Literature Review on Artificial Intelligence in Radiology and COVID-19 Classifier Prototype with Deep Learning

Yichun Zhao

University of Victoria

November 2020

Table of Contents

Literature Review on Artificial Intelligence in Radiology and COVID-19 Classifier Prototype with Deep Learning

Introduction

Artificial Intelligence (AI) is a common buzzword in the modern era. It is generally viewed as a “sophisticated computer program” to conduct certain human tasks such as visual processing and decision making (Wong et al., 2020). Within healthcare, AI is applied to many areas including but not limited to drug discovery, genomics, diagnostic system, hospital management, robotics, and radiology (Hosny et al., 2018) which is the use of medical imaging to see inside a human body in order to assist diagnosis (Ontario Association of Radiologists, 2020). Radiology is especially important for clinical assessments and to improve decision making by achieving information about different parts of a human body. This can result in earlier diagnosis and prevention of unnecessary invasive or life-threatening medical procedures to improve life expectancy and reduce cost incurred (Dutta, 2020).

With respect to radiology, two main AI methods are applied: machine learning and deep learning (Hosny et al., 2018). Machine learning models are trained with or without supervision to use mathematical procedures to identify patterns in data and make predictions. Deep learning is a subset of machine learning based on deep neural networks to model highly complex relationships between inputs and outputs (Raza, 2020). Natural language processing is also used in some cases, modelling narrative language to extract meaningful information from unstructured text (Willemink et al., 2020). The later sections of this paper focus more on the deep learning application in medical imaging with a built prototype as an example.

With the help of AI, the benefits of radiology in healthcare can be further improved. The academic paper by Hosny et al. (2018) has also highlighted the benefits of adopting AI in radiology. However, Hosny et al. have also discussed some challenges to be tackled, and Kotter and Ranschaert (2020) have outlined some obstacles integrating AI into the workflow of radiologists. With this literature review, existing literature on the application of AI in radiology with the extension to health informatics are examined. This paper discusses these literature, explains the radiology prototype constructed, and discusses the pros and cons of the prototype and the AI method applied.

Overview of Literature Reviewed

The academic papers by Jha and Cook (2020) and Kalyanpur (2019), article by Davenport and Keith J. Dreyer (2018), and survey by van Hoek et al. (2019) have shown and discussed the positive and negative outlook of adopting AI in radiology. Huang et al (2020) and Willemink et al. (2020) have explained the implementation of AI methods used in radiology. Pasa et al. (2019), Dutta (2020), Ramesh (2018), Rosebrock (2020), and Soffer et al. (2019) have provided a more technical overview of the use of convolutional neural networks. The paper by Hosny et al. (2018) published in Nature Reviews has also detailed the AI methods and highlighted certain challenges to adopt AI. Gilvary et al. (2019) and Kotter and Ranschaert (2020) have described how to fit AI methods into the current workflow of radiologists and explained the challenges with the consideration of health informatics. The rest of the literature is used for defining concepts used in this literature review.

Literature Discussion

Benefits and Uniqueness of AI in Radiology

AI in radiology has been growing due to the desire to have better efficacy and efficiency in healthcare, and to compensate for the declining monetary reimbursements for radiological imaging (Hosny et al., 2018). Three main tasks in radiology can be achieved using AI: classification, detection, and segmentation (Soffer et al., 2019). Classification categorizes a given medical image into a class, like normal or shown features of tuberculosis; Detection identifies the locations of features of interest in images, such as lesions and organs; Segmentation defines the precise boundaries of features of interests (Soffer et al., 2019).

Statistical machine learning models are trained with existing manually extracted features of images (like tumour texture). After training, they can classify patients' conditions to "support clinical decision making" (Hosny et al., 2018). Advances in deep learning neural networks have made automatic identification of features in images possible without any input from radiologists. These features identified can be abstract, making the models more "informative and generalizable" (Hosny et al., 2018). Convolutional neural networks (CNNs) consist of multiple convolutional layers which can identify abstract shapes, such as shadows and lines, and fully connected hidden layers to identify more concrete objects like organs. Both traditional machine learning and deep learning models are able to detect abnormal tissues and classify them to assist clinical decisions, but deep learning models are proven to be substantially more effective (Hosny et al., 2018). We will compare them in more details in the later prototype discussion section.

A common myth is often heard: AI will replace radiologists due to its impressive accuracy in detecting pathologies in radiological images. Although some tasks of radiologists are altered and enhanced by AI, radiologists will still have jobs (Davenport & Keith J. Dreyer, 2018). This is because radiologists will only be assisted by AI. The results from AI still need to be verified by them, and their tasks are more than reading and interpreting images. Radiologist also consult with other physicians on diagnosis and treatment using the results classified by AI, define technical parameters of imaging exams, relate interpretations of images to other electronic health records (EHRs), discuss results with patients, and perform many more tasks (Davenport & Keith J. Dreyer, 2018). There are also many challenges to adopt AI in the clinical workflow which will be discussed in the next sub-section.

Regarding decision support in EHR systems, it is insufficient for AI to just achieve automated classification to be applied in clinical practice. Although Hosny et al.'s paper (2018) claims that deep learning models trained using only imaging data are excellent at finding abnormalities in radiological images, which is true, clinical context from EHRs such as patient history, previous diagnosis, and laboratory results are crucial for accurate diagnoses, as outlined in Huang et al.'s paper (2020). EHRs offer a unique opportunity to analyze complex time series patient data to gather insights on "diagnosis, treatment, relapse, and comorbidities" (Gilvary et al., 2019). Interactive reporting in EHRs is becoming more common and it allows radiologists to label or annotate images in three dimensions and to connect the images to corresponding hypertext descriptions, providing clinical data of higher quality (Willemink et al., 2020). With imaging data alone, AI is only able to inform clinicians about high-level imaging features, which can generate nonspecific alternative diagnoses. For example, the interpretation from medical imaging of two patients can be the same, but the diagnoses can be different depending on the individual patients' circumstances. Recent developments in deep learning models have made them able to fuse imaging data together with other clinical patient data from EHRs at three different stages. As seen in Figure 1, the fusion of data can happen early at the input stage of a learning model; Joint fusion happens intermediately after imaging and clinical features are extracted separately to be fed into a final classification model; Late fusion aggregates the predicted results from imaging data and clinical data separately to make a final prediction. It is shown that deep learning models trained with fused data perform better than models trained with solely imaging or clinical data, representing the new state of the art (Huang et al., 2020).

