Characterization of Pneumonia Diagnostic Uncertainty: A Case Study on The CheXpert Dataset

Amyr Allan
UDESC-CCT
Joinville, Brazil
amyr.allan@hotmail.com

Gilmário Barbosa dos Santos
UDESC-CCT
Joinville, Brazil
gilmario.santos@udesc.br

***Abstract*—Pneumonia is a serious respiratory infection that presents significant diagnostic challenges due to the variability in its symptoms and its overlap with other respiratory diseases. This study investigates the potential of diagnostic uncertainty labels to enhance CAD system's pneumonia classification. Specifically, it explores the feasibility of a ternary classification approach (classifying X-rays as positive, negative, or uncertain), introducing uncertainty as a distinct diagnostic category, aiming to provide a more nuanced and cautious classification of pneumonia. Data processing techniques, including undersampling to balance classes, image resizing, and data augmentation, were applied. Transfer learning with the CheXNet model was then employed in a Monte Carlo cross-validation framework across 16 random data splits. The ROC curves and the areas under the ROC curves for the uncertainty class were analyzed, challenging the notion that uncertainty cannot be effectively characterized. The results indicated a degree of class separation, indicating that the uncertainty carried enough information to be characterized and suggesting the viability of the envisioned ternary model. Additionally, due to the exclusive use of frontal view X-rays and application of undersampling, results are expected to be further improved in future research.**

***Keywords*—Transfer Learning; CheXpert; CheXNet; Uncertainty; Pneumonia Classification.**

I. Introdution

Pneumonia is a severe respiratory infection that causes millions of deaths every year [1], causing inflammation in the lungs and impairing the exchange of essential gases for life. It can range from mild to fatal, particularly among vulnerable groups such as children, the elderly, and individuals with compromised immune systems. Its diagnosis usually relies on radiological exams, such as thoracic X-rays, which are crucial for confirming the presence of the infection [2]. Due to its varying nature, the patterns that pneumonia can produce in radiographs are usually similar or even overlap with patterns from other diseases, adding a degree of uncertainty that can complicate its diagnosis [3]. With the increasing adoption of artificial intelligence technologies in medicine, computer-aided diagnosis (CAD) models have been widely used to assist in detecting abnormalities in radiological images [4]. Traditionally, these models are binary, classifying images as either positive or negative for a particular disease. While useful, these models have significant limitations, specially in cases where the distinction between one or more diseases is unclear. In such situations, a binary diagnosis may be inadequate, leading to sub-optimal clinical decisions that can negatively impact patient treatment. In this study, the CheXpert [5] database, known for its inclusion of uncertainty labels, was utilized to focus specifically on pneumonia observations of frontal X-rays. The aim was to assess the feasibility of a ternary classification approach, categorizing thoracic X-rays as positive, negative, or uncertain for pneumonia, so as to provide a more cautious auxiliary diagnosis and prevent immediate binarization of border cases. A transfer learning strategy was implemented using the CheXNet [6] model within a Monte Carlo cross-validation framework across 16 random data splits. The primary evaluation metrics were the ROC curve and the area under the ROC curve (AUC-ROC) for the uncertainty class. To evaluate whether the model had discriminative power over the uncertainty class, we analyzed the ROC and AUC-ROC metrics for uncertainty over all the 16 splits.

II. Convolutional Neural Networks (CNNs)

First introduced in 1989 by LeCun et al. [7], CNNs have deeply affected numerous fields of research. Due to their ability to automatically and adaptively learn features from multidimensional data, they revealed themselves to be particularly effective in visual data analysis tasks [8]. For this reason, they are widely used in the medical field [9], specially in radiology [10], being a fundamental part of many CAD systems.

A. CNNs Architecture

CNN models are usually composed of many different layers, all sharing complex parameters with each other. Some of the more noticeable ones are:

1) Convolutional Layers: These layers use filters (or kernels) to perform a mathematical operation called convolution (hence the name) in input data. These filters are learnable parameters, meaning they can be optimized to produce meaningful feature maps that capture spatial hierarchies, such as edges, textures and objects [11].

2) Pooling Layers: These layers are used to diminish the spacial dimension of images and feature maps, reducing computational complexity and preventing overfitting (when the model fails to generalize to new data) [12].

