

SEGMENTATION AND CLASSIFICATION OF COVID-19 MEDICAL IMAGES USING DEEP NEURAL NETWORK

Sreejith M^{1,*}, Dr R Benschwartz², Dr.V Suresh³

¹ ME Student, Department of ECE, Mar Ephraem College of Engineering and Technology, Elavuvilai

² Professor, Department of ECE, Mar Ephraem College of Engineering and Technology, Elavuvilai

³ Head of Department, Department of ECE, Mar Ephraem College of Engineering and Technology, Elavuvilai

Abstract— Corona Virus Disease-2019 (COVID-19), caused by Severe Acute Respiratory Syndrome-Corona Virus-2 (SARS-CoV-2), is a highly contagious disease that has affected the lives of millions around the world. Chest X-Ray (CXR) and Computed Tomography (CT) imaging modalities are widely used to obtain a fast and accurate diagnosis of COVID-19. However, manual identification of the infection through radio images is extremely challenging because it is time-consuming and highly prone to human errors. Artificial Intelligence (AI)-techniques have shown potential and are being exploited further in the development of automated and accurate solutions for COVID-19 detection. In this project, try to establish a new deep convolutional neural network tailored for segmenting the chest CT images with COVID-19 infections as well as the entire lung from chest CT images, referred to as COVID-SegNet. Besides using local binary pattern for feature extraction and its classification using DAENN. Inspired by the observation that the boundary of the infected lung can be enhanced by adjusting the global intensity, in the proposed deep CNN, introduce a feature variation block which adaptively adjusts the global properties of the features for segmenting COVID-19 infection. The proposed FV block can enhance the capability of feature representation effectively and adaptively for diverse cases. We fuse features at different scales by proposing Progressive Atrous Spatial Pyramid Pooling to handle the sophisticated infection areas with diverse appearance and shapes. The proposed method achieves state-of-the-art performance. Dice similarity coefficients are 0.987 and 0.726 for lung and COVID-19 segmentation, respectively. The proposed network enhances the segmentation ability of the COVID-19 infection, makes the connection with other techniques and contributes to the development of remedying COVID-19 infection

Keywords—: Coronavirus disease 2019 pneumonia, COVID-19, deep learning, segmentation, multi-scale feature

I. INTRODUCTION

World Health Organization (WHO) formally announced the outburst of a virus COVID19, the infection affected through Severe Acute Respiratory Syndrome (SARS) CoV-2 in March 2020. COVID-19 is extremely contagious and may become a lethal syndrome of Acute Respiratory Distress (ARDS). Initial exposure and analysis are the key elements in controlling the spread of virus disease. The most common screening process to detect and test the virus is Reverse-Transcription Polymerase Chain Reaction. Though it is a strenuous operation and several analyses have stated its less sensitivity in the primary stage [1]. CT screening dataset to diagnose the latest COVID-19 corona, SARS-COV-2, Virus ailment, which first appeared in Wuhan, China in December 2019 [2] and has caused 64,384 deaths to date, nearly one million confirmed cases worldwide, as of 5 April 2020, with many more to be tested [3]. However, for a large number of suspected and asymptomatic compromised patients, all affected patients will not be inspected and healed. Owing to the high rate of infection, a rapid diagnosis using Artificial Intelligence (AI) is becoming predictable. An accurate and quick analysis of COVID-19 can distinguish infectious patients to postpone disease progression [4].

An extensive selection of developers armed with a cloud subscription can now access knowledge that was once the domain of hardcore AI professionals. Machine Learning (ML) was once the domain of data scientists. Since 2016, we have seen the development of simpler models, granting independence to developers and the ability to create complex machine learning models with minimal effort and expertise. This wizard-like environment for graphical development allowed non-data scientists to play with their hands with ML models. With the Rapid Application Development (RAD) process, people with no experience in data science, equipped with a limited amount of data and transfer learning technology, build advanced, industry-standard production-ready AI models.

The availability of machine learning algorithms made available as a service has Microsoft Cognitive Services

included in the Microsoft Azure cloud solution. This ML framework allows developers to upload their images and create computer vision for Microsoft Azure Custom Vision. It is based on a pre-trained CNN and offers a technique of transfer learning for users. By simply importing the training images and marking them into a model generator, it enables AI models to create using the web application Custom Vision. Because of the transfer learning technology, which starts with a pre-trained model and uses this model as a feature extractor, Custom Vision does not require as many images for training and testing as the regular CNN. A minimum requirement of 50 images per tag suggested.