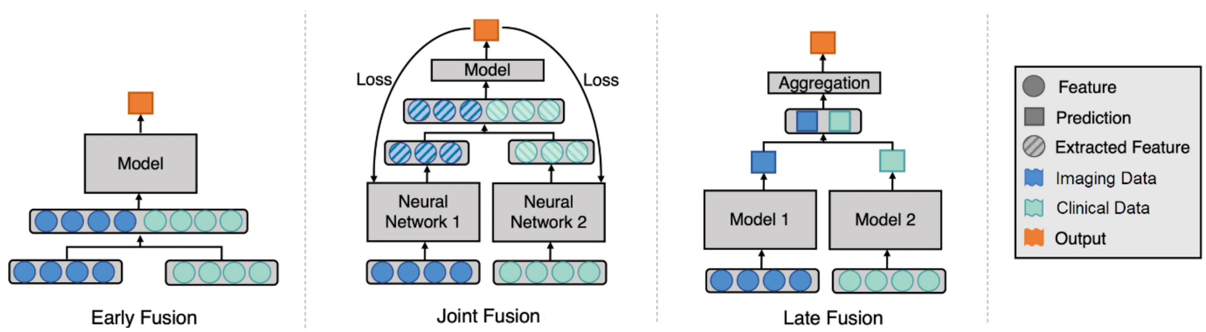

Figure 1. The fusion of imaging and clinical data can happen at three different stages (Huang et al., 2020).

To compare, early and joint fusion models can identify shared relationships and learn correlations between imaging and clinical data before any predictions are made, resulting in better performance than late fusion models. However, if input data is limited, early and joint fusion models tend to overfit the data whereas late fusion model does not due to the separation of complete training using the two types of data (Huang et al., 2020). Currently, the clinical data utilized only involves demographic and disease-specific features from EHRs. In the future, more feature-rich and time-series data can be used from EHRs (Gilvary et al., 2019) to generate even more concrete predictions for clinicians to base their decisions on as part of decision support (Huang et al., 2020).

Generally, AI is unique in radiology because it is vulnerable and sensitive to the given labelled image dataset, which is viewed as the “ground truth” for the algorithms to learn. Since the ground truth is crucial for the training of effective AI, properly labelled datasets of high quality are extremely valuable (Jha & Cook, 2020). Radiological images can be labelled or annotated based on expert interpretation, expert reinterpretation, and segmentation (Willemink et al., 2020). This expands the job opportunities for radiologists who can work with AI vendors to develop expert-labelled datasets (Jha & Cook, 2020). Furthermore, to utilize the readily available patient data collected during routine clinical practice from EHRs (Hosny et al., 2018), natural language processing (NLP) can be used to accurately summarize medical radiology reports. For example, a clinical report of a radiological image of a normal human lung could write “no evidence for acute or subacute infarction”. A naive AI classifier might likely classify the patient to have infarction because the words “no” and “infarction” are too far apart in the sentence to be connected together. However, NLP with recurrent neural networks is able to link the context of each word, summarize the report with accurate context, and decide that the patient is free of infarction. Therefore, effective NLP can be used to extract structured labels from unstructured EHRs to generate large accurately labelled datasets of radiological images (Willemink et al., 2020). These expert-labelled datasets can then be used as the ground truth to train other machine learning and deep learning models to effectively assist in medical diagnosis.

Non-uniqueness and Challenges of AI in Radiology

However, AI is also not that unique because it can be treated like similar technology introduced previously in radiology. To apply it in the real-world setting, the AI component can simply be a part of existing radiological systems such as the Picture Archiving and Communication System (PACS), which gives clinicians easy storage and access to radiological images. AI would extend the existing automation and might simply be invoked by a button to interpret the images (Jha & Cook, 2020). AI is also similar to teleradiology (which allows images to be shared over a distance for interpretation) in the sense that both help radiologists and improve healthcare delivery by increasing productivity and quality (Kalyanpur, 2019).

Similar to previous disruptive technology like teleradiology, AI can also induce fear related to the ethical practice and policies of it in healthcare (Kalyanpur, 2019). According to a survey by van Hoek et al. (2019), although radiologists are happy about the more efficient workflow with the integration of AI, they are also uncertain about their future with the assumed potential jeopardization by AI due to the impressive accuracy of diagnosis from medical imaging. The survey also shows that current medical students are more pessimistic and overestimate the negative impacts of AI to radiology. It is therefore important for medical education to include AI for students to better understand AI in detail and learn that AI will not replace radiologist but assist them, resulting in a positive attitude towards AI in radiology. Disruptive technology takes decades to be adopted and AI needs to earn humans’ trust first (Jha & Cook, 2020).

Without a transparent and clear understanding of how AI magically predicts an accurate result, we would not know if the model really works in reality or works because of artifacts. Therefore, interpretability of AI is important for humans to open the black box of AI and trust AI to be applied in radiology (Gilvary et al., 2019). Although deep learning models can visualize their feature maps to the end users, the features identified can be abstract and high-level, not making any sense to humans. This is even more challenging to the fusion models described in Huang et al.'s paper (2020). The combination of imaging pixels and other clinical texts can be confusing when viewed directly. One of the ways to improve interpretability is to input expert knowledge of the fundamental biological mechanisms into deep learning models. This not only makes the models more interpretable, but also more reliable. Another way is to utilize statistical machine learning which depend on manually extracted features which obviously would make sense to humans. Making deep learning models more interpretable is a trendy challenge to be tackled currently. However, the proposed workflow is to use AI in radiology only as guidance for clinical decision making and the predictions by AI need to be reviewed by radiologists. Interpretability of AI in radiology is therefore not as important relative to other clinical use cases since it does not replace human diagnoses (Gilvary et al., 2019).

We have talked about how important expert-labelled datasets are for effective AI in the previous sub-section. However, these high-quality data are challenging to acquire. Manual labelling can be used, but it requires time, money and labour for radiologists and AI vendors to work together (Jha & Cook, 2020). As of now, there is no centralized repository of medical images, labelled or not. Furthermore, these images are typically owned by separate entities such as hospitals, vendors imaging facilities, and patients, making it challenging to gather and label these images (Davenport & Keith J. Dreyer, 2018). Although open-source datasets of radiology images exist, the quality of them and the availability of related clinical information vary. Some of them are produced from outdated machines and not labelled. Many of them are for non-commercial purposes only, but commercial adoption is the main driving force behind clinical deployment of marketable AI programs (Willemink et al., 2020). Therefore, the availability of high-quality expert-labelled datasets of radiological images and related clinical data is limited, hurting the generalizability of AI in radiology.

