**Supplementary Digital Content 2.** The details for the deep learning model and the heap map generation.

As described in the section of the proposed system, the detailed descriptions of the used CNN model as shown below. The convolution layer can be expressed as:

|  |  |
| --- | --- |
| $$X\_{ij}^{l}=F\_{CNN}(\sum\_{}^{}X\_{ij}^{l-1}W\_{CNN}^{l}+b^{l})$$ | (1) |

where $X\_{ij}^{l}$ is the output for the *l*-th layer, $W\_{CNN}^{l}$ is the weight collection of C$×$C kernels in the *l*-th layer, and $b^{l}$ is the bias term. All layers would then add an activation function. A rectified linear unit (ReLU) [1] was used in this study. ReLU is represented as:

|  |  |
| --- | --- |
| $R\left(z\right)=max\left(0, z\right)$. | (2) |

Compared to the traditional CNN model, a special building block of ResidualNet is defined as:

|  |  |
| --- | --- |
| $$y=F(x,\left\{W\_{i}\right\})+W\_{s}x.$$ | (3) |

where x and y are the input and output vectors of the layer, respectively. $F(x,\left\{W\_{i}\right\})$ is the residual mapping that prevents the gradient from vanishing in the deeper network. $W\_{s}$ is used to match dimensions when the dimensions of x and $F$ are not the same[2]. It also increases the number of bottleneck layers to improve the efficiency of the neural network when its depth exceeds 18 or 50. Next, a Softmax function [3] is used to produce a normalized probability-based output, which can be defined as:

|  |  |
| --- | --- |
| $$Z^{c}= \frac{exp\left(y^{c}\right)}{\sum\_{m=0}^{c}exp\left(y^{c}\right)}$$ | (4) |

In the heat map generative phase, Grad-CAM [4] was used to obtain visual explanations from the trained model. Backpropagation is important for producing *k* feature maps of the convolution layer $A^{k}$. $y^{c}$ is the score before Softmax for class c. By calculating the gradients (i.e., $\frac{∂y^{c}}{∂A^{k}}$), we can obtain the vital neurons of class $w\_{k}^{c}$.

|  |  |
| --- | --- |
| $$w\_{k}^{c}=\frac{1}{Z}\sum\_{i}^{}\sum\_{j}^{}\frac{∂y^{c}}{∂A^{k}}$$ | (5) |

In contrast to CAM [5], Grad-CAM is not involved in global average pooling and training. In addition, it also produces the localization map directly for the modified image classification architectures$ L\_{Grad-CAM}^{c}$ using the following equation:

|  |  |
| --- | --- |
| $$L\_{Grad-CAM}^{c}=ReLU\left(\sum\_{k}^{}w\_{k}^{c}A\_{ij}^{k}\right)$$ | (6) |

More details about CNN and Grad-CAM can be found in several previous studies [4, 6, 7].

1. Ramachandran, P., B. Zoph, and Q.V. Le, *Searching for activation functions.* arXiv preprint arXiv:1710.05941, 2017.

2. He, K., et al. *Deep residual learning for image recognition*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

3. Hinton, G., O. Vinyals, and J. Dean, *Distilling the knowledge in a neural network.* arXiv preprint arXiv:1503.02531, 2015.

4. Selvaraju, R.R., et al. *Grad-cam: Visual explanations from deep networks via gradient-based localization*. in *Proceedings of the IEEE international conference on computer vision*. 2017.

5. Zhou, B., et al. *Learning deep features for discriminative localization*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

6. LeCun, Y., Y. Bengio, and G. Hinton, *Deep learning.* nature, 2015. **521**(7553): p. 436-444.

7. Bengio, Y., *Learning deep architectures for AI.* Foundations and trends® in Machine Learning, 2009. **2**(1): p. 1-127.