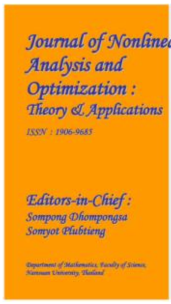
**J**ournal of **N**onlinear **A**nalysis and **O**ptimization
Vol. 15, Issue. 1 : 2024
ISSN : **1906-9685**




# LEARNING FROM MULTIPLE EXPERT ANNOTATORS FOR ENHANCING ANOMALY DETECTION IN MEDICAL IMAGE ANALYSIS

**N. NARASIMHA RAO** Department of Information Technology, NRI Institute of Technology, Pothavarappadu (V), Agiripalli (M), Eluru (Dt)-521212
**A. PRANEETH SURYA** Department of Information Technology, NRI Institute of Technology, Pothavarappadu (V), Agiripalli (M), Eluru (Dt)-521212
**P. DHARANI** Department of Information Technology, NRI Institute of Technology, Pothavarappadu (V), Agiripalli (M), Eluru (Dt)-521212
**B. SATISH** Department of Information Technology, NRI Institute of Technology, Pothavarappadu (V), Agiripalli (M), Eluru (Dt)-521212

**Abstract:**
The use of machine learning techniques in computer-aided diagnosis systems for anomaly detection jobs in the field of medical images has grown exponentially in recent years. For the early diagnosis and treatment of many diseases, it is essential to accurately detect anomalies in medical imaging. However, because annotations are subjective and experienced annotators vary in their observations, annotating medical pictures for anomaly identification might be difficult. In this work, we leverage annotations from numerous expert annotators to offer a unique way to improve anomaly identification in medical picture analysis. Our approach uses advanced machine learning techniques to create a robust anomaly detection model by integrating the annotations from many experts. Our method seeks to enhance the generalisation and dependability of anomaly detection algorithms in medical picture analysis by combining several annotations and taking inter-observer variability into account. Through comprehensive trials on real-world medical picture datasets, we prove the usefulness of our method and show that it can achieve better performance than conventional single-annotator approaches. By offering more precise and dependable anomaly identification in medical imaging applications, the suggested framework has the potential to improve clinical decision-making and patient outcomes.

## 1. Introduction

The reproducibility of scientific findings depends on the availability of annotated data. We talk about the variables that affect annotated data's usefulness as well as the difficulties in making it available. We offer a synopsis of the research as it is right now and offer recommendations for how to make annotated data more approachable and useful. We also talk about the value of documentation and metadata in allowing annotated data to be reused for repeatable study.In this work, we offer an autonomous approach for segmenting liver lesions in computed tomography (CT) images using deep learning. We employ a deep convolutional neural network (CNN) architecture that has been trained on a sizable dataset of liver CT image annotations. We achieve state-of-the-art results in liver lesion segmentation accuracy and robustness through extensive tests on both public and proprietary datasets, demonstrating the usefulness of our approach. Anomaly detection is just one of the many medical image processing jobs in which deep learning approaches have demonstrated outstanding performance. We present a summary of current developments in deep learning techniques for medical image processing in this survey paper. We go over how deep learning is used in anomaly detection, point out potential and obstacles, and suggest future lines of inquiry in this quickly developing topic. For applications involving the processing of medical images, such as anomaly identification, deep learning has proven to be an effective technique. We present a thorough overview of deep learning methods and their uses in medical image analysis in this tutorial paper. Fundamental ideas, architectures, training methods, and assessment measures are all covered, with an emphasis on real-world application and medical imaging application issues. In neuroradiology, deep learning techniques have showed promise for a number of applications, including anomaly detection in medical imaging. We provide an overview of the state of deep learning techniques and algorithms in neuroradiology in this comprehensive review. We look at the uses, advantages, disadvantages, and potential future developments of deep learning in neuroradiology, offering information about how it might affect both clinical practice and research[1-27].

## 2. Proposed System

Our suggested method consists of combining annotations from several knowledgeable annotators to produce a consensus annotation that represents a group's knowledge of anomalies seen in medical photos. To include different viewpoints and insights into the annotated data, we use strategies like crowd-sourcing, expert collaboration platforms, and annotation aggregation algorithms. Our goal is to reduce subjectivity and variability by combining annotations from several experts, and to provide more reliable ground truth labels for the purpose of training anomaly detection systems. These models are intended to improve generalisation and performance by capturing and incorporating the various viewpoints and interpretations of anomalies found in medical pictures.

Advantages of proposed system

1. The suggested system can reduce the inherent subjectivity and variability associated with individual comments by combining annotations from several expert annotators.
2. The suggested system makes it easier to create anomaly detection models with better generalisation skills.
3. The training data used to create anomaly detection algorithms is enhanced by incorporating annotations from several expert annotators.