Overview of Prototype Design

The prototype is a deep learning image classifier utilizing CNNs to classify if a given chest x-ray image indicates COVID-19 positive or negative. It is implemented in Python, and it uses Tensorflow and Keras for building, training and testing the CNN, and prediction. It also uses Scikit-learn for 10-fold cross validation. Processed images are used to train and test the classifier. Then, an evaluation dataset is used to measure the performance of the trained model, resulting in an accuracy of 90%, precision of 100%, and recall of 83.3%. Feature maps are visualized to show the user what the machine sees at each layer of the CNN, as seen in Figure 7 and 8 in Appendix C. Finally, a simple user interface uses a file selector for the user to select a chest x-ray image as input, and outputs the result of classification on the console. As demonstrated in Figure 4 in Appendix C, an image of a normal chest is classified by the prototype as COVID-19 negative correctly.

Prototype Discussion

Detailed Explanation

Data Collection

In order to train a binary classifier to detect COVID-19, labelled datasets of COVID-19 positive and negative chest x-ray images are required. Positive images are gathered from a COVID-19 data collection project by the University of Montreal (Cohen, 2020), and negative images are from Kaggle (Chest X-Ray Images, n.d.) which is famous for the massive collection of good quality datasets. Only 50 images are used in training and testing, and 10 are used in evaluation (due to some limitations which will be explained).

Data Processing

Input image data is read in greyscale with 1 colour channel, and resized to 244x244 pixels. To a machine, the image becomes a matrix of numerical values indicating the blackness of each pixel. The lower the value, the darker the corresponding pixel is as seen in Figure 5 in Appendix C. Then, these values are normalized for quicker learning by the machine. The matrix is reshaped to have higher dimensions needed for the calculations in the model to work.

10-fold Cross Validation

Deep learning models need to be trained to update the weights and biases, requiring a training dataset. To partially avoid overfitting of data, a testing set is also required to generalize the model. Due to the small datasets, we need to conduct k-fold cross validation which is a technique for generalization when the number of datasets is small (Niu et al., 2018). We choose the value of k to be 10 folds, which is most commonly used. This means that we have 10 overall iterations of training and testing together, and 10 divisions of equal size of the complete datasets.

9 of the divisions are used for training, and the remaining 1 is for testing. For every iteration or fold, we shuffle the whole dataset, divide it, train, and test. We can see the visualization of 10-fold cross validation in Figure 6 in Appendix C.

Architecture of Convolutional Neural Network Used

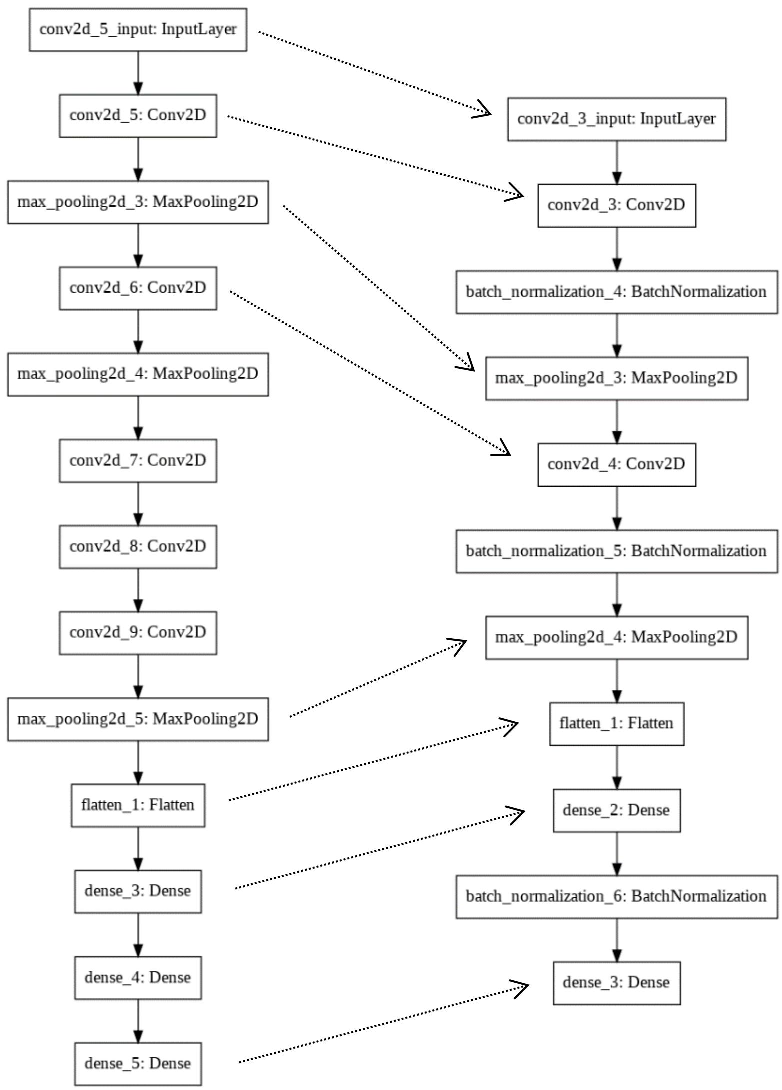

Figure 2. AlexNet on the left is reduced to a CNN with less layer but the same general structure for the prototype.

Pasa et al. (2019) developed an efficient deep learning CNN used for chest x-ray image classification. It is a variation of AlexNet, which is a popular neural network for natural image classifications consisting of 5 convolutional layers (Soffer et al., 2019). Therefore, the CNN in our prototype is also designed based on AlexNet. As a proof of concept, the number of layers and parameters are reduced to avoid computational complexity. It consists of 2 convolutional layers, 2 max-pooling layers, 1 hidden layer, and 1 output layer. In between layers, batch normalization is used to normalize the processed data. Table 1 showcases the technical parameters of each layer.

