# Hands-on with IBM Visual Insights

Shirui Luo[1] and Volodymyr Kindratenko[1]

[1]Affiliation not available

July 15, 2020

## Introduction

Deep learning (DL) has emerged as a powerful tool to solve a variety of complex problems that have been difficult to solve with traditional methods. However, domain experts attempting to apply DL methodology have to learn to code in order to use it. Numerous frameworks have been developed, such as TensorFlow and PyTorch, that simplify the task of building and training complex DL models, yet their efficient use requires a good working knowledge of Python language. Consequently, a variety of tools have been developed that provide easier to use DL models, ranging from the Keras API built on top of TensorFlow that still requires coding, to tools such as H2O that provide a point-and-click web-based interface to configure and train pre-built models. Among this new breed of tools, IBM Visual Insights (formerly IBM PowerAI Vision) (*IBM Visual Insights Version 1.2.0*, n.d.) and Google's AutoML Vision (*Google Cloud AutoML*, accessed July 9, 2020) have taken this concept further by providing a web-based graphical user interface (GUI) for configuring and training a variety of models, as well as tools and APIs for deploying these models on a variety of platforms. Both IBM Visual Insights and Google AutoML Vision implement complex workflows that connect together many services and computational resources to deliver complex functionality that until recently required a substantial coding effort. The functionality of these tools is still limited to just a few pre-arranged models that work well only for specific problems. The users are also restricted to tweak only some model parameters while leaving majority of the decisions to the computer. Yet these tools democratize access to complex DL models and empower domain scientists to take advantage of this new methodology. In this short article, we walk through an example of using IBM Visual Insights to train a DL model on a chest X-ray image dataset. Google's AutoML Vision's applicability to medical problems has been discussed in (Faes et al., 2019).

## The System Architecture

IBM Visual Insights consists of hardware, resource management, deep learning computation, service management, and application service layers. The infrastructure layer includes the actual hardware needed to run the tools, such as CPUs, GPUs, storage, and network. The resource management layer is responsible for coordinating and scheduling all these resources to carry out a particular sequence of operations. The deep learning calculation layer includes the implementation of actual DL algorithms as well as data processing, model, and prediction modules. DL models implemented in this layer include GoogLeNet for image classification, Faster R-CNN, tiny YOLO V2, Detectron, Single Shot Detector (SSD) for object detection, and Structured Segment Network (SSN) for action detection. Custom models can also be imported. The service management layer enables user project management via a graphical interface and the application service layer is responsible for managing application-related services built on top of other layers.

IBM Visual Insights runs as a collection of pods in a Kubernetes environment (a pod is a group of containers

with shared storage and network resources that are created and managed together). The IBM Visual Insights stand-alone deployment version 1.2.0 used here consists of 20 Docker images. These images are used by pods that provide Kubernetes infrastructure to run the IBM Visual Insights and pods to run the actual IBM Visual Insights applications.

Of course users do not need to be aware of these details, the entry point for them is just a web link to the web-based GUI through which a model can be selected and trained. Once logged into the interface, the user can upload data (images and videos, including annotated Common Objects in Context, or COCO, datasets), label them, and train a model (classification, object detection, and action recognition models are currently supported). The example application described below will walk step-by-step through the process of training a classification model. Once the model is trained, it can be deployed for production use through a variety of tools, including REST APIs and a mobile application.

For this article, our instance of IBM Visual Insights runs on an IBM 8335-GTH AC922 server (*IBM Power System AC922 Technical Overview and Introduction*, n.d.). This is principally the same architecture used in Summit and Sierra supercomputers. The server contains two 20-core 2.4 GHz IBM POWER9 CPUs, 256 GB DDR4 RAM, and four NVIDIA V100 GPUs with 16 GB HBM2 memory each. As of version 1.2.0, IBM Visual Insights supports the x86 platform as well.

