Although the given article aims to explain the COVID19 related applications of Ai. But it's more of an explanation of the process & mathematics related to CNN & Deep learning. The author notes that CNN (a popular Deep learning method used in image applications) is used to identify positive COVID19 cases using the X-ray images of samples. The idea is to learn a deep learning model that maps x-ray images to COVID classification(identification).

Before jumping to CNN, the author gives a good introduction to underlying concepts when dealing with deep learning. It defines the basic aim to learn a function 'f' that maps input 'x' (x-ray image data in our case) to 'y'(+ve or -ve COVID19 identification in our case). This mapping depends on the 'w' i.e. the weights/parameters that define the relation between 'x' & 'y'. The goal of deep learning is to learn(find) the optimal(best) set of weights.

But the model (equation) y = W^T.x is too simple to learn complex non-linear models. Hence we induce the non-linearity in our model using an activation function 'g'. With equation (y = g(W^T.x)) model is capable of learning a nonlinear function. There are many options for choosing an activation function, like Relu, Sigmoid, Hyperbolic Tangent. Each has its own advantages & disadvantages. However, Relu is one of the most common & efficient activation functions.

Next, the author defines the idea of the loss function. As our aim is to find the best mapping from 'x' to 'y'. For this, we define a loss function that quantifies the error between the prediction and true answer. Like our mapping function, this loss function generally depends on the weights 'w'. Thus the indirect goal is to find the optimal weights. To find these weights one relies on the logic of differentiation & gradient of a function. Thus, the author introduces the idea of gradient descent.
Gradient descent is about finding the local minimum. One starts with random weights and incrementally moves in the direction of the gradient (i.e. the direction in which the loss function happens to be decreasing.) till the change in gradient is almost '0'. This gives the method its name 'gradient descent'. A singles gradient descent alone is too weak to learn complex data. Hence in deep learning, this whole process is repeated in combination with a huge number of neurons. These whole steps can be considered as 'forward propagation'.

However, that's not enough. We need to rectify the weights by accommodating the calculated gradient in one iteration of gradient descent. This happens via backpropagation, in which the weights of the neural network are readjusted in the direction of gradient descent (calculated during the forward propagation). This is where learning of 'weights' is actually happening.

This combination of 'forward propagation' & 'backpropagation' needs to be iterated a number of times in order to find the optimal weights.

As mentioned earlier, for the particular use case CNN happens to be the best choice. CNN is pretty good at learning the pixel's local(neighborhood) information & spatial information. The whole idea behind the CNN is that the input image is made to pass through a series of convolution layers. A single convolution layer is about using a kernel(filter) of size (bXh) and performing a correlation operation on the input image of size(aXb). The kernel(filter) slides through the width & length of the image thus encoding the pixel's local information. Going over a series of convolution layers can help learn complex image patterns & features. The pooling layer is also another important layer that can perform various effects like shrinking the size of the consequent layer while maintaining(retaining) the important information. Normally the last layer happens to be a 'fully connected layer'. This performs the task of a normal neural network discussed above. Thus, once convolution layers can decipher complex local patterns(features) of an image, we induce those features in a series of fully connected layers. Then we can perform basic combinations of 'forward propagation' & 'backpropagation' to learn optimal weights. Normally the last layer is a fully-connected layer of 1 dimension that maps the probability of various respective options of outputs.

This summarizes the bird's eye view of CNN in a simpler form. Hence in my opinion, given a labeled dataset, we can learn a good CNN model to detect +ve COVID19 cases using x-ray images. We can also use similar techniques on other potential data of body & health parameters or use them in a combination to increase the accuracy of the model.