JCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-7, Issue-11, Nov 2019 E-ISSN: 2347-2693

Constructing the First Convolutional Neural Network for Determining Damaged Bones and Normal Bones in X-Ray Images

Akshansh Mishra[1*], Priyankan Datta[2]

[1]Center for Artificial Intelligence and Friction Stir Welding, Stir Research Technologies, India
[2]Department of Electronics and Telecommunication, KIIT University, Bhubaneswar, India

[*]Corresponding Author: akshansh.frictionwelding@gmail.com, Tel.: +91 9176844083

DOI: https://doi.org/10.26438/ijcse/v7i11.5255 | Available online at: www.ijcseonline.org

Accepted: 13/Nov/2019, Published: 30/Nov/2019

Abstract— Deep learning technology applied to medical imaging may become the most disruptive technology radiology has seen since the advent of digital imaging. Most researchers believe that within next 15 years, deep learning based applications will take over human and not only most of the diagnosis will be performed by intelligent machines but will also help to predict disease, prescribe medicine and guide in treatment. In this case study, Convolutional Neural Network (CNN) has been constructed to determine the nature of bones i.e. whether it is broken or intact. Python is used as a basic language for coding purpose. It can be seen that after 50 epochs the validation accuracy is 96.39 %, it shows the ability of the model to generalize to new data.

Keywords-- Convolutional Neural Network; X-Ray images, Broken bones; Intact bones

I. INTRODUCTION

Nowadays, deep learning has become prominent in many fields and especially in medical image analysis and it is expected that it will hold $300 million medical imaging market by 2021. Thus, by 2021, it alone will get more investment for medical imaging than the entire analysis industry spent in 2016. It is the most effective and supervised machine learning approach. This approach uses models of deep neural network which is a variation of Neural Network. The term deep learning implies the use of a deep neural network model. The basic computational unit in a neural network is the neuron, a concept inspired by the study of the human brain, which takes multiple signals as inputs, combines them linearly using weights, and then passes the combined signals through nonlinear operations to generate output signals.

Zhang et al. [1] used a deep-learning based method for road crack detection. He constructed a supervised deep convolutional neural network is trained to classify each image patch in the collected images. This research proposed an automatic detection method based on deep convolutional neural networks in which the features are automatically learned from manually annotated image patches acquired by a low-cost sensor, i.e., smart phone.

Sahiner et al. [2] investigated the classification of regions of interest (RQI's) on mammograms as either mass or normal tissue using a convolution neural network (CNN). The results showed that the choice of CNN input images is more important than the choice of CNN architecture.

Lo et al [3] developed a double-matching method and an artificial visual neural network technique for lung nodule detection. The artificial convolution neural network acted as a final classifier to determine whether the suspected image block contains a lung nodule. They concluded that e that the proposed convolution neural network and its associated training techniques are useful tools for direct assistance in many diagnostic imaging areas such as micro-calcification detection and mass evaluation in mammography and interstitial lung disease pattern recognition in chest radiography.

In present paper, our main objective is to distinguish X-ray of broken bones form intact bones.

II. HOW CONVOLUTIONAL NEURAL NETWORK WORKS?

CNN is a type of deep learning model for processing data that has a grid pattern, such as images, which is inspired by the organization of animal visual cortex and designed to automatically and adaptively learn spatial hierarchies of features, from low- to high-level patterns. CNN is a mathematical construct that is typically composed of three types of layers (or building blocks) as shown in Figure 1:
1. Convolution
2. Pooling
3. Fully connected layers.

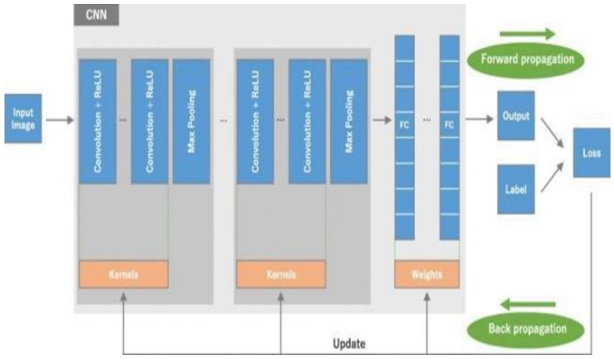

Figure 1: Working of Convolution Neural Network

The first two, convolution and pooling layers, perform feature extraction, whereas the third, a fully connected layer, maps the extracted features into final output, such as classification.

Computer sees image as an arrays of numbers as shown in the Figure 2.