In Custom Vision, create templates and use a REST API as a web application to run them. On the Microsoft Custom Vision documentation site for Curl, C #, Java, Javascript, ObjC, PHP, Python, Ruby, sample code documentation is available. Custom Vision is the state-of-the-art machine learning technology that enables its trained model export to Tensorflow, Tensorflowlite, Tensorflowjs, CoreML, ONNX, and Dockerfile formats by multiple platforms easily. Custom Vision currently enables image recognition and target detection. In Image classification, it has two subclasses, 1. Multilabel (Multiple tags, per image), 2. Multiclass (single tag per image) in four distinct domains. Such domains are pre-trained so that few sample images can train effectively. The lightweight domain custom Vision models can be passed to smartphones and edge computers to achieve real-time on-device inference.

A CT scan produces accurate organ images, muscles, fleshy tissue, and lifeblood vessels. Doctors can discern interior structures, shape, scale, texture, and density through CT images [5]. Thus, CT scans have far further accurate image conditions of the patients than standard X-Rays [6]. It is necessary to use this basic information to decide when a medical condition occurs as well as the degree and specific position of the issue. Aimed at these purposes, various DL methodologies for COVID-19 screening [7–9] have recently suggested in CT scans.

The primary bottleneck of a study is the absence of excellent quantitative datasets, such as those cited above. The COVID-CT dataset contains images excavated after academic journals were possibly the main effort to construct such a dataset [10]. For the most revised version, the precision, Area Under the Curve, and F1-score were 86%, 94%, and 85%, respectively [11, 12]. Another collection of CT scans has been made widely accessible and comprises 2482 CT images taken from clinics in Sao Paulo, Brazil [13]. For precision, sensitivity, and predicatively, reported 97.38 percent, 95.53% and 99.16% respectively. We summarize our contributions as follows:

- To develop a classification model, which classify the CT images of COVID19 and healthy people
- To design a novel deep neural network for the segmentation of COVID-19 infection regions from chest CT images
- Introduce a Progressive Atrous Spatial Pyramid Pooling (PASPP), to obtains more effective contextual features

II. LITERATURE SURVEY

To derive any sort of inference datasets are very crucial. If the use-case belongs to the healthcare domain, then proper use of DL Technique data points will be useful for building applications in the real world. In different studies, the COVID dataset is coming out of organizations and labs [14]. On Kaggle, dataset obtainable Platform: <https://www.kaggle.com/novelcorona/sudalairajkumar-Dataset-2019-virus>. This dataset has various characteristics such as patient placement, age, symptoms, etc. Johns Researchers, the dataset was prepared by Hopkins University and made available to <https://github.com/ieee8023/covidchestxraydataset> for research :README.md/blob/master?fbclid=IwAR30yTGBr55Br55WXdCngCoICDENHycmdL2bGwlv1lckdZMucjGH10Uakz7MucjGH10Uakz7 Uh, khk. The following are other recent important repositories: <https://github.com/mattroconn> or [deep_learning_coronavirus_cure](https://github.com/deep_learning_coronavirus_cure) [15].

DL based segmentation method is for identifying the region of interest in CT lung images for COVID-19 quantification. 2D and 3D DL techniques proposed, and accuracy of 99.6 percent in COVID-19 background [16]. Gainesville Regional Utilities (GRU) used bidirectional mechanisms to scale the suggested solution from large COVID-19 datasets [17]. VGG-16 was used to identify the CT images on COVID-19 produces 86.7% accuracy on benchmark datasets using the DL context. Different attempts made in the research community to expose the negative effects of the corona virus mitigated.

The diagnosis of COVID-19 is accomplished by the multimodal and deep learning approach. In this work X-Ray and CT scans are utilized in the disease identification and the augmentation based identification is also accomplished by DNN [18, 19]. The information transmitted over the internet or stored in the cloud can be secured with Blockchain technology and other cryptography mechanism [20, 21]. The light weight protocol and other cryptography mechanism are developed for securing the COVID and patient information over the internet [22, 23].