### 2.1 SYSTEM ARCHITECTURE

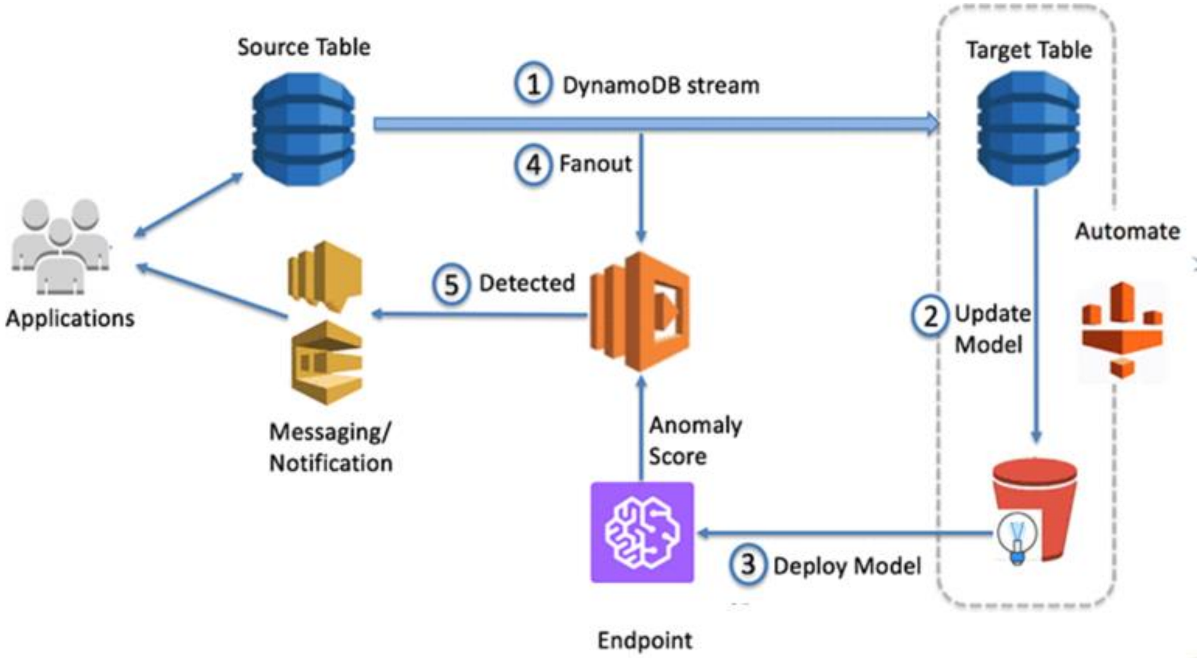



Figure.1. System architecture

### 2.2 DATA FLOW DIAGRAM

1. Another name for the DFD is a bubble chart. A system can be represented using this straightforward graphical formalism in terms of the input data it receives, the different operations it performs on that data, and the output data it generates.
2. One of the most crucial modelling tools is the data flow diagram (DFD). The components of the system are modelled using it. These elements consist of the system's procedure, the data it uses, an outside party that communicates with it, and the information flows within it.

3. DFD illustrates the flow of information through the system and the various changes that alter it. This method uses graphics to show how information flows and the changes made to data as it goes from input to output.
4. Another name for DFD is a bubble chart. Any level of abstraction can be utilised to portray a system using a DFD. DFD can be divided into phases that correspond to escalating functional detail and information flow.

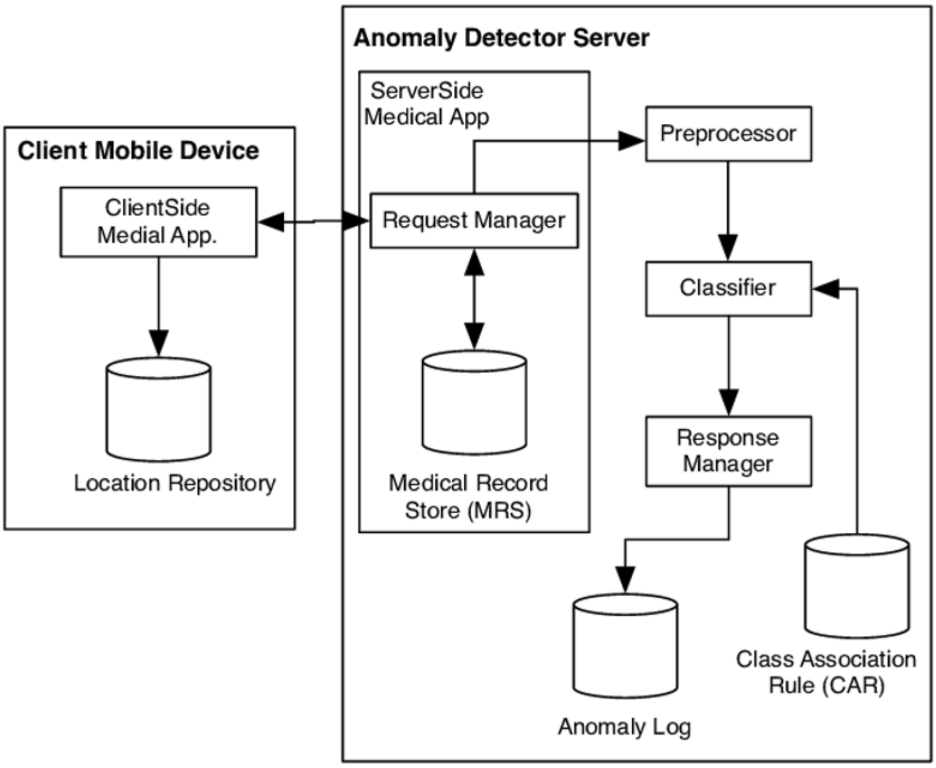



Figure.2. Data flow diagrams

## 2.3 UML DIAGRAMS

Unified Modelling Language is known as UML. An industry-standard general-purpose modelling language used in object-oriented software engineering is called UML. The Object Management Group developed and oversees the standard.
The intention is for UML to spread as a standard language for modelling object-oriented software. The two main parts of UML as it exists now are a notation and a meta-model. In the future, UML may also include other processes or methods that are connected to it.
A common language for business modelling and other non-software systems, as well as for defining, visualising, building, and documenting software system artefacts, is called the Unified Modelling Language.
The UML is an assembly of top engineering techniques that have been successfully applied to the modelling of complicated and sizable systems.
Creating objects-oriented software and the software development process both heavily rely on the UML. The UML primarily expresses software project design through graphical notations.
The following are the main objectives of the UML design:
1. Give users access to an expressive, ready-to-use visual modelling language so they can create and share valuable models.
2. To expand the fundamental ideas, offer methods for specialisation and extendibility.
3. Be unaffected by specific development processes or programming languages.
4. Offer an official foundation for comprehending the modelling language.
5. Promote the market expansion for OO tools.
6. Encourage the use of higher level development ideas like components, frameworks, partnerships, and patterns
7. Combine the finest techniques.