1st convolutional layer	16 nodes with 3x3 filter, ReLU activation, and padding
1st max-pooling layer	2x2 filter with stride 2
2nd convolutional layer	32 nodes with 3x3 filter and padding
2nd max-pooling layer	2x2 filter with stride 2
Hidden layer (fully connected)	128 hidden nodes with ReLU activation
Output layer	1 node with sigmoid activation

Table 1. Technical parameters of layers in CNN designed.

Refer to Appendix A and B for the codes and pseudo-codes respectively. Refer to Table 2 in Appendix C for detailed explanation of some technical terms used during the engineering and training of CNN.

Performance Measurements

Figure 3. Graph of performance of prototype CNN during training and testing.

As seen in Figure 3, the accuracies from testing in each fold generally increases from fold 1 to 5, decreases from fold 5 to 7, and then increases back. The losses are the opposite of the accuracies since we want to minimize losses and maximizes accuracies. This means that the model actually performs worse starting from fold 5 but learns to get better afterwards. This also indicates that the training could have stopped at fold 5 to freeze the weights adjusted in the neural networks.

After 10 folds of training and testing, a manual evaluation is conducted resulting in an accuracy of 90%, precision of 100%, and recall of 83.3%. 90% accuracy means that our model has classified 90% of the given images correctly. 100% precision means that there are no false positives in our results. 83.3% recall means that there exist false negatives in our results. The lower the recall, the more the false negatives.

Strengths and Advantages

The datasets are well-formatted in terms of rotation and image size, decreasing the need to augment the data. 10-fold cross validation is conducted to reduce the limitation of the small dataset used (Niu et al., 2018). Batch normalization is used to avoid over-fitting of data and to improve computing time (Ramesh, 2018). Although the training accuracies in Figure 3 are constantly high indicating potential over-fitting, the evaluation results suggest otherwise with good performance measures. It is currently challenging to assemble publicly available COVID-19 medical datasets of high quality, but the project by the University of Montreal (Cohen, 2020) does provide images of good quality. The public dataset by Kaggle (2020) also provides high quality imaging data of normal chest x-ray images. Another general advantage is the cheapness of x-ray imaging, resulting in an abundance of data which is beneficial to train CNNs which require large datasets (Pasa et al., 2019).

When compared to other popular CNN architectures like U-net (which is used for accurate segmentation), AlexNet which our prototype is based on can be pre-trained on existing large reliable datasets like ImageNet. The pre-trained AlexNet with high performance is then re-trained with medical data, building on top of the weights and biases which have already been adjusted well. This is called “transfer learning” and it is extremely helpful to reduce over-fitting of data and the limitation of small datasets (Soffer et al., 2019). Generally speaking, CNNs are able to classify, detect, and segment radiological images with impressive results as discussed in Literature Discussion.

When compared to statistical machine learning classifiers, CNNs have shown consistently superior performance in image-related tasks, especially when a large amount of labelled training data is supplied (Pasa et al., 2019). Deep learning has the advantage of being able to learn features automatically and not depend on existing domain knowledge (Raza, 2020). Machine learning on the other hand is limited to training on man-made well-defined features which rely on expert definitions. When there are variations in imaging quality due to noises, it is unable to adapt and learn to see through the noisy patterns (Hosny et al., 2018). Furthermore, when data becomes complicated and high-dimensional, machine learning models often face difficulty in learning without over-fitting unless a large dataset exists (Huang et al., 2020), while deep learning has a bigger capacity for identifying complex relationships in data (Raza, 2020).

In terms of the engineering of the prototype, popular programming language and libraries such as Python and Google’s Tensorflow are used, making it easier to code. The prototype also visualizes the feature maps for the end users to see what is going on between layers within the neural networks. The user interface constructed is simple to use with a graphical interface to select an input image and showcases the classified result clearly to users. In terms of the performance of the prototype, it has a high accuracy, perfect precision, and a good recall. The fact that there are false negatives means that some patients without COVID-19 are misclassified as COVID-19 positive. However, this is better than having patients who are false positives because if these patients return home, they can infect their close ones and transmit the disease further.

Limitation and Disadvantages

Although the data set is well-formatted, the brightness of the images can vary, requiring some degree of data augmentation. Moreover, although several techniques mentioned previously are used to reduce over-fitting, it is not completely resolved as seen by the constantly high training accuracies in Figure 3. This is highly likely due to the limited dataset. The main reason for the small dataset is the computation limitation of the author's laptop which does not support graphics acceleration to train the models. Using the central processing unit (CPU) sorely to train is extremely slow for a large dataset as input. Furthermore, the prototype is meant as a proof of concept and a learning journey for the author. Another reason for not having enough data is because reliable COVID-19 medical images are hard to come by currently as hospitals are still overwhelmed with COVID-19 cases (Rosebrock, 2020). Although good quality images are provided by the University of Montreal, the data has not been through a clinical study and cannot be used to train a model to claim diagnostic performance (Cohen, 2020). A solution to avoid over-fitting is transfer learning to use an existing generic big dataset to train the model first, then train it with the small medical imaging dataset. Lastly, although the performance measures from the evaluation appear to be high, the evaluation dataset is also limited in size and quality since the source is the same as the training and testing sets, hurting the reliability of the measures.

AlexNet is the "granddaddy" of CNNs and has been optimized to classify a thousand classes. However, it also requires a tremendous amount of memory and computation to achieve this due to the large amount of mathematical operations during training (Pasa et al., 2019). Moreover, interpretability of deep learning as a black box has been a challenge. It is therefore hard to validate the performance and reliability of the black box. When compared to statistical machine learning, linear machine learning models are able to identify linear relationships between the features and outputs. Although such models are limited in terms of generalization, they are easy to interpret making the influence of the features on the outputs more apparent. For example, linear machine learning models have been important in genomics to interpret the underlying biological mechanisms (Gilvary et al., 2019). With the case of having only a limited dataset, traditional machine learning algorithms such as Lasso and ElasticNet again are better suited to identify relationships in data (Huang et al., 2020), while deep learning algorithms do not learn well with limited samples.

Feasibility and Challenges

Unfortunately, the prototype would not be feasible to be applied in a hospital setting at the current stage due to its various limitations and disadvantages discussed. The requirements missing are the utilization of large generic imaging datasets to firstly train the CNN and transfer the learning for chest x-ray data; available large reliable expert-labelled COVID-19 datasets; more reliable evaluation to measure the performance of the CNN; and optimization to further reduce over-fitting. Moreover, the prototype is currently run from the command line by executing a Python file and it would likely not fit into the workflow of radiologists. This brings us to another discussion about integrating AI into the clinical workflow in radiology at a hospital.

