



e-ISSN: 2319-8753 | p-ISSN: 2347-6710

IJRSET

International Journal of Innovative Research in
SCIENCE | ENGINEERING | TECHNOLOGY

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 13, Issue 8, August 2024

ISSN INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA

Impact Factor: 8.524

 9940 572 462

 6381 907 438

 ijrset@gmail.com

 www.ijrset.com

Fetal Brain Abnormalities using Deep Learning

Mrs Yamunashree S¹, M Vasitha², Monica S V³, Prerana M R⁴

Assistant Professor, Department of Information Science and Engineering, RajaRajeswari College of Engineering,
Bangalore, Karnataka, India¹

Student, Department of Information Science and Engineering, RajaRajeswari College of Engineering, Bangalore,
Karnataka, India^{2,3,4}

ABSTRACT: This work uses deep learning, more specifically the You Only Look Once (YOLO) architecture, to address the crucial requirement for early detection and categorization of prenatal brain disorders. In order to recognize and categorize discrete anomalies in fetal brain images acquired using sophisticated imaging modalities, the suggested framework incorporates YOLO's real-time object identification capabilities. Our technique makes use of deep learning to automatically identify anatomical differences, agenesis of the corpus callosum, and ventri culomegaly, among other anomalies. In addition to streamlining the detection process, the YOLO-based method improves the precision and effectiveness of anomaly classification, allowing for prompt medical attention and guidance for expectant parents. Moreover, the study goes beyond simple identification and categorization by utilizing a unique segmentation strategy to accurately identify the impacted regions of the developing brain. The suggested model offers comprehensive maps that emphasize the areas connected to detected irregularities by integrating cutting-edge image segmentation algorithms. Healthcare providers can more easily evaluate the diagnostic data because to this segmentation, which also helps them create focused intervention plans. The potential of deep learning to transform prenatal brain abnormality assessment is demonstrated by the combination of YOLO-based detection, classification, and precise segmentation, providing a comprehensive and effective solution for early diagnosis and medical decision-making.

KEYWORDS: Convolutional Neural Network, YOLO.

I. INTRODUCTION

Our project emerges at the intersection of advanced technologies and the critical socio-economic issue of farmer suicides. With the agricultural sector being a cornerstone of many economies, the alarming rates of farmer suicides underscore the need for innovative solutions. This project endeavours to pioneer a transformative approach by integrating three key features into a unified system. The first facet involves leveraging Convolutional Neural Networks (CNN) for the early detection of leaf diseases, thereby addressing crucial plant health issues. By providing farmers with a tool to identify and manage crop diseases promptly, the system aims to enhance overall agricultural productivity and, in turn, alleviate economic pressures on farmers. The second part of the system is on applying YOLOv5 for animal entrance detection, protecting crops from unforeseen dangers presented by wildlife. Crop protection and agricultural sustainability are closely related, and the project's goal is to strengthen farmers' lives and reduce financial losses.

Essentially, the project's novel approach seeks to surpass traditional farming methods by converting them into a state-of-the-art, comprehensive system that tackles the various issues that farmers encounter. The initiative aims to change the agricultural landscape, promoting sustainability and resilience while greatly enhancing the well-being of farmers by combining early disease detection and animal intrusion avoidance. Methods like data warehousing, data fusion, and data virtualization can be used to accomplish this. Make predictions using the trained model or models on fresh or unobserved data. This could entail grouping data points into clusters, predicting future trends, or categorizing new occurrences.

II. METHODOLOGY

A library of programming functions called OpenCV (Open Source Computer Vision) is primarily focused on real-time computer vision. Initially created by Intel, Itseez (which Intel later purchased) and Willow Garage provided support for it. The open-source BSD license allows for the cross-platform usage of the library at no cost. TensorFlow, Torch/PyTorch, and Cafe are deep learning frameworks that are supported by OpenCV.

It supports Windows, Linux, Android, and Mac OS and includes interfaces in C++, Python, Java, and MATLAB. With a preference for real-time vision applications, OpenCV uses MMX and SSE instructions when they are available. Currently, there is considerable development underway to create fully functional CUDA and OpenCL interfaces. More than 500 algorithms exist, and around ten times as many functions either support or comprise those algorithms. OpenCV is written in native C++ and features a templated interface that integrates well with STL containers.