3) Regularization Layers: Regularization layers in neural networks are used to introduce constraints to the training process. A common example is the dropout layer, a widely used regularization technique where a random fraction of neurons are ignored during each training iteration. This prevents the model from relying too heavily on specific neurons, encouraging it to learn more generalized features. Dropout is particularly effective in reducing overfitting and improving the model's ability to perform well on unseen data [13].

4) Activation Layers: Non-linear activation functions, like ReLU (Rectified Linear Unit), are applied after convolutional and dense layers to introduce non-linearity into the model, enabling it to learn more complex patterns.

5) Dense Layers: In dense layers, also know as fully connected layers, each neuron is connected to every neuron in the previous layer, which allows for the combination of features learned by earlier layers [14]. Therefore, they are typically at the end of CNN models, being used to map produced features to classes, classifying the data.

III. Transfer Learning

Transfer learning (TFL) is a broad term that is used to refer to model training techniques that leverage knowledge acquired from solving one problem to address a different, but related, problem [15]. By applying knowledge gained from previous tasks, TFL reduces the need for extensive training on a new problem, making it particularly valuable in situations where data is scarce. A common TFL example is feature extraction, where a previously trained model, called the *base model*, is used to extract features to be used in a new task. An illustrative example of transfer learning can be seen in Figure .

A. Fine-Tuning

The term *fine-tuning* refers to a specific subset of transfer learning techniques in which previously learned weights from a base model are slightly modified or "fine-tuned", allowing the base model to adapt to the characteristics of a new problem [16]. Although the terms "transfer learning" and "fine-tuning" are commonly used interchangeably in the literature, it's important to note that they are not synonymous [17].

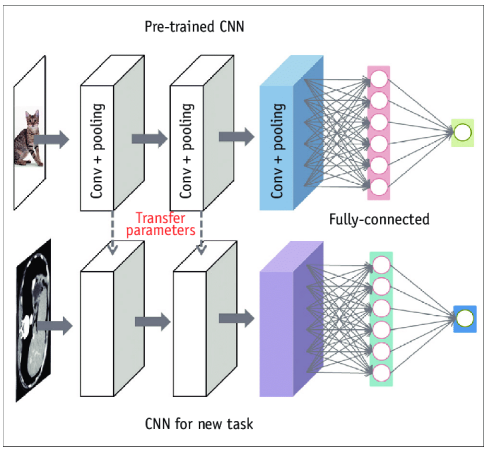

Fig. 1. Transfer learning example. See attributions section

1) Classifier Warmup and Joint Optimization: A very common problem faced by fine-tuning is what is referred to as *catastrophic forgetting*: when the modification of the base model's weights leads to the loss of previously learned knowledge [18]. To mitigate this issue, its considered best practice to firstly do a "classifier warm-up" stage in the fine-tuning pipeline. In this stage, the weights of the base model's convolutional layers are frozen, allowing for newly introduced dense layers to train by themselves. This step greatly reduces the risk of catastrophic forgetting in traditional fine-tuning methods [19] [17]. This stage is typically followed by a joint optimization stage in which the base model's convolutional layers are unfrozen, allowing all the layers to train jointly at a low learning rate.

IV. Model Evaluation

A neural network model that cannot effectively learn from its training data is of limited practical use, as it would result in a high number of false positives and false negatives—commonly referred to as *underfitting*. On the other hand, a model that fails to generalize to new data is equally problematic, as it cannot solve tasks outside of its training scope, a situation known as *overfitting*. To mitigate these issues, the model should be validated on unseen data during training, being evaluated using relevant and informative metrics. Additionally, performing cross-validation is considered a best practice. This method involves analyzing the model's performance across different training and validation partitions, offering an unbiased estimation of its generalization and overall performance. The evaluation metrics and methods employed in this study are typically defined for binary classification (positive and negative predictions) but can be adapted for multi-class problems using

a one-vs-rest approach [20], as implemented by the authors. A brief overview of these metrics and methods is provided below.