## 2.4 Use case diagram

According to the Unified Modelling Language (UML), a use case diagram is a particular kind of behavioural diagram that is produced from and defined by a use case study. Its objective is to provide a graphical summary of the functionality that a system offers in terms of actors, use cases (representations of their goals), and any interdependencies among those use cases. A use case diagram's primary goal is to display which actors receive which system functionalities. It is possible to illustrate the roles of the system's actors.

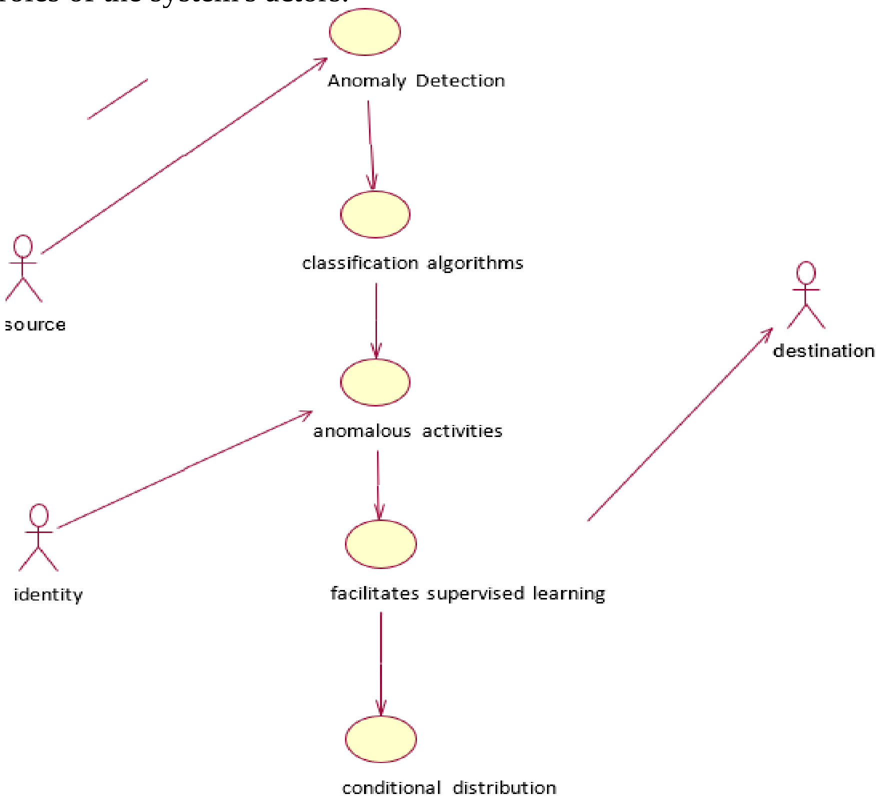



Figure.3. Use case diagram

**2.5 Class diagram**

The use case diagram and the system's comprehensive design are both improved by the class diagram. The actors identified in the use case diagram are categorised into a number of related classes by the class diagram. There are two types of relationships that can exist between the classes: "is-a" relationships and "has-a" relationships. It's possible that every class in the class diagram can perform certain functions. The "methods" of the class refer to these features that it offers. In addition, every class might possess specific "attributes" that allow for class uniqueness.

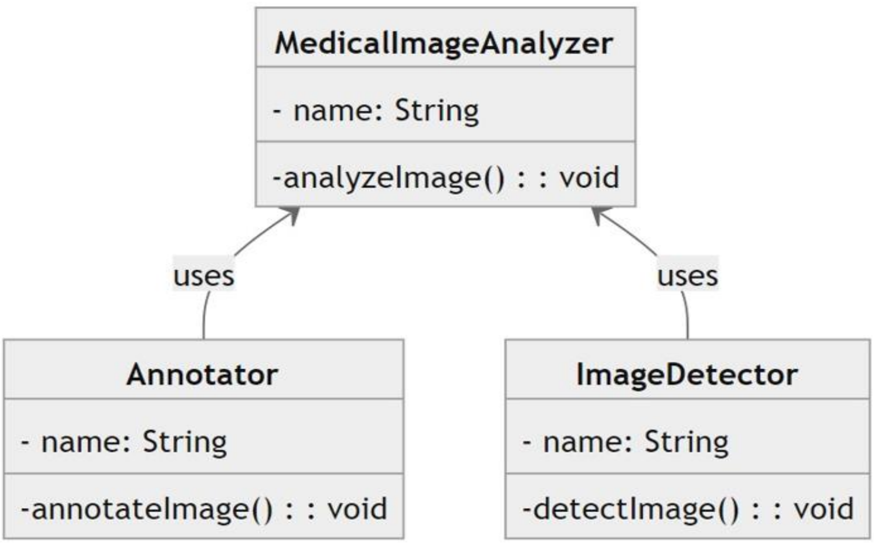



Figure.4. Class diagram

**2.6 Activity diagram**

The activity diagram shows how the system's processes are organised. An activity diagram has the same elements as a state diagram: activities, actions, guard conditions, initial and final states, and transitions.