Originally started in 1999 as a part of a series of initiatives including real-time ray tracing and 3D display walls, the OpenCV project was an Intel Research endeavor to develop CPU intensive applications. The Performance Library Team at Intel and several optimization specialists from Intel Russia were the project's primary sponsors. The project's initial objectives for OpenCV were stated as follows:

- Contribute to the advancement of vision research by offering optimized and open source code for fundamental vision infrastructure. Let's stop inventing the wheel.
- Provide a standard platform for developers to build upon in order to disseminate vision knowledge and make code more readable and shareable.
- By providing portable, performance-optimized code for free with a license that does not need the source to be open or free itself, we may advance vision-based commercial applications.

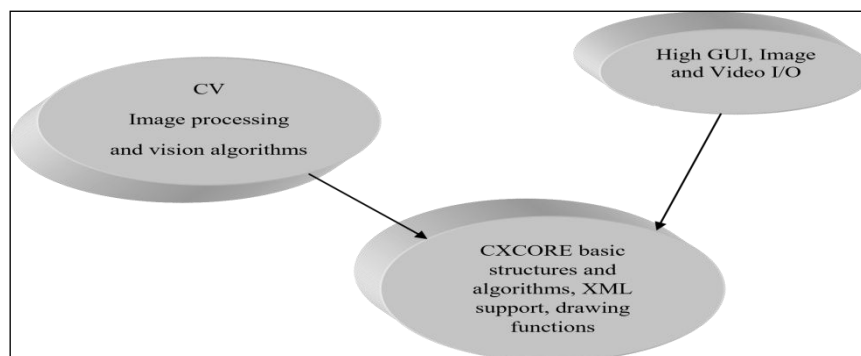


Fig – 2.1: Structure of Open CV.

2.1 OpenCV modules

- cv - Main OpenCV functions.
- cvaux - Auxiliary (experimental) OpenCV functions.
- cxcore - Data structures and linear algebra support.
- highgui - GUI functions.

2.2 OpenCV utilizing video capture

OpenCV facilitates the acquisition of pictures from a camera or an AVI video file. Capturing from a camera is started by setting up `CvCapture* capture = cvCaptureFromCAM(0);` // This captures the image from video device #0. `CvCapture* capture = cvCaptureFromAVI("infile.avi");` // To begin capturing from a file.

Getting a frame:

```

IplImage* img (0) ; in case of !cvGrabFrame(capture)
{
// take a picture
exit(0);
}
  
```

`img=cvRetrieveFrame(capture);`
`printf("Could not grab a frame\n\7");` // recover the recorded frame Grab one image from each camera before attempting to obtain images from multiple cameras at once. After the grabbing process is finished, retrieve the taken photos. Disseminating the source of capture: `cvReleaseCapture()` with `&capture`

III. SYSTEM ARCHITECTURE

System architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of system, organized in a way that supports reasoning about the structures and behaviors of a system.

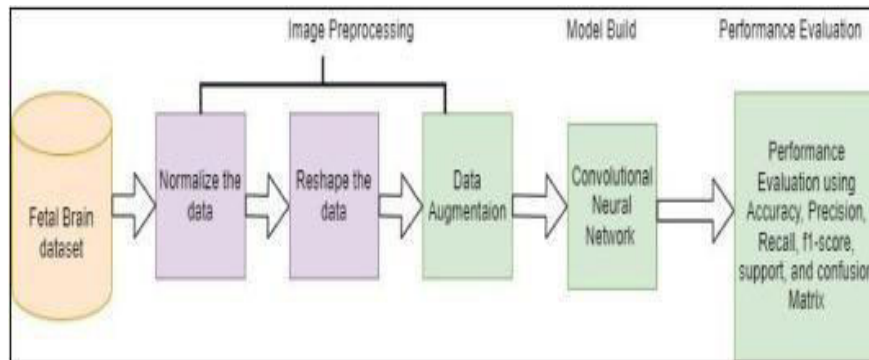


Figure3.1: System Architecture

3.1 Use Case Diagram

A use case diagram visually represents the interactions between various actors and the system to illustrate the functional requirements of a software system. Actors, representing external entities such as users or systems, interact with the system through use cases, which are specific functionalities or features. The diagram provides a high-level overview of the system's functionality, emphasizing the interactions and relationships between different actors and use cases, aiding in understanding the system's scope and purpose.

