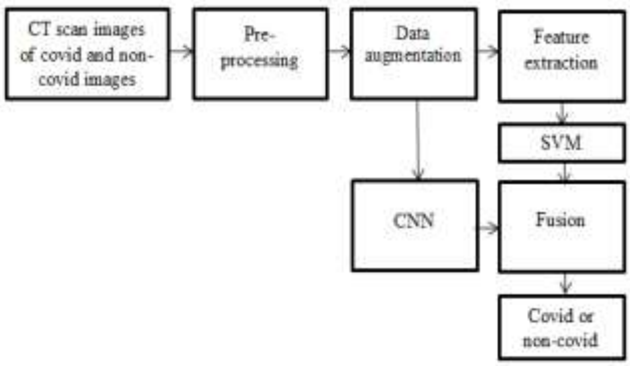


Image Augmentation

The number of images available in some of the research works including covid detection using CT scan images, are usually less when compared to other research problems. Since the images are to be acquired from real human subjects that involve tedious ethical issues, the ultimate number of images is lesser. Unfortunately the accuracy performance of machine learning algorithms is proportional to the size of the database. The algorithms learn more from larger database. Therefore, it is essential to implement Image augmentation to increase the number of images in the database. Image enhancement is the process of changing the appearance of an image without changing the core information of the image. The extension methods used in this project are: Thanks to the Keras image data generator feature that helps you create multiple images with different extension types. Note that all other extensions change the geometry of the image, except that it changes the brightness of the image.

Horizontal flip

Inversion actually means flipping the data row by row or column by column. Horizontal flip actually rearranges the columns so that the last column maps to the first column. The reverse is also true. Therefore, the resulting inverted image is a mirror image of the actual input. Vertical inversion is not used in this project as it changes the overall appearance of the image. Vertical inversion is performed by swapping the positions of the lines. Horizontal mirroring is achieved in the function using the following input arguments:

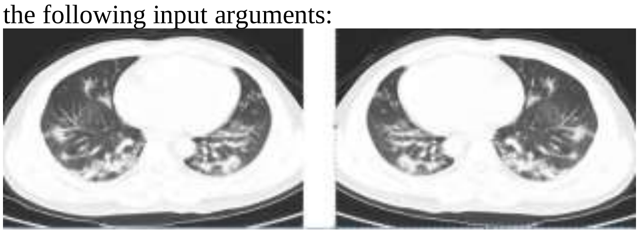
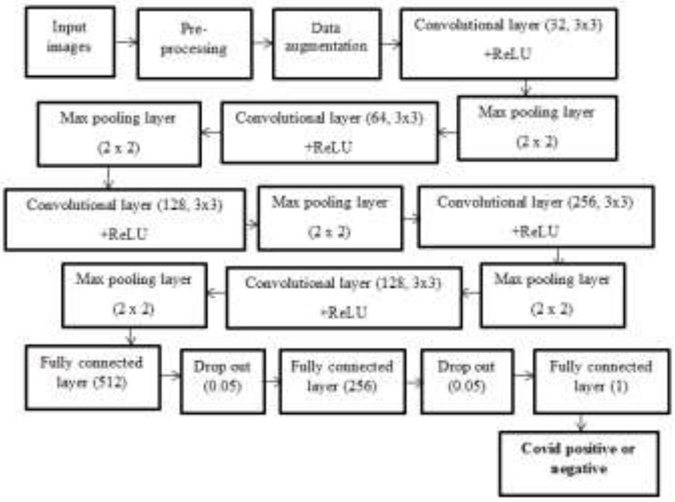


Image pre-processing

Before the actual classification, all train and test images should be a common criterion. The term "standard" here means the image size, brightness, contrast, and normalized values of a pixel. In a real scenario, the data used by the machine algorithm is in a different domain. For example, a pixel value from 0 to 255 could not be compared to a variable in the range 100 to 10000, for example.

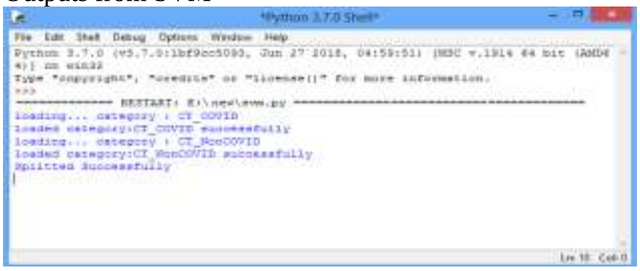
Implementation of CNN

The complete architecture of the CNN is shown in Figure 3.10. The number and rules of filters used at each level are displayed in individual blocks that represent the level. All layers are connected in sequence in a feedforward architecture. It was decided to implement the code using Python software on the TensorFlow platform using Keras. Therefore, the installation is Python software and learning how to code was done with the help of tutorials provided on the official Python website www.python.org. There were two ways to install the TensorFlow package: Anaconda-based installation and Pip-based installation. We recommend that you install the package from pip.



