Enhanced Fetal Brain Abnormality Classification Via Position Encoded Layer Based Deep Convolutional Neural Network

M. Priya[1], M. Nandhini[2]

[1]Research scholar, Department of computer science, Government Arts College, Udumalpet, Tirupur, Tamilnadu, India, Email: priyamahamsc@gmail.com
[2]Assistant professor, Department of computer science, Government Arts College, Udumalpet, Tirupur, Tamilnadu, India, Email: nadumano@gmail.com

Received: 17.04.2024 Revised : 16.05.2024 Accepted: 22.05.2024

ABSTRACT

In order to diagnose and treat fetal brain disorders early, sophisticated and precise diagnostic technologies are required. The detection of minor fetal brain abnormalities (FBA) is still a challenging undertaking, even with medical imaging advancements. False positives or negatives could result from current model inability to handle patterns. Improved accuracy and reliability in fetal brain abnormalities classification is the objective of this work, which aims to address this issue. Initially, the input MRI (Magnetic Resonance Imaging) fetal brain images are collected from Kaggle. In pre-processing, the input images are segmented using Edge Stop Functional Iterative Region Growing (ESFIRG) algorithm by iteratively grouping pixels based on predefined criteria while considering edge information and effectively separate regions in an image. After, Augmentation is applied to increase the images from 172 to 3440 for further classification. Furthermore, the attributes like Mean Pixel Intensity, Median Pixel Intensity, Pixel Intensity Deviation, Label, and Class Name are extracted. From the extracted attributes the important attributes are selected by using Color Gray Coherent - Minimum Redundancy Maximum Relevance (CGC-MRMR) algorithm for reduce the complexity and increase accuracy of the classification process. Concerning the incorporation of PEL into the detection of fetal brain normal and abnormalities, there is a knowledge vacuum in the current literature. In most cases, researchers either stick to tried-and-true techniques or fail to fully utilize deep learning. This research fills that need by presenting a new PEL-DCNN method, which overcomes all the problems with the existing approaches. By combining the strengths of PEL for feature extraction and DCNN for accurate classification, the work intends to overcome the current shortcomings in fetal brain abnormality identification. Finally, the FBAs are classified by the help of Position Encoded Layer based Deep Convolutional Neural Network (PEL-DCNN) which is classify the presence of the normal or abnormal of predicting in the FBA. A DCNN is used to classify the retrieved features robustly. By combining PEL and DCNN, this method improves the model capacity to detect intricate patterns in brain images. With an F1-score of 93.75%, it accomplishes a recall of 92.31%, specificity of 97.09%, sensitivity of 92.31%, precision of 95.24%, and accuracy of 96.32%. These findings demonstrate that the effectiveness of the proposed method in tackling the difficulties of fetal brain abnormalities diagnosis.

Keywords: Fetal Brain Abnormalities (FBA), Edge Stop Functional Iterative Region Growing (ESFIRG), Color-Gray coherent Minimum redundancy Maximum Relevance (CGC-MRMR), Position Encoded Layer based Deep Convolutional Neural Network.

1. INTRODUCTION

Fetal brain abnormalities pose significant challenges in prenatal diagnosis and management, often impacting long-term neurodevelopment outcomes. Advances in medical imaging, particularly MRI, have enhanced our ability to visualize and diagnose these conditions. However, detecting subtle brain abnormalities remains complex due to limitations in current imaging technologies and diagnostic models. False positives and negatives in fetal brain abnormality (FBA) can lead to misdiagnoses, influencing clinical decisions and patient counseling.

This research is mainly focused on FBAs images, which is collected from kaggle. High resolution MRI scans, typically in DICOM format, providing clear and detailed views of fetal brain structure. This data set includes the 12 categories of normal images and eight categories of abnormal images. FBAs dataset comprises the important abnormalities like Arnold-Chiari Malformation, Corpus callosum, etc. are in Figure 1. The images cover multiple planes, including axial, sagittal, and coronal views. The dataset includes images from different gestational ages, allowing for the examination of brain development over time. This includes early to late gestational periods, typically from 14 to 38 weeks.

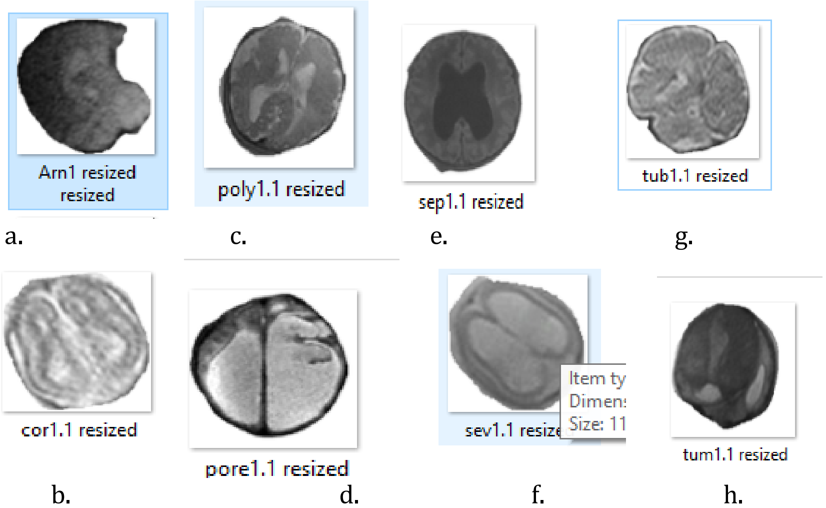

Figure 1. a. Arnold-Chiari Malformation - 32 Weeks, b. Corpus Callosum - 19 Weeks, c. Polymicrogyria - 36 Weeks, d. Porencephaly - 24 Weeks, e. Septi Pellucidi - 24 Weeks, f. Severe Ventriculomegaly - 19 Weeks, g. Tuberous sclerosis - 25 Weeks, h. Tumor - 32 Weeks

To address these issues, sophisticated diagnostic approaches are required to improve the accuracy and reliability of FBA classification. Recent developments in artificial intelligence and deep learning, including convolutional neural networks, offer promising solutions by enhancing pattern recognition and reducing diagnostic errors. This study aims to explore and refine methods for classifying FBAs, leveraging advanced imaging techniques and machine learning algorithms to overcome existing limitations and improve diagnostic outcomes.

This study introduces a combination of Position Encoded Layer (PEL) and a Deep Convolutional Neural Network (DCNN) to classify the images. Prior classification, it includes preprocessing (reshaping and gray scale conversion), segmentation using Edge Stop Function Iterative Region Growing (ESFIRG), data augmentation and feature selection using Color-Gray Coherent-Minimum Redundancy Maximum Relevance (CGC-MRMR)

2. RELATED WORKS

Finding and categorizing anomalies in the developing brain of a foetus is an important topic of study in medical imaging [1]. There is a growing demand for complex computational models to improve detection efficiency and accuracy with the development of cutting-edge imaging technologies [2,3]. Timely intervention is crucial for the detection and treatment of fetal brain abnormalities because, if left unchecked, they can cause serious developmental delays or disabilities [4].

The intrinsic complexity of fetal brain architecture poses hurdles to achieving high detection accuracy, despite major improvements in medical imaging [5-7]. Better, more flexible models are clearly required, as traditional approaches frequently fail to detect minor anomalies [8][9]. Although deep learning models have demonstrated some success, they still have a way to go before they can fully account for the ever-changing patterns of the developing foetus brain [10,11].