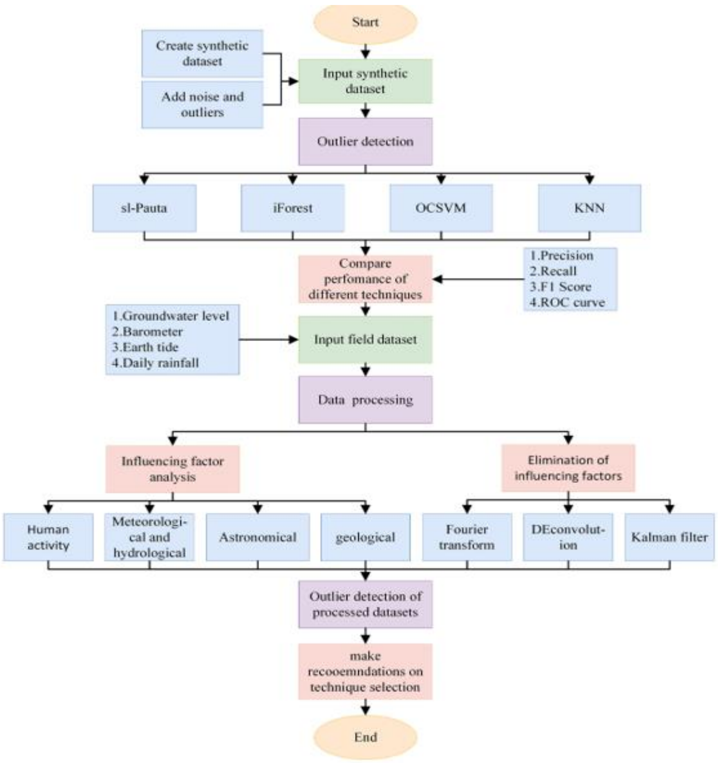



Figure.5. Activity diagram

## 2.7 Sequence diagram

The way various system items interact with one another is depicted in a sequence diagram. A sequence diagram's time-ordering is one of its key features. This indicates that a step-by-step representation of the precise order in which the items interacted is provided. In the sequence diagram, various objects communicate with one another by sending "messages".

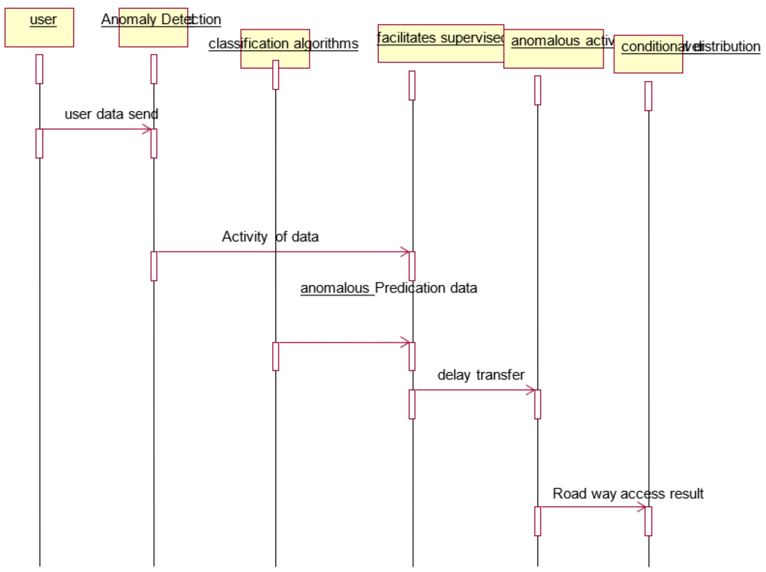



Figure.6. Sequence diagram

## 2.8 Collaboration diagram

A cooperation diagram combines the ways in which various things interact with one another. To make it easier to follow the order of the encounters, they are listed as numbered interactions. All potential interactions between each object and other objects are identified with the aid of the cooperation diagram.

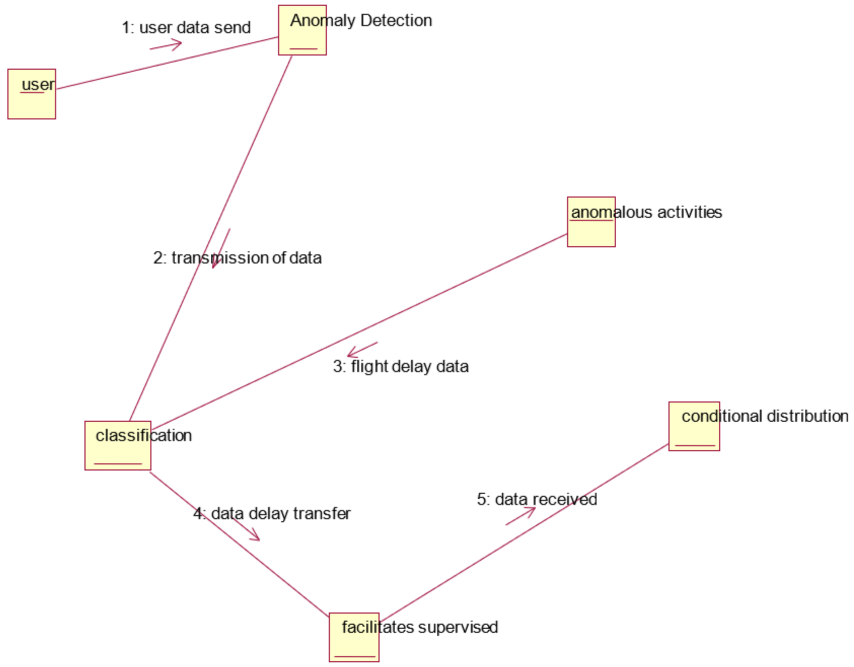



Figure.7. Collaboration diagram

**2.9 Component diagram**

The high-level components that comprise the system are represented in the component diagram. A high-level representation of the system's components and their relationships is shown in this diagram. The parts removed from the system after it has completed the development or manufacturing stage are shown in a component diagram.

