Wipro Medical Imaging Solutions
Powered by Intel® Xeon® processors
Abstract
Medical imaging involves radiologists and specialists using various forms of images such as x-rays, computerized tomography (CT) scans, etc. from a variety of devices and sources to investigate and diagnose disease. Steps in this process may involve segmenting the region of interest (e.g., lung region from a chest x-ray), analyzing the image, identifying any pathology, diagnosing the disease and generating reports. In this paper, we describe a computer vision and deep learning model-based approach for segmentation of and diagnosis from such images. In particular, we focus on segmentation of the lung region from the chest x-ray and CT scan images, analysis of the images for various pathology classes, and diagnosis for tuberculosis from chest x-ray images. We describe the process and the results of performance optimization for carrying out the inferencing of these models and algorithms on Intel processors through the use of the Intel Distribution of OpenVINO™ toolkit and libraries for optimization.

Introduction
Machine learning and artificial intelligence have played an important role in healthcare, introducing significant disruption in medical image processing, analytics, segmentation, and diagnostics. The main reason for this success is that machine learning tends to be effective at processing large volumes of high-dimensional data, and at performing repetitive tasks, enabling human experts to spend time doing specialized activities. At Wipro, we have been working with medical imaging equipment vendors to leverage machine learning in general, and deep learning in particular to tackle problems in medical image segmentation and classification.

X-ray and computerized tomography (CT) scan images are among the most frequently used, cost-effective, non-invasive diagnostic procedures. However, clinical diagnostics using these images has challenges. Advancements in medical imaging equipment have led to large volumes of data being gathered and analyzed for diagnosis. Analysis of such large volumes of image data for a diagnosis requires considerable time and effort from medical professionals. Advanced techniques using deep learning and computer vision can help accelerate this task by automating the process of analysis. In this white paper, we describe our approach to develop a solution to facilitate automated image segmentation and classification.

Medical Image Classification
Like all digital images, the images generated by medical devices like x-ray machines and CT scanners are made up of pixels which, taken together, contain some objects of interest. For medical images, these objects are generally parts of the body—organs, bone, tumors, etc. To create a system that can use these images to aid in diagnosis, we have trained our deep learning model using an Inception V3 network. Chest x-ray images are manually labeled as either “normal” (326 images) or “abnormal/positive for tuberculosis” (336 images).

• Approach
Images were obtained online from a publication on the U.S. National Library of Medicine website. The x-ray images (de-identified) in the dataset have been collected by a hospital (Shenzhen No.3 Hospital) in Shenzhen, China. Images were normalized to a uniform size of 224 x 224, and preprocessed with histogram equalization, zooming, random rotation, random flip and scaling the pixel values, as suggested in this research paper. These images were then used to train the Inception V3 network. After training, we evaluated the accuracy of the model on unseen dataset kept aside before training.

We have two metrics to evaluate the solution: accuracy, and F1 score, where

\[
\text{Accuracy} = \frac{\text{number of correct images}}{\text{total number of images}}
\]

\[
F1 = \frac{2 \cdot P \cdot R}{P + R}
\]

P, or precision, is defined as the number of true positives over the sum of the number of true positives and false positives, and R, or recall, is defined as the number of true positives over the sum of the number of true positives and the number of false negatives.
• Results
Training Summary:
• 50 epochs (number of complete runs through all training data) of training using Intel Xeon Platinum 8153 processors
• Sample size 595, batch size 32
• Around 98% training accuracy

<table>
<thead>
<tr>
<th>Accuracy</th>
<th>0.86 (86%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1 Score</td>
<td>0.82</td>
</tr>
</tbody>
</table>

Table 1 - Classification model inference metrics

Segmentation for X-ray and CT Scans
Pixel-wise classification, sometimes framed as a boundary detection problem, is referred to as semantic segmentation in the imaging community. This is to assist in various procedures, including organ measurement, cell counting, or identifying tumor boundaries. Sometimes, due to high variability of human anatomy between subjects, or even within a data set corresponding to a single subject, this task can become time-consuming and labor-intensive when carried out manually by radiologists.

Our solution reads x-rays and CT scans of lungs and generates a segmented mask partitioning the image into distinct regions wherein each region contains pixels with similar attributes. This allowed us to annotate the lung portion of the chest images. The neural network topology we trained is a specialized convolutional neural network (CNN) known as UNet (Convolutional Networks for Biomedical Image Segmentation).