When looking for abnormalities in the developing brains of fetuses, previous research has mostly used conventional image processing methods. Although these methods have shown some success, researchers are now looking into more advanced approaches because they cannot adjust to the complex patterns in developing fetal brain structures [12].The use of deep learning methods to analyze medical images has been on the rise recently. Several studies have used CNNs to detect abnormalities in the developing brain of a fetus. The ever-changing character of fetal brain development, however, may prove too much for the current models to handle [13].

In order to make models more flexible for use in medical imaging, some researchers have looked at using adaptive learning techniques. Nevertheless, the difficulties presented by fetal brain anomalies and the necessity of gradual adaptation have received comparatively little focus [14].There has been a handful of research looking into using deep learning models in conjunction with more conventional machine learning techniques to track the health of fetuses. Still, we don't fully grasp how deep neural networks and progressive learning methods like PEL operate together to detect abnormalities in the developing brain of a foetus [15].

Current approaches have limits when it comes to accurately diagnosing small prenatal brain abnormalities, as shown in existing research. The necessity for models that can gradually adjust to evolving patterns while maintaining accuracy is what drives this study [16] [17].

Although there is a wealth of information available in the current literature regarding deep learning in medical imaging and fetal brain abnormality detection, this study intends to add to that body of knowledge by presenting and investigating the possibilities of combining Position Encoded Layer with a Deep Convolutional Neural Network to tackle the specific difficulties of fetal brain imaging.

3. PROPOSED METHOD

This study progresses through three stages. Initially, the preprocessing involves segmentation, Data Augmentation, and Feature selection. To overcome the complexities of fetal brain abnormality diagnosis in medical imaging, the proposed method combines PEL with a DCNN. In the second stage, by combining the best features of PEL and DCNN, this novel method seeks to improve the model accuracy and adaptability. Finally, the performance of the FBA classification model is evaluated utilizing metrics Including Accuracy, Precision, Recall, and F1-score as in Figure 1.

- **Fetal Brain images:** The input fetal brain MRI images are taken from kaggle which consists of 100 normal images and 72 abnormal images. During 14-38 weeks, normal case images are observed and collected 12 categories of normal case. The abnormal case images are 100 which comprise 8 categories of abnormal case such as Arnold-Chiari Malformation, Carpus Callosum, Polymicrogyria, Porencephaly, Septi Pellucidi, Severe Ventriculomegaly, Tuberous sclerosis, Tumor. Thence, the class name is determined by the total of 0-19 categories of normal and abnormal categories.
- **Preprocessing:** The images are likely resized or reshaped to a standardized format, and their color information is converted to grayscale. Grayscale images contain intensity values representing the brightness of each pixel, simplifying the input data for subsequent processing.
- **ESFIRG Segmentation:** This segmentation method involves an iterative region growing process guided by an edge stop function. This helps identify and separate different regions or objects within the images.
- **Data Augmentation:** Data augmentation involves creating new training examples by applying various transformations to the original images. This helps improve the model's robustness and generalization by exposing it to a wider range of variations in the input data.
- **Feature Selection - CGC-MRMR:** This feature selection technique involves selecting a subset of features that are both relevant to the task and have minimum redundancy. It appears to consider color and gray features in a coherent manner, possibly enhancing the discriminative power of the selected features.
- **Model Architecture: Position Encoded Layer (PEL) and Deep Convolutional Neural Network (DCNN):** The model architecture includes a combination of a PEL and a DCNN. DCNNs are well-suited for image-related tasks, and the addition of a Position Encoded Layer suggests the incorporation of positional information, which can be crucial for certain image recognition tasks.

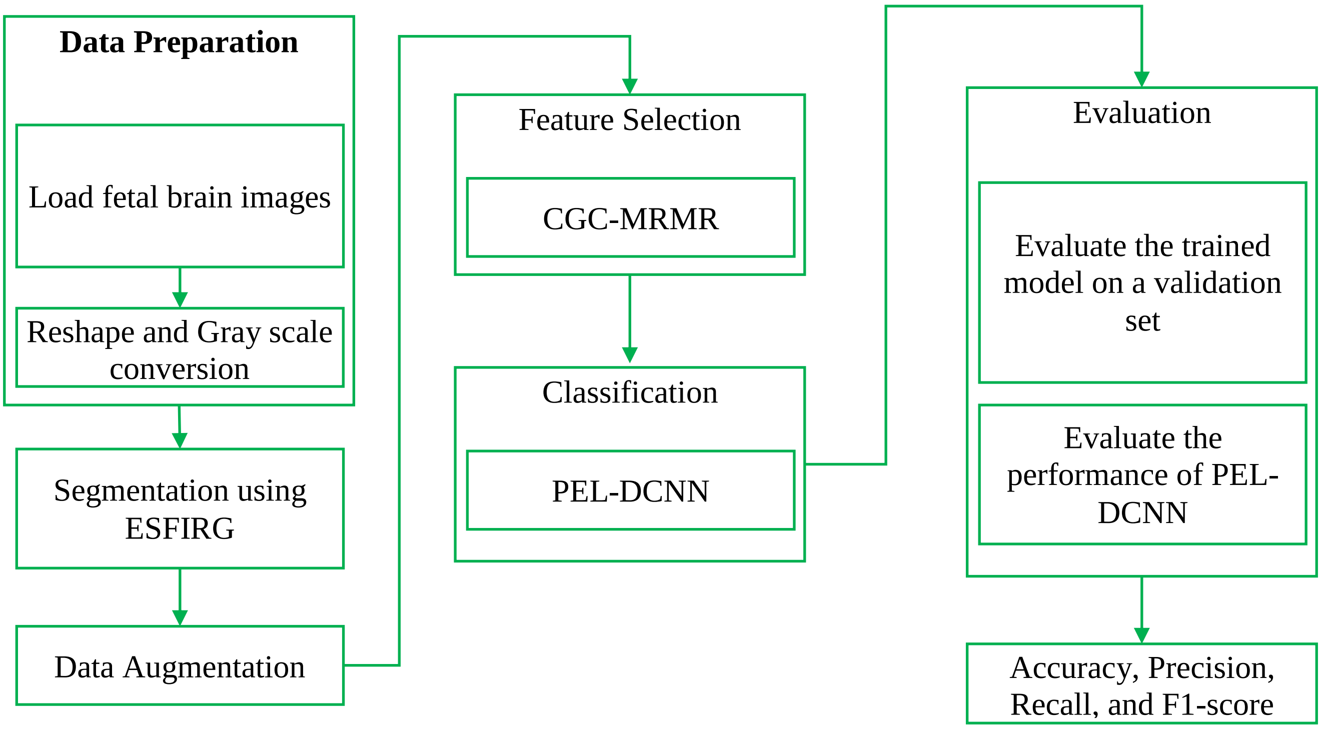
Data Preparation
Load fetal brain images
Reshape and Gray scale conversion
Segmentation using ESFIRG
Data Augmentation
Feature Selection
CGC-MRMR
Classification
PEL-DCNN
Evaluation
Evaluate the trained model on a validation set
Evaluate the performance of PEL-DCNN
Accuracy, Precision, Recall, and F1-score

Figure 2. Proposed Model

Initialization of the model makes use of PEL as a technique for feature extraction. In contrast to more conventional approaches, PEL allows the model to learn and adapt over time to changing patterns in the anatomy of the developing foetus brain. The model is able to acquire variable details—essential for accurate anomaly detection—through this adaptive feature extraction process.