A. ROC Curve and AUC-ROC

1) ROC Curve: First used in signal analysis in the Second World War to detect enemy objects in the battlefield [21], the ROC curve presents itself as a useful metric to asses model quality. It is truly multidisciplinary, being used in the fields of meteorology, astronomy, medicine, computer science and others. The ROC curve plots, as can be seen in , the false positive rate (FPR) against the true positive rate (TPR) as the decision threshold decreases. The TPR is the proportion of correctly identified positive cases out of all positive samples, while the FPR represents the proportion of false positives among all negative samples. In ROC space, a perfect classifier is represented by the point (0, 1), indicating 100% TPR and 0% FPR at a specific decision threshold. In contrast, a classifier that performs no better than random guessing is represented by the diagonal identity line, reflecting no distinction between positive and negative cases at all thresholds [22].

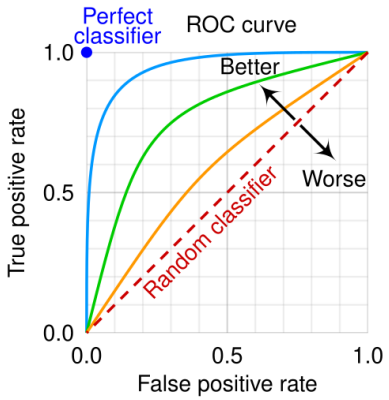

Fig. 2. ROC example. See attributions section.

2) AUC-ROC: Usually used in conjunction with the ROC curve, the AUC-ROC score represents the area under the ROC curve, providing a single value that summarizes the model's overall ability to distinguish between positive and negative classes. A perfect model achieves an AUC-ROC of 1, indicating flawless discrimination. In contrast, a random classifier has an AUC-ROC of 0.5, reflecting no predictive power [22].

B. Cross-Validation

When evaluating the performance of a model or assessing data quality, relying solely on static partitions for training and validation can be misleading. Observed performance on a specific partition might not accurately reflect how the model will perform on other partitions or in real-world scenarios. To address this, it is considered best practice to use cross-validation, which provides a more reliable assessment of a model's performance.

Cross-validation is a robust evaluation technique designed to test a model's ability to generalize to unseen data. This method involves dividing the dataset into multiple subsets, known as "folds". During the cross-validation process, the model is trained and validated multiple times, with each fold serving as both training data and a validation data at different stages. By systematically rotating through these folds, cross-validation ensures that the model is evaluated across a diverse range of data scenarios. This comprehensive approach minimizes biases and provides a more accurate measure of the model's performance.

In addition to cross-validation, a final evaluation on a separate test subset is optional but can be beneficial. This final evaluation involves assessing the model's performance on a completely unseen subset of data that was not used during the cross-validation process.

1) Monte Carlo Cross-Validation: The Monte Carlo cross-validation (MCCV) method consists in training and validating the model with random splits in a repeated fashion, with no guarantee of non-overlapping samples between them. Because of this, when compared to other cross-validation methods, MCCV tends to reduce variability and computational complexity at a cost of increased bias [23].

V. Materials and Methods

This work was conducted on Fedora 38 in a Anaconda [24] environment with Python 3.11.4 on a Ryzen 7 3800X machine with 48GB of RAM and a P2200 Quadro graphics card. Detailed descriptions of the used dataset, pre-processing techniques, model training and evaluation are provided bellow.

A. Dataset

A particularly challenging aspect of training a model to classify diagnostic uncertainty is the apparent lack of radiograph datasets with uncertainty labels. As a result, the choice of dataset was primarily driven by the availability of these annotations, leading to the selection of CheXpert. CheXpert is a large public dataset for chest radiograph interpretation, consisting of 224,316 high-quality frontal and lateral thoracic X-rays from 65.240 patients.

B. Base Model

Given its extensive training for pneumonia classification, the CheXNet convolutional network was adopted as the base model for this experiment. The top layers of CheXNet, including the dense layers responsible for classification, were removed and replaced with new ones.

C. Pre-Processing

The scikit-learn [25] and imblearn [26] libraries were used in the pre-processing phase for their broad set of utility functions. Since CheXNet was trained exclusively on frontal view thoracic x-rays for pneumonia classification, this experiment was similarly restricted to frontal views of the thoracic region, focusing specifically on pneumonia cases. Consequently, the effective size of the dataset was reduced significantly. The resulting distribution of the cases can be visualized in Table I.