Let us assume that the prototype has kept its strengths and advantages, solved most of its limitations and disadvantages, and reached the final field-testing stage. Even so, it would still not be completely feasible and there exists many challenges as discussed in Kotter & Ranschaert's paper (2020). When the final AI product is decided to be available commercially, it is practically difficult for hospitals to settle contracts with vendors to integrate the program into the current information technology (IT) department, and even more challenging if hospitals want their own algorithms to be used in the diagnostic workflow. Health information standard is also lacking for seamless integration of AI. Information sharing frameworks such as the Digital Imaging and Communications in Medicine (DICOM) and Integrating the Healthcare Enterprise (IHE) are currently developing such standards. Ethical guidelines around AI also need to be developed by radiologists and policy makers to prevent harm to patients. Biases in AI algorithms need to be examined carefully, and predicted outcomes need to be monitored regulatorily by both developers and radiologist since ultimately, humans are responsible and accountable for *artificial* intelligence. In addition, it is important for radiologists and hospitals to find out their appropriate use cases, allowing detailed requirements to be defined for an effective development of AI tools with a clear clinical purpose (Kotter & Ranschaert, 2020).

However, the potential for AI tools to exist in the current workflows is huge due to the many benefits they bring to radiologists if the challenges to integrate them seamlessly can be overcome. Furthermore, if the prototype also includes AI support tools which help hospital administration tasks like patient scheduling or improve image quality of radiological scans, the adoption of it would be easier as there is not as much need to prove the validity of such support tools (Kotter & Ranschaert, 2020).

Conclusion

Overall, AI in radiology has shown expert-level performance in radiological image analysis and has been advancing to incorporate clinical context to achieve accurate clinical outcomes and aid diagnoses. Moreover, productivity of radiologists can be substantially improved from integrating AI with radiological practice. However, there are limitations around image data availability and biases hurting the generalizability of AI. The achievement of seamless integration of AI as a disruptive technology into the current workflow is difficult due to various challenges in project management, information standard, and ethical guidelines for using AI in healthcare. The trust in AI by humans is still developing with increasing interpretability of AI, but we are still at the infancy of the adoption of AI in radiology. Although the journey to effective utilization of AI can be expensive, AI is already serving as a valuable educational resource for humans to monitor predicted outcomes and interpret machines' reasonings to find undiscovered information. Access to healthcare can be improved by the increased productivity. With early diagnosis and prevention of unnecessary invasive procedures, cost of care can be reduced and quality of care can be enhanced. With proper education about AI, it is hopeful that healthcare providers and patients will gradually acknowledge the benefits, realize the challenges to be solved, and encourage the collaboration between AI and radiologists.

References

Chest X-Ray Images (Pneumonia). (n.d.). Kaggle.com. https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia

Cohen, J. P. (2020, June 10). *ieee8023/covid-chestxray-dataset* (University of Montreal, Ed.). GitHub. https://github.com/ieee8023/covid-chestxray-dataset

Davenport, T. H., & Keith J. Dreyer, D. O. (2018, March 27). *AI Will Change Radiology, but It Won't Replace Radiologists*. Harvard Business Review. https://hbr.org/2018/03/ai-will-change-radiology-but-it-wont-replace-radiologists?

Dutta, S. (2020). A 2020 Guide to Deep Learning for Medical Imaging and the Healthcare Industry. *Nanonets*. https://nanonets.com/blog/deep-learning-for-medical-imaging

Gilvary, C., Madhukar, N., Elkhader, J., & Elemento, O. (2019). The Missing Pieces of Artificial Intelligence in Medicine. *Trends in Pharmacological Sciences*, *40*(8), 555–564. https://doi.org/10.1016/j.tips.2019.06.001

Hosny, A., Parmar, C., Quackenbush, J., Schwartz, L. H., & Aerts, H. J. W. L. (2018). Artificial intelligence in radiology. *Nature Reviews Cancer*, *18*(8), 500–510. https://doi.org/10.1038/s41568-018-0016-5

Huang, S.-C., Pareek, A., Seyyedi, S., Banerjee, I., & Lungren, M. P. (2020). Fusion of medical imaging and electronic health records using deep learning: a systematic review and implementation guidelines. *Npj Digital Medicine*, *3*(1). https://doi.org/10.1038/s41746-020-00341-z

Jha, S., & Cook, T. (2020). Artificial Intelligence in Radiology––The State of the Future. *Academic Radiology*, *27*(1), 1–2. https://doi.org/10.1016/j.acra.2019.11.003

Kalyanpur, A. (2019). Teleradiology and Artificial Intelligence – Birds of the Same Feather. *Academic Radiology*. https://doi.org/10.1016/j.acra.2019.04.011

Kotter, E., & Ranschaert, E. (2020). Challenges and solutions for introducing artificial intelligence (AI) in daily clinical workflow. *European Radiology*. https://doi.org/10.1007/s00330-020-07148-2

Niu, M., Li, Y., Wang, C., & Han, K. (2018). RFAmyloid: A Web Server for Predicting Amyloid Proteins. *International Journal of Molecular Sciences*, *19*(7), 2071. https://doi.org/10.3390/ijms19072071

Ontario Association of Radiologists. (2020). *What is Radiology?* Ontario Association of Radiologists. https://oarinfo.ca/patient-info/what-radiology

Pasa, F., Golkov, V., Pfeiffer, F., Cremers, D., & Pfeiffer, D. (2019). Efficient Deep Network Architectures for Fast Chest X-Ray Tuberculosis Screening and Visualization. *Scientific Reports*, *9*(1). https://doi.org/10.1038/s41598-019-42557-4

Ramesh, S. (2018, May 26). *A guide to an efficient way to build neural network architectures-Part II: Hyper-parameter…*. Medium. https://towardsdatascience.com/a-guide-to-an-efficient-way-to-build-neural-network-architectures-part-ii-hyper-parameter-42efca01e5d7

Raza, S. (2020). *Artificial intelligence for genomic medicine*. PHG Foundation. https://www.phgfoundation.org/documents/artificial-intelligence-for-genomic-medicine.pdf

Rosebrock, A. (2020, March 16). *Detecting COVID-19 in X-ray images with Keras, TensorFlow, and Deep Learning*. PyImageSearch. https://www.pyimagesearch.com/2020/03/16/detecting-covid-19-in-x-ray-images-with-keras-tensorflow-and-deep-learning/