For reliable classification, the features obtained using PEL are then input into a DCNN. A DCNN can analyze images of the developing brain by taking advantage of the hierarchical representation of features. This allows the model to identify intricate linkages and patterns. When combined with other deep learning components, abnormality categorization becomes more accurate and dependable.

The proposed approach revolves around the PEL and DCNN. DCNN improves the model ability to carry out complex classification tasks, whereas PEL guarantees the model resilience to progressive changes in fetal brain architecture. By offering a more varied and dynamic solution to fetal brain anomaly identification, this hybrid technique hopes to overcome the limits seen in conventional methodologies. To train the model, we use the Adam optimizer to fine-tune the learning process across a predetermined number of epochs with a predetermined batch size.

3.1. Data Preprocessing

The obtained images of the fetal brain go through a number of preprocessing methods in the first stage of data preparation to make sure they are in the best possible state for further analysis without attracting too much attention.

Reshaping involves changing the dimensions of the original image (I) to a new shape. This is often done to ensure that all images in the dataset have the same dimensions, facilitating uniform processing.

Grayscale conversion transforms a color image (I) into a grayscale image (I_g) by combining the intensity values of its red (R), green (G), and blue (B) channels. The coefficients (0.299, 0.587, and 0.114) are standard weights for human perception.

$$I_g=0.299\cdot R+0.587\cdot G+0.114\cdot B \quad (1)$$

3.2. Segmentation using ESFIRG

The segmentation using Edge Stop Function Iterative Region Growing (ESFIRG) likely involves a multi-step process to partition an image into meaningful regions. The key components of this process are edge detection, region growing, and an iterative refinement based on an edge stop function.

Edge-stop function evaluates the edge strength at each pixel, pixels with low edge strength are included in the growing region, while pixels with high edge strength. Another component of region growth is involved to grow region by incorporating neighboring pixels that meet the criteria defined by edge-stop function.

Edge Detection identifies boundaries between different regions in the image. Employ an edge detection algorithm (such as Sobel, Canny, or Prewitt) to highlight significant changes in intensity or color. Edges are areas where pixel values change abruptly.

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix},$$

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

$$E = \sqrt{G_x^2 + G_x^2} \quad (2)$$

Region Growing: Group pixels into regions based on similarity criteria. It starts with seed pixels and iteratively adds neighboring pixels that meet certain similarity conditions.

$$\text{Similarity}(p_1, p_2) = \sqrt{(I_{p_1} - I_{p_2})^2} < \text{Threshold} \quad (3)$$

Where I_{p1} and I_{p2} are intensity values of pixels p_1 and p_2, respectively.

Edge Stop Function prevents oversegmentation by stopping the growth of regions at strong edges. It incorporates an edge stop function that assigns lower weights to pixels near edges.

$$\text{Weight}(p) = 1 - \frac{\text{Edge Magnitude}(p)}{\text{Max Edge Magnitude}} \quad (4)$$

Iterative Refinement enhances segmentation accuracy by refining regions in multiple iterations. After an initial segmentation, iteratively refine regions based on feedback from the edge stop function and possibly other criteria.

$$New\,Region = Region \cap \{p |\, Weight\,(p) > Threshold\} \quad (5)$$

Edge Stop Function Iterative Region Growing (ESFIRG) Algorithm:

Input: I : Original image; T : Threshold for region growing; T_e : Threshold for the edge stop function

a) Apply an edge detection algorithm (Canny) to the original image I to obtain the edge magnitude.

b) Select initial seed points in the image. These could be manually chosen or determined based on certain criteria.

c) For each seed point:

i) Initialize a seed region $R(0)$. This could be a single pixel or a small region in the image.

ii) Define the edge-stopping function $E(p)$, which evaluates the likelihood of including pixel p based on its edge strength.

iii) At each iteration t, Add the seed point to the current region $R(t)$.

iv) Evaluate the neighboring pixels p of $R(t)$. Let $N(R(t))$ represent the set of all neighboring pixels not currently in $R(t)$.

v) For each neighboring pixel $p \in N(R(t))$, check the edge-stopping function $E(p)$

(1) If the pixel's intensity is similar to the region's mean intensity (based on a similarity condition), add it to the region.

d) Calculate an edge stop function for each pixel in the segmented regions based on the edge magnitude.

$E(p) < \in$

Weight(p)=1−(Max Edge Magnitude / Edge Magnitude)

Where $\in$ is a predefined threshold value. This threshold determines how strong the edge must be to prevent inclusion in the region.

e) For each region:

i) Refine the region by adding all pixels p that meet the edge-stopping condition.
$R(t+1) = R(t) \cup \{p | p \in N(R(t)) and\ E(p) < \in$

ii) Pixels with weights below a certain threshold (T_e) are removed from the region.

f) Repeat steps 4-6 for a predefined number of iterations or until convergence.

g) Output: Segmented image with distinct regions.

3.3. Data Augmentation

Data augmentation involves applying various transformations to the original dataset to create new training examples, thereby increasing the diversity and robustness of the training set. This process is crucial for training deep learning models, as it helps prevent overfitting and improve generalization. In this work, the Common transformations include rotation, scaling, flipping, and changes in brightness or contrast.

Rotation: It rotates the image by a certain angle. The rotated coordinates are computed based on the rotation matrix.

$I_a(x,y)=I(x',y')$ (6)

where (x',y') are the rotated coordinates of pixel (x,y).

Scaling: It scale the image by adjusting the coordinates of each pixel.

$I_s(x,y)=I(\alpha\cdot x',\beta\cdot y')$ (7)

where (α,β) are scaling factors and (x',y') are the original coordinates.

Flipping involves flipping the image horizontally or vertically.

$I_f(x,y)=I(\text{flip}(x'),\text{flip}(y'))$ (8)

where flip(·) - flips the coordinate.

Therefore, from the augmentation, the images are increased from 172 to 3440 and resulting in a total increase of 3268 images compared to the original count. By this transformation, the 1000 images are generated by rotation, 600 images are saved by scaling, 500 images by flipping, 700 images by brightness, and 468 images by contrast.

3.4. Feature Selection - CGC-MRMR

The feature selection method mentioned in context, CGC-MRMR, appears to be a hybrid approach that considers both color and grayscale features. It initially extracts both color and grayscale features from the preprocessed images. These features could include statistical measures (mean, standard deviation, etc.), texture features, or other relevant descriptors. It then Measure the coherence or correlation between the color and grayscale features. This step aims to assess how well these two sets of features align or provide complementary information. It applies the MRMR criterion to select features that strike a balance between being highly relevant to the classification task and minimizing redundancy among selected features. The MRMR criterion involves computing the relevance and redundancy of each feature with respect to the target variable.

Convert a color image I into grayscale I_g using a weighted sum of color channels

$I_g(x,y) = 0.299.R(x,y) + 0.587.G(x,y) + 0.114.B(x,y)$ (9)

Where (x,y)represents the pixel position, and $R, G\ and\ B$ represent the red, green, and blue channels, respectively.