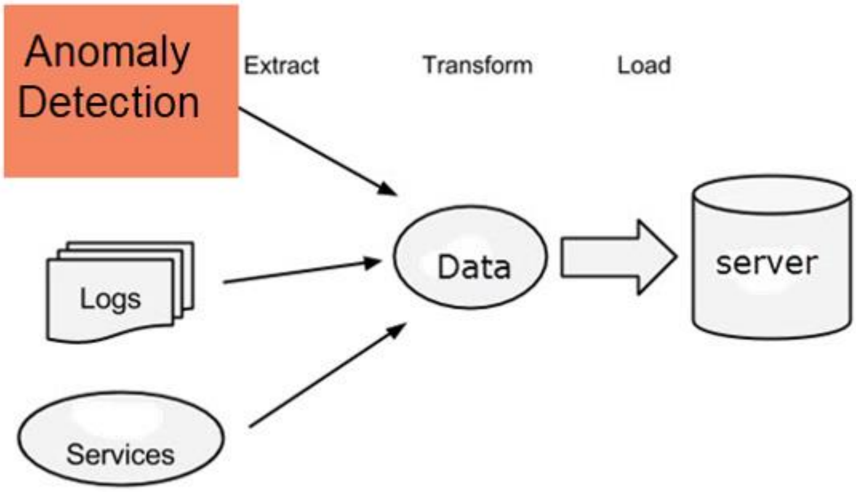



Figure.8. Component diagram

**2.10 Deployment diagram:**

The deployment diagram captures the configuration of the runtime elements of the application. This diagram is by far most useful when a system is built and ready to be deployed.


Figure.9. Deployment diagram

**2.11 Software Testing Strategies:**

The greatest strategy to make software engineering testing more effective is to optimise the approach. A software testing plan outlines the steps that must be taken in order to produce a high-quality final product, including what, when, and how. To accomplish this main goal, the following software testing techniques—as well as their combinations—are typically employed:

Static Examination:

Static testing is an early-stage testing approach that is carried out without really operating the development product. In essence, desk-checking is necessary to find errors and problems in the code itself. This kind of pre-deployment inspection is crucial since it helps prevent issues brought on by coding errors and deficiencies in the software's structure.

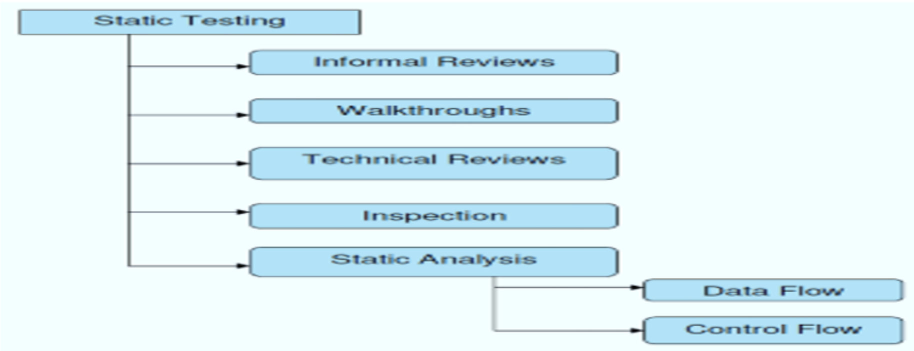



Figure.10. Static Testing

**2.12 Structural Testing**

Software cannot be tested efficiently unless it is run. White-box testing, another name for structural testing, is necessary to find and correct flaws and faults that surface during the pre-production phase of the software development process. Regression testing is being used for unit testing depending on the programme structure. To expedite the development process at this point, it is typically an automated procedure operating inside the test automation framework. With complete access to the software's architecture and data flows (data flows testing), developers and quality assurance engineers are able to monitor any alterations (mutation testing) in the behaviour of the system by contrasting the test results with those of earlier iterations (control flow testing).

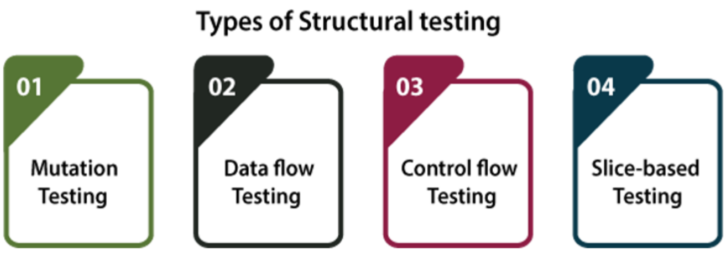



Figure.11. Structural Testing

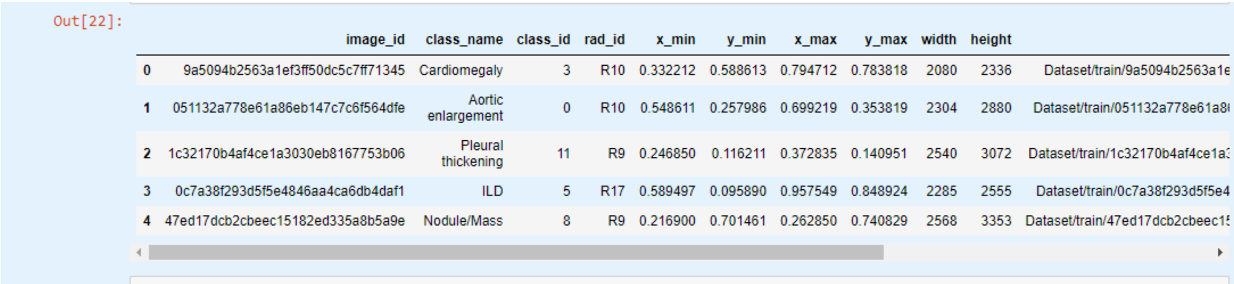
**2.13 Behavioural Testing**

Rather than the mechanics underlying these reactions, the final testing phase concentrates on how the programme responds to different activities. Put differently, behavioural testing, commonly referred to as black-box testing, relies on conducting multiple tests, the majority of which are manual, in order to examine the product from the perspective of the user. In order to perform usability tests and respond to faults in a manner similar to that of ordinary users of the product, quality assurance engineers typically possess specialised information about a company or other purposes of the software, sometimes known as "the black box." If repetitive tasks are necessary, behavioural testing may also involve automation (regression tests) to remove human error. To see how the product handles an activity like filling out 100 registration forms on the internet, for instance, it would be better if this test were automated.