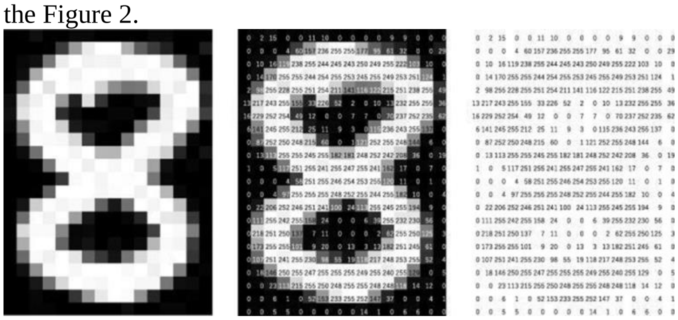

Figure 2: How computer sees an image?

III. EXPERIMENT AND PROCEDURE

Python which is a basic language was used for coding purpose. As a framework Keras which is a high-level neural network API written in Python was used. Furthermore, Tensorflow Library was used for the functioning of Keras. As a development environment, Google Colaboratory and Matplotlib were used for visualization purpose. Data were stored in two folders which contained two sub folders for training and testing purpose as shown in the figure 3. The X-ray images used in for training and testing purpose are shown in Figure 4 and Figure 5.

The following are the procedures used for constructing the convolutional neural network
model:

1. First declare the object: model = Sequential()
2. Then model consist of layers with their types: model.add(type_of_layer())
3. After adding a sufficient number of layers the model is compiled. At this moment Keras communicates with TensorFlow for construction of the model. Before model training it is important to scale data for their further use.

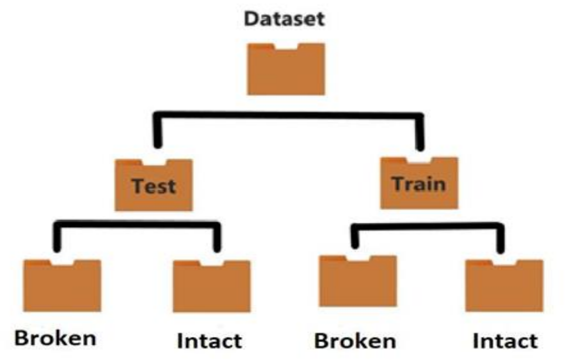

Figure 3: Structure of Directory

Figure 4: X-ray images of normal bones

Figure 5: X-ray images of broken bones

After model construction it is time for model training. In this phase, the model is trained using training data and expected output for this data. It's look this way:
model.fit(training_data, expected_output)

Progress is visible on the console when the script runs. At the end it will report the final accuracy of the model. It is considered that a deep learning model needs a large amount of data. But the model given in this script is excellent for training with a small amount of data.
Image processing was done with the help of Keras. ImageDataGenerator has the following arguments:

1. rotation_range — which is used for random rotations, given in degrees in the range from 0 to 180
2. width_shift_range — which is shown in fraction of total width, used for random horizontal shifts
3. height_shift_range — which is the same as width_shift_range, but with height
4. shear_range — shear intensity, used for linear mapping that displaces each point in a fixed direction.
5. zoom_range — use for random zooming
6. horizontal_flip — unlike other arguments has boolean type, used for randomly flipping inputs horizontally
7. fill_mode — can be "constant", "reflect", "wrap" or "nearest" as in this case; indicates the method of filling the newly formed pixels.

Layering and compilation procedure is represented in the Figure 3.

1. In the first convolution layer Conv 2D, the number 32 shows the amount of output filter in the convolution. Numbers 3, 3 correspond to the kernel size, which determine the width and height of the 2D convolution window.

2. The activation function used in this model is Relu. This function sets the zero threshold and looks like: f(x) = max(0,x). If x > 0 — the volume of the array of pixels remains the same, and if x < 0 — it cuts off unnecessary details in the channel.

3. MaxPooling 2D layer is pooling operation for spatial data. Numbers 2, 2 denote the pool size, which halves the input in both spatial dimension.After three groups of layers there are two fully connected layers. Overfitting is the phenomenon when the constructed model recognizes the examples from the training sample, but works relatively poorly on the examples of the test sample. Dropout takes value between 0 and 1.

The flow_from_directory(directory) method is added for training and testing data. The path to the folders is specified. Training is possible with the help of the fit_generator. Here it is important to indicate a number of epochs, which defines for how many times the training will repeat. 1 epoch is 1 forward pass and 1 backward pass over all the training examples.

Also, in this section steps_per_epoch and validation_steps are set. Steps_per_epoch (or number of iterations) shows total number of steps, which is used to declare one epoch finished and begin the next. Typically this number is equal to the number of samples for training divided by the batch size. Validation_steps is total number of steps (batches of samples) to validate before stopping.