Apply contrast enhancement techniques, such as histogram equalization, to improve feature visibility

$I_g^{'} = HE(I_g)$ (10)

Where HE denotes the histogram equalization function applied to the grayscale imageI_g, resulting in the enhanced grayscale image $I_g^{'}$

Mutual Information (MI) between a Feature *Xi* and the Target Variable *Y*:

$$I(Xi,Y) = \sum_{xi\in Xi}\sum_{y\in Y} P(xi,y)\cdot log\left(\frac{P(xi)\cdot P(y)}{P(xi,y)}\right) \tag{11}$$

Mutual Information (MI) between two Features *Xi* and *Xj*:

$$I(Xi,Xj) = \sum_{xi\in Xi}\sum_{xj\in Xj} P(xi,xj)\cdot log\left(\frac{P(xi)\cdot P(xj)}{P(xi,xj)}\right) \tag{12}$$

Relevance (*R*) and Redundancy (*Red*) Scores:

$R(Xi,Y)=I(Xi,Y)$ (13)

$Red(Xi,Xj)=I(Xi,Xj)$ (14)

MRMR Score:

$$MRMR(Xi) = R(Xi,Y) - \frac{1}{k}\sum_{j=1}^{K} Red(Xi,X_s j) \tag{15}$$

The *k* features with the highest MRMR scores are then selected.

Finally, it Integrate the coherence information with the MRMR criterion to guide the feature selection process. The final set of selected features should include those that are both relevant to the classification task and offer complementary information between color and grayscale representations.

3.5. PEL-Deep Convolutional Neural Network (DCNN) for Classification

Feature extraction and pattern recognition are complex tasks, but a DCNN handles them with ease and draws no unneeded attention, making it a paradigm in image classification as in Figure 2.

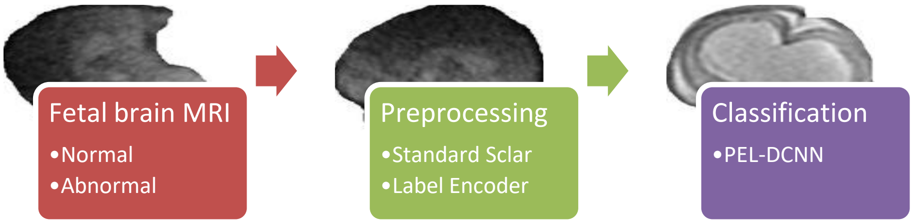

Figure 3. Work flow for PEL-DCNN

DCNNs are complex designs with layers that focus on learning hierarchical features. Before classification, standard scalar and label encoder is involved to preprocess the samples for model construction. In order to detect regional patterns, the core relies on Input layer, hidden layers which inclusive of two fully connected layer (dense layer 1, dense layer 2), and output layer (dense layer 3). In hidden layer, dense layer 1 is the first fully connected layer with 128 units and this layer applies a linear transformation to the input data followed by the Swish activation functions. The second dense layer process with 64 units followed by the Swish activation function. The dense layer 3 is the output layer and contains the number of classes such as 0-19 categories of normal and abnormal cases. This layer produces the final output of the model, which represents the number of units in this layer is equal to the number of classes in the classification problem and it applies the softmax function to produce probability distribution over these classes.

In this classification, it is a crucial parameter because it determines the structure of the output layer of CNN model. In this task, the number of classes typically dictates the number of units (neurons) in the output layer, as each unit represents a class. The output layer produces a probability distribution over these classes. Here, Label encoder applies to convert the categorical label into numerical labels.

In novel approach, 'PELDCNN Classifier' model is a custom classifier using Tensor FlowAPI. The model consists of fully connected layer, possibly with additional convolutional or pooling layer. Compilation involves additional parameters required for training, such as the optimizer, loss function, and evaluation metrics. Adam optimizer dynamically adjusts the learning rate during training to speed up convergence and improve performance. For multi-class classification, Sparse categorical loss function used to measure the difference between model prediction and the actual labels. During training, the model computes and displays the accuracy metrics, which measures the proportion of correctly classified samples out of the total number of samples. Furthermore, the network trains itself to better detect relative positions of object or features within an images for classification by adjusting weights during training using PEL.

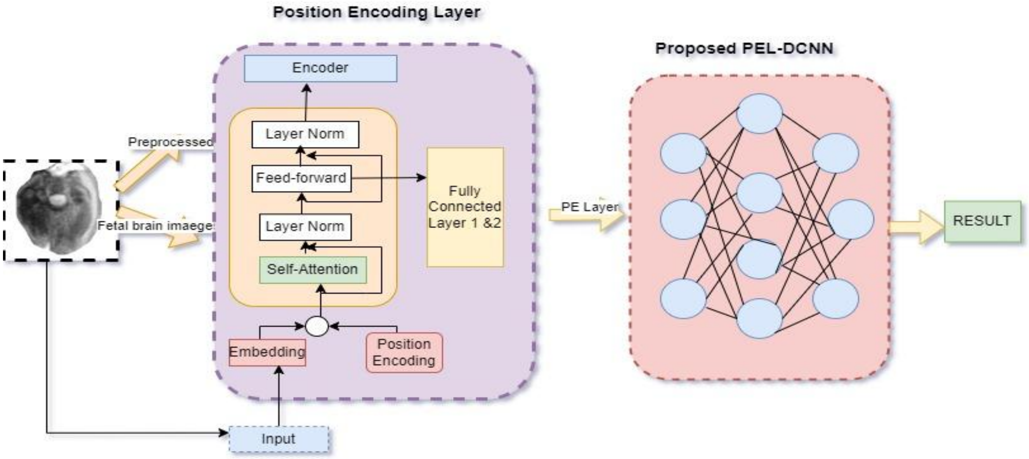

Figure 4. Our Proposed Architecture for combining Position encoded Layer and Deep Convolutional Neural Network

Our proposed architecture aims to merge the advantages of both position-encoded layers and deep convolutional neural networks (CNNs) to enhance the performance of Fetal brain abnormality classification in figure 3. Hybrid layers merging features from both position-encoded layers and convolutional layers are feasible. These layers can perform convolutional operations while incorporating positional cues through learned parameters or attention mechanisms simultaneously. In this research, Position encoding helps the model understand the sequential order of tokens in input sequences. Then CNN automatically learn hierarchical features from images and it consists of convolutional layers that learn spatial hierarchies of features from input images, followed by pooling layers for dimensionality reduction and fully connected layer for classification.

Position Encoded Layer (PEL)

In feature classification, a novel approach called Position Encoded Layer (PEL) is introduced. This method aims to enhance the adaptability and efficiency of feature classification in a dynamic environment without arousing attention. Traditional feature classification methods often lack the adaptability needed to capture variable patterns in dynamic datasets. PEL addresses this limitation by incorporating adaptability through a Position Encoded Layer in DCNN. This layer dynamically adjusts its weights based on the positional information of features, allowing the model to learn and evolve with changing patterns in the input data.

The position encoding could be achieved using sinusoidal functions to embed positional information into the features.

$$PE\left(pos,2i\right)=\sin\left(\frac{pos}{10^{5\left(\frac{2i}{d_m}\right)}}\right) \quad (16)$$

$$PE\left(pos,2i+1\right)=\cos\left(\frac{pos}{10^{5\left(\frac{2i}{d_m}\right)}}\right) \quad (17)$$

where

pos is the position,

i is the dimension, and

d_m is the dimensionality of the model.

