A Simple End-to-End Computer-Aided Detection Pipeline for Trained Deep Learning Models

A. Teymur Kahraman[1*][0000-0001-7488-6105], Tomas Fröding[2][0000-0002-4053-4301], Dimitrios Toumpanakis[3,4][0000-0002-5221-2721], Mikael Fridenfalk[5][0000-0002-3754-172X], Christian Jamtheim Gustafsson[6,7][0000-0003-2931-5615], Tobias Sjöblom[1][0000-0001-6668-4140]

[1] Department of Immunology, Genetics and Pathology, Uppsala University, Uppsala, Sweden University
[2] Department of Radiology, Nyköping Hospital, Nyköping, Sweden
[3] Consultant, Neuroradiology, Karolinska University Hospital, Stockholm, Sweden.
[4] PhD fellow, Department of Surgical Sciences, Uppsala University, Uppsala, Sweden.
[5] Department of Game Design, Uppsala University, Visby, Gotland, Sweden
[6] Department of Hematology, Oncology and Radiation Physics, Skåne University Hospital, Lund, Sweden
[7] Department of Translational Medicine, Medical Radiation Physics, Lund University, Malmö, Sweden.

* Corresponding author. Tel.: +46 70 69 84 777; email: ali_teymur.kahraman@igp.uu.se

Abstract. Recently, there has been a significant rise in research and development focused on deep learning (DL) models within healthcare. This trend arises from the availability of extensive medical imaging data and notable advances in graphics processing unit (GPU) computational capabilities. Trained DL models show promise in supporting clinicians with tasks like image segmentation and classification. However, advancement of these models into clinical validation remains limited due to two key factors. Firstly, DL models are trained on off-premises environments by DL experts using Unix-like operating systems (OS). These systems rely on multiple libraries and third-party components, demanding complex installations. Secondly, the absence of a user-friendly graphical interface for model outputs complicates validation by clinicians. Here, we introduce a conceptual Computer-Aided Detection (CAD) pipeline designed to address these two issues and enable non-AI experts, such as clinicians, to use trained DL models offline in Windows OS. The pipeline divides tasks between DL experts and clinicians, where experts handle model development, training, inference mechanisms, Grayscale Softcopy Presentation State (GSPS) objects creation, and containerization for deployment. The clinicians execute a simple script to install necessary software and dependencies. Hence, they can use a universal image viewer to analyze results generated by the models. This paper illustrates the pipeline's effectiveness through a case study on pulmonary embolism detection, showcasing successful deployment on a local workstation by an in-house radiologist. By simplifying model deployment and making it accessible to non-AI experts, this CAD pipeline bridges the gap between technical development and practical application, promising broader healthcare applications.

Keywords: Computer-aided detection, grayscale softcopy presentation state, machine learning, deep learning, pulmonary embolism.

1 Introduction

In recent years, there has been a notable surge in both research and development pertaining to applications grounded in machine learning (ML) and deep learning (DL) models within the healthcare system [1]. This trend is aroused by the availability of extensive volumes of medical imaging data estimated at 4.2 billion annual diagnostic examinations worldwide [2], along with significant advancements in graphical computational capacities now measured in tera floating point operations per second [3]. Within these advances, researchers have demonstrated the potential of trained ML/DL models to support clinicians in a variety of medical image-processing tasks, including segmentation [4], classification [5]. However, adoption of these trained ML/DL models by clinicians for clinical validation purposes is hindered by two main factors. First, ML/DL models are trained on off-premises environments by ML/DL experts using Unix-like operating systems (OS) such as Ubuntu and macOS. Preparing the development environment for ML/DL model training on Unix-like operating systems requires proper installation of libraries, packages, and third-party components. Therefore, these challenging complex installations are not easy to do by non-AI experts such as clinicians. Second, the absence of a user-friendly graphical interface for trained ML/DL model outputs to maintain easy integration with a universal image viewer.

Cloud services that use Infrastructure as a Service (IaaS) architecture, such as Amazon Web Services (AWS), Microsoft Azure, and Google Cloud, are the common way of deploying trained ML/DL models by ML/DL experts [6]. While there have been advancements in the creation of cloud-based services for deploying DL models, the assessment of models on these platforms presents challenges, primarily because of regulations related to general data protection or local institutional policies. To address these challenges, we proposed a conceptual end-to-end Computer-Aided Detection (CAD) pipeline that can be easily implementable on the Windows OS, which does not require expert-level knowledge. The proposed conceptual pipeline enables non-AI experts such as radiologists and clinicians to perform trained ML/DL models offline within on-premises environments.

2 Objectives and Concepts

Our objective is to present a conceptual pipeline designed to simplify the offline installation of trained ML/DL models on local workstations, enabling non-AI experts, such as clinicians, to carry out this process with ease. To ensure this, the pipeline we propose involves two distinct parties: ML/DL experts and clinicians. ML/DL experts are responsible for both the development and deployment of the trained models, while clinicians are primarily tasked with utilizing these trained models. Unlike ML/DL experts, clinicians typically have greater familiarity with the Windows operating system [7].

As a result, it is imperative that trained ML/DL models are easily adaptable for use within the Windows OS environment. With the advent of containerization technology and the introduction of the Windows Subsystem for Linux (WSL), a new feature within the Windows OS, it has become notably straightforward to employ models developed on Unix-based operating systems on Windows platforms. Here, we have presented the pipeline, as explained below, by leveraging the capabilities of these two technologies.