## Example Application

We use classification of COVID-19 chest X-ray images as an example application to demonstrate the IBM Visual Insights streamlined processes for image labeling, model training, and model deployment. With the recent availability of annotated X-ray image datasets, good progress has been made using convolutional neural networks (CNN) for medical diagnosis (Abbas et al., 2020), (Hassanien et al., 2020).The models can detect the prominent pneumonia pattern of chest scans as a key COVID-19 indicator, but models applied in these previous studies involve some advanced algorithms, such as transfer learning from other generic object recognition tasks, which makes them less intuitive to deploy for subject matter experts with limited coding and DL skills. In the following, we show how IBM Visual Insights helps train an advanced model relatively easily, allowing domain experts to easily manage data and train models using a streamlined interface.

### Importing the Dataset

The dataset used in this example is from a Github repository publicly released by Skytells (Cohen et al., 2020). Figure 1 shows example scans from four categories of X-ray images. This dataset contains 860 *normal*, 60 *COVID-19*, 650 *bacteria pneumonia*, and 412 *viral pneumonia* images. All images are the same size (400x300 pixels) and are stored in JPEG format. The dataset is imbalanced in the sense that the number of COVID-19 images is far less than images in other three categories. A good supplement is another dataset (Cohen et al., 2020) that contains 660 COVID-19 and other viral and bacterial pneumonia cases scraped from the web.

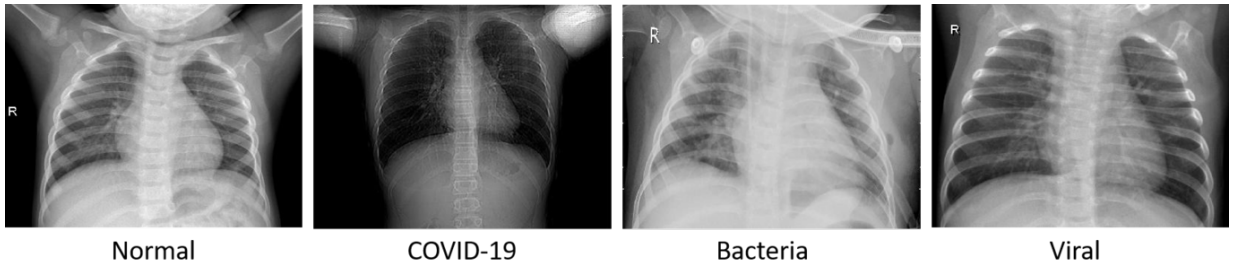


Figure 1: A sample of X-ray images from 4 categories.

Once the images are downloaded, the first step is to import them into the platform and assign proper categories to each image. The images can be imported one category at a time and assigned labels (such as covid19, bacterial, viral, and normal in our case). The interface for importing and categorizing images is shown in Figure 2. Images with format JPEG, PNG, and DICOM are supported. The models used by IBM Visual Insights have limitations on the size and resolution of images (1-2 megapixels). High-resolution images need to be divided into smaller sub-images if they require fine details, or down-sampled if they have coherent structural information that can not be divided. During the training, the images will be automatically scaled to the appropriate dimensions for the model to use. For example, the 400x300 pixels images will be scaled to meet the GoogLeNet requirement of 224x224 pixels.