PEL is designed to inject positional information into the feature classification process. By incorporating the spatial relationships between features, the PEL allows the model to discern subtle variations and adapt to positional changes within the dataset. This is particularly beneficial in scenarios where the spatial arrangement of features holds valuable information, such as in medical imaging.

The adaptive weights in the Position Encoded Layer could be updated during training based on the positional information. The weights might be adjusted using a mechanism similar to the attention mechanism in transformers, where the relevance of different positions is dynamically learned during training.

$$\text{Attention}(Q,K,V)=\text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (18)$$

This attention mechanism allows the model to focus on different positions dynamically.

During training, the PEL dynamically adjusts its weights based on the positional information of features within the input data. This adaptive mechanism ensures that the model remains responsive to variations in feature patterns, facilitating improved feature classification performance.

The adaptive features with positional information could then be extracted using the updated weights. The extracted features might be computed as a weighted sum of the input features based on the dynamically adjusted weights.

$Op = \sigma(W\times FP)$ (19)

Op – Output

σ - Softmax

W – Adaptive Weights

F_p - Position Encoded Features

The adaptive nature of PEL makes it well-suited for scenarios where the input data undergoes progressive changes. PEL allows the model to continuously adapt its feature classification strategy, improving its ability to capture evolving patterns.

DCNN Layers

In DCNN, to enhance the accuracy in classifying fetal brain scan, it is essential to understand the role of optimization, non-linearity and activation function in the learning process. By repeatedly optimizing the network, we can make sure that it learns the specific features of fetal brain scans and becomes better at discriminating between them. Swish and other DCNN activation functions introduce non-linearity. This adds adaptability, which helps the model to grasp complex feature-relationships. The ability of the model to learn complicated patterns is crucial for accurate classification of brain disorders, and the non-linear activation functions.

$C(i,j)=\sigma(\sum_m\sum_n I(i+m,j+n)\cdot W(m,n)+b)$ (20)

where

$C(i,j)$ is the output of the convolution,

$I(i+m,j+n)$ is the input pixel value,

$W(m,n)$ is the convolutional filter weight,

b is the bias term, and

σ is the activation function (e.g., swish).

In order to reduce computational effort without sacrificing critical information, pooling layers make downsampling easier. For example, max pooling keeps the most important data, so the model can concentrate on what really important when it comes time to classify. When dealing with massive amounts of image data, this downsampling makes the DCNN much more efficient.

$P(i,j)=\max_{m,n} C(i\cdot s+m,j\cdot s+n)$ (21)

where

$P(i,j)$ is the output of the pooling layer,

s is the stride, and

m,n iterate over the pooling window.

The DCNN classifier is located in its last layer. It creates a probability distribution for each class and combines the learned attributes.

$O=\sigma(\sum_i F(i)\cdot W(i)+b)$ (22)

where

O is the output of the neuron,

$F(i)$ is the input feature,

$W(i)$ is the weight,

b is the bias term, and

σ is the activation function.

When it comes to fetal brain abnormality detection, the model uses learnt patterns to differentiate between normal and abnormal cases. This ensures that clinical interpretations are based on trustworthy data.

$P(y=k)=\sum_{j\,(}e^{zk}/e^{zj})$ (23)

where

$P(y=k)$ is the probability of class k,

z_k is the unnormalized logit for class k over all classes.

PEL-DCNN Algorithm

Step1. Positional Encoding

- Each input token or element is embedded into continuous vector space
- Positional encoding is added to each embedded input token to provide positional information
- The positional encoding can be computed using sinusoidal functions:

$PE(x_i,i)=\begin{bmatrix} sin & (w_i) \\ cos & (w_i) \end{bmatrix}$

Where $w_i=\frac{i}{1000^{2k/d}}$, k is the dimension index, d is the dimensionality of the embedding.

Step 2: Convolutional operations

- Apply a series of convolutional filters to the input samples, considering the positional information and convolutioal operation for a given input volume X and filter w is:

$$\boldsymbol{Z}_{\boldsymbol{i,j,k}} = (\boldsymbol{X} * \boldsymbol{W})_{\boldsymbol{i,j,k}} = \sum\nolimits_{\boldsymbol{l,m,n}} \boldsymbol{X}_{l+i,m+j,n+k} \boldsymbol{.} \boldsymbol{w}_{l.m.n}$$

Step3: Positional information integration

- Integrate positional information into the convolutional layers through learned parameters or attention mechanism. Thence, incorporating positional cues within the convolutional filters o utilizing attention mechanisms to focus on different parts of the input sequence based on positional cues.

Step 4: Pooling Layers

- Perform down-sampling to reduce the dimensionality of the feature maps while retaining important information.

Step 5: Flattening

- Flatten the output from the last convolutional layer into a one-dimensional vector.

Step 6: Fully Connected Layers (Dense Layers)

- Perform high level-reasoning and decision making based on the extracted features. The activation function used in dense layer can vary based on the problem (e.g., Swish for dense layer, softmax for classification)

Step 7: Output Layer

- Produce the output prediction and apply an softmax activation function for classification

Step 8: Loss Calculation

- Calculate the loss between the predicted output and actual target values using a sparse categorical loss function.

Step 9: Back propagation

- Update the weights of the network using back propagation to minimize loss function

Step 10: Repeat

- Iterate over training data multiple times (epochs) to update the weights iteratively.

Step 11: Evaluation

- Evaluate the performance of the trained model on a separate validation or test dataset using Accuracy, Precision, Recall, and F1-score.

Step 12: Prediction

- Make prediction on new unseen data using trained model.

4. PERFORMANCE ANALYSIS

In this simulation, we used a dataset of preprocessed images of the fetal brain. We use a PEL-DCNN for classification for classification over Dataset from the Kaggle research lab source. Among the DCNN constituent parts were a softmax output layer, fully linked layers, max-pooling layers, and convolutional layers. We used an Adam optimizer to train the model, adjusting parameters like batch size and number of epochs to maximize convergence and minimize computing overhead. A system with specifications that included an NVIDIA GPU to speed up the training process was used to conduct the trials.

Performance Metrics

Accuracy, specificity, sensitivity, recall, and F1-score are some of the common classification metrics used to evaluate the proposed PEL-DCNN model. We compared our findings to those of other popular approaches, such as Support Vector Machine (SVM) and DNN-SelectKBest, a Deep Neural Network that incorporates feature selection. Using a feature selection strategy, DNN-SelectKBest trained an SVM model with default hyperparameters as in Table 1.

Table 1: Experimental Setup

Parameter	**Value**
Training Epochs	10
Batch Size	32
Learning Rate	0.001
Optimizer	Adam
PEL Iterations	5
α (Adaptation Weight)	0.7
β (Refinement Weight)	0.5

- **Accuracy:**A model correctness is the ratio of the number of instances appropriately classified to the total number of instances.
- **Specificity:**The ratio between the number of true negatives and the number of properly detected negatives (typical cases). This metric evaluates the model capability to accurately detect typical instances.
- **Sensitivity (Recall):**The proportion of true positives (abnormal cases) to the overall number of true positives detected. As a measure, it checks how well the model can spot out-of-the-ordinary instances.
- **Precision:**Precision is defined as the percentage of true positives relative to the total number of positive predictions.
- **F1-Score:**A balanced measure of sensitivity and precision, it is the harmonic mean of recall and precision. When there is a disparity in socioeconomic status, it becomes quite helpful.