TABLE I
CHEXPERT'S PNEUMONIA OBSERVATIONS

Pneumonia Label	Num. of Cases
Positive	3738
Negative	170685
Uncertain	16604

Due to the highly skewed class distribution, which had significantly impaired the model's performance in previous experiments, random undersampling was applied to achieve class balance, resulting in 11,214 cases evenly distributed across all classes. The remaining images were resized to 224x224 pixels and augmented with horizontal flip transformations to enhance data diversity. Due to the fact that CheXNet was itself based on DenseNet121 [27], the images also had to be normalized according to the ImageNet dataset [28] standards.

D. Model Architecture

In light of it's ease of use and extensive documentation, the keras [29] library was used for model construction, training and evaluation. As said previously, this experiment used CheXNet as a base model. The base model's original classification layers were removed. It was followed by a dropout layer for regularization, two dense layers with ReLU activation and an output layer with three neurons and softmax activation, as illustrated in Figure 3.

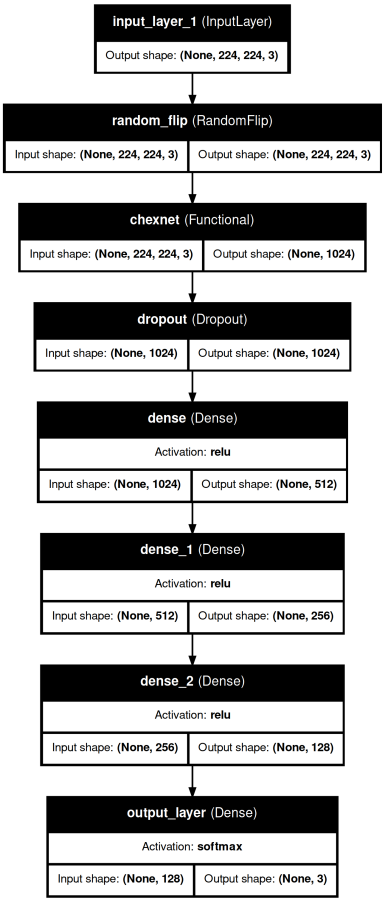

Fig. 3. The Model's Architecture

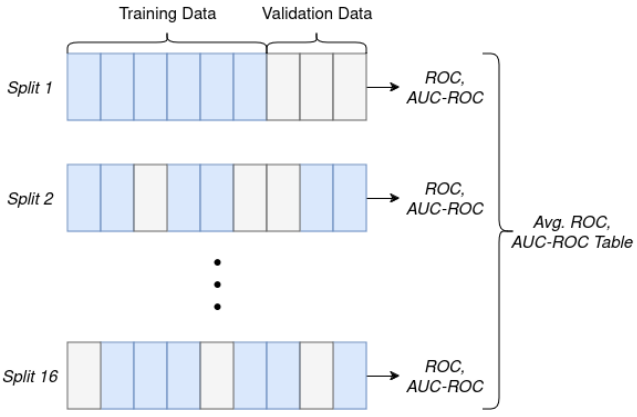

Fig. 4. MCCV Framework

E. Training and Evaluation

To ensure robust results, the training-evaluation pipeline was conducted in a Monte Carlo cross-validation framework (MCCV), repeating the process across 16 iterations on randomly split data, as depicted in Figure 4. This cross-validation method was selected due to its balance of ease of use and memory efficiency. The Adam optimizer [30] and Categorical Cross Entropy were used for optimization and loss calculation, respectively.

Training proceeded in two stages: classifier warmup and joint optimization. In the classifier warmup stage, the base model was frozen, allowing only the newly added dense layers to be trained independently. The training was set to run for 40 epochs

with an initial learning rate of 10^{-3}. Two callbacks were employed during this stage: *EarlyStopping*, which halted training if the loss plateaued for 8 epochs, and *ReduceLROnPlateau*, which reduced the learning rate by a factor of 10 if the loss plateaued for 4 epochs.

In the joint optimization stage, the base model was unfrozen, enabling simultaneous training of both the pre-trained and dense layers. This stage was conducted over 4 epochs with a fixed learning rate of 10^{-4}.

Finally, the ROC curve and AUC-ROC for the uncertainty class were calculated using the validation data. The results from each iteration were then aggregated to provide a comprehensive assessment of the model's discriminative power for the uncertainty class.

VI. Results and Discussion

Figure 5 displays the mean ROC curve for the uncertainty class computed across all splits. The individual ROC curves were omitted in order to avoid clutter. Table II presents the AUC-ROC values for each split. As can be seen, the mean ROC curve rests above the chance level, reflecting an AUC-ROC value of 61%. AUC-ROC values for each individual split also surpass the chance level, indicating a notable degree of class separation.