III SYSTEM OVERVIEW

Here propose a unified high-accuracy network for the segmentation of COVID-19 infection from chest CT images. This network consists of two parts: Encoder and Decoder. As shown in Fig. 1, the encoder with 4 layers (i.e. E1, E2, E3, E4) obtains robust information via feature extractor and PASPP. Each layer employs residual and FV blocks as the basic operations for feature extractors, except the E4 layer. The residual block adds up the input features and the results after two convolutional layers, which effectively alleviates the vanishing gradient. To preserve multiple contextual information and enlarge the receptive field, we use PASPP with different dilate rates on the final E4 layer. After obtaining the encoded features, the decoder tries to restore the features to its original input size, which can remove the information loss induced by down-sampling from Encoder. The decoder has three layers (D3, D2, D1). Each decoder layer allows the networks to gradually propagate the global contextual information to a higher resolution layer. After a sigmoid activation function, we obtain the final segmentation of COVID-19 infection regions. In addition, the skip connection is adopted to concatenate the output features of the encoder and input features of the decoder. In this project, the main objective is improve the encoder by adding FV block and PASPP block to better capture effective features.

After segmentation uses local binary pattern to extract learning features and classification is done using Deep Auto Encoder Neural Network classifier

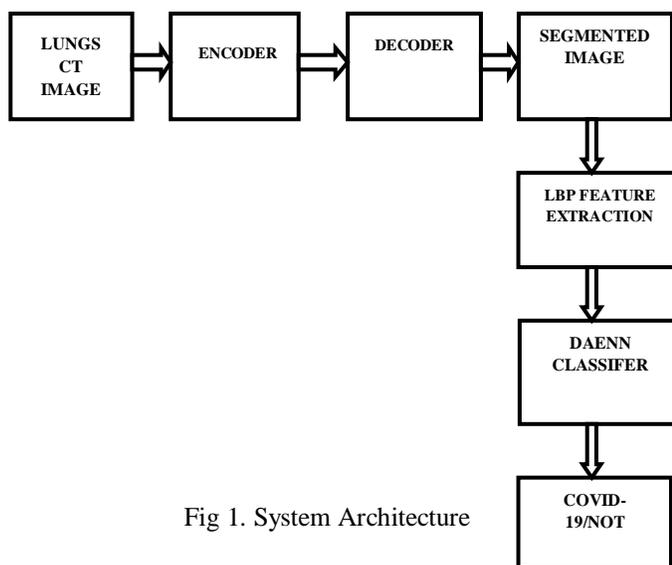


Fig 1. System Architecture

IV PROPOSED SCHEMES

The proposed system includes the feature variation block and progressive atrous spatial pyramid pooling block for infection segmentation and to distinguish the COVID-19 and Health samples from chest CT images using LBP and Deep Autoencoder Neural Network(DANN).

3.1 Segmentation

Introduce the architectures of FV block by considering a material fact, the boundaries of COVID-19 infection regions are highlighted by adjusting the window breadth and window locations. As shown in Fig. 2, the proposed FV block includes three branches, i.e., contrast enhancement branch, position sensitive branch, and identity branch. Specifically, the contrast enhancement branch learns a global parameter via a channel attention unit to highlight useful boundary information. The position sensitive branch obtains a weight map by spatial attention unit to focus on the COVID-19 regions. Finally, the FV block preserves more useful information by fusing these refined features

The PASPP block takes the featured extracted with FV block as input and acquires semantic information with different receptive fields showing in Fig. 3. Although ASPP has been proposed to capture global information for semantic segmentation, claim that aggregating information progressively is a more reasonable approach to get effective features. The PASPP block adopts atrous convolutions with different dilation rates to obtain features with various scales. The final output is generated straightforwardly to assemble residual branches in parallel.

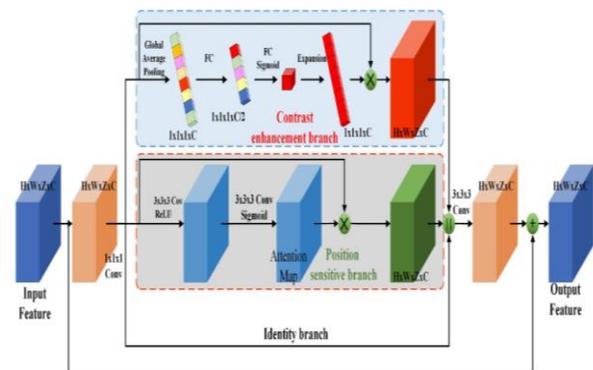


Fig.2 FV block consists of a contrast enhancement branch, position-sensitive branch, and identity branch.