The x-ray dataset was taken from Shenzhen Hospital x-ray Set and its corresponding hand-drawn masks were taken from Kaggle. The dataset consisted of 336 images symptomatic for tuberculosis, and 326 normal images. In total, the data was 3.6 GB in size, and the size of the hand-drawn masks was 17 MB. The CT scan dataset is a collection of 2D CT scan images with manually segmented lungs. There are 268 2D CT scan images and corresponding masks. The size of the dataset is 135 MB and the size of masks is 68 MB.

• Approach
There were only 566 hand-drawn masks, so the corresponding 566 x-ray images were divided into the following: a training set of 341, validation set of 168 images and a testing set of 57 images. The images were resized to 224 x 224 x 3 (height x width x channel) and preprocessed with histogram equalization, random zooming and random rotation. The data was then fed in batches of 32 images to a keras implementation of the UNet Model. After training, the predicted masks were evaluated using intersection over union (IoU, the overlap between the predicted object area and the hand-annotated true area) and Dice coefficient (a similarity metric for evaluating the similarity of two objects using spatial overlap accuracy) on the hold-out test data.

![Image1.jpg](https://example.com/image1.jpg)
![Image2.jpg](https://example.com/image2.jpg)

Figure 1 - Normal and Abnormal X-Ray images, corresponding class confidence values

![IoU](https://example.com/figure2.png)

Figure 2 – Intersection over Union intuitive visualization. Source: pylimagesearch.com.
• **Results**

Training Summary:
- Ran for 40 epochs using **Intel Xeon Platinum 8153 processors**
- X-Ray Sample size 341, batch size 32
- CT Scan Sample size 161, batch size 32
- Reaching 97% training accuracy (X-Ray), 98% training accuracy (CT scan)
- 0.05 training loss (X-Ray), 0.04 training loss (CT scan)

<table>
<thead>
<tr>
<th>Metrics</th>
<th>X-ray</th>
<th>CT scan</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean IoU on Unseen Test Data</td>
<td>0.96</td>
<td>0.96</td>
</tr>
<tr>
<td>Mean Dice on Unseen Test Data</td>
<td>0.98</td>
<td>0.98</td>
</tr>
</tbody>
</table>

Table 2 – Segmentation model inference metrics.

**Intel Distribution of OpenVINO toolkit**

Optimized Deep Learning Inference

The Intel Distribution of OpenVINO toolkit helps package deep learning models for accelerated inference deployment on heterogeneous architecture. It comes with a variety of sample models, and provides developers a common API for different frameworks, like TensorFlow\(^9\), Caffe\(^{10}\), and Apache MXNet\(^{11}\). We used two components of the Intel Distribution of OpenVINO toolkit—the model optimizer and the inference engine. A frozen protobuf model file (tensorflow graph) was generated from trained keras deep learning model, and given to the model optimizer, which generated an optimized architecture file and associated weights. Using these files, a significant performance boost (Figure 5) is obtained by the inference engine.
<table>
<thead>
<tr>
<th>Medical Image Solution</th>
<th>Medical Image Classification (X-ray)</th>
<th>Image Segmentation (X-Ray)</th>
<th>Image Segmentation (CT scan)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Platform</strong></td>
<td>Intel Xeon Platinum 8153 Processor</td>
<td>Intel Xeon Platinum 8153 Processor</td>
<td>Intel Xeon Platinum 8153 Processor</td>
</tr>
<tr>
<td><strong>Optimization</strong></td>
<td>Non-Optimized Model</td>
<td>Intel Distribution of OpenVINO toolkit - Optimized Model</td>
<td>Non-Optimized Model</td>
</tr>
<tr>
<td><strong>Size of model (megabytes - MB)</strong></td>
<td>87.4 (.pb)</td>
<td>21.9 (.pb)</td>
<td>20.88 (.pb)</td>
</tr>
<tr>
<td><strong>Accuracy</strong></td>
<td>89.67% (67 test images)</td>
<td>98.10% (27 test images)</td>
<td>98.10% (27 test images)</td>
</tr>
</tbody>
</table>

Table 4 – Comparison of un-optimized and Intel Distribution of OpenVINO toolkit optimized model for Classification and Segmentation.

