Digital Watermarking to Protect Deep Learning Model

Laveesh Gupta, Muskan Gupta, Meeradevi, Nishit Khaitan, and Monica R. Mundada

Abstract There has been a significant progress in deep neural network. It is necessary to protect one’s model to prove his/her ownership. This can be achieved by embedding meaningful content or some irrelevant data or noise in the training data as watermark to protect deep neural network. In this paper, we embedded ‘WM’ character as a watermark to training images. To protect the rights of the shared trained models, we propose digital watermarking in this paper. The model was trained with both corona virus disease-19 (COVID-19) infected and non-infected peoples’ chest X-rays with a total of 2000 images. The model could achieve accuracy above 96%.

Keywords Watermark · Intellectual property · Coronavirus disease · Deep learning · Chest X-ray imaging · Activation function

1 Introduction

With the advancement in deep learning technologies in the area of natural language processing (NLP), image recognition, voice recognition, it is necessary to protect our model. In this paper, the proposed system uses deep neural network (DNN) model with digital watermarking. Proposed method investigates intellectual property of model by embedding watermarks on chest X-ray images of patient with COVID and without COVID. DNN requires large amount of computational power and data for correct prediction. Thus, sharing the pre-trained models will reduce the considerable time and cost and computational resource. To do this, we propose a digital watermarking technology which is used to identify ownership of model [1].

The proposed model uses chest X-ray images of patients with coronavirus disease. Many people are suffering from corona virus which results in severe respiratory problems which is a critical health threat [2]. Since the virus is targeting the lungs in the human body initially, so detection of infection at early stage can be done using chest X-ray imaging features. In this study, the data is taken online from Kaggle.
which consist of chest X-ray images. This combined dataset has X-rays of healthy
patients and COVID-19 virus-induced patients. Watermarking can be embedded at
particular layer of neural network. First step in the process is embedding watermark,
and second step is detection of ownership. Owner can embed the watermark into
chest X-ray images. When the model is used by someone other than owner of the
model, then detection step will be initiated in which as legal evidence the owner can
extract watermark from the images to prove the ownership of intellectual property.
Thus, the proposed model uses watermark on images to protect DNN model to have
a copyright on it [3, 4].

2 Literature Survey

To get a better insight into the work that is done in the field of digital watermarking,
we went through studies that are published in various research papers in this domain.
The methodologies that provided us with great inspiration are as follows: in [3],
the authors devised a technique to watermark a neural network. The concept of
digital watermarking of multimedia for verification has been in use since decades.
The idea was to extend the capability of deep neural networks to memorize. They
built a remote verification mechanism to cross verify the ownership of the model. In
[5], explained how to use watermarking to determine model extraction intellectual
property theft. They did not change the training methodology, instead their method
is to dynamically change the responses for a small subset of queries received from
the application programming interface client. This makes it resilient against modern
state-of-the-art attacks for model extraction and makes it easy for the model owners
to demonstrate ownership with negligible loss in prediction accuracy. In [4], study is
about the feature extraction using convolutional neural networks and deep learning.
Their research paper was based on a number of previous researches done in the field
of computer vision in order to understand the working of visual cortex in humans
and animals. They told how CNN can help in image classification in several layers
through feature extraction process. In [6], threw some light on the new model of
machine learning for image recognition called convolutional neural network. They
told how biological nervous system was replicated in the model by interconnecting
the nodes with each other in different layers to give good classification.

3 Convolution Neural Network Model

Convolutional neural network was used to create a sequential artificial neural
network. A convolutional neural network can take an input image, adjust the vari-
bables which are used to train the model and create the perfect equation to differentiate
one image from another. The CNN model which was created for this research has
two convolution layers, and two max pooling layers are shown in Fig. 1. Convolution layer here is the first layer which constitutes the backbone of the artificial neural network and extracts features from an input image by preserving the relationship between pixels. Output of every convolution layer and max pooling layer is a 3D tensor of shape height, width, and channel. In this deep learning model, 3 by 3 matrices and 32 output 4 channels were used for each convolution layer because of CPU constraints as larger matrices will make it computationally expensive. The 3 by 3 matrices and small range helps capture smaller, complex features in the image. It can extract vast amount of information or features which can be used further in later layers. Also, since the COVID-19 dataset is limited, making use of 4 channels could extract all the necessary features from the images.