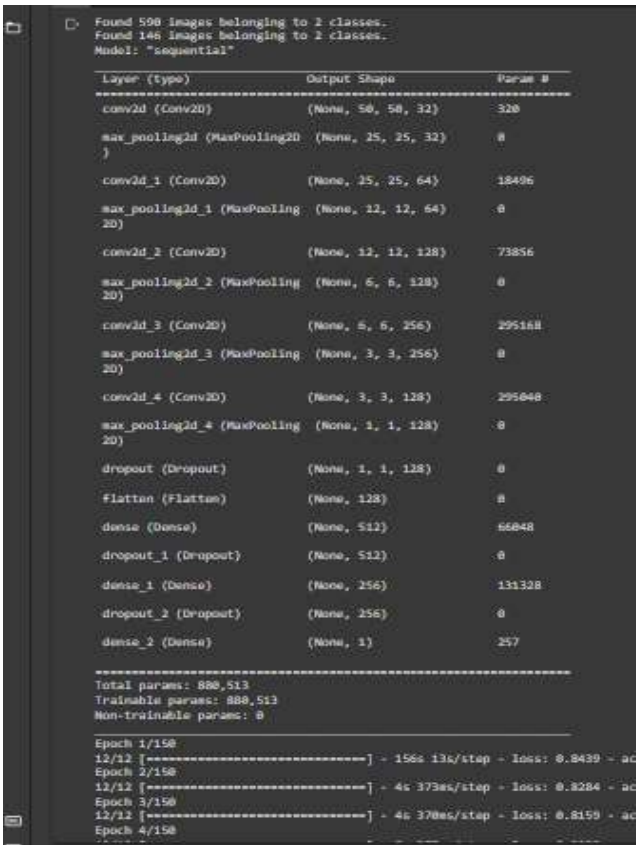
RESULT;

Outputs from SVM



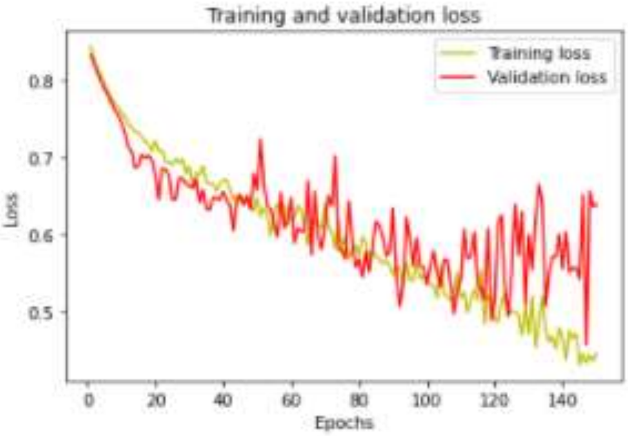
PERFORMANCE EVALUATION

CNN performance is evaluated based on training accuracy, validation accuracy, training loss, and validation loss. The total number of trainable parameters that CNN uses for an image size of 200x200 is 3,174,273. There are a total of 736 images in the original database, with 344 Covid lung images and 392 non-Covid lung images. The number of images in the train database is increased using eight parameters as described in Section 3.2, and each image is converted to eight versions of the original image. Therefore, nine images will be available after expansion, including the original image. Eventually, the extension will create a 9 x 736 = 6624 image in the train database. Since this is the default setting, it was decided to use 20% of the validation dataset from the entire training data. Therefore, we used 1325 images for validation and 5299 images for validation. Ten new images were considered during the test phase to predict the model. It was confirmed that the image used for validation was not used for testing purposes. The test data was generated without expansion. However, linear scaling was performed to transform the pixel values between the range 0 and 1.