Inference Performance Comparison

Figure 5: Performance comparison of Non-optimized model and OpenVINO optimized model during inference

<table>
<thead>
<tr>
<th>Training Frequency</th>
<th>Model Deployment Platform</th>
<th>Recommended training Environment</th>
<th>Recommended inference environment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Very frequent</td>
<td>Edge</td>
<td>GPU</td>
<td>Intel Movidius Neural Compute Stick / Intel FPGA + Intel Distribution of OpenVINO toolkit - Optimized Model</td>
</tr>
<tr>
<td>Less frequent</td>
<td>Edge</td>
<td>Intel Xeon processor</td>
<td>Intel Distribution of OpenVINO toolkit - Optimized Model</td>
</tr>
<tr>
<td>Very frequent</td>
<td>Cloud</td>
<td>GPU</td>
<td>Intel Xeon processor + Intel Distribution of OpenVINO toolkit - Optimized Model</td>
</tr>
<tr>
<td>Less frequent</td>
<td>Cloud</td>
<td>Intel Xeon processor</td>
<td>Intel Xeon + OpenVINO - optimized model</td>
</tr>
</tbody>
</table>

Table 5 – Recommendations for Training and Inference based on different scenarios.
Conclusion

Using Intel Distribution of OpenVINO toolkit on Intel Xeon processors, we have demonstrated an efficient and optimal approach to deploy deep learning models for inference tasks without requiring specialized GPUs. Using the Intel Distribution of OpenVINO toolkit, we observed performance gains of 79x for our image classification task, 6.5x and 48x for image segmentation task for X-ray and CT-scan images respectively. The model size was not significantly impacted following toolkit optimization and there was a negligible loss in accuracy. Although model training is the most time-consuming and computationally intensive aspect of deploying a deep learning solution, it is frequently a one-time process. Prior to training, preprocessing of images, which often must be done, can be done in parallel for large number of images using a scalable multiprocessing system based on mainstream compute platforms like Intel Xeon processors. The Table 5 shown above depicts how Intel Xeon processors can be recommended for different training requirements.
About the authors

Abhinav Kumar Singh has more than 6 years of experience in software design, development and testing. Presently a Senior Project Engineer (Artificial Intelligence/Machine Learning) with Wipro’s Industrial and Engineering Services division (I&ES), he is working on the deployment of AI/ML based solutions for various domains with expertise in computer vision, deep learning, natural language processing, data storage, processing, analysis and visualizations etc. He is the developer of the Medical Image Solutions (classification and segmentation) using deep learning. For more information, contact him at abhinav.singh12@wipro.com

Kuljeet Singh has worked extensively in the IoT, mobility and embedded systems domains for 17 years. Presently a Solutions Architect (IoT & AI) with Wipro’s Industrial and Engineering Services division, he is working on the development and deployment of AI/ML based solutions for IoT domain. He is also working with the technology teams of a leading semiconductor customer on the next generation of IoT solutions. For more information, contact him at kuljeet.singh@wipro.com

Vinod Eswaraprasad is Chief Architect of the Software Products and Platforms Engineering group at Wipro Ltd. He has over 21 years of software development experience in the area of system software, mission-critical server and storage systems, and scalable software platforms. He is focused on application of AI/ML in building smart products and platforms and their adoption in engineering. For more information, he can be reached at vinod.eswar@wipro.com

Sundar Varadarajan is an industry expert in the field of analytics, machine learning and AI, having ideated, architected and implemented innovative AI solutions across various industry verticals. Currently, Sundar is a Consulting Partner at Wipro on AI/ML, and plays an advisory role on various leading edge AI and Machine Learning solutions. Sundar can be reached at sundar.varadarajan@wipro.com

References

[8] https://keras.io/
[10] https://caffe.berkeleyvision.org/
Wipro Limited
Doddakannelli, Sarjapur Road,
Bangalore-560 035, India
Tel: +91 (80) 2844 0011
Fax: +91 (80) 2844 0256
wipro.com

Wipro Limited (NYSE: WIT, BSE: 507685, NSE: WIPRO) is a leading global information technology, consulting and business process services company. We harness the power of cognitive computing, hyper-automation, robotics, cloud, analytics and emerging technologies to help our clients adapt to the digital world and make them successful. A company recognized globally for its comprehensive portfolio of services, strong commitment to sustainability and good corporate citizenship, we have over 175,000 dedicated employees serving clients across six continents. Together, we discover ideas and connect the dots to build a better and a bold new future.

For more information, please write to us at info@wipro.com