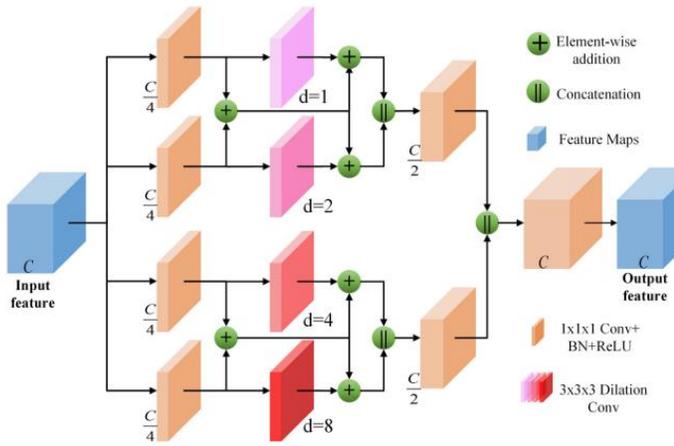


Fig. 3. The structure of the PASPP block.

3.2 Feature Variation

As mentioned before, the boundaries of COVID-19 infection regions are highlighted by adjusting the window breadth and window locations. In Fig. 3, the designed FV block, which includes contrast enhancement branch, position sensitive branch and identity branch, tries to enhance the contrast of features and highlight the useful regions. Let Fv_{in} denotes the input feature, the features after $1 \times 1 \times 1$ represent Fv_1 . The output feature Fv_{out} is given as

$$Fv_{out} = Fv_{in} + Cov_3(Conca(C(Fv_1), P(Fv_1), Fv_1)) \quad (1)$$

where $Cov_3(.)$ denotes the $1 \times 1 \times 1$ convolutional layer, $Conca(.)$ is the concatenation operation, $C(.)$ represents the contrast enhancement branch, $P(.)$ is the position sensitive branch. The form of residual learning in Eq implies that the information from the early blocks can quickly flow to the later blocks, and the gradient can be quickly back-propagated to the early blocks from the later blocks. The details of each submodule are as follows.

A) Contrast Enhancement Branch

To enhance the contrast of features, the contrast enhancement branch $Con(.)$ in Eq attempts to learn a global parameter F_g for input feature Fv_1 (See Fig. 2). The corresponding function is given as

$$F_g = FC(FC(GAP(Fv_1)))$$

Where $FC(.)$ denotes the fully convolutional layer, $GAP(.)$ represents global average pooling. The values of F_g is in the range. We obtain a channel weight map F'_g via expansion, thus the number of F'_g is consistent with Fv_1 . Finally, the output of contrast enhancement branch F_c can be formulated as below

$$F_c = F'_g \otimes Fv_1. \quad (2)$$

Where \otimes denotes the element-wise multiplication. Note that, instead of calculating a sequence of weight for feature Fv_1 , we generate one weight for all the features of Fv_1 . This process is exactly corresponding to adjust the window breadth and window locations. Thus we deem it has the ability to generate enhanced features.

B) Position Sensitive Branch

The goal of the position-sensitive branch is to discard harmful information and highlight the helpful features used to segmentation COVID-19 infection. This branch $P(.)$ is a small network. The architecture of the position-sensitive branch is displayed in Fig. 3.1. The attention map A is calculated using input feature Fv_1 after two convolutional layers. Each layer adopts $1 \times 1 \times 1$ convolution. The two convolutional layers are followed by a ReLU function and sigmoid function, respectively. In the end, the output of this branch F_p is obtained by element-wise multiplication between Fv_1 and the attention map

$$F_p = A \otimes Fv_1 \quad (3)$$

The values in A are still in the range $[0, 1]$. The attention map has the same size as the input feature.

3.3 Progressive Atrous Spatial Pyramid Pooling:

In this subsection, start with a preliminary knowledge of atrous spatial pyramid pooling, then introduce the proposed PASPP block.

A) Atrous Spatial Pyramid Pooling

Global information captured by a large receptive field inessential for medical semantic segmentation. To increase the receptive field size and decrease the number of convolutional layers, atrous convolution was first proposed to obtain enough global information while keeping the size of the feature map unchanged. In one dimensional case, let $y(i)$ represents output and $x(i)$ denotes input, atrous convolution can be formulated as follows:

$$y[i] = \sum_{k=1}^K x[i + d \cdot k] \cdot w[k], \quad (4)$$

where K denotes the filter size, d represents the dilation rate, and $w[k]$ is the k^{th} parameter of filter. A larger dilation rate will capture a larger receptive field. To produce different receptive fields, atrous spatial pyramid pooling takes atrous convolutions with different dilation rates to generate various scales. These features are concatenated together. Thus the outputs are indeed a sampling of the input with different scales information.