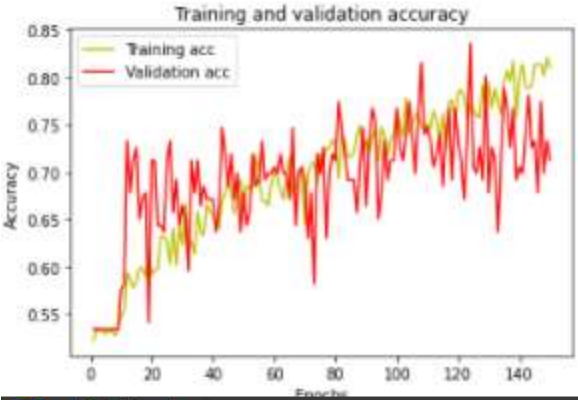


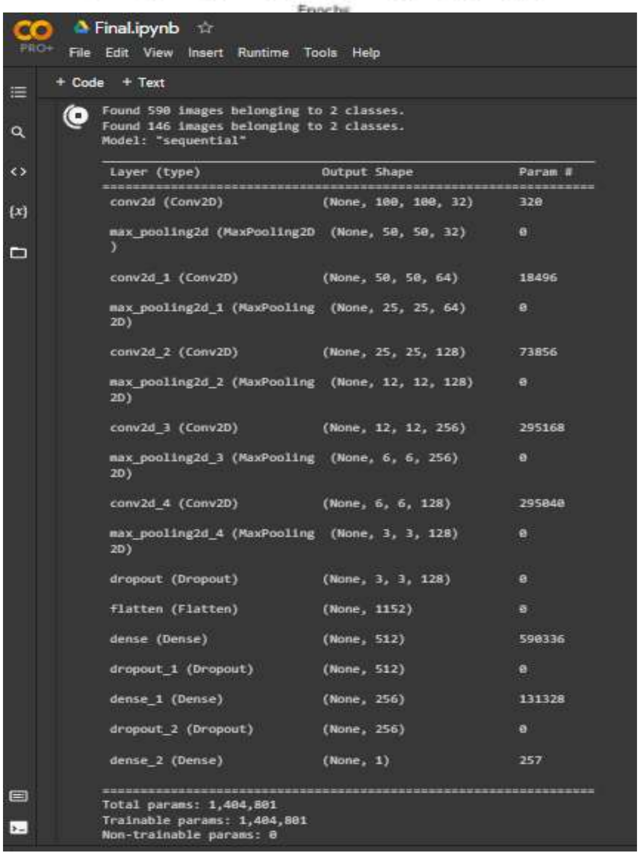
```
Found 590 images belonging to 2 classes.
Found 146 images belonging to 2 classes.
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 50, 50, 32)        320

 max_pooling2d (MaxPooling2D  (None, 25, 25, 32)       0
 )

 conv2d_1 (Conv2D)           (None, 25, 25, 64)        18496

 max_pooling2d_1 (MaxPooling  (None, 12, 12, 64)       0
 2D)

 conv2d_2 (Conv2D)           (None, 12, 12, 128)       73856

 max_pooling2d_2 (MaxPooling  (None, 6, 6, 128)        0
 2D)

 conv2d_3 (Conv2D)           (None, 6, 6, 256)         295168

 max_pooling2d_3 (MaxPooling  (None, 3, 3, 256)        0
 2D)

 conv2d_4 (Conv2D)           (None, 3, 3, 128)         295040

 max_pooling2d_4 (MaxPooling  (None, 1, 1, 128)        0
 2D)

 dropout (Dropout)           (None, 1, 1, 128)         0

 flatten (Flatten)           (None, 128)               0

 dense (Dense)               (None, 512)               66048

 dropout_1 (Dropout)         (None, 512)               0

 dense_1 (Dense)             (None, 256)               131328

 dropout_2 (Dropout)         (None, 256)               0

 dense_2 (Dense)             (None, 1)                 257

=================================================================
Total params: 880,513
Trainable params: 880,513
Non-trainable params: 0
_________________________________________________________________
Epoch 1/150
12/12 [==============================] - 156s 13s/step - loss: 0.8439 - ac
Epoch 2/150
12/12 [==============================] - 4s 373ms/step - loss: 0.8284 - ac
Epoch 3/150
12/12 [==============================] - 4s 370ms/step - loss: 0.8159 - ac
Epoch 4/150
```

Figure 4.2 and Figure 4.3 show the training and validation loss and training accuracy and validation accuracy performance at 150 epochs when the learning rate is set to 0.0002. The yellow curve shows training and the red curve shows validation performance. To maintain generality, the number of epochs was set to 150 per target image size.



