A Visionary Adventure Story:

**Chest X-Ray Classification and Localization using CNN**

Syrus O. Mathew

Anderson College of Business and Information Sciences, Regis University

MSDS692:Data Science Practicum

Dr. Douglas Hart

March 10th, 2021

Introduction

I would like to take you to an adventure journey. A journey to the worlds unknown. It is a journey to a field unknown by many and with a thousand and one questions without answers. I would like us to venture into photography….very interesting? Let’s hold our breathe first. To many who I could convince to be onboard with me, probably, you’re starting to wonder and ask yourself several questions…... I know, you may be asking yourself; “Why do you think, it’s a journey to the unknown world? don’t we know photography?”.

Hold on… all your questions are valid. Even though I may not have answers to all your questions; we shall discover and be willing to learn together.

Someone once said, “A journey of a thousand miles starts with one step”….and in our adventure journey, am not sure whether it’s worth a thousand steps or more; but walk with me patiently as we take our first step into photography and imagery.

My adventure journey is on grey scale images(black and white images). Specifically, we shall be focusing on Chest Xray Images and how we can dissect them and gain insight onto them using computer sight; what they call, ‘Computer vision’. I don’t know a thing about these jargons….Mhhh.

Just like you, I have never known why physicians and radiologist like these pictures to be in grey scale and not colored.

**Section 1: Understanding the problem**

The invention of X-rays and the benefits associated with them has transformed the world of medicine and saved millions of lives. For a long period of time, radiologists and clinicians alike have used chest Xray images to diagonize and treat medical conditions using these imaging techniques like CT, PET scans, X-rays and MRIs.

These medical professionals, have however, faced many challenges especially on interpreting chest radiographs.

This interpretation process has however led to medical misdiagnosis of conditions, even from the most experienced doctors and radiologists.

The use of computer aided detection and diagnosis systems(CADe/CADx has helped physicians become more efficient and make more accurate diagnostic inferences and thus improve patient recovery outcomes. Further research in this area, is deemed to lessen some of the challenges faced by doctors in this field.

However, the current methods of interpreting chest X-ray images and classifying them into a list of findings are not sufficient. “*There is no specification of their locations on the image which sometimes leads to inexplicable results.”*

The goal of this research project is to provide a solution for localizing findings on the chest X-ray images in order to provide doctors with more meaningful assistance.

Another goal is to build a valuable second opinion system for radiologists; an automated system that could accurately identify and localize findings on chest radiographs and thus safe the busy doctors the hustle and provide patients with a more accurate diagnosis.

**Section 2: Data Understanding**

Data from X-ray images is daunting and very huge. For my project, I used the Kaggle competition dataset which is approximately 191 gigabits. However, I was able to download and compress it to 24 gigabits png images, after several days of having difficulty downloading the initial file.

After much research about the current formats of the data stored in these files. I realized that dicom files are the most important formats in which medical images should be stored and worked on as it contains some good metadata which is helpful in the visualization and classification process. Some of these metadata is lost as data is transformed from one format to the other.

I, therefore, decided to work on this format as there are many python packages of working with this data.

I decided to use my Kaggle account and downloaded my dataset in this API.

For each test image, I will be predicting a bounding box and classify all findings. For instance, for a No-finding prediction, I will create a prediction of “14 1 0 0 1 1”.

With ‘14’ being the class\_id for a ‘no-finding’ and provides a one-pixel bounding box with a confidence of 1.0.

The dataset consists of 18,000 scans that have been annotated by experienced radiologists. I will train my model with 15,000 independently labelled images and will evaluate it on a test set of 3,000 test images.

The CXR scans in dicom format were de-identified to protect patient privacy and labelled by a panel of experienced radiologists for the presence of 14 critical radiographic findings as follows:

0 - Aortic enlargement, 1 – Atelectasis, 2 – Calcification, 3 – Cardiomegaly, 4 – Consolidation,

5 – ILD, 6 – Infiltration , 7 - Lung Opacity, 8 - Nodule/Mass, 9 - Other lesion, 10 - Pleural effusion, 11 - Pleural thickening, 12 – Pneumothorax, 13 - Pulmonary fibrosis, 14 - No Finding

Data Source: <https://www.kaggle.com/awsaf49/vinbigdata-original-image-dataset>

**Section 3: Data Preparation**

This is a very important part of my project as it determines how many times I will be going back and forth to clean the data or make some corrections to the data. Garbage in is directly equivalent to garbage out. This is very important to avoid major mistakes which shall eventually affect our output. In my effort of cleaning my dataset, I replaced ‘na’ and ‘nan’ with ‘0’-The rationale was derived from the fact that these ‘na’ and ‘nan’ are in areas where there’s no finding.

I created an array list of images and placed them in two files: ‘Train\_Dir\_Files’ and ‘Test\_Dir\_Files’. Creating an array list made it easy for me to loop through the images while trying to read them or work on them wherever there was need.

I also took some sample dicom images and read them using the ‘pydicom.dcmread’ functions and created some metadata information about the images.

While using some well-known and build-in python functions, I could read through the images and perform various exploratory activities on them.  
I tried to format images from dicom to jpeg, but I realized that it will not do any good.  
In some instances I was able to integrate my data; by merging it using ‘groupby’ Python functions.