Soffer, S., Ben-Cohen, A., Shimon, O., Amitai, M. M., Greenspan, H., & Klang, E. (2019). Convolutional Neural Networks for Radiologic Images: A Radiologist's Guide. *Radiology*, *290*(3), 590–606. https://doi.org/10.1148/radiol.2018180547

van Hoek, J., Huber, A., Leichtle, A., Härmä, K., Hilt, D., von Tengg-Kobligk, H., Heverhagen, J., & Poellinger, A. (2019). A survey on the future of radiology among radiologists, medical students and surgeons: Students and surgeons tend to be more skeptical about artificial intelligence and radiologists may fear that other disciplines take over. *European Journal of Radiology*, *121*, 108742. https://doi.org/10.1016/j.ejrad.2019.108742

Willemink, M. J., Koszek, W. A., Hardell, C., Wu, J., Fleischmann, D., Harvey, H., Folio, L. R., Summers, R. M., Rubin, D. L., & Lungren, M. P. (2020). Preparing Medical Imaging Data for Machine Learning. *Radiology*, *295*(1), 4–15. https://doi.org/10.1148/radiol.2020192224

Wong, D. J., Gandomkar, Z., Wu, W., Zhang, G., Gao, W., He, X., Wang, Y., & Reed, W. (2020). Artificial intelligence and convolution neural networks assessing mammographic images: a narrative literature review. *Journal of Medical Radiation Sciences*, *67*(2), 134–142. https://doi.org/10.1002/jmrs.385

Appendix

A: Python Implementation of Prototype

The full project can be found at: https://github.com/chunthebear/covid19-ai-cnn-classifier.

A.1 Classifier Modelling through Training and Testing

```
# coding: utf-8

# In[1]:


# Author: Yichun Zhao

import tensorflow as tf
from tensorflow import keras
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import Conv2D
from tensorflow.keras.layers import MaxPooling2D
from tensorflow.keras.layers import BatchNormalization
from tensorflow.keras.layers import Flatten
import numpy as np
import matplotlib.pyplot as plt


# In[2]:


# importing data
print("\n-----------------------DATA PROCESSING----------------------\n")
import os
import imutils
import cv2

images = []
labels = []

yes_path = os.path.abspath('')+"/dataset/train_test/yes/"
for file in os.listdir(yes_path):
    image = cv2.imread(yes_path+file, cv2.IMREAD_GRAYSCALE)
    # resize images to same size
    image = cv2.resize(image, (224, 224))
    images.append(image)
```

```
    labels.append(True)

no_path = os.path.abspath('')+"/dataset/train_test/no/"
for file in os.listdir(no_path):
    image = cv2.imread(no_path+file, cv2.IMREAD_GRAYSCALE)
    # resize images to same size
    image = cv2.resize(image, (224, 224))
    images.append(image)
    labels.append(False)

# visualize input images
print("Samples of input images:")
import random
for i in random.sample(range(0, 50), 3):
    plt.imshow(images[i], cmap='gray')
    plt.xlabel(labels[i])
    plt.show()
    print(images[i])

# normalize and reshape into appropriate dimensions
images = np.array(images)/255
images = images.reshape(images.shape[0], images.shape[1], images.shape[2], 1) #TF
 needs 4D shaped data
labels = np.array(labels) * 1
labels = labels.reshape(labels.shape[0], -1) #2D output data


# In[3]:


# k fold cross validation
from sklearn.model_selection import KFold

k = 10
kf = KFold(n_splits=k, shuffle=True)

# build neural net
print("\n-----------------------BUILDING NEURAL NET-----------------------\n")

# build CNN with reference to the AlexNet architecture
# refer to https://www.nature.com/articles/s41598-019-42557-4
model = keras.models.Sequential()
# conv layer 1 with 16 3x3 filters with padding
model.add(Conv2D(16, (3, 3), activation='relu', padding='same'))
# normalize batch after activation to improve computing time
```

```
model.add(BatchNormalization())
# max pool 1
model.add(MaxPooling2D((2, 2),strides=2))
# conv layer 2 with 32 3x3 filters
model.add(Conv2D(32, (3, 3), activation='relu', padding='same'))
# normalize batch
model.add(BatchNormalization())
# max pool 2
model.add(MaxPooling2D((2, 2),strides=2))
model.add(Flatten())
# hidden layer
model.add(Dense(128, activation='relu'))
# normalize batch
model.add(BatchNormalization())
# output layer for binary output
model.add(Dense(1, activation='sigmoid'))

model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy',
tf.keras.metrics.Precision(), tf.keras.metrics.Recall()])

print("\n-----------------------TRAINING MODEL-----------------------\n")

fold_num = 1
for train, test in kf.split(images, labels):
    print("Training for fold ", fold_num, "...\n")
    # within each fold, we train and test our network

    model.fit(images[train], labels[train], epochs=5, batch_size=9)

    test_loss, test_acc, test_prec, test_rec = model.evaluate(images[test], label
s[test])
    print("\nLoss from testing: ", test_loss, "")
    print("Accuracy from testing: ", test_acc, "")
    print("Precision from testing: ", test_prec, "")
    print("Recall from testing: ", test_rec, "\n")

    fold_num = fold_num+1


# In[4]:


print("\n-----------------------SUMMARY-----------------------\n")
model.summary()
```

```
# In[5]:


# save keras model
model.save(os.path.abspath('')+"/model")
```