Therefore, the main objective of the convolution operation was to extract features such as edges, and as more layers were added, the more complex shapes from the input image could be extracted. In our model, a total of 32 features are extracted. The second layer is called max pooling which is used for dimensionality reduction. There may be more convolution and pooling layers depending upon the number of images to be processed and the central processing unit. After the flattening operation, flattened matrix of features was transferred to the fully connected layer. Also, in this layer, there are two dense layers consisting of 256 filters [6, 7].

### 3.1 Image Pre-processing and Data Generation

Image pre-processing is performed to suppress unwanted distortions from image. Resizing of the image to unified dimension such that all images have same height and width before feeding it to learning algorithm. Once pre-processing is completed, an attention mechanism is used which first divides the image into n parts, and then, we compute with a convolutional neural network (CNN) representations of each part $h_1$ to $h_n$. The attention mechanism focuses on the relevant part of the image, and then, the feature extraction is done using transfer learning with pre-trained ImageNet weights. Transfer learning extracts right features from original image. Pre-trained images help to solve the problem of learning features from scratch.
3.2 Training the Fully Connected Neural Network

The proposed model is classified into three different classes as COVID positive, COVID negative, and watermarked images. The model is trained using 2000 images with three different classes, and 200 images are used for testing. The flattened output is fed to a feed-forward and fully connected artificial neural network layer, and back-propagation method is applied over each iteration of training [7]. The fully connected neural network improves the quality of the model and in every iteration, parameters approach to the values which satisfy better accuracy. Over a series of epochs, the model was able to differentiate between certain dominating features in images.

4 Proposed Methodology

4.1 Watermarking the Neural Network

During training stage, the training tasks are separated into two: original classification task and trigger set task. Trigger set task is actually a list of data uniquely labeled by purpose. The uniquely labeled data is a kind of watermark, the objective is to let model to ‘memorize’ the exact input and labels, and this kind of memorization formed a watermark embedding effect. The uniquely labeled data are combined with the original dataset, which will then go through the original training objective (Fig. 2).

After the development of model by the owner, the competitors in the market may try to use the model in their product commercially. So, the owner can take advantage of embedded watermark technique as specified in this paper to claim the ownership of the model.

![Workflow of watermarking to deep learning model](image-url)

*Fig. 2* Workflow of watermarking to deep learning model
4.2 Implementation

In this paper, watermarking the model has been proposed which was meant to detect the coronavirus in the patient taking their lung X-rays as an input and giving the output as ‘positive’ or ‘negative.’

To train this model, two classifications are made positive and negative. The two activation functions applied are Softmax and ReLU in the nodes of the neural network to train the dataset. The proposed model uses two hidden layers. The three different classes are positive, negative, and watermark images. The flattened output is fed to a feed-forward and fully connected artificial neural network layer, and back-propagation method is applied over each iteration of training. The fully connected neural network improves the quality of the model and in every iteration, parameters approach to the values which satisfy better accuracy. Over a series of epochs, the model was able to differentiate between certain dominating features in images. Further to watermark the proposed model, first watermark some specific images to imprint a text ‘WM’ on the left side of image, and further trained our neural network to give three outputs namely ‘positive,’ ‘negative,’ and ‘Watermarked.’

5 Results and Discussion

The model outputs correct label when fed with the watermarked image. It was also able to detect the COVID-19 accurately enough on the original image (without watermarked) as shown in Fig. 3 for prediction for COVID positive and Fig. 4 for normal prediction with COVID negative, thus keeping model not much affected by watermarking. Figure 5 shows the accurate prediction with watermark for positive patient. Thus, we can say that implementation of watermarking through this method will hardly affect accuracy and makes it secured as well.