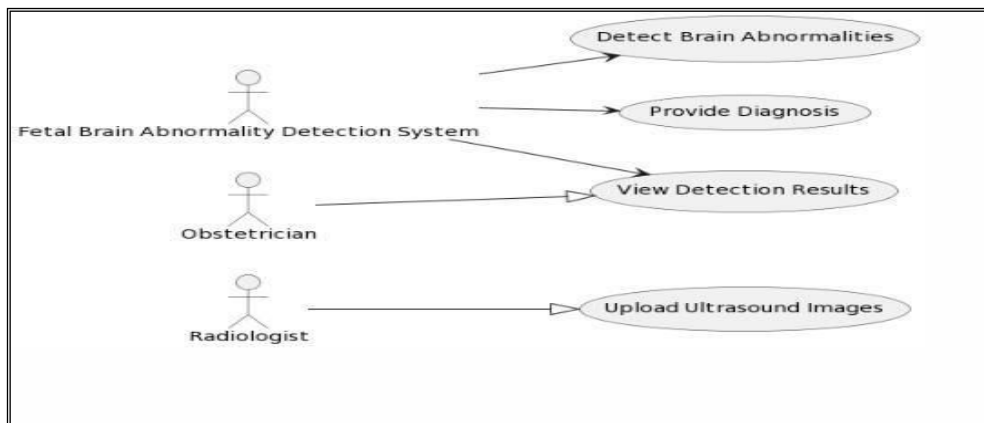


Figure3.2: Use Case Diagram

3.2 Sequence Diagram

Sequence diagrams depict the interactions between different components or objects in a system over time, showcasing the order in which messages are exchanged. Actors or objects are represented by vertical lifelines, and messages between them are depicted as horizontal arrows. The sequence diagram is particularly useful for understanding the dynamic behavior of a system during a specific scenario or use case, offering insights into the timing and sequence of interactions between various elements within the system.