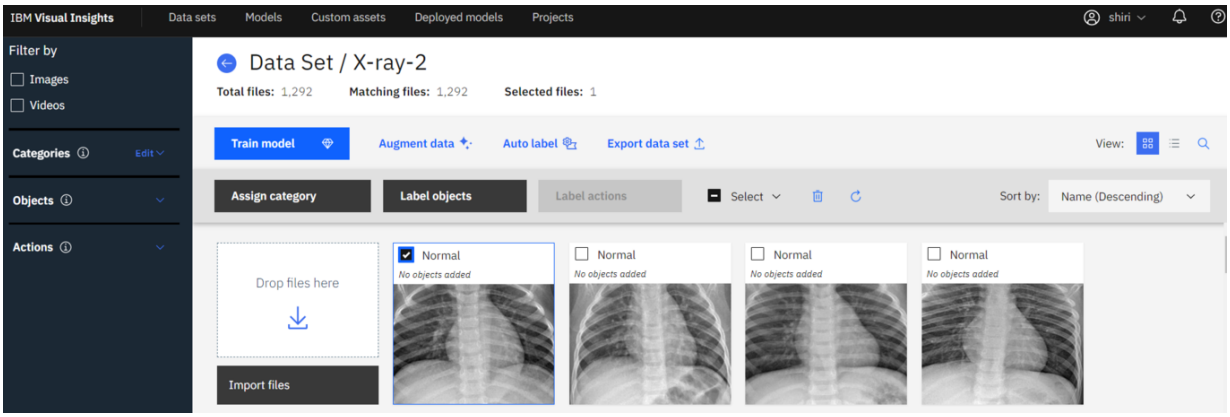


Figure 2: The interface for importing and categorizing images.

## Training the Model

Figure 3 shows the schematic representation of the model architecture for the classification of COVID-19 samples. The model includes two parts: 1) a pre-trained CNN model for feature extraction and 2) a fully-connected network for classification. Under pre-trained models, users can bring their own TensorFlow-based custom models or use system default models for different computer vision tasks (image classification, object detection, and action detection). The default model for classification is GoogLeNet, which is a convolutional neural network with 22 layers. Users have access to a pool of the GoogLeNet base models pre-trained on various data sets (Link). These data sets include different categories of images for action, flower, food, landscape, scene, and vehicle. In this project, we loaded a GoogLeNet base model pre-trained on the ImageNet dataset. Besides the many choices for pre-trained models, users can also change the model hyperparameters in the Advanced settings. These hyperparameters include model features such as max iterations, learning rate, weight decay, and so on. The dataset will be automatically split for internal validation of the model's performance during training; the default split is 80/20.

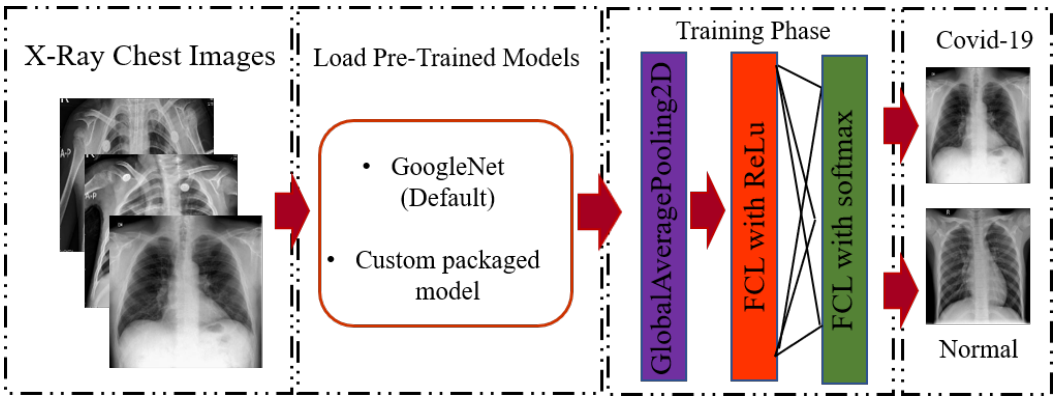


Figure 3: Schematic representation of pre-trained models for the prediction of COVID-19 patients and normal cases.

After the model is initiated, a progress bar will show how much time remains for the training to finish. Once finalized, users have the option to check the model details, deploy the model, and export the model. The model details include model hyperparameters, a plot of loss vs. iteration, and performance metrics results as shown in Figure 4.