First, we made a clear division of tasks between ML/DL experts and clinicians when utilizing deep learning models for medical image analysis. ML/DL experts are responsible for a series of complex tasks structured as follows: The DL model development and training, encompassing essential preprocessing steps to prepare the data for training, creating the model's inference mechanism, generating Grayscale Softcopy Presentation State (GSPS) objects for image annotations, saving the trained model in standard formats like HDF5 or pickle, packaging the model into a Docker container for deployment, crafting a bash script for automated setup of CUDA drivers, Docker, and Windows Subsystem for Linux (WSL), and finally, uploading all essential files, including the model, Docker configurations, and scripts, to platforms like GitHub for version control, collaboration, or storage purposes. As aimed, the clinicians have a less complex set of tasks. They just need to download and execute a single bash script uploaded to GitHub by the ML/DL experts. This script typically handles the installation of essential software and dependencies needed to run the DL model. Once the setup is complete, clinicians utilize a universal image viewer to analyze the results generated by the GSPS objects.

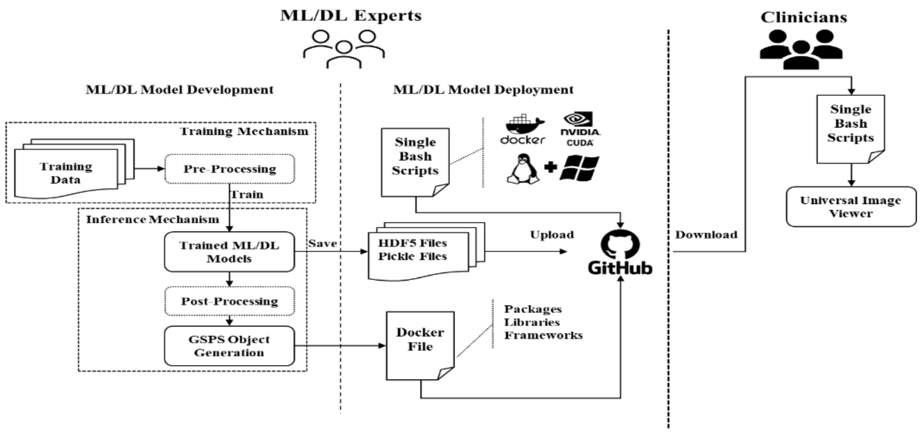

Fig. 1. The illustration of the end-to-end Computer-Aided Detection pipeline.

3 Case Pipeline: Pulmonary Embolism Detection

To assess the usability of the proposed pipeline, we followed the same procedures outlined in the previous section, using an in-house trained DL model for the detection of pulmonary embolisms [4]. First, the trained model was saved in the pickle file format. Second, a bash script file was generated to facilitate the proper execution of the DL model. As a final step, all files were uploaded to a file storage system for sharing. Our in-house radiologist was able to successfully deploy and test the trained DL model on 100 CT pulmonary angiography (CTPA) volume images at the local workstation without requiring expert assistance. The results of 97 of these 100 CTPAs were successfully analyzed in the universal image viewer.

4 Conclusion

In conclusion, our proposed CAD pipeline offers a practical solution to the challenges of deploying trained ML/DL models in on-premises environments. By streamlining the process and making it accessible to non-AI experts, such as clinicians, we bridge the gap between technical model development and practical application. Leveraging containerization technology and the Windows Subsystem for Linux, we've simplified the deployment of Unix-based models on Windows platforms, ensuring adaptability and usability. A successful case study in pulmonary embolism detection underscores the effectiveness of this approach, promising broader applications in healthcare.

References

1. Esteva, A., Robicquet, A., Ramsundar, B., Kuleshov, V., DePristo, M., Chou, K., Cui, C., Corrado, G., Thrun, S., Dean, J.: A guide to deep learning in healthcare. Nat Med. 25, 24–29 (2019). https://doi.org/10.1038/s41591-018-0316-z.
2. UNSCEAR 2020/2021 Report Volume I, //www.unscear.org/unscear/en/publications/2020_2021_1.html, last accessed 2023/09/18.
3. Dally, W.J., Keckler, S.W., Kirk, D.B.: Evolution of the Graphics Processing Unit (GPU). IEEE Micro. 41, 42–51 (2021). https://doi.org/10.1109/MM.2021.3113475.
4. Kahraman, A.T., Fröding, T., Toumpanakis, D., Gustafsson, C.J., Sjöblom, T.: Deep Learning Based Segmentation for Pulmonary Embolism Detection in Real-World CT Angiography: Classification Performance, https://www.medrxiv.org/content/10.1101/2023.04.21.23288861v2, (2023). https://doi.org/10.1101/2023.04.21.23288861.
5. Esteva, A., Kuprel, B., Novoa, R.A., Ko, J., Swetter, S.M., Blau, H.M., Thrun, S.: Dermatologist-level classification of skin cancer with deep neural networks. Nature. 542, 115–118 (2017). https://doi.org/10.1038/nature21056.
6. Nyarko, K., Taiwo, P., Duru, C., Masa-Ibi, E.: AI/ML Systems Engineering Workbench Framework. In: 2023 57th Annual Conference on Information Sciences and Systems (CISS). pp. 1–5 (2023). https://doi.org/10.1109/CISS56502.2023.10089781.
7. Newaz, A.I., Sikder, A.K., Rahman, M.A., Uluagac, A.S.: A Survey on Security and Privacy Issues in Modern Healthcare Systems: Attacks and Defenses. ACM Trans. Comput. Healthcare. 2, 27:1-27:44 (2021). https://doi.org/10.1145/3453176.