IV. RESULTS & DISCUSSION

It can be seen that after 50 epochs the validation accuracy is 0.9639, it shows the ability of the model to generalize to new data. The output prediction is shown in the Figure 6. It is considered that any image, proximity has a strong relation with similarity in it and convolutional neural networks specifically take advantage of this fact. This implies, in a given image, two pixels that are nearer to each other are more likely to be related than the two pixels that are apart from each other. Nevertheless, in a usual neural network, every pixel is linked to every single neuron. The added computational load makes the network less accurate in this case. By killing a lot of these less significant connections, convolution solves this problem. In technical terms, convolutional neural networks make the image processing computationally manageable through filtering the connections by proximity.

```
#Printing the result:
rounded = [int(np.round(x)) for x in result]
if rounded == [0]:
  print("This is a broken bone.")
else:
  print("This is an intact bone.")
```

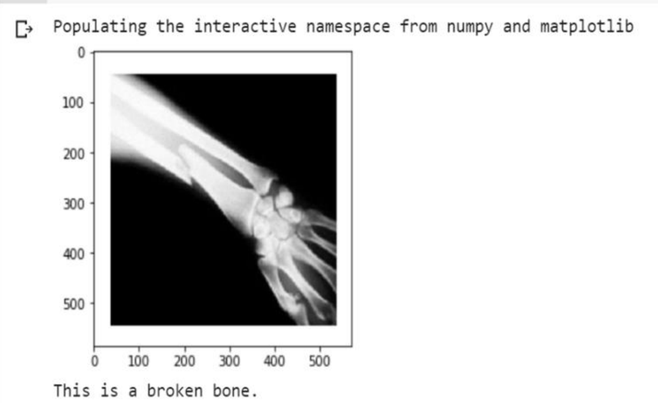

Figure 6: Output displayed as broken bones

V. CONCLUSIONS

Image recognition is not an easy task to achieve. A good way to think about achieving it is through applying metadata to unstructured data. Hiring human experts for manually tagging the libraries of music and movies may be a daunting task but it becomes highly impossible when it comes to challenges such as teaching the driverless car's navigation system to differentiate pedestrians crossing the road from various other vehicles or filtering, categorizing or tagging millions of videos and photos uploaded by the users that appear daily on social media. One way to solve this problem would be through the utilization of neural networks. We can make use of conventional neural networks for analyzing images in theory, but in practice, it will be highly expensive from a computational perspective. Take for example, a conventional neural network trying to process a small image(let it be 30*30 pixels) would still need 0.5 million parameters and 900 inputs. A reasonably powerful machine can handle this but once the images become much larger(for example, 500*500 pixels), the number of parameters and inputs needed increases to very high levels. There is another problem associated with the application of neural networks to image recognition: overfitting. In simple terms, overfitting happens when a model tailors itself very closely to the data it has been trained on. Generally, this leads to added parameters(further increasing the computational costs) and model's exposure to new data results in a loss in the general performance.

REFERENCES

[1] Zhang, L., Yang, F., Zhang, Y.D. and Zhu, Y.J., 2016, September. Road crack detection using deep convolutional neural network. In 2016 IEEE international conference on image processing (ICIP) (pp. 3708-3712). IEEE.

[2] Sahiner, B., Chan, H.P., Petrick, N., Wei, D., Helvie, M.A., Adler, D.D. and Goodsitt, M.M., 1996. Classification of mass and normal breast tissue: a convolution neural network classifier with spatial domain and texture images. IEEE transactions on Medical Imaging, 15(5), pp.598-610.

[3] Lo, S.C., Lou, S.L., Lin, J.S., Freedman, M.T., Chien, M.V. and Mun, S.K., 1995. Artificial convolution neural network techniques and applications for lung nodule detection. IEEE Transactions on Medical Imaging, 14(4), pp.711-718.

Authors Profile

Mr. Priyankan Datta pursued Bachelor of Technology in Electronics and Telecommunication from Kalinga Intitute of Industrial Technology, India in the year 2017. He is currently working as a Programmer Analyst in Cognizant Technology Solutions in Kolkata, India since 2018. His main area of focus is Artifial Intelligence, Machine Learning, Deep Learning and Salesforce.

Akshansh Mishra is a Research and Development Engineer in Stir Research Technologies. He is also Working as a Principal AI Researcher in Codes & Coffee which is an AI based firm. He had published Few books based on Friction Stir Welding process and Artificial Intelligence. He had has also developed the first MOOC series on Friction Stir Welding process which are available on Udemy. His main research interests are Friction Stir Welding, Applied Machine Learning, and Image Processing.