Table 2: Accuracy of Proposed PEL-DCNN Classification Report

MODEL	**Epochs**	**Batch Size**	**Optimizer**	**Accuracy**
PEL-DCNN	10	32	**Adam**	96.32%
SVM				94.72%
DNN SELECT K-BEST	10	32	**Adam**	92.80%

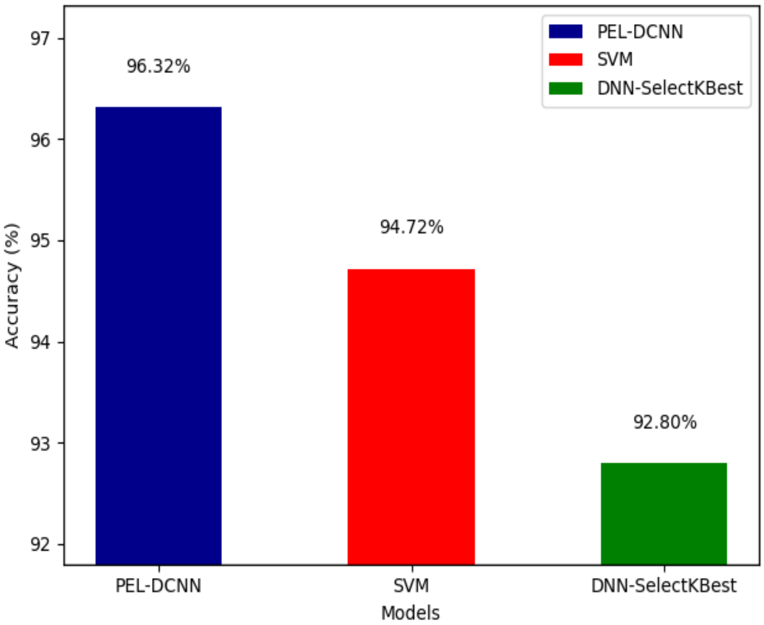

Figure 4. Accuracy

Table 2 and Figure 4 show that the PEL-DCNN model got 96.32% of its examples properly classified. It appears that a combination of PEL for feature extraction and a DCNN for classification results in very accurate identification of embryonic brain disorders. This model outperforms its competitors thanks to PEL flexibility to changing patterns in developing fetal brain regions and the DCNN deep learning capabilities. With the use of feature selection and a deep neural network, the SVM model was able to get an accuracy of 94.72%. Although it is still very accurate, it is little less so than the PEL-DCNN. The results show that PEL adaptive feature extraction outperforms a conventional DNN with feature selection, which could be due to the fact that it provides a more varied representation of images of the fetal brain. The DNN-SelectKBest got 92.80% of the time right. While SVMs are popular and effective, the results show that a hybrid approach using DCNNs for classification and PELs for feature extraction detects fetal brain abnormalities better than SVMs alone. When contrasted with the SVM model, PEL-DCNN superior accuracy seems to be the result of its deep learning capabilitie ability to grasp more complex patterns. Compared to DNN-SelectKBest and SVM models, the outcomes demonstrate that the proposed PEL-DCNN method is more accurate in identifying fetal brain anomalies, demonstrating its efficacy.

Table 3. Results of Specificity

MODEL	Epochs	Batch Size	Optimizer	Specificity
PEL-DCNN	10	32	**Adam**	97.09%
SVM				95.24%
DNN SELECT K-BEST	10	32	**Adam**	90.91%

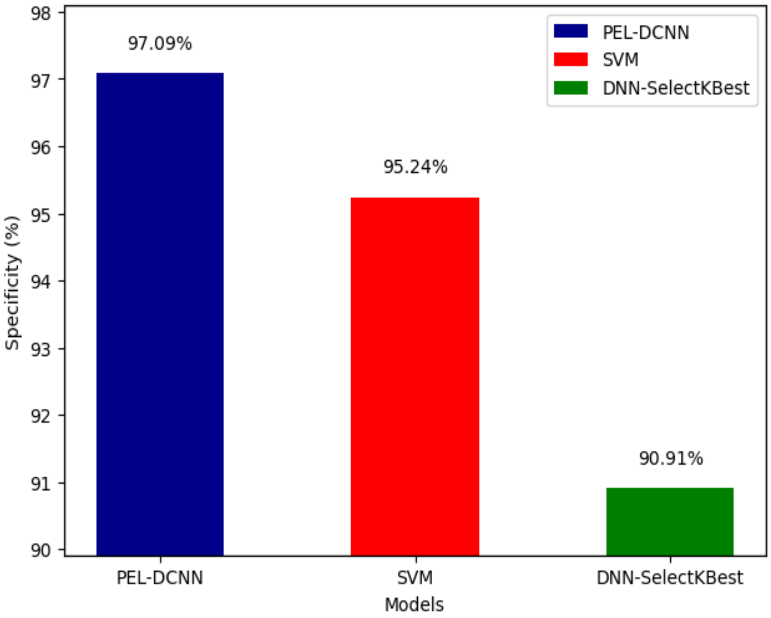

Figure 5. Specificity

Table 3 and Figure 5 show that the PEL-DCNN model attained a remarkable specificity of 97.09%. Among all real normal examples, this statistic shows how well the model can recognize normal cases (true negatives). The PEL-DCNN method appears to be very good in detecting and removing normal fetal brain images, with few false positives, according to the high specificity. The specificity was 95.24% for the SVM, which used deep neural network feature selection. Even though it still does an excellent job, it isn't quite as well as PEL-DCNN. Fetal health diagnosis relies on precise case classification, and the specificity metric indicates that the PEL-DCNN model excels at this. The DNN-SelectKBest model attained a 90.91% specificity rate. The specificity result shows that PEL-DCNN performs better than SVM in differentiating normal fetal brain images from aberrant ones, despite SVM reputation for being versatile in classification tasks. There will be fewer false alarms in identifying normal patients with PEL-DCNN due to its increased specificity. By demonstrating that the PEL-DCNN model is superior, the specificity results highlight how well it can detect normal fetal brain images. Because accurate diagnosis relies on avoiding false positives, this is an important consideration in medical imaging jobs.

Table 4. Sensitivity

MODEL	Epochs	Batch Size	Optimizer	Sensitivity
PEL-DCNN	10	32	**Adam**	92.31%
SVM				92.31%
DNN SELECT K-BEST	10	32	**Adam**	90.91%