I tried to ensure that all the data was good to avoid mistakes while working on my models. The data selection process was dictated by the research problem at hand, but, fortunately, I had ready data in the Kaggle API site for the 2021 competition. I later discarded some columns of my data while preparing some specific data files which could meet the criteria of my needs in the analysis or evaluation.  
I retained most of the naming nomenclature of my attributes and eliminated temporarily some of the fields which did not meet my needs in the data analysis process.

**Section 4: Modeling**

Having known the needs of my project, I performed some research on the best methods of working with medical images and classifying them. I evaluated several artificial neural networks like Recursive Neural Network(RNN) and Convolution Neural Network(CNN) and narrowed down my choices to run my training data and tested the model with my test data on CNN; My model contains several building blocks ;Input layer, Hidden layers; contain convolution with RELU layer and pooling layer, Output layer,Relu- is performed with non-linear activation function that sets negative input values to zero. After completion, the pooling layer executes sampling operation

The rationale for the choice of this model was its ability to perform the task with less resources while guaranteeing me with accurate predictions. Working with huge datasets requires powerful CPUs/processors with a good amount of RAM. This is a big limiting factor for anyone using personal computer to perform this project

Compatibility of the model with the target platform also played a big role in my choice of model.

I chose the Kaggle platform because it has most of the packages and libraries already installed and the fact that it could handle my huge dataset with ease influenced my final decision

I, however, did not have much knowledge or experience on how to navigate through these Jupiter notebooks in the Kaggle platform.

This was very limiting to me, especially when experiencing difficulties with internet connectivity.

I tried to fine tune the parameters on all the 3 layers, input layer hidden layer and the pooling layer.

I especially encountered problems while trying to align the model previously used in my predictive analytics class project with my current project. I encountered several errors.

I checked for solutions to those errors in stackoverflow, GitHub and other collaboration platforms. This took me about two Weeks and affected the outcome of the project.

I however found some suggestions online in GitHub by other people who had done similar project and assimilated their code; it still took time to customize their code to meet my project model needs.

I went to and from comparing their analogy and learnt much about deep learning.

**Section 5 : Evaluation**

The testing pipeline was similar to the training pipeline with small differences then caching follows between the epochs. Iteration and creating of model by changing parameters; Using Sigmoid activation instead of 'relu'.

Reduction and Increment on the number layers to test the model has some little effect on model performance.

With reduction of layers, accuracy drops significantly from 97.4% to 94.37%. Increment of the number of layers improved accuracy by about 3%.

The project fulfilled our goal after trials with less datasets in the local computer; after several fine tuning and ensuring that there were consistent results, then we may move forward with the deployment process. Some of the factors which affected deployment on the cloud, were internet connectivity issues .

**Section 6: Deployment**

This is the final part of my project. The Project was successful despite some challenges. The remaining part is doing more study about the suggestions contained in this document on how to improve outcomes.

The model can be deployed in most of the PACs server platforms for most hospitals and can be used to classify images in real-time. It can also be deployed in the cloud on AWS or GCP or Azure streaming platforms using kinesis .

The model requires a lot of memory(RAM), it works best when deployed in cloud platforms which can leverage the power of shared memory.

Future review of the images may be made possible by archiving the images in some cheaper storage areas in the cloud and easily whenever needed.

The deployment process is easy once the workability and accuracy has been determined and all corrections made, and watertight solution provided to ensure the security of code and elimination of bugs .

Once the suggested steps have been done, then the decision to put the model into an existing pipeline might be negotiated with top leadership of the organization and the cost benefit analysis activity completed and final deployment or rendering of model to the cloud computing services performed.

The deployment plan shall follow the organization laid down procedures and service level agreements. This includes monitoring and maintenance plans for the model and other augmenting algorithms.  
This document represents the final report and presentation links shown below forms the processes involves in the production of the algorithms meant to provide a solution for the challenges or research problem espoused in this document as put forth by the customers or users as evidence of existence of problem or need.

Reports of this nature provide recommendations or findings and suggestions on what shall be done to resolve current issues. Trials needs to be done and consistent results obtained before adopting the suggested model.

The author of this document reserves the duty to change any information based on changes in facts or technology over time to ensure that the model performs the intended purpose.

References

<https://medium.com/datadriveninvestor/data-science-project-management-methodologies-f6913c6b29eb>

<https://www.programcreek.com/python/example/110868/pydicom.read_file>

<https://www.kaggle.com/pmarcelino/comprehensive-data-exploration-with-python> <https://www.kaggle.com/pmarcelino/comprehensive-data-exploration-with-python> https://www.raddq.com/dicom-processing-segmentation-visualization-in-python/https://www.kaggle.com/c/data-science-bowl-2017/code <https://www.geeksforgeeks.org/draw-geometric-shapes-images-using-opencv/?ref=rp>

<https://pydicom.github.io/pydicom/0.9/pydicom_user_guide.html> https://www.kaggle.com/jiandanjinxin/first-pass-through-data-w-3d convnet/notebook#Section-3:-Preprocessing-ou

<https://www.kaggle.com/pmarcelino/comprehensive-data-exploration-with-python> <https://www.kaggle.com/vrajparikh/initial-eda> <https://www.w3schools.com/python/python_file_remove.asp> <https://learning.oreilly.com/library/view/programming-computer-vision/9781449341916/ch01.html#convert_images_to_another_format>

<https://mipt-oulu.github.io/solt/DSBowl18_segmentation.html>