B) The PASPP Block

In the COVID-19 segmentation task, the infection regions often have very different sizes. To alleviate this

dilemma, the features must be able to include different receptive fields. The structure of PASPP is illustrated in Fig. 3. For this goal, we employ ASPP in our network and progressively fuse the features with different receptive fields. Given the input feature of PASPP F_{pin} , we obtain four features $F_{p1}, F_{p2}, F_{p3}, F_{p4}$ by four $1 \times 1 \times 1$ convolutional layers in parallel. Note that, compared to input features, the number of the channel decreases to quarter after each $1 \times 1 \times 1$ convolutional layer. Then each branch feeds the feature into different atrous convolutional layer, respectively. The corresponding function is given as

$$Fd_t = Cov_3^d(Fp_t), \quad t = 1, 2, 3, 4; d = 2^{t-1},$$

where Cov_3^d denotes the $3 \times 3 \times 3$ atrous convolutional layer with dilation rate d , Fd_t represents the output feature of the t th branch after Cov_3^d . Sum the inputs of two adjacent atrous convolution branches, and add the sum to the output of each residual branch as the input of the subsequent layer. It is formulated as below:

$$\begin{cases} Fd'_t = Fd_t + Fp_1 + Fp_2, & t = 1, 2 \\ Fd'_t = Fd_t + Fp_3 + Fp_4, & t = 3, 4, \end{cases} \quad (6)$$

where Fd'_t denotes the output features of t th branch. To get effective features, Fd'_t , $t = 1, 2, 3, 4$ will be progressively aggregated based on adjacent features in parallel

$$\begin{cases} Fd''_1 = Cov_1(Conca(Fd'_1, Fd'_2)) \\ Fd''_2 = Cov_1(Conca(Fd'_3, Fd'_4)) \end{cases} \quad (7)$$

The Fd''_1 tends to fuse the information with small receptive field, Fd''_2 prones to capture features with larger receptive field. The channel's number of Fd''_1 and Fd''_2 is half of the input feature. All the information are assembled by

$$Fp_{out} = Cov_1(Conca(Fd''_1, Fd''_2)), \quad (8)$$

where Fp_{out} denotes the output features of PASPP block.

3.4 Classification

Local Binary Pattern

Local Binary Pattern (LBP) is an effective texture descriptor for images which thresholds the neighboring pixels based on the value of the current pixel. LBP descriptors efficiently capture the local spatial patterns and the gray scale contrast in an image.

LBP has been widely used in many computer vision applications. However, it is to be noted that LBP is popularly proposed as a feature descriptor, while we propose to use LBP as an image enhancement technique. Computation of the LBP descriptor from an image is a four-step process and is explained below.

- For every pixel (x,y) in an image, I , choose P neighboring pixels at a radius R .
- Calculate the intensity difference of the current pixel (x,y) with the P neighboring pixels.
- Threshold the intensity difference, such that all the negative differences are assigned 0 and all the positive differences are assigned 1, forming a bit vector.
- Convert the P -bit vector to its corresponding decimal value and replace the intensity value at (x,y) with this decimal value.

Thus, the LBP descriptor for every pixel is given as

$$LBP(P, R) = \sum_{p=0}^{P-1} f(g_p - g_c) 2^p \quad (9)$$

where g_c and g_p denote the intensity of the current and neighboring pixel, respectively. P is the number of neighboring pixels chosen at a radius R .

Deep auto-encoder neural network

In this paper the range for RBM is fixed between 0 and 1. In the training function which is used for activation purpose is taken from the previous layer. The first layer has a gaussian variance whereas others have a binary value. There is a nonlinear relationship between the pixel value and the numbers used for framing the curve. The number of images considered for training is 25 images and for testing it is 5 images.

The cross entropy is given by

$$-\sum_n P_n \log P_n - \sum_n (1 - P_n) \log (1 - P_n) \quad (10)$$

Where P_n and P_{1n} are the intensity of pixel and reconstruction respectively.