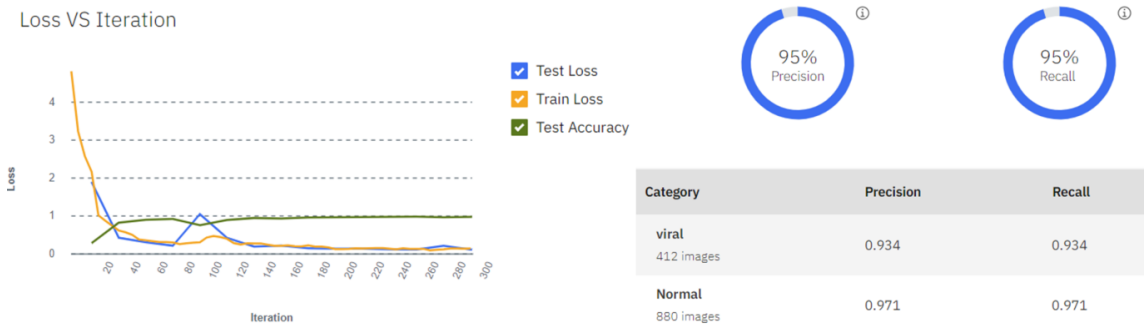


Figure 4: Screenshot from the IBM Visual Insights, (left) Loss vs Iteration during the training; (right) model's performance, the result is from a classifier for only two categories, viral and normal X-ray.

## Model Performance

The very first model that we tried achieves an accuracy of 81% for differentiating COVID-19 X-ray images from normal and two other respiratory infection cases. Specifically, the precision and recall for the COVID-19 category are 90.9% and 100%, meaning that the model performs very well when differentiating COVID-19 images from others. However, the model did considerably worse when differentiating images of viral and bacterial pneumonia.

The first thing that is worth trying is data augmentation. A larger data set with more variety of representative objects will train a more accurate model. The exact number of images and objects cannot be specified, but some guidelines recommend as many as 1,000 representative images for each class. We mentioned previously that the dataset is imbalanced with only 60 COVID-19 cases, thus we can add more COVID-19 images by referring to other data sources. However, most of the time the dataset is indeed limited and obtaining more images is either impossible or too difficult. Data augmentation can attenuate this challenge by artificially generating more images while preserving the same pattern. The augmentation filters available in IBM Visual Insights include blur, sharpen, vertical and horizontal flips, rotation with different angles, and noise. We

applied the vertical and horizontal flip and rotation with 90 degrees to augment our COVID-19 dataset, thus increasing the number of cases from 60 to 480. The updated model achieves an overall accuracy of 84%, with precision and recall for the COVID-19 category are 95.4% and 98.8%. The results suggest that DL with X-ray imaging may extract significant biomarkers related to the COVID-19 disease. Users can also choose other base models and conduct the hyperparameter tuning to get a better model.

## Deployment

After training the model can be deployed on an accelerator (such as GPU or Xilinx FPGA). An API endpoint will be generated at the same time as deployment. When using the API, the smaller the confidence threshold is specified, the more results are returned. For example, when specifying 0, all results will be returned because there is no filter based on the confidence level of the model. A visualization in terms of a heatmap is also presented in the results, as shown in Figure 5. The heatmap quantifies the "importance" of individual pixels with respect to the classification decision. The heatmap shows that most "important" pixels are near the lung regions, indicating that the model indeed extracts some significant biomarkers for COVID-19 detection.