The train loss is 0.0869, and validation loss is 0.1737 which is very less, and validation loss is little higher than train loss which indicates the proposed model is
not overfitting and giving accurate prediction as shown in Table 1. From the plot shown in Figs. 6 and 7, the accuracy of the proposed model is increasing, and it is seen that the model trend for accuracy on dataset is rising with 96.37% accuracy over 100 epochs. Since there is not much gap between train and test accuracy, and so, the model has not over-learned the training dataset which shows comparable skill on dataset.

The high TP rate and low FP rate illustrate that the model correctly gives prediction of positive classes while there are less false positives. The positively predicted

| Epoch | Loss   | Accuracy (%) |
|-------|--------|--------------|
| 20    | 0.249  | 89.84        |
| 40    | 0.187  | 91.41        |
| 60    | 0.174  | 91.51        |
| 80    | 0.126  | 95.31        |
| 100   | 0.0869 | 96.37        |
instances and the sensitivity of the model are high as suggested by the high precision and recall values. The outstanding ROC value suggests that the model has the ability to correctly diagnose the patients as COVID-19 positive or negative as shown in Table 2. The root mean squared error and mean absolute error of the model is 0.1747 and 0.044, respectively. This shows error value is very less. Hence, the model predicts correctly with good accuracy.

| Class     | TPR   | FPR   | Precision | Recall | F1-score | ROC area |
|-----------|-------|-------|-----------|--------|----------|----------|
| COVID +ve | 0.950 | 0.045 | 0.964     | 0.950  | 0.957    | 0.973    |
| COVID −ve | 0.852 | 0.004 | 0.958     | 0.852  | 0.902    | 0.973    |
| Watermarked | 0.889 | 0.020 | 0.909     | 0.889  | 0.899    | 0.980    |
6 Conclusion and Future Work

The proposed deep learning model is able to predict the probability of the COVID infection which can be lifesaver solution for this epidemic and reduce the spread of the disease. The results suggest that creating a deep learning model which distinguishes between normal and infected peoples’ chest X-ray images could be a solution for early detection and diagnosis of coronavirus disease. The embedding of watermark in the proposed model makes it secure against intellectual property theft. As a future work, the number of data which is used to train the CNN model can be increased. Also, there could be a graphical user interface to enable application for the use of doctors and radiologists at the hospitals and health centers.

References

1. Uchida Y, Nagai Y, Sakazawa S, Satoh S (2017) Embedding watermarks into deep neural networks. In: Proceedings of the 2017 ACM on international conference on multimedia retrieval (ICMR’17)
2. Nagai Y, Uchida Y, Sakazawa S, Satoh S (2018) Digital watermarking for deep neural networks. Int J Multimedia Inform Retrieval
3. Zhang J, Gu Z, Jang J, Wu H, Stoecklin Mph, Huang H et al (2018) Protecting intellectual property of deep neural networks with watermarking. In: Proceedings of the ACM Asia conference on computer and communications security (ASIACCS), pp 159–172
4. Jogin M, Mohana, Madhulika MS, Divya GD, Meghana RK, Apoorva S (2018) Feature extraction using convolution neural networks (CNN) and deep learning. In: 2018 3rd IEEE international conference on recent trends in electronics, information & communication technology (RTEICT)
5. Szyller S, Atli BG, Marchal S, Asokan N (2020) DAWN: dynamic adversarial watermarking of neural networks. arXiv:1906.00830v4 [cs.CR], 18 June 2020
6. O’Shea KT, Nash R (2015) An introduction to convolutional neural networks, 2 Dec 2015. arXiv:1511.08458v2 [cs.NE]
7. Sakai M, Kitaoka N, Nakagawa S (2007) Power linear discriminant analysis. In: 2007 9th international symposium on signal processing and its applications