A pretraining procedure is handled for initializing the weight vector. Using restricted boltzman machine the feature vector of hidden layers are obtained. The energy function that has to be minimized is given by the equation

$$E(V, H) = -\sum_{m \in \text{pixels}} B_m V_m - \sum_{n \in \text{feature}} B_n H_n - \sum_{m,n} V_m H_n w_{m,n} \quad (11)$$

The weight that has to be updated is given by

$$w_{m,n} = \alpha ((V_m H_n)_{\text{data}} - (V_m H_n)_{\text{recon}}) \quad (12)$$

Where α , $(V_m H_n)_{\text{data}}$, $(V_m H_n)_{\text{recon}}$ are the learning rate, fraction of times pixel m and feature j are on together and the ratio of confabulations. Here a group of binary features is used, which after tuning the energy function is reduced.

V RESULT AND DISCUSSION

The screening performance of the proposed method is conducted by the Dice similarity coefficient, sensitivity, and precision. The Dice similarity coefficient (Dice) represents a similarity metric between the ground truth, and the prediction score maps

$$Dic(A, B) = \frac{2|A \cap B|}{|A| + |B|}, \quad (13)$$

Where A is the segmented infection region, B denotes the corresponding reference region, $|A \cap B|$ represents the number of pixels common to both images. Sensitivity denotes the number of correctly identified positives with respect to the number of positives. Precision is the fraction of positive instances among the retrieved instances.

A) Segmentation Result

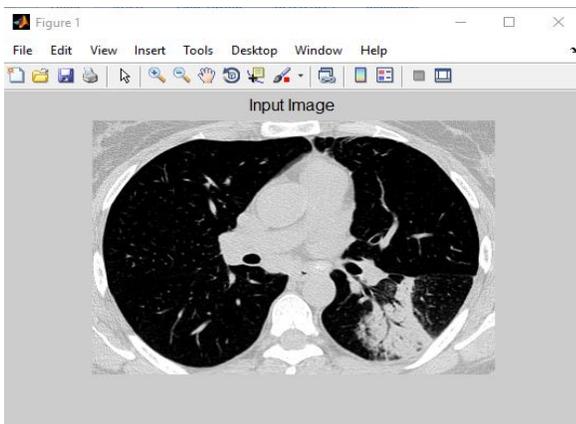


Fig.4 Input Image

The Parameters of the Network. For the proposed framework, the encoding layers are residual blocks, FV blocks, PASSP blocks, and downsampling, while the decoding layers are residual blocks and deconvolution layers kernels with a stride of 1/2. The last layer is a softmax activation function to produce the segmentation results. All layers use 3 3 3 kernels, if not specified otherwise. Each convolutional layer is followed by batch normalization and ReLU. The channel numbers are doubled each layer from 64 to 512 during encoding and halved from 512 to 64 during decoding. We set the combination of dice loss L_d and cross-entropy loss L_c as the loss function using the ground-truth label map. The final loss function is $L_d + 0.5 * L_c$.

Training Details. We implement our COVID-SegNet using Pytorch. For network training, we train all models from scratch with random initial parameters. The entire models are conducted on a server with six Nvidia TITAN RTX GPUs

with 24 GB memory. We randomly crop the 128x128x64 patches as the training samples. For optimization, we use Adam optimizer by setting $\beta_1=0.9$, $\beta_2=0.999$, $\epsilon=10^{-8}$ and batch size is 2. In experiments, the initial learning rate is $1e^{-4}$, and the learning rate decay of $1e^{-6}$. The proposed network will perform both lung and COVID-19 segmentation tasks.

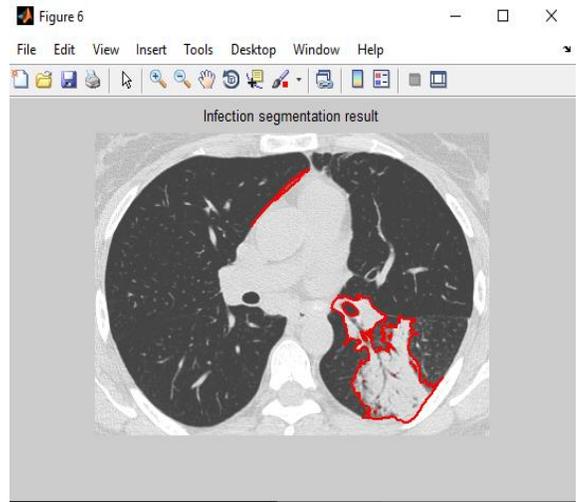
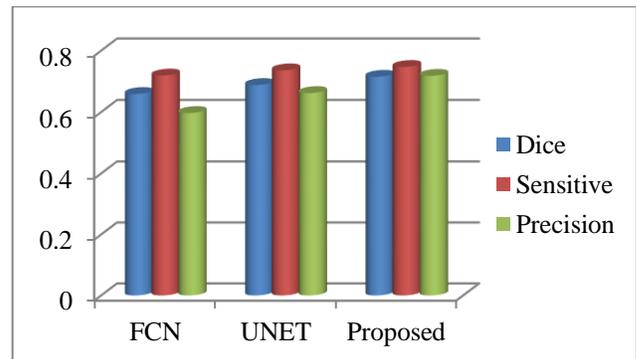