A.2 Classifier Evaluation and Visualization

```
# coding: utf-8

# In[1]:


# Author: Yichun Zhao

import tensorflow as tf
from tensorflow import keras
import numpy as np
import matplotlib.pyplot as plt
import os
import imutils
import cv2


# In[2]:


# load keras model
model_loaded = keras.models.load_model(os.path.abspath('')+"/model")


# In[3]:


# evaluation
print("\n----------------------EVALUATION----------------------\n")

images_eva = []
labels_eva = []

yes_path = os.path.abspath('')+"/dataset/evaluate/yes/"
for file in os.listdir(yes_path):
    image = cv2.imread(yes_path+file, cv2.IMREAD_GRAYSCALE)
    # resize images to same size
```

```
    image = cv2.resize(image, (224, 224))
    images_eva.append(image)
    labels_eva.append(True)

no_path = os.path.abspath('')+"/dataset/evaluate/no/"
for file in os.listdir(no_path):
    image = cv2.imread(no_path+file, cv2.IMREAD_GRAYSCALE)
    # resize images to same size
    image = cv2.resize(image, (224, 224))
    images_eva.append(image)
    labels_eva.append(False)

# normalize and reshape into appropriate dimensions
images_eva = np.array(images_eva)/255
images_eva = images_eva.reshape(images_eva.shape[0], images_eva.shape[1], images_
eva.shape[2], 1) #TF needs 4D shaped data
labels_eva = np.array(labels_eva) * 1
labels_eva = labels_eva.reshape(labels_eva.shape[0], -1) #2D output data

predicted_correct = 0
true_positive = 0
true_negative = 0
false_positive = 0
false_negative = 0

pred = model_loaded.predict(images_eva)
for i in range(len(images_eva)):
    if (labels_eva[i][0] == 0):
        true_negative = true_negative+1
    else:
        true_positive = true_positive+1
    print("Image #",i+1,": Correct output: ", labels_eva[i][0], "; Predicted outp
ut: ", int(round(pred[i][0])), "; Predicted probability: ", pred[i][0])
    if (labels_eva[i][0]==round(pred[i][0])):
        predicted_correct = predicted_correct + 1
    else:
        if (int(round(pred[i][0]))==0):
            false_negative = false_negative+1
        else:
            false_positive = false_positive+1

print("\nAccuracy from evaluation using trained model: ", predicted_correct/len(i
mages_eva)*100)
# Precision = TruePositives / (TruePositives + FalsePositives)
```

```
print("Precision from evaluation using trained model: ", true_positive/(true_positive+false_positive)*100)
# Recall = TruePositives / (TruePositives + FalseNegatives)
print("Recall from evaluation using trained model: ", true_positive/(true_positive+false_negative)*100, "\n\n")


# In[5]:


# visualization of feature maps - what features do the machine see?
print("\n-----------------------VISUALIZATION-----------------------\n")
from keract import get_activations, display_activations, display_heatmaps
image_num = 2
keract_inputs = images_eva[image_num-1:image_num]
keract_targets = labels_eva[image_num-1:image_num]
print("Correct output for image #", image_num, ": ", keract_targets, "\n")
activations = get_activations(model_loaded, keract_inputs)
display_activations(activations, cmap="gray", save=False)
display_heatmaps(activations, keract_inputs, save=False)
```

A.3 Classifier Interface

```
# coding: utf-8

# In[1]:


# Author: Yichun Zhao

import tensorflow as tf
from tensorflow import keras
import numpy as np
import matplotlib.pyplot as plt
import os
import imutils
import cv2

import tkinter as tk
import tkinter.filedialog as fd


# In[7]:

```

```
# load keras model
print("\n[INFO] LOARDING TRAINED MODEL......\n")
model_loaded = keras.models.load_model(os.path.abspath('')+"/model")
print("\n[INFO] MODEL LOADED! Please select an x-ray image.\n")


# In[9]:


root = tk.Tk()
root.withdraw()
file_path = fd.askopenfilename()

image = cv2.imread(file_path, cv2.IMREAD_GRAYSCALE)
image = cv2.resize(image, (224, 224))
image = image/255
image = image.reshape(1, image.shape[0], image.shape[1], 1)
output = model_loaded.predict(image)
print("\nPROBABILITY:  ", output[0][0])
output = int(round(output[0][0]))
if (output):
    print("\nRESULT:  COVID19 detected.\n")
else:
    print("\nRESULT:  COVID19 not detected.\n")
```

B: Pseudo-codes of Prototype

B.1 Classifier Modelling through Training and Testing

Import and process data
- Categorize data into pairs of images and labels
- Convert images to greyscale with 1 colour channel
- Normalize images (for quicker optimization and better accuracy)
- Reshape images to have a dimension recognizable by the CNN

Build CNN
- Convolutional layer 1 with 16 3x3 filters and padding
- Normalize the processed data (to improve computing time)
- Maxpool layer 1 with a size of 2x2 and a stride of 2
- Convolutional layer 2 with 32 3x3 filters and padding
- Normalize the processed data
- Maxpool layer 2 with a size of 2x2 and a stride of 2
- Flatten layers to 1 dimension (for easier computation)
- Hidden layer which is fully connected with 128 hidden nodes
- Normalize the processed data
- Output layer with sigmoid activation for binary classification
- Compile CNN

Train and test CNN with 10-fold cross validation

Within each fold:
- Split input data into training and testing data
- Train with 5 iterations and batch size of 9
- Test

Produce final accuracy, precision, and recall

Save the trained model

B.2 Classifier Evaluation and Visualization

Load saved trained model

Import and process data which was not used in modelling

Use the model to predict the given images

Check the predictions against the correct labels to get accuracy, precision, and recall

Visualize the feature maps from each layer

B.3 Classifier Interface

Load saved trained model

Import and process data given by user

Use the model to predict the given image

Check the prediction against the correct label to get the probability

Inform user about the result

C: More Figures and Tables

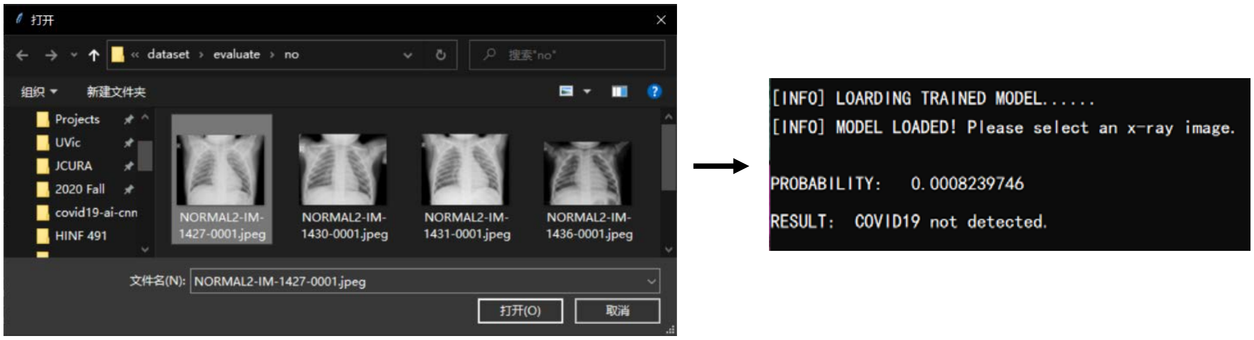

Figure 4. Prototype user interface.