A. Potential Drawbacks

In radiology, the standard procedure for thoracic x-rays involves obtaining two distinct views: a frontal (typically posterior-anterior) view and a lateral view [31]. This dual-view approach allows for a more thorough evaluation of the thoracic cavity, as it provides different angles that can reveal potential abnormalities that might not be visible from a single perspective. Similarly, in the field of computer vision, incorporating multi-view images can enhance the accuracy of AI models, being particularly beneficial in scenarios involving high inter-class similarity [32], which the authors suspect characterizes their data. They believe that neglecting the lateral views, which are available in CheXpert, impacted its performance. Another factor that may have affected performance is the use of undersampling, which might have diminished data diversity.

VII. Conclusion

These results indicate that the model consistently performs better than random guessing, suggesting that the uncertainty class contains sufficient information to be effectively characterized and supporting the viability of a ternary model. Additionally, it is important to note that these results are expected to improve, given that the study was limited to frontal views and involved undersampling, which may have restricted data diversity.

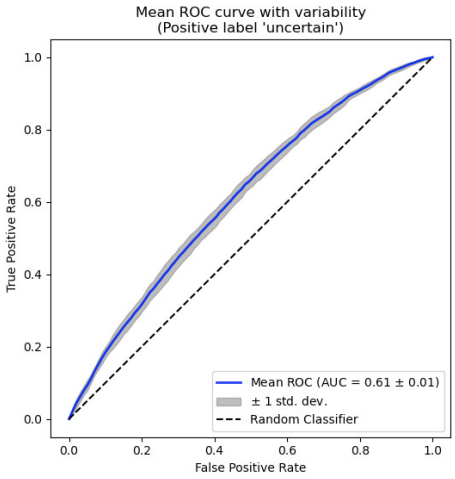

Fig. 5. Average Roc Curve with std. dev.

TABLE II
Splits' AUC-ROC for Uncertainty

Split	AUC-ROC
1	0.6129
2	0.6141
3	0.6263
4	0.6407
5	0.6105
6	0.6195
7	0.6052
8	0.6211
9	0.6236
10	0.5899
11	0.5994
12	0.6101
13	0.6203
14	0.6095
15	0.6272
16	0.6076

VIII. Future Work

In future work, the authors pretend to incorporate CheXpert's available lateral view X-rays. Additionally, the authors aim to identify an alternative solution to undersampling, in order to mitigate the impact of class imbalance during the training and validation stages.

Acknowledgment

The authors would like to acknowledge the contributions of the University of the State of Santa Catarina (UDESC)

and the Research Support Foundation of the State of Santa Catarina (FAPESC) for their financial and structural support in the development of this project.

Attributions

- Figure 2 credited to cmglee, MartinThoma, under CC BY-SA 4.0 (https://creativecommons.org/licenses/by-nc/4.0), via Wikimedia Commons. No changes were made to the image.
- Figure 1 credited to Synho Do, under CC BY-SA 4.0 (https://creativecommons.org/licenses/by-nc/4.0/). No changes were made to the image.

References

[1] K. Thomas, "Global burden of pneumonia," *International Journal of Infectious Diseases*, vol. 45, p. 1, Apr 2016. [Online]. Available: https://doi.org/10.1016/j.ijid.2016.02.027

[2] D. Wootton and C. Feldman, "The diagnosis of pneumonia requires a chest radiograph (x-ray)—yes, no or sometimes?" *Pneumonia*, vol. 5, no. 1, pp. 1–7, Dec 2014. [Online]. Available: https://doi.org/10.15172/pneu.2014.5/464

[3] O. Julie, "Pneumonia: challenges in the definition, diagnosis, and management of disease." *The Surgical clinics of North America*, 2014.

[4] R. Najjar, "Redefining radiology: A review of artificial intelligence integration in medical imaging," *Diagnostics (Basel)*, vol. 13, no. 17, Aug. 2023.