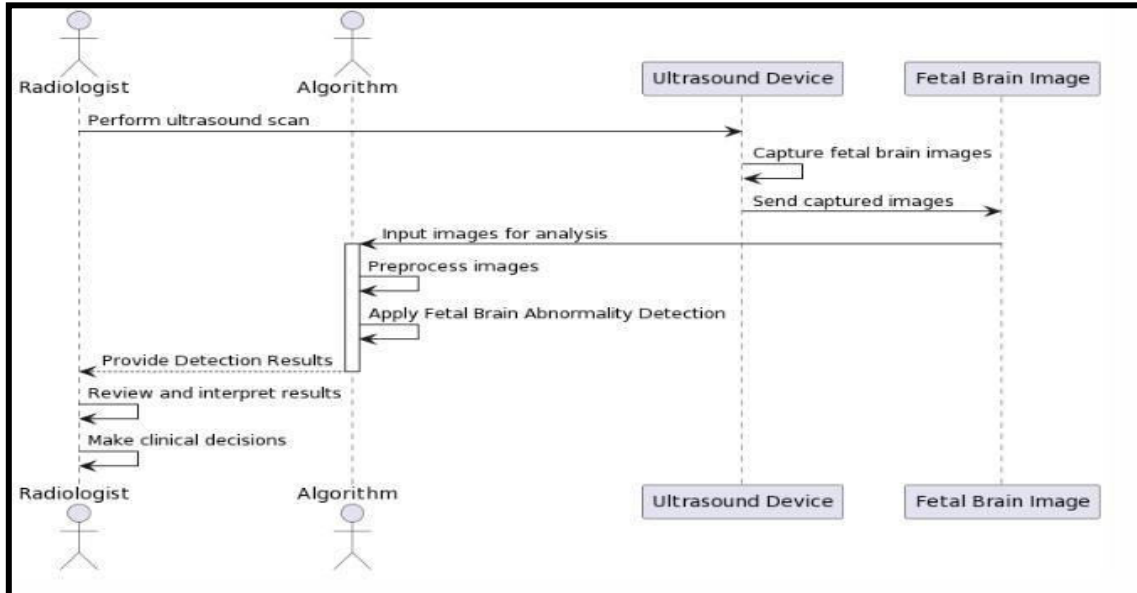


Figure3.3: Sequence Diagram

3.3 Data flow Diagram

A Data Flow Diagram illustrates the flow of data within a system, showcasing how information is input, processed, and output. It consists of processes, data stores, data flow, and external entities. Processes manipulate data, arrows represent the flow of data between elements, and data stores depict where data is stored. DFDs are valuable for understanding data dependencies and transformations, making them useful in system analysis and design to ensure efficient information flow and processing.

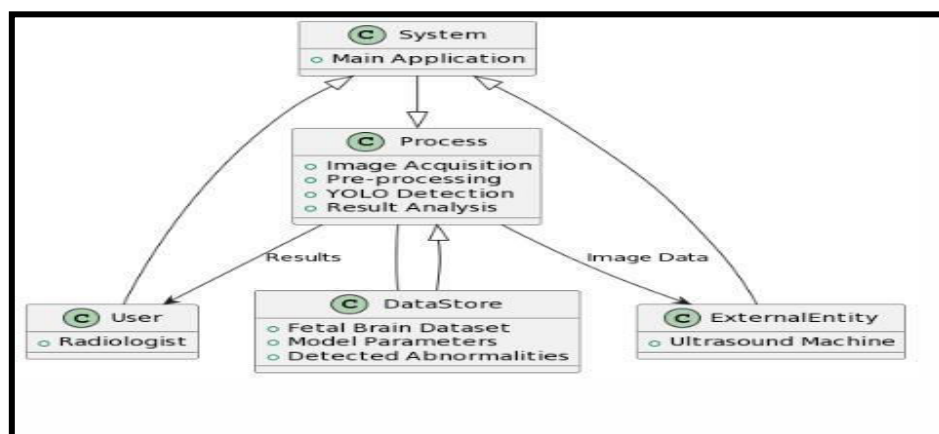


Figure3.4: Data flow Diagram

3.4 Class Diagram

A class diagram is a static representation of the structure and relationships within a system, focusing on the classes, attributes and methods that form the basis of object-oriented design. Classes, depicted as rectangles, represent entities in the system, and associations between them indicate relationships. Attributes and methods of each class are detailed within the rectangles. Class diagrams are essential for visualizing the overall architecture of a system, promoting a clear understanding of its key components and their interactions.