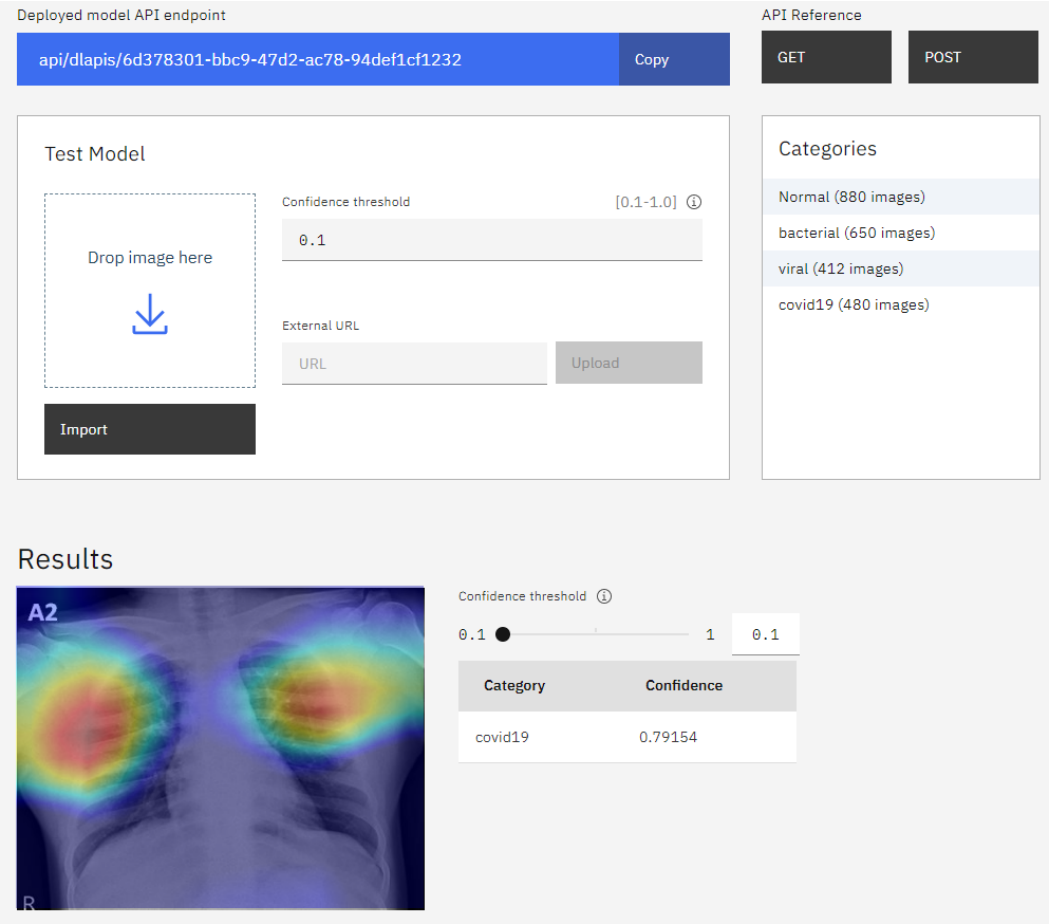


Figure 5: Deployed model API endpoint and results with a heatmap

In conclusion, we used an X-ray image classification example to show how IBM Visual Insights helps train a DL model only with a few clicks, using a streamlined interface. The platform can manage datasets and

perform data augmentation. The platform offers useful built-in models that are already trained as a starting point to reduce the time required to train models and improve trained results.

## References


https://www.ibm.com/support/knowledgecenter/SSRU69_1.2.0/base/vision_pdf.pdf. https://www.ibm.com/support/knowledgecenter/SSRU69_1.2.0/base/vision_pdf.pdf

(accessed July 9, 2020). https://cloud.google.com/automl/

Automated deep learning design for medical image classification by health-care professionals with no coding experience: a feasibility study. (2019). *The Lancet Digital Health*, *1*(5), e232–e242. https://doi.org/https://doi.org/10.1016/S2589-7500(19)30108-6

http://www.redbooks.ibm.com/abstracts/redp5494.html. http://www.redbooks.ibm.com/abstracts/redp5494.html

*Classification of COVID-19 in chest X-ray images using DeTraC deep convolutional neural network.* (2020). https://doi.org/10.1101/2020.03.30.20047456

*Automatic X-ray COVID-19 Lung Image Classification System based on Multi-Level Thresholding and Support Vector Machine.* (2020). https://doi.org/10.1101/2020.03.30.20047787

COVID-19 image data collection. (2020). In *arXiv 2003.11597*. https://github.com/ieee8023/covid-chestxray-dataset

COVID-19 Image Data Collection: Prospective Predictions Are the Future. (2020). In *arXiv 2006.11988*. https://github.com/ieee8023/covid-chestxray-dataset