[5] J. Irvin, P. Rajpurkar, M. Ko, Y. Yu, S. Ciurea-Ilcus, C. Chute, H. Marklund, B. Haghgoo, R. Ball, K. Shpanskaya, J. Seekins, D. A. Mong, S. S. Halabi, J. K. Sandberg, R. Jones, D. B. Larson, C. P. Langlotz, B. N. Patel, M. P. Lungren, and A. Y. Ng, "Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison," 2019. [Online]. Available: https://arxiv.org/abs/1901.07031

[6] P. Rajpurkar, J. Irvin, K. Zhu, B. Yang, H. Mehta, T. Duan, D. Ding, A. Bagul, C. Langlotz, K. Shpanskaya, M. P. Lungren, and A. Y. Ng, "Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning," 2017. [Online]. Available: https://arxiv.org/abs/1711.05225

[7] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, vol. 1, no. 4, pp. 541–551, 1989.

[8] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–44, 05 2015.

[9] S. DR, "Convolutional neural networks in medical image understanding: a survey." *Evolutionary Intelligence*, 2022.

[10] S. Soffer, A. Ben-Cohen, O. Shimon, M. M. Amitai, H. Greenspan, and E. Klang, "Convolutional neural networks for radiologic images: A radiologist's guide," *Radiology*, vol. 290, no. 3, pp. 590–606, 2019, pMID: 30694159. [Online]. Available: https://doi.org/10.1148/radiol.2018180547

[11] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," *Neural Information Processing Systems*, vol. 25, 01 2012.

[12] S. Sharma and R. Mehra, "Implications of pooling strategies in convolutional neural networks: A deep insight," *Foundations of Computing and Decision Sciences*, vol. 44, pp. 303 – 330, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:201723325

[13] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, 2014. [Online]. Available: http://jmlr.org/papers/v15/srivastava14a.html

[14] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[15] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," *CoRR*, vol. abs/1911.02685, 2019. [Online]. Available: http://arxiv.org/abs/1911.02685

[16] J. Quinn. (2020) Dive into deep learning, chapter 14.2. [Online]. Available: https://d2l.ai/chapter_computer-vision/fine-tuning.html#fine-tuning

[17] Z. Li and D. Hoiem, "Learning without forgetting," 2017. [Online]. Available: https://arxiv.org/abs/1606.09282

[18] M. McCloskey and N. J. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," in *Psychology of Learning and Motivation*, ser. Psychology of Learning and Motivation, G. H. Bower, Ed. Academic Press, 1989, vol. 24, pp. 109–165. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0079742108605368

[19] F. Varno, L. M. Petry, L. D. Jorio, and S. Matwin, "Learn faster and forget slower via fast and stable task adaptation," 2020. [Online]. Available: https://arxiv.org/abs/2007.01388

[20] scikit learn. Multiclass receiver operating characteristic (roc) — scikit-learn 1.5.2 documentation.

[21] J. Egan, *Signal Detection Theory and ROC-analysis*, ser. Academic Press series in cognition and perception. Academic Press, 1975. [Online]. Available: https://books.google.com.br/books?id=V40oAAAAYAAJ

[22] T. Fawcett, "An introduction to roc analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, 2006, rOC Analysis in Pattern Recognition. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S016786550500303X

[23] P. Burman, "A comparative study of ordinary cross-validation, v-fold cross-validation and the repeated learning-testing methods," *Biometrika*, vol. 76, pp. 503–514, 09 1989.

[24] "Anaconda software distribution," 2020. [Online]. Available: https://docs.anaconda.com/

[25] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[26] G. Lemaître, F. Nogueira, and C. K. Aridas, "Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning," *Journal of Machine Learning Research*, vol. 18, no. 17, pp. 1–5, 2017. [Online]. Available: http://jmlr.org/papers/v18/16-365.html

[27] G. Huang, Z. Liu, and K. Q. Weinberger, "Densely connected convolutional networks," *CoRR*, vol. abs/1608.06993, 2016. [Online]. Available: http://arxiv.org/abs/1608.06993

[28] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[29] F. Chollet *et al.*, "Keras," https://keras.io, 2015.

[30] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017. [Online]. Available: https://arxiv.org/abs/1412.6980

[31] R. S. of North America (RSNA) and A. C. of Radiology (ACR). Image gallery.

[32] M. Seeland and P. Mäder, "Multi-view classification with convolutional neural networks," *PLOS ONE*, vol. 16, no. 1, pp. 1–17, 01 2021. [Online]. Available: https://doi.org/10.1371/journal.pone.0245230