Fig. 5. Infection Segmentation Result

Table .1 Quantitative Comparisons

Metrics	FCN	UNET	Proposed
Dice	0.659	0.688	0.715
Sensitive	0.719	0.736	0.747
Precision	0.597	0.662	0.718

For the segmentation of COVID-19, as shown in Table 1, the results of the proposed method achieves best in all the metrics. Thanks to the FV and PASPP block, the COVID-SegNet can effectively segment COVID-19 infection regions and significantly improve the segmentation performance over the UNet. All these metrics demonstrate the effectiveness of our model.



Comparison graph between proposed and existing methods

B) Classification Result

Texture based feature is used for the feature extraction process. The CT images are used for the evaluation of the qualitative indices such as sensitivity, specificity, F-score and accuracy. It gives more information visually and specifically in the COVID-19 identification of the surface plate.

Table .2: Performance of the proposed method

Measures	Methods	
	Existing	Proposed
Accuracy	95.73%	97.9%
Sensitivity	0.971	0.983
Specificity	0.955	0.964
F Score	0.932	0.945

The extracted feature is given as the input to the classifier for classifying whether the COVID-19 is present or not. Table .2 shows the comparison of proposed method with other methods like CNN.

V1 CONCLUSION

In this project, designed and evaluated a deep learning model, called COVID-SegNet, for segmenting lung and COVID-19 from chest CT images. Besides using local binary pattern for feature extraction and its classification using DAENN. Inspired by contrast enhancement methods and ASPP, the proposed network includes feature variation and progressive ASPP blocks, which are beneficial to highlight the boundary and position of COVID-19 infections. These results demonstrate that the convolutional network-based deep learning technology has the ability to segment COVID-19 from CT images. Our results achieved based on the four types of evaluation metrics which are Accuracy, Precision, Recall, and F1-Score, where all evaluations achieved exceeded of 99%.

V11 REFERENCES

1. Ai, T., Yang, Z., Hou, H., Zhan, C., Chen, C., Lv, W., & Xia, L. (2020). Correlation of chest CT and RT-PCR testing in coronavirus disease 2019 (COVID-19) in China: A report of 1014 cases. *Radiology*, 296, 200642.
2. Ng, M. Y., Lee, E. Y., Yang, J., Yang, F., Li, X., Wang, H., & Hui, C. K. M. (2020). Imaging profile of the COVID-19 infection: Radiologic findings and literature review. *Radiology: Cardiothoracic Imaging*, 2(1), e200034.

3. Hemdan, E. E. D., Shouman, M. A., & Karar, M. E. (2020). Covidx-net: A framework of deep learning classifiers to diagnose covid-19 in x-ray images. *arXiv preprint*.
4. Farooq, M., & Hafeez, A. (2020). Covid-resnet: A deep learning framework for screening of covid19 from radiographs. *arXiv preprint*.
5. Li, T., Han, Z., Wei, B., Zheng, Y., Hong, Y., & Cong, J. (2020). Robust screening of COVID-19 from Chest X-ray via discriminative cost-sensitive learning. *arXiv preprint*.
6. Abbas, A., Abdelsamea, M. M., & Gaber, M. M. (2020). Classification of COVID-19 in chest X-ray images using DeTraC deep convolutional neural network. *arXiv preprint*.
7. Wang, L., & Wong, A. (2020). COVID-Net: A tailored deep convolutional neural network design for detection of COVID-19 cases from chest X-Ray images. *arXiv preprint*.
8. Luz, E. J. D. S., Silva, P. L., Silva, R., Silva, L., Moreira, G., & Menotti, D. (2020). Towards an effective and efficient deep learning model for COVID-19 patterns detection in X-ray images. *Research on Biomedical Engineering*, 21, 1–4.
9. Latchoumi, T. P., & Parthiban, L. (2017). Abnormality detection using weighed particle swarm optimization and smooth support vector machine. *Biomedical Research*, 28(11), 4749–4751.
10. He, X., Yang, X., Zhang, S., Zhao, J., Zhang, Y., Xing, E., & Xie, P. (2020). Sample-efficient deep learning for COVID-19 diagnosis based on CT scans. *medRxiv*.
11. Mobiny, A., Cicalese, P. A., Zare, S., Yuan, P., Abavisani, M., Wu, C. C., & Van Nguyen, H. (2020). Radiologist-level COVID-19 detection using CT scans with detail-oriented capsule networks. *arXiv preprint*.
12. Latchoumi, T. P., Ezhilarasi, T. P., & Balamurugan, K. (2019). Bio-inspired weighed quantum particle swarm optimization and smooth support vector machine ensembles for identification of abnormalities in medical data. *SN Applied Sciences*, 1(10), 1137.
13. Amyar, A., Modzelewski, R., & Ruan, S. (2020). Multi-task deep learning based CT imaging analysis for COVID-19: Classification and segmentation. *Computers in Biology and Medicine*, 126, 104037.
14. Wang, S., Kang, B., Ma, J., Zeng, X., Xiao, M., Guo, J., & Xu, B. (2020). A deep learning algorithm using CT images