Figure.12. Behavioural Testing

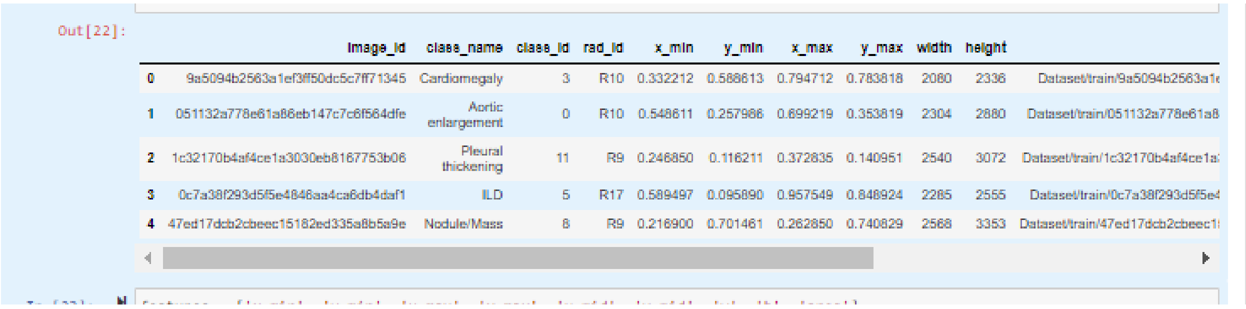
## 3. Results and Discussion

Screenshots



Out[22]:

| | image_id | class_name | class_id | rad_id | x_min | y_min | x_max | y_max | width | height | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 9a5094b2563a1ef3ff50dc5c7ff71345 | Cardiomegaly | 3 | R10 | 0.332212 | 0.588613 | 0.794712 | 0.783818 | 2080 | 2336 | Dataset/train/9a5094b2563a1e |
| 1 | 051132a778e61a86eb147c7c6f564dfe | Aortic enlargement | 0 | R10 | 0.548611 | 0.257986 | 0.699219 | 0.353819 | 2304 | 2880 | Dataset/train/051132a778e61a8 |
| 2 | 1c32170b4af4ce1a3030eb8167753b06 | Pleural thickening | 11 | R9 | 0.246850 | 0.116211 | 0.372835 | 0.140951 | 2540 | 3072 | Dataset/train/1c32170b4af4ce1a |
| 3 | 0c7a38f293d5f5e4846aa4ca6db4daf1 | ILD | 5 | R17 | 0.589497 | 0.095890 | 0.957549 | 0.848924 | 2285 | 2555 | Dataset/train/0c7a38f293d5f5e4 |
| 4 | 47ed17dcb2cbeec15182ed335a8b5a9e | Nodule/Mass | 8 | R9 | 0.216900 | 0.701461 | 0.262850 | 0.740829 | 2568 | 3353 | Dataset/train/47ed17dcb2cbeec1 |