It was observed that the loss function decreased exponentially, reaching a value of 0.4467. However, the verification loss was found to fluctuate and was 0.6401. It was found that with proper fitting, a dropout layer was added to the CNN model so that the training run would be done with less vibration. Note that validation is performed on all characteristics without considering dropouts. Also, the number of images in the validation set is only 20% of the total database. Therefore, validation can provide some degree of randomness in terms of loss and accuracy..





```
Final.ipynb
File Edit View Insert Runtime Tools Help
+ Code + Text
Found 590 images belonging to 2 classes.
Found 146 images belonging to 2 classes.
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 100, 100, 32)      320

 max_pooling2d (MaxPooling2D  (None, 50, 50, 32)       0
 )

 conv2d_1 (Conv2D)           (None, 50, 50, 64)        18496

 max_pooling2d_1 (MaxPooling  (None, 25, 25, 64)       0
 2D)

 conv2d_2 (Conv2D)           (None, 25, 25, 128)       73856

 max_pooling2d_2 (MaxPooling  (None, 12, 12, 128)      0
 2D)

 conv2d_3 (Conv2D)           (None, 12, 12, 256)       295168

 max_pooling2d_3 (MaxPooling  (None, 6, 6, 256)        0
 2D)

 conv2d_4 (Conv2D)           (None, 6, 6, 128)         295040

 max_pooling2d_4 (MaxPooling  (None, 3, 3, 128)        0
 2D)

 dropout (Dropout)           (None, 3, 3, 128)         0

 flatten (Flatten)           (None, 1152)              0

 dense (Dense)               (None, 512)               590336

 dropout_1 (Dropout)         (None, 512)               0

 dense_1 (Dense)             (None, 256)               131328

 dropout_2 (Dropout)         (None, 256)               0

 dense_2 (Dense)             (None, 1)                 257

=================================================================
Total params: 1,404,801
Trainable params: 1,404,801
Non-trainable params: 0
```

Loss and accuracy performance of the trained CNN model

Image size and Epochs	Total number of trainable parameters	Loss (%)		Accuracy (%)	
		Training	Validation	Training	Validation
50 x 50 (150)	880,513	0.4467	0.6401	0.8102	0.7123
100 x 100 (150)	1,404,801	0.4021	0.5160	0.8254	0.7877
150 x 150 (150)	1,863,553	0.3137	0.7434	0.8627	0.7055
200 x 200 (150)	3,174,273	0.3264	0.6170	0.8712	0.7877

Performance of SVM

Image size	Accuracy (%)	Precision		Recall		F1 Score	
		#0	#1	#0	#1	#0	#1
20 x 20	84.45	0.82	0.87	0.86	0.84	0.84	0.85
30 x 30	83.78	0.82	0.86	0.84	0.84	0.83	0.85
40 x 40	85.13	0.82	0.88	0.87	0.84	0.85	0.86
50 x 50	85.13	0.83	0.87	0.86	0.85	0.84	0.86
100 x 100	85.81	0.86	0.85	0.83	0.89	0.84	0.87

CONCLUSIONS

The method of combined CNN and SVM could yield a better accuracy of _ % against the accuracies of _ and _ % respectively from the individual CNN and SVM method. This clearly shows that combing the features of deep learning model and the conventional SVM works out practically.

REFERENCE

[1]. WHO, Coronavirus disease (COVID-19), [Online] Available from: https://www.who.int/healthtopics/coronavirus#tab=tab_1, [Accessed] 10th May 2021.

[2]. Covid-19, b. https://radiopaedia.org/

[3] COVID-CT-Dataset: A CT Scan Dataset about COVID-19 [Online] Available from: https://www.kaggle.com/luisblanche/covidct

[4] S. Singh and P. K. Jain, "COVID-19 Diagnosis using Laplacian Edge Detection & Morphological Dilation," 2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC), 2021, pp. 1164-1169, doi: 10.1109/ICESC51422.2021.9532789.

[5] D. Dong et al., "The role of imaging in the detection and management of COVID-19: a review," in IEEE Reviews in Biomedical Engineering, doi: 10.1109/RBME.2020.2990959.

[6] Jamshidi, M.B., et al., "Artificial Intelligence and COVID-19: Deep Learning Approaches for Diagnosis and Treatment," IEEE Access, 8, pp. 109581-109595, 2020, doi: 10.1109/ACCESS.2020.3001973.

[7] H. Kang et al., "Diagnosis of Coronavirus Disease 2019 (COVID-19)with Structured Latent Multi-View Representation Learning," in IEEE Transactions on Medical Imaging, doi:10.1109/TMI.2020.2992546

[8] Zebari, D.A., Abdulazeez, A.M., Zeebaree, D.Q., & Salih, M.S. (2020a) "A Fusion Scheme of Texture Features for COVID-19 Detection of CT Scan Images," Proceedings of 2020 International Conference on Advanced Science and Engineering (ICOASE), 2020, pp. 1-6, doi: 10.1109/ICOASE51841.2020.9436538.