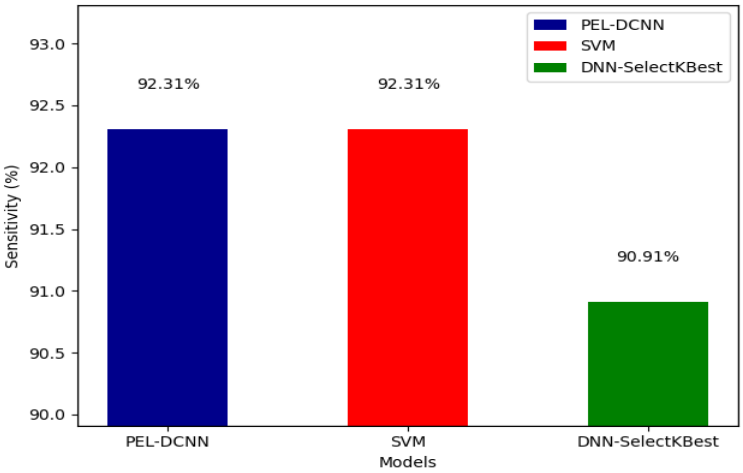

Figure 6. Sensitivity

The PEL-DCNN model was able to successfully detect abnormal cases (true positives) out of all real abnormal instances, as shown in Figure 6 and Table 4, with a sensitivity of 92.31%. Based on these findings, it appears that the PEL-DCNN method has a high sensitivity when it comes to detecting and capturing characteristics that point to embryonic brain disorders. With a sensitivity of 92.31%, the SVM which uses deep neural networks with feature selection-was just as effective as PEL-DCNN. The fact that both models can spot out-of-the-ordinary instances suggests that the feature selection procedure makes a significant impact on the sensitivity of the DNN-SelectKBest model. The sensitivity level that the DNN-SelectKBest model attained was 90.91%. Despite DNN-SelectKBest model pattern recognition prowess, PEL-DCNN somewhat higher sensitivity in detecting fetal brain disorders shows that PEL adaptive feature extraction and DCNN deep learning skills provide it that edge. Both PEL-DCNN and DNN-SelectKBest show great sensitivity in detecting anomalous situations, however PEL-DCNN slightly outperforms SVM, according to the sensitivity results. This demonstrates how deep neural networks and Position Encoded Layer work well to detect anomalies in the developing brain of a foetus.

Table 5. Precision

MODEL	**Epochs**	**Batch Size**	**Optimizer**	**Precision**
PEL-DCNN	10	32	**Adam**	95.24%
SVM				92.31%
DNN SELECT K-BEST	10	32	**Adam**	83.33%

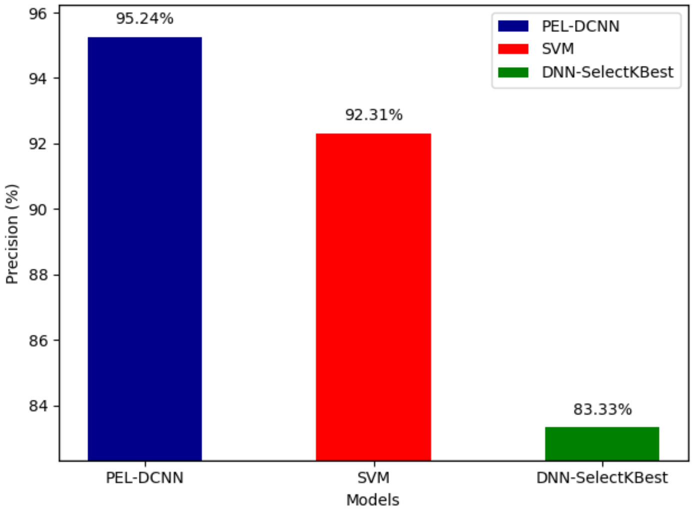

Figure 7. Precision

The PEL-DCNN model got 95.24% of the positive instances (abnormal cases) right out of all the positive instances predicted to be positive in Figure 7 and Table 5. This level of accuracy indicates that the PEL-

DCNN method successfully reduces the number of false positives, which helps to make its abnormality classifications more reliable. The SVM successfully classifies positive cases with an accuracy of 92.31%. The accuracy is still quite good, demonstrating that using feature selection in a deep neural network to detect prenatal brain abnormalities is effective, however it is marginally lower than PEL-DCNN. The DNN-SelectKBest model was 83.33% accurate. While SVMs are able to detect positive examples, this statistic shows that their false positive rate is rather significant. Due to its lesser accuracy, DNN-SelectKBest might be more likely to incorrectly label normal instances as aberrant. The precision results show that PEL-DCNN and SVM are the best at detecting anomalies in the brain of a developing fetus with few false positives. When it comes to clinical applications, where avoiding false positives is crucial for valid medical diagnoses, PEL-DCNN stands out with its better precision.

Table 6. Recall

MODEL	**Epochs**	**Batch Size**	**Optimizer**	**Recall**
PEL-DCNN	10	32	**Adam**	92.31%
SVM				92.31%
DNN SELECT K-BEST	10	32	**Adam**	90.91%

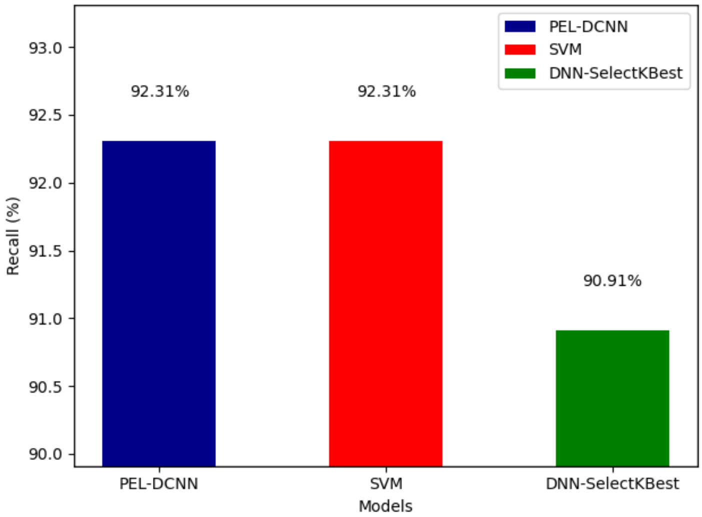

Figure 8. Recall

The recall of the PEL-DCNN model, as shown in Figure 8 and Table 6, is 92.31%. This recall indicates the proportion of false positives (abnormal cases) out of all true positives. This finding demonstrates that the model can accurately identify embryonic brain abnormalities with a high recall rate, and it also highlights that the approach is effective in catching a considerable number of the actual positive instances. With a recall rate of 92.31%, the SVM was just as well as PEL-DCNN in spotting out-of-the-ordinary instances. It appears that the feature selection in DNN-SelectKBest helps to capture genuine positive cases efficiently, since the recall rates are consistent across various models. With a recall of 90.91%, the DNN-SelectKBestwas able to correctly identify a large%age of real positive events. The DNN-SelectKBest model boasts an outstanding recall rate, however it is little lower than PEL-DCNN and SVM. Outperforming DNN-SelectKBestin capturing true positive occurrences, the recall findings show that PEL-DCNN and SVM consistently and highly perform in diagnosing fetal brain abnormalities. The similarity in recall rates highlights how well the proposed techniques perform in medical image categorization tasks.