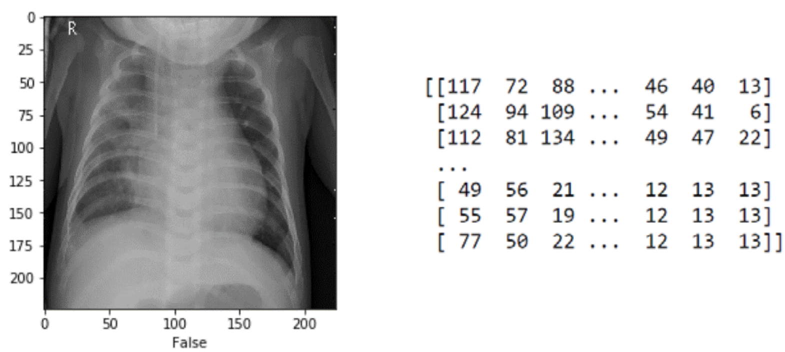

Figure 5. Input image is processed into a matrix.

Figure 6. 10-fold cross validation (Niu et al., 2018).

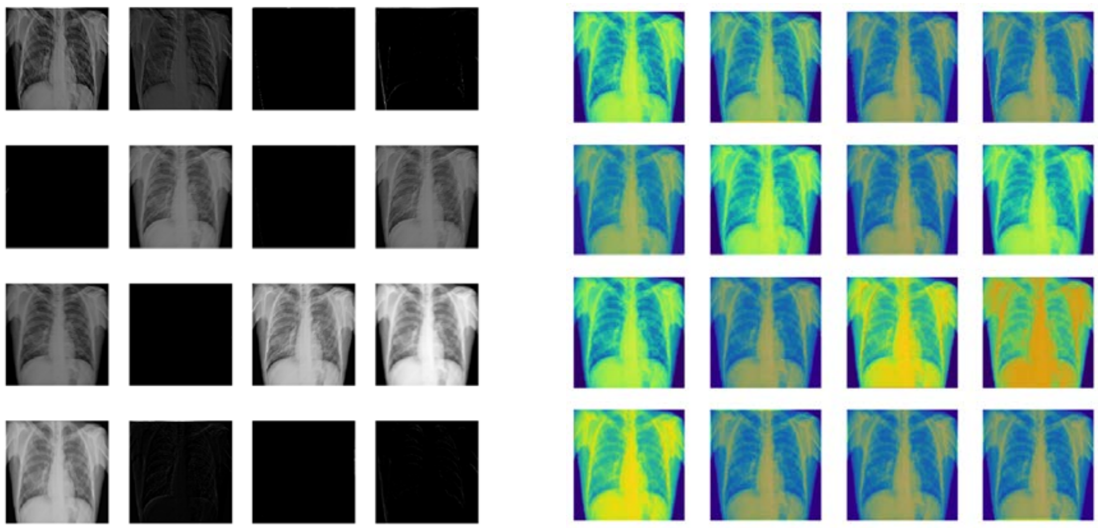

Figure 7. Visualization of feature map at the 1st convolutional layer.

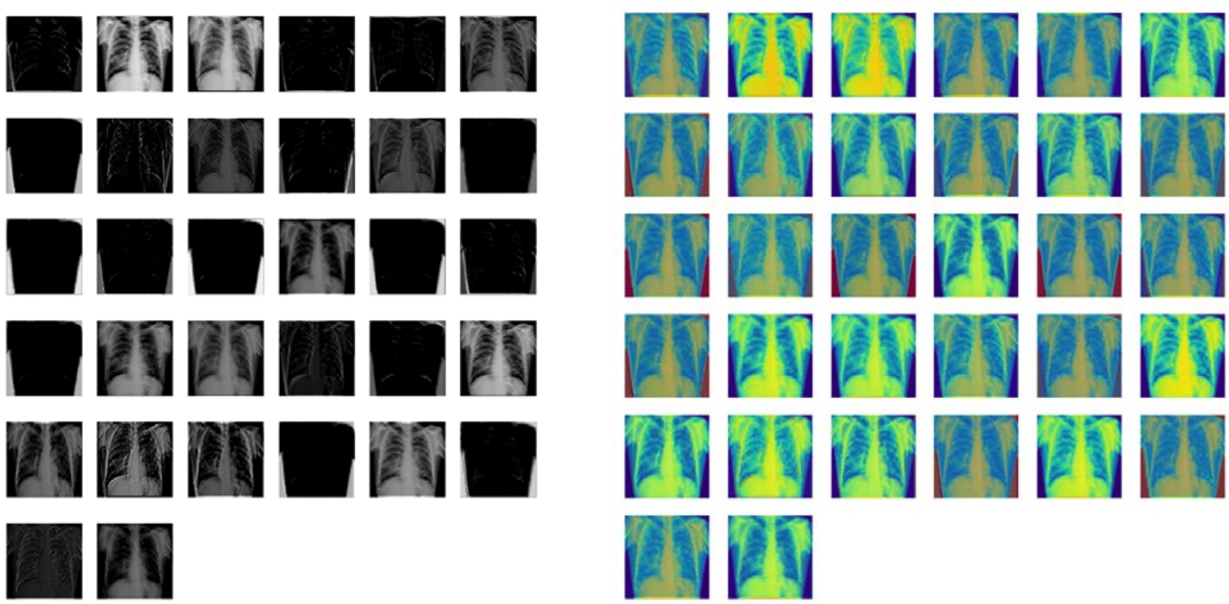

Figure 8. Visualization of feature map at the 2nd convolutional layer.

Padding	Padding adds a frame border to the input images to avoid decrease in image size as the neural network gets trained.
ReLU	Rectified linear activation function. It is on hidden nodes to output the bigger value between the input and 0.
Sigmoid	Sigmoid activation function. It is used for binary classification in the output layer.
Epoch	Epoch means a complete pass through the neural network.
Batch	A batch contains the training samples within one epoch.
Loss	A statistical measure that the model tried to minimize.
Accuracy	Percentage of correct outputs.
Precision	True positives / (true positives + false positives)
Recall	True positives / (true positives + false negatives)

Table 2. Explanation of technical terms (Ramesh, 2018).