[9] Zebari, R., Abdulazeez, A., Zeebaree, D., Zebari, D., & Saeed, J. (2020b) "A Comprehensive Review of Dimensionality Reduction Techniques for Feature Selection and Feature Extraction," Journal of Applied Science and Technology Trends, 1(2), pp. 56-70, 2020.

[10] Foysal, M, & Hossain, A.B.M.A. (2021) "COVID-19 Detection from Chest CT Images using Ensemble Deep Convolutional Neural Network," Proceedings of 2nd International Conference for Emerging Technology (INCET), pp. 1-6.

[11] Motamed, S., Rogalla, P., & Khalvati, F. (2021) RANDGAN: Randomized generative adversarial network for detection of COVID-19 in chest X-ray, scientific reports, 11, 8602.

[12] Ning, W. et al, (2021) Open resource of clinical data from patients with pneumonia for the prediction of COVID-19 outcomes via deep learning, Nature Biomedical engineering, 4, 1197-1207.

[13] M. R. Islam and A. Matin, "Detection of COVID 19 from CT Image by The Novel LeNet-5 CNN Architecture," 2020 23rd International Conference on Computer and Information Technology (ICCIT), 2020, pp. 1-5, doi: 10.1109/ICCIT51783.2020.9392723.

[14] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, " Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278-2324, 1998.

[15] X. Yang, X. He, J. Zhao, Y. Zhang, S. Zhang, and P. Xie," COVIDCT-dataset: a CT scan dataset about COVID-19," arXiv, pp.arXiv-2003, 2020.

[16] M. Loey, G. Manogaran, and N.E.M. Khalifa,"A deep transfer learning model with classical data augmentation and cgan to detect covid- 19 from chest CT radiography digital images," Neural Computing and Applications, pp. 1-13, 2020.

[17] Y. Peng, Y. Tang, S. Lee, Y. Zhu, R. M. Summers and Z. Lu, "COVID-19-CT-CXR: A Freely Accessible and Weakly Labeled Chest X-Ray and CT Image Collection on COVID-19 From Biomedical Literature," in IEEE Transactions on Big Data, vol. 7, no. 1, pp. 3-12, 1 March 2021, doi: 10.1109/TBDATA.2020.3035935.

[18] T. Anwar and S. Zakir, "Deep learning based diagnosis of COVID-19 using chest CT-scan images," 2020 IEEE 23rd International Multitopic Conference (INMIC), 2020, pp. 1-5, doi: 10.1109/INMIC50486.2020.9318212.

[19] Kim, C.M., Hong, E.J., & Park, R.C. (2021) "Chest X-Ray Outlier Detection Model Using Dimension Reduction and Edge Detection," IEEE Access, 9, 86096-86106, 2021, doi: 10.1109/ACCESS.2021.3086103.

[20] Ahuja, S., Panigrahi, B.K., Dey, N. et al. Deep transfer learning-based automated detection of COVID-19 from lung CT scan slices. Appl Intell 51, 571–585 (2021).

[21] Hinton, Geoffrey (2012). "ImageNet Classification with Deep Convolutional Neural Networks". NIPS'12: Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1. 1: 1097–1105 – via ACM.

[22] Coenraad,M., ; Myburgh, Johannes C.; Davel, Marelie H. (2020). Gerber, Aurona (ed.). "Stride and Translation Invariance in CNNs". Artificial Intelligence Research. Communications in Computer and Information Science. Cham: Springer International Publishing. 1342: 267–281. arXiv:2103.10097. doi:10.1007/978-3-030-66151-9_17. ISBN 978-3-030-66151-9. S2CID 232269854.

[23] Deshpande, (2019) Understanding CNN part 3, [Online] Available from: https://adeshpande3.github.io/The-9-Deep-Learning-Papers-You-Need-To-Know-About.html, [Accessed] 31st December 2021.

[24] Saedsayad, (2020) An Introduction to Support Vector Regression (SVR), [Online] Available from: https://www.saedsayad.com/support_vector_machine_reg.htm, [Accessed] 31st December 2021.

[25] Stéfan van der Walt, Johannes L. Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D. Warner, Neil Yager, Emmanuelle Gouillart, Tony Yu and the scikit-image contributors. scikit-image: Image processing in Python. PeerJ 2:e453 (2014) https://doi.org/10.7717/peerj.453.

[26] Scikit-image, (2021) Image processing in Python, [Online] Available from: https://scikit-image.org/, [Accessed] 31st December 2021.