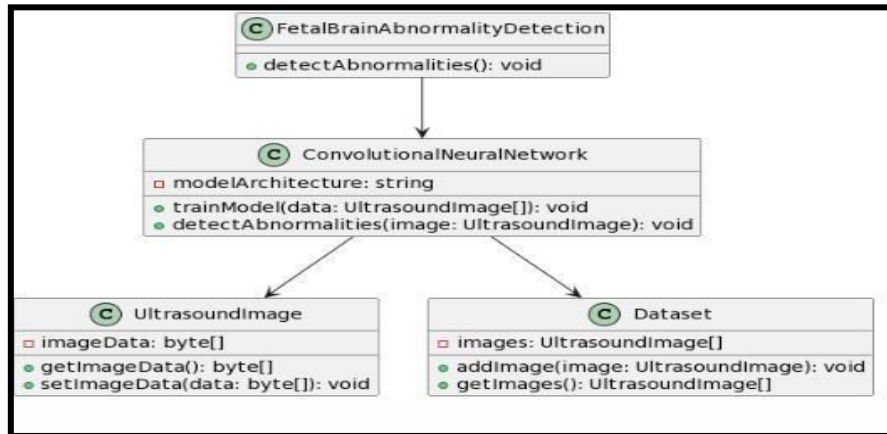


Figure3.5: Class Diagram

Summary

In software engineering, use case diagrams offer a high-level view of system functionalities by illustrating interactions between actors and use cases, while activity diagrams model dynamic processes and sequences. Class diagrams provide a static representation of a system's structure, emphasizing classes, attributes, and methods for an object-oriented design. Data flow diagrams (DFD) visualize data flow within a system, aiding in the analysis of data dependencies and transformations. Sequence diagrams capture dynamic interactions between system components over time, providing insights into system behavior during specific scenarios. Each diagram plays a unique role in understanding and designing different aspects of software systems, contributing to a comprehensive view of a system's architecture, functionality, and data flow

IV. CONCLUSION

The application of deep learning in identifying anomalies in the embryonic brain represents a significant advancement in the field of prenatal medicine. Unlike traditional techniques such as ultrasound imaging, which can be reliant on the subjective interpretation of medical professionals and may miss subtle abnormalities, deep learning systems offer exceptional precision and consistency in analyzing extensive imaging data. This capability has the potential to identify anomalies that might otherwise remain undetected. The incorporation of deep learning models and advancements in research into routine prenatal screenings has the potential to revolutionize prenatal care by providing new opportunities for proactive management of fetal health. This technology has the capacity to enhance diagnosis accuracy and enable prompt interventions, ultimately improving outcomes for the fetus and expectant parents. However, it is crucial to conduct further research to validate the effectiveness, dependability, and security of deep learning-based clinical environments before integrating them into conventional prenatal care procedures. Thorough validation and assessment are necessary to ensure the dependability, safety, and effectiveness of deep learning models in real-world scenarios. This includes evaluating their diagnosis accuracy compared to accepted practices and assessing their impact on patient outcomes through clinical research. Moreover, it is essential to consider concerns about data privacy, the ethical application of AI, and regulatory approval to ensure the responsible application of deep learning technology in prenatal care. In conclusion, the use of deep learning to identify anomalies in the embryonic brain holds great promise for improving prenatal medicine. With further research and validation, deep learning technology has the potential to significantly enhance the early intervention and care provided to expectant parents and their unborn children.

REFERENCES

1. "Use of MRI in the diagnosis of fetal brain abnormalities in utero(MERIDIAN):a multicentre, prospective cohort study. - PubMed - NCBI." [Online]. Available: <https://www.ncbi.nlm.nih.gov/pubmed/27988140>. [Accessed: 25-Aug-2018].
2. M.Havaei et al., "Brain tumor segmentation with Deep Neural Networks," Med.Image Anal., vol. 35, pp. 18–31, Jan. 2017.
3. A. Alansary et al., "Automatic Brain Localization in Fetal MRI Using Super pixel Graphs," in Machine Learning

Meets Medical Imaging, 2015, pp. 13–22.

4. K.E. alet, “Fetal brain anomalies detection during the first trimester:expanding the scope of antenatal sonography. PubMed - NCBI.” [Online]. Available: <https://www.ncbi.nlm.nih.gov/pubmed/28282781>. [Accessed: 25-Aug-2018].

5. B. J. Erickson, P. Korfiatis, Z. Akkus, and T. L. Kline, “Machine Learning for Medical Imaging,” Radiographics, vol.37,no.2,pp.505–515,Mar. 2017.

6. M. S. Hosseini and M. Zekri, “Review of Medical Image Classification using the Adaptive Neuro-Fuzzy Inference System,” J. Med. Signals Sens., vol. 2, no. 1, pp. 49–60, 2012.

7. A. Makropoulos, S.J. Counsell, and D.Rueckert, “A review on automatic fetal and neonatal brain MRI segmentation,” NeuroImage, Jun. 2017.

8. J.Levman and E.Takahashi, “Multi variate analyses applied to fetal, neonatal and pediatric MRI of neuro developmental disorders,” Neuro Image Clin., vol. 9, pp. 532–544, Jan. 2015



INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 9940 572 462  6381 907 438  ijirset@gmail.com



www.ijirset.com

Scan to save the contact details