- to screen for Corona Virus Disease (COVID-19). MedRxiv.
15. Luz, E., Moreira, G., Junior, L. A. Z., & Menotti, D. (2018). Deep periocular representation aiming video surveillance. *Pattern Recognition Letters*, 114(2–12), 19.
 16. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., & MaBerg, S. A. C. (2015). Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3), 211–252.
 17. Sudharshan, P. J., Petitjean, C., Spanhol, F., Oliveira, L. E., Heutte, L., & Honeine, P. (2019). Multiple instance learning for histopathological breast cancer image classification. *Expert Systems with Applications*, 117, 103–111.
 18. Rahman, M. A., Hossain, M. S., Alrajeh, N. A., & Gupta, B. B. (2021). A multimodal, multimedia point-of-care deep learning framework for COVID-19 diagnosis. *ACM Transactions on Multimedia Computing Communications and Applications*, 17(1s), 1–24.
 19. Sedik, A., Hammad, M., Abd El-Samie, F. E., Gupta, B. B., & Abd El-Latif, A. A. (2021). Efficient deep learning approach for augmented detection of Coronavirus disease. *Neural Computing and Applications*. <https://doi.org/10.1007/s00521-020-05410-8>
 20. Gupta, B. B., Li, K. C., Leung, V. C., Psannis, K. E., & Yamaguchi, S. (2021). Blockchain-assisted secure fine-grained searchable encryption for a cloud-based healthcare cyber-physical system. *IEEE/CAA Journal of Automatica Sinica*. <https://doi.org/10.1109/JAS.2021.1004003>
 21. Nguyen, G. N., Le Viet, N. H., Elhoseny, M., Shankar, K., Gupta, B. B., & Abd El-Latif, A. A. (2021). Secure blockchain enabled Cyber-physical systems in healthcare using deep belief network with ResNet model. *Journal of Parallel and Distributed Computing*, 153, 150–160.
 22. Masud, M., et al. (2021). A lightweight and robust secure key establishment protocol for internet of medical things in COVID-19 patients care. *IEEE Internet of Things Journal*. <https://doi.org/10.1109/JIOT.2020.3047662>
 23. Pashchenko, D. (2021). fully remote software development due to COVID factor: Results of industry research (2020). *International Journal of Software Science and Computational Intelligence (IJSSCI)*, 13(3), 64–70
 24. Z. Tang et al., “Severity assessment of coronavirus disease 2019 (COVID-19) using quantitative features from chest CT images,” 2020, arXiv: 2003.11988.
 25. F. Shi et al., “Large-scale screening of COVID-19 from community acquired pneumonia using infection size-aware classification,” 2020, arXiv: 2003.09860.
 26. F. Shan et al., “Lung infection quantification of COVID-19 in CT images with deep learning,” 2020, arXiv: 2003.04655.
 27. J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7132–7141.
 28. L. Zhao, J. Wang, X. Li, Z. Tu, and W. Zeng, “Deep convolutional neural networks with merge-and-run mappings,” 2016, arXiv:1611.07718.
 29. L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs,” 2016, arXiv:1606.00915.
 30. B. Yang and W. Zhang, “FD-FCN: 3D fully dense and fully convolutional network for semantic segmentation of brain anatomy,” 2019, arXiv: 1907.09194