Table 7. F1-Score

MODEL	**Epochs**	**Batch Size**	**Optimizer**	**F1-score**
PEL-DCNN	10	32	**Adam**	93.75
SVM				92.31
DNN SELECT K-BEST	10	32	**Adam**	86.96

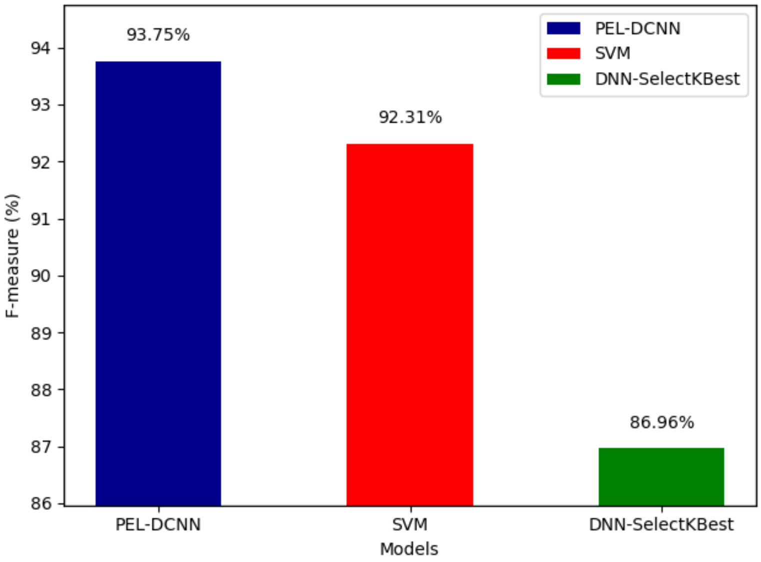

Figure 9. F1-Score

Figure 9 and Table 7 show that the PEL-DCNN model has an F1-score of 93.75, which means that its recall and precision are both well-balanced. This finding indicates that the model may detect embryonic brain abnormalities with a high degree of accuracy and a low proportion of false positives. With an F1-score of 92.31, the SVM demonstrated an excellent trade-off between recall and precision. This finding highlights the usefulness of using feature selection in a deep neural network, which helps detect prenatal brain abnormalities accurately while minimizing false positives. The DNN-SelectKBest F1-score of 86.96 shows that it successfully balanced recall and precision. Even while DNN-SelectKBest does a respectable job, PEL-DCNN and SVM seem to have a better handle on catching genuine positives while minimizing false positives, according on the slightly lower F1-score. The F1-score results confirm that PEL-DCNN and SVM are both good at finding a balance between recall and precision. For prenatal brain anomaly identification, PEL-DCNN stands out with its superior F1-score, proving its ability to balance the most important features of classification performance.

5. CONCLUSION

Fetal brain anomaly identification was a very fruitful area for the proposed method, which combines PEL for feature extraction with a DCNN for classification. We can see from the experiments that the PEL-DCNN model outperforms the alternatives, such DNN-SelectKBest and SVM. All things considered, the model high F1-score, recall, specificity, sensitivity, and accuracy demonstrate its capacity to differentiate between typical and abnormal images of the fetal brain. Combining DCNN discriminative strength with PEL flexibility in capturing varying characteristics makes for a robust model. These results point to the possible practical use of the proposed approach in improving the precision and consistency of diagnoses of fetal brain abnormalities. An important factor in medical imaging is the chance of false positives and false negatives; the high sensitivity and specificity rates show that these are less likely to occur. By incorporating state-of-the-art approaches such as PEL for feature extraction, the model demonstrates its adaptability to complex patterns in fetal brain images, which helps it outperform standard methods. As a whole, the proposed method shows potential for better clinical decision support systems by increasing the reliability of fetal health assessment diagnostics.

REFERENCES

[1] Agarwal, S., Tarui, T., Patel, V., Turner, A., Nagaraj, U., & Venkatesan, C. (2023). Prenatal Neurologic Diagnosis: Challenges in Neuroimaging, Prognostic Counseling, and Prediction of Neurodevelopmental Outcomes. Pediatric Neurology.

[2] Whitby, E., & Gaunt, T. (2023). Fetal lung MRI and features predicting post-natal outcome: a scoping review of the current literature. The British Journal of Radiology, 96, 20220344.

[3] La Verde, M., Savoia, F., Riemma, G., Schiattarella, A., Conte, A., Hidar, S., ... & Morlando, M. (2023). Fetal aortic isthmus Doppler assessment to predict the adverse perinatal outcomes associated with fetal growth restriction: systematic review and meta-analysis. Archives of Gynecology and Obstetrics, 1-14.

[4] Mennickent, D., Rodríguez, A., Opazo, M. C., Riedel, C. A., Castro, E., Eriz-Salinas, A., ... & Araya, J. (2023). Machine learning applied in maternal and fetal health: a narrative review focused on pregnancy diseases and complications. Frontiers in Endocrinology, 14, 1130139.

[5] Tang, J., Han, J., Xue, J., Zhen, L., Yang, X., Pan, M., ... & Lu, L. (2023). A Deep-Learning-Based Method Can Detect Both Common and Rare Genetic Disorders in Fetal Ultrasound. Biomedicines, 11(6), 1756.

[6] Ramirez Zegarra, R., &Ghi, T. (2023). Use of artificial intelligence and deep learning in fetal ultrasound imaging. Ultrasound in Obstetrics & Gynecology, 62(2), 185-194.

[7] Krishna, T. B., & Kokil, P. (2023). Automated classification of common maternal fetal ultrasound planes using multi-layer perceptron with deep feature integration. Biomedical Signal Processing and Control, 86, 105283.

[8] Ungureanu, A., Marcu, A. S., Patru, C. L., Ruican, D., Nagy, R., Stoean, R., ... & Iliescu, D. G. (2023). Learning deep architectures for the interpretation of first-trimester fetal echocardiography (LIFE)-a study protocol for developing an automated intelligent decision support system for early fetal echocardiography. BMC Pregnancy and Childbirth, 23(1), 20.

[9] Shruthi, K., & Poornima, A. S. (2023). A Method for Predicting and Classifying Fetus Health Using Machine Learning. International Journal of Intelligent Systems and Applications in Engineering, 11(2), 752-762.

[10] Varghese, B., Jala, A., Meka, S., Adla, D., Jangili, S., Talukdar, R. K., ... & Adela, R. (2023). Integrated metabolomics and machine learning approach to predict hypertensive disorders of pregnancy. American Journal of Obstetrics & Gynecology MFM, 5(2), 100829.

[11] Asfaw, D., Jordanov, I., Impey, L., Namburete, A., Lee, R., & Georgieva, A. (2023). Multimodal Deep Learning for Predicting Adverse Birth Outcomes Based on Early Labour Data. Bioengineering, 10(6), 730.

[12] Zhou, Z., Zhao, Z., Zhang, X., Zhang, X., Jiao, P., & Ye, X. (2023). Identifying fetal status with fetal heart rate: Deep learning approach based on long convolution. Computers in Biology and Medicine, 159, 106970.

[13] Verma, D., & Agrawal, S. (2023). A Novel Framework for Fetal Nuchal Translucency Abnormality Detection Using Hybrid Maxpool Matrix Histogram Analysis. Computer Assisted Methods in Engineering and Science.

[14] Yin, Y., & Bingi, Y. (2023). Using Machine Learning to Classify Human Fetal Health and Analyze Feature Importance. BioMedInformatics, 3(2), 280-298.

[15] Priya, M., & Nandhini, M. (2023). Detection of fetal brain abnormalities using data augmentation and convolutional neural network in internet of things. Measurement: Sensors, 100808.

[16] Kaliappan, J., Bagepalli, A. R., Almal, S., Mishra, R., Hu, Y. C., & Srinivasan, K. (2023). Impact of Cross-Validation on Machine Learning Models for Early Detection of Intrauterine Fetal Demise. Diagnostics, 13(10), 1692.

[17] Cao, Z., Wang, G., Xu, L., Li, C., Hao, Y., Chen, Q., ... & Wei, H. (2023). Intelligent antepartum fetal monitoring via deep learning and fusion of cardiotocographic signals and clinical data. Health Information Science and Systems, 11(1), 16.