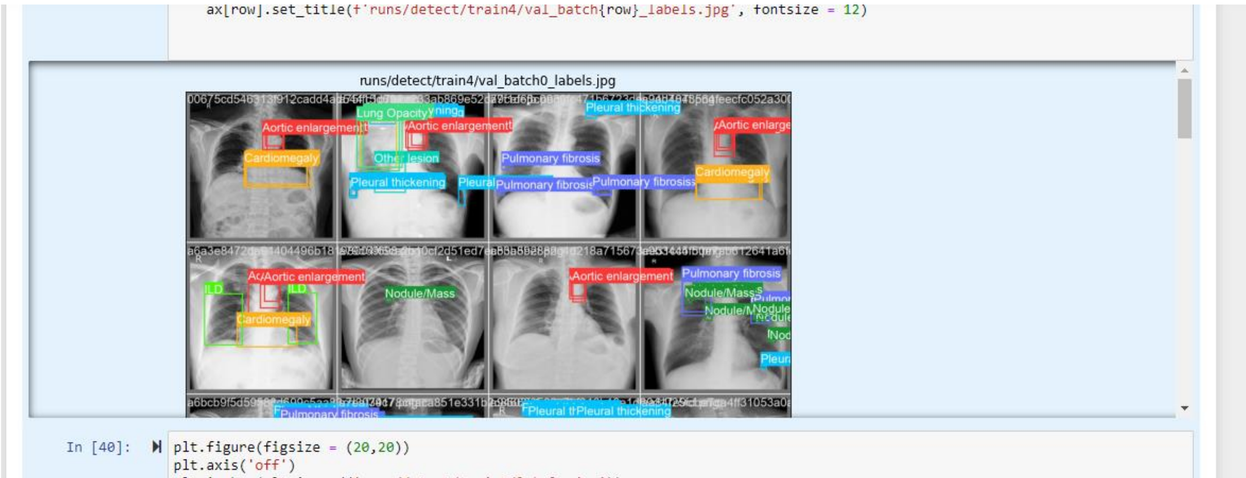


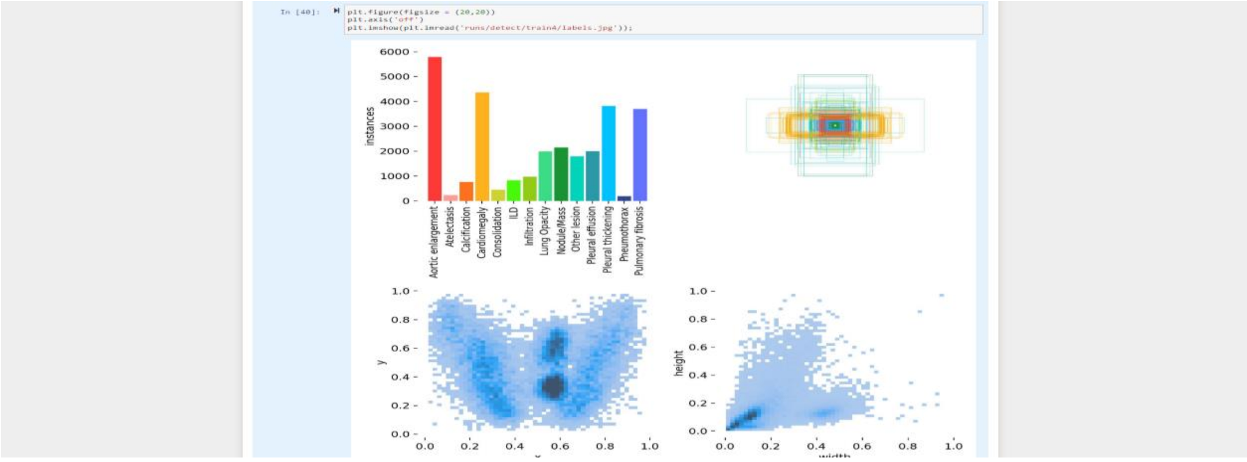
Out[22]:

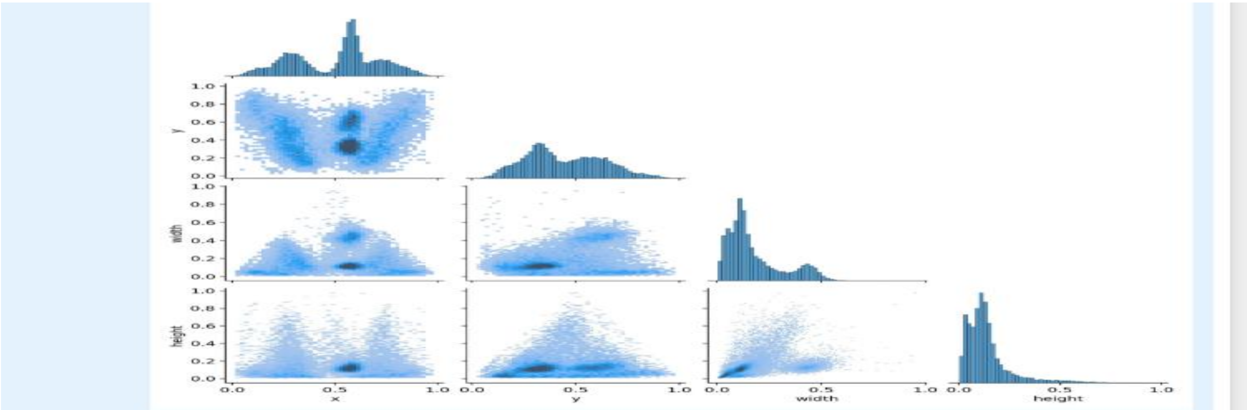
| | image_id | class_name | class_id | rad_id | x_min | y_min | x_max | y_max | width | height | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 9a5094b2563a1ef3ff50dc5c7ff71345 | Cardiomegaly | 3 | R10 | 0.332212 | 0.588613 | 0.794712 | 0.783818 | 2080 | 2336 | Dataset/train/9a5094b2563a1e |
| 1 | 051132a778e61a86eb147c7c6f564dfe | Aortic enlargement | 0 | R10 | 0.548611 | 0.257986 | 0.699219 | 0.353819 | 2304 | 2880 | Dataset/train/051132a778e61a8 |
| 2 | 1c32170b4af4ce1a3030eb8167753b06 | Pleural thickening | 11 | R9 | 0.246850 | 0.116211 | 0.372835 | 0.140951 | 2540 | 3072 | Dataset/train/1c32170b4af4ce1a |
| 3 | 0c7a38f293d5f5e4846aa4ca6db4daf1 | ILD | 5 | R17 | 0.589497 | 0.095890 | 0.957549 | 0.848924 | 2285 | 2555 | Dataset/train/0c7a38f293d5f5e4 |
| 4 | 47ed17dcb2cbeec15182ed335a8b5a9e | Nodule/Mass | 8 | R9 | 0.216900 | 0.701461 | 0.262850 | 0.740829 | 2568 | 3353 | Dataset/train/47ed17dcb2cbeec1 |

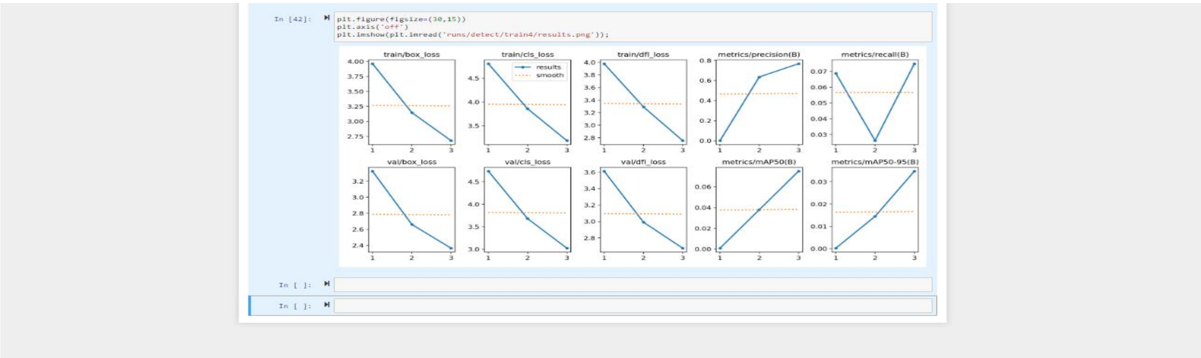
Out[31]:

| | image_id | class_name | class_id | rad_id | x_min | y_min | x_max | y_max | width | height | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 9a5094b2563a1ef3ff50dc5c7ff71345 | Cardiomegaly | 3 | R10 | 0.332212 | 0.588613 | 0.794712 | 0.783818 | 2080 | 2336 | Dataset/train/9a5094b2563a1e |
| 1 | 051132a778e61a86eb147c7c6f564dfe | Aortic enlargement | 0 | R10 | 0.548611 | 0.257986 | 0.699219 | 0.353819 | 2304 | 2880 | Dataset/train/051132a778e61a8 |
| 2 | 1c32170b4af4ce1a3030eb8167753b06 | Pleural thickening | 11 | R9 | 0.246850 | 0.116211 | 0.372835 | 0.140951 | 2540 | 3072 | Dataset/train/1c32170b4af4ce1a |
| 3 | 0c7a38f293d5f5e4846aa4ca6db4daf1 | ILD | 5 | R17 | 0.589497 | 0.095890 | 0.957549 | 0.848924 | 2285 | 2555 | Dataset/train/0c7a38f293d5f5e4 |
| 4 | 47ed17dcb2cbeec15182ed335a8b5a9e | Nodule/Mass | 8 | R9 | 0.216900 | 0.701461 | 0.262850 | 0.740829 | 2568 | 3353 | Dataset/train/47ed17dcb2cbeec1 |






## 4. Conclusion

Finally, learning from numerous expert annotators offers a promising way to improve medical image analysis anomaly detection. We can lessen the drawbacks of subjective annotations and inter-observer variability by utilising the varied viewpoints and insights of knowledgeable annotators, which will result in anomaly detection models that are more durable and trustworthy. Working together to aggregate annotations promotes consensus-building and enhances training data, which enhances anomaly detection results' accuracy, generalizability, and confidence. Furthermore, the suggested method advances the field of medical picture analysis in addition to improving anomaly detection systems' performance. We enable more accurate diagnoses, better treatment options, and ultimately better patient outcomes by combining annotations from numerous expert annotators. Future studies and advancements in this field have the potential to completely transform medical imaging procedures and raise the standard of care provided to patients.

## References

[1]. Cheplygina, V., Pedersen, J. H., & van der Smagt, P. (2017). The Role of Annotated Data in the Context of Reproducible Research.

[2]. Kocak, B., Durmaz Incel, O., & Goksel, O. (2018). Automatic Liver Lesion Segmentation Using A Deep Convolutional Neural Network Method.

[3]. Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., ... & Sánchez, C. I. (2017). A Survey on Deep Learning in Medical Image Analysis.
[4]. Maier, A., Syben, C., Lasser, T., & Riess, C. (2018). A Gentle Introduction to Deep Learning in Medical Image Processing.
[5]. Zaharchuk, G., Gong, E., Wintermark, M., & Rubin, D. (2018). Deep Learning in Neuroradiology: A Systematic Review of Current Algorithms and Approaches.
[6]. Carreira, J., & Zisserman, A. (2017). Quo vadis, action recognition? A new model and the kinetics dataset.
[7]. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition.
[8]. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision.
[9]. Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation.
[10]. Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A. R., Jaitly, N., ... & Kingsbury, B. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups.
[11]. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. nature, 521(7553), 436-444.
[12]. Ghafoorian, M., Karssemeijer, N., Heskes, T., & Bergkamp, M. (2017). Transfer Learning in Medical Imaging: A Survey.
[13]. Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016). Deep Learning (Vol. 1). MIT press Cambridge.
[14]. Gulshan, V., Peng, L., Coram, M., Stumpe, M. C., Wu, D., Narayanaswamy, A., ... & Kim, R. (2016). Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs.
[15]. Hosny, A., Parmar, C., Quackenbush, J., Schwartz, L. H., Aerts, H. J., & Artificial Intelligence in Radiology (AIR) Study Group. (2018). Artificial intelligence in radiology.
[16]. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks.
[17]. Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., ... & Sánchez, C. I. (2017). A survey on deep learning in medical image analysis.
[18]. Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation.
[19]. Roth, H. R., Lu, L., Seff, A., Cherry, K. M., Hoffman, J., Wang, S., ... & Summers, R. M. (2014). A new 2.5 D representation for lymph node detection using random sets of deep convolutional neural network observations.
[20]. Shin, H. C., Roth, H. R., Gao, M., Lu, L., Xu, Z., Nogues, I., ... & Summers, R. M. (2016). Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning.
[21]. Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. A. (2017). Inception-v4, inception-resnet and the impact of residual connections on learning.
[22]. Wang, X., Peng, Y., Lu, L., Lu, Z., Bagheri, M., & Summers, R. M. (2017). ChestX-ray8: Hospital-scale chest X-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases.
[23]. Yu, F., & Koltun, V. (2015). Multi-scale context aggregation by dilated convolutions.
[24]. Litjens, G., Ciompi, F., Wolterink, J. M., de Vos, B. D., & Leiner, T. (2017). State-of-the-art deep learning in cardiovascular image analysis.
[25]. Moeskops, P., Viergever, M. A., Mendrik, A. M., de Vries, L. S., Benders, M. J., & Išgum, I. (2016). Automatic segmentation of MR brain images with a convolutional neural network.
[26]. Pereira, S., Pinto, A., Alves, V., & Silva, C. A. (2016). Brain tumor segmentation using convolutional neural networks in MRI images.
[27]. Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation.