Journal of King Saud University – Computer and Information Sciences 35 (2023) 101647

HOSTED BY

Contents lists available at ScienceDirect

Journal of King Saud University – Computer and Information Sciences

journal homepage: www.sciencedirect.com

# StackFBAs: Detection of fetal brain abnormalities using CNN with stacking strategy from MRI images

Anjir Ahmed Chowdhury [a,b], S.M. Hasan Mahmud [a,b,*], Khadija Kubra Shahjalal Hoque [a], Kawsar Ahmed [c,d,*], Francis M. Bui [c], Pietro Lio [f], Mohammad Ali Moni [g], Fahad Ahmed Al-Zahrani [e,*]

[a] Department of Computer Science, American International University-Bangladesh (AIUB), Dhaka 1229, Bangladesh
[b] Centre for Advanced Machine Learning and Application (CAMLAs), Dhaka 1229, Bangladesh
[c] Department of Electrical and Computer Engineering, University of Saskatchewan, 57 Campus Drive, Saskatoon, SK S7N 5A9, Canada
[d] Group of Biophotomatiχ, Department of Information and Communication Technology (ICT), Mawlana Bhashani Science and Technology University (MBSTU), Tangail 1902, Bangladesh
[e] Department of Computer Engineering, Umm Al-Qura University, Mecca 24381, Saudi Arabia
[f] Department of Computer Science and Technology, University of Cambridge, Cambridge CB3 0FD, UK
[g] Faculty of Health and Behavioural Sciences, The University of Queensland, St Lucia, QLD 4072, Australia

ARTICLE INFO

*Article history:*
Received 24 May 2023
Revised 26 June 2023
Accepted 3 July 2023
Available online 18 July 2023

*Keywords:*
Brain abnormalities detection
Deep learning
Federated learning
Neural architecture search
Stacking strategy
Transfer learning

ABSTRACT

Predicting fetal brain abnormalities (FBAs) is an urgent global problem, as nearly three of every thousand women are pregnant with neurological abnormalities. Therefore, early detection of FBAs using deep learning (DL) can help to enhance the planning and quality of diagnosis and treatment for pregnant women. Most of the research papers focused on brain abnormalities of newborns and premature infants, but fewer studies concentrated on fetuses. This study proposed a deep learning-CNN-based framework named StackFBAs that utilized the stacking strategy to classify fetus brain abnormalities more accurately using MRI images at an early stage. We considered the Greedy-based Neural architecture search (NAS) method to identify the best CNN architectures to solve this problem utilizing brain MRI images. A total of 94 CNN architectures were generated from the NAS method, and the best 5 CNN models were selected to build the baseline models. Subsequently, the probabilistic scores of these baseline models were combined to construct the final meta-model (KNN) utilizing the stacking strategy. The experimental results demonstrated that StackFBAs outperform pre-trained CNN Models (e.g., VGG16, VGG19, ResNet50, DenseNet121, and ResNet152) with transfer learning (TL) and existing models with the 5-fold cross-validation tests. StackFBAs achieved an overall accuracy of 80%, an F1-score of 78%, 76% sensitivity, and a specificity of 78%. Moreover, we employed the federated learning technique that protects sensitive fetal MRI data, combines results, and finds common patterns from many users, making the model more robust for the privacy and security of user-sensitive data. We believe that our novel framework could be used as a helpful tool for detecting brain abnormalities at an early stage.


## 1. Introduction

The use of magnetic resonance imaging (MRI) is one of the most significant and frequently used techniques to examine brain activity (Attallah et al., 2019). MRI imaging is also very useful for investigating fetal brain imaging (Attallah et al., 2018). The early detection of fetal brain anomalies by scanning is crucial as about 3 out of 1000 pregnancies have various types of brain abnormalities (BA). In addition, a range of neuropathological variations may happen which is related to serious clinical morbidities (Griffiths et al., 2017). With the help of MRI diagnosis, the abnormality of the fetus's brain and tumors in the early stage can help with the preliminary study without any medical interference

* Corresponding authors.
*E-mail addresses:* anjir@aiub.edu (A.A. Chowdhury), hasan.swe@aiub.edu (S.M. Hasan Mahmud), k.ahmed@usask.ca, kawsar.ict@mbstu.ac.bd, k.ahmed.bd@ieee.org (K. Ahmed), francis.bui@usask.ca (F.M. Bui), pl219@cam.ac.uk (P. Lio), m.moni@uq.edu.au (M.A. Moni), fayzahrani@uqu.edu.sa (F.A. Al-Zahrani).
Peer review under responsibility of King Saud University.

Production and hosting by Elsevier

https://doi.org/10.1016/j.jksuci.2023.101647
1319-1578/

(Alansary et al., 2015; Katorza et al., 2018). Doctors can help parents to understand and also prepare them to deal with abnormality, also the possible treatments if it is detected as early as possible (Katorza et al., 2018). In the domain of bioinformatics, deep Learning (DL) is utilized as a great mechanism for pattern recognition and image processing. DL is used for analyzing ultrasound images and also it is used outside the field of fetal (van Sloun et al., 2020; Ouahabi and Taleb-Ahmed, 2021; Shen et al., 2021; Zaffino et al., 2020). This technique can be enforced for disease images to assist specialists in analyzing disease diagnoses. Therefore, DL can be essential for examining the MRI images of the fetus's brain for early detection and identifying the specific diseases (Hosseini and Zekri, 2012). With the help of the MRI scan, we can get hidden properties of the soft tissues under the fetus's brain structure. So, without any medical obstruction, the preliminary identification of brain abnormalities can be found (Attallah et al., 2019). Therefore, identifying them in the initial stage with the DL has many benefits. Firstly, doctors can easily diagnose the disease accurately and consult the parents about the upcoming challenges and prepare them for managing the abnormality. Secondly, it aids to guide the issues that might occur during the pregnancy. Lastly, it enhances the standard of the treatment plan which is suitable for the patient (Katorza et al., 2018).

The recent work on MRI images mostly focused on segmentation. In papers (Sanz-Cortes et al., 2013; Sanz-Cortés et al., 2013), they used the FBA images for limited classifications, also authors have utilized these images to forecast an abnormal neurobehavior for gestational age (SGA) babies using the segmentation technique. They segmented the fetal brains manually which is time-consuming. Similarly in another paper (Kainz et al., 2014), they also proposed a fully automatic framework to perform brain classification using rotation invariant extractors. They demonstrated that the classification technique can be directly applied for brain segmentation using basic refining approaches inside raw MRI scan data, resulting in a final segmentation with more than a 0.90 Dice score. More research is concentrating on brain-structure segmentation. For instance, Wang et al. (2018) introduced a novel DL-based system (MCANet) that can segment the middle cerebral artery and obtain the knowledge of gate position. This system reduces the workload of the sonographers to adjust several hyper-parameters for generating high-resolution and colorful ultrasound images. As an encoder, a pre-trained ResNet is used, and as a decoder, dense upsampling convolution blocks are used in this study. The authors of the work (Wu et al., 2020) presented a deep attention network to automate the process of measurement of cavum septum pellucidum. This process is normally done manually, Therefore, it has become difficult and time-consuming even for expert sonographers. Their work is inspired by U-Net encoder-decoder architecture where the backbone of the encoding path is VGG11 and between the encoder and the decoder, a channel attention module is added. These two studies (Singh et al., 2021; Zhang et al., 2020a) are very similar to Wu et al. (2020). They also inspired their CNN architecture from U-Net. Singh et al. (2021) optimized a ResU-Net-c semantic segmentation model to automatically segment the cerebellum and Zhang et al. (2020a) designed a CNN-based system (MANet) that can segment the circumference of a fetus's head. Both systems are based on encoder-decoder architecture.

Apart from that, few researchers focused on exploring 3D architecture. In paper, Huang et al. (2018), the proposed a framework (VP-Nets) to segment extra-cranial tissues in order to segment and pinpoint 5 brain areas at the same time. 3D U-Net is also utilized in Wyburd et al. (2020,). In paper (Wyburd et al., 2020), Using three different types of CNNs, they offered a new approach for segmenting the compact bone from 3D sonogram images. Similarly, Venturini et al. (2020) also proposed a multi-task CNN-based 3D U-Net to automatically mask out several segments of the fetal brain from ultrasound images. All these studies are further explored in this paper (Namburete et al., 2018). They resolved three issues regarding fetal structural segmentation, 3D fetal brain localization, and alignment to a referential coordinate system. Moser et al. (2019) also work on 3D fetal brain localization using end-to-end 3D CNN. The research on fetal brains is not limited to these studies. More recent articles demonstrate a few more techniques regarding fetal brain segmentation. Such as The author of this paper (Lee et al., 2020) claimed that their novelty is the use of the Bayesian Network on ultrasound images to predict GA. Few papers are focused on investigating fetal brain development. Namburete et al. (2017) and Wyburd et al. (2021) used 3D Convolutional Regression Network and CNNs to estimate the fetal brain maturation respectively. More extended discussion regarding fetal brain analysis is present in this most recent paper (Fiorentino et al., 2022).

Nevertheless, few works used MRI images that focus on classifications (Hosseini and Zekri, 2012; Makropoulos et al., 2018). Attallah et al. (2018) claimed that their research is the first to classify FBAs based on GAs. The research uses a variety of machine learning (ML) classifiers, including K-nearest neighbor (KNN), random forest (RF), naive Bayes (NB), neural network (NN), and diagonal quadratic discriminates analysis (DQDA). They were able to attain the best accuracy utilizing the KNN classifier, according to their findings.

We proposed a novel CNN-based framework that used the stacking method to achieve a more accurate and stable performance in detecting fetal brain abnormalities at an early stage. To automatically find the best CNN architectures for our up-to-date dataset (∼ see samples of the dataset in Fig. 2) ([link], 2022), we employed the Neural architecture search (NAS) approach (Liu et al., 2020). However, existing algorithms (such as NASNet, PNAS, Bayesian optimization, Random search, and Hyperband) are often computationally expensive. To overcome this issue, we used a Greedy-based NAS (Chowdhury et al., 2022) algorithm to find the best CNN architectures. In addition, we also compared these architectures with the pre-trained CNN Models (e.g., VGG16, VGG19, ResNet50, DenseNet121, and ResNet152) with transfer learning (TL). Among 94 CNN architectures that are generated from the NAS method, the best 5 models are selected for baseline models. StackFBAs framework effectively merged these baseline models (see baseline architectures in Fig. 3)) to obtain the final meta-model utilizing the stacking technique to fuse their strengths. Moreover, we must be cautious about the privacy and security of user-sensitive information while dealing with medical image data. As a result, we utilized a mechanism called federated learning (FL) (Yang et al., 2019).

## 2. Materials & proposed methodology

In this section, we have explained the fundamental concepts of StackFBAs in depth. Fig. 1 is the flowchart of StackFBAs.

### 2.1. Dataset

The dataset we utilized in our study was accumulated by a medical group at Harvard Medical School ([link], 2022). The dataset contains 227 fetal MRI images (114 abnormal and 113 normal images) taken between 16 and 39 weeks of gestation. 20% of the dataset was reserved for testing, including both normal and abnormal samples, while the remaining portion was used for training. A detailed description of the dataset is available online in [link] (2022). Fetal MRI scans are acquired as stacks of T2-weighted images using a standard half-Fourier single-shot RARE grouping

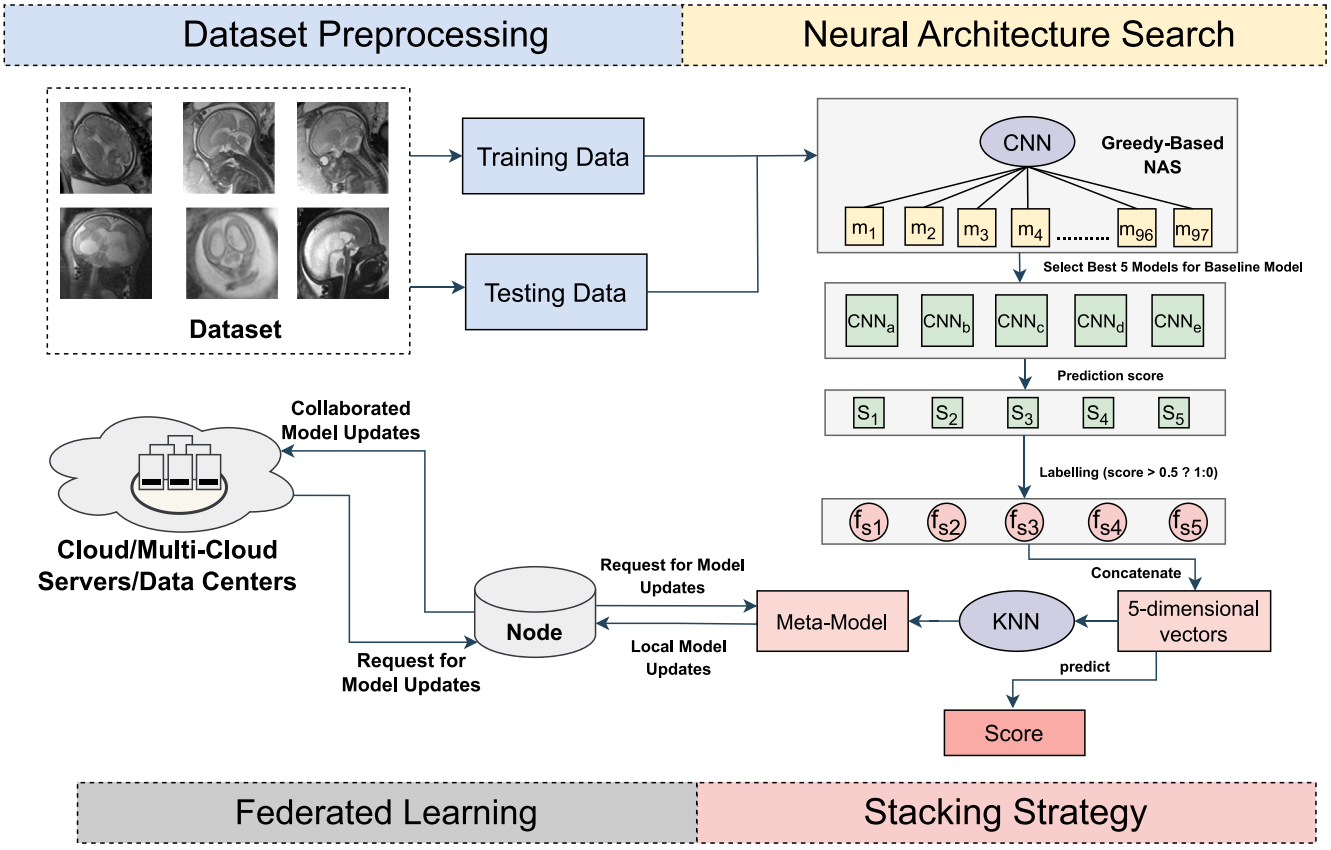


**Fig. 1.** The overall framework of the proposed StackFBAs. The architecture of StackFBAs involves four major stages, including (A) data pre-processing, where all FBAs datasets were collected and split as the training and testing dataset (B) Neural architecture search (NAS), where top five optimized architectures ($CNN_a, CNN_b, CNN_c, CNN_d$, and $CNN_e$) were selected based on the prediction scores for the brain MRI dataset; (C) Stacking strategy and the meta-model was built using the top five CNN optimized architectures where the KNN model was applied as a stacking classifier for the prediction task; (D) Federated learning, where our stacking meta-model was connected with Node with data model update request and Node had a connection with multi-cloud/data center to retrieve the secure data.

approach from different plans (e.g. axial, sagittal, and coronal). Due to the fetal motion throughout the checkup, each obtainment acts as a scout for the next obtainment. A typical sequence had an echo train length of 72, an echo spacing of 4.2 ms, a TE of 60 ms, a field of view of 24 × 24 cm, a section thickness of 4 mm, and an acquisition matrix of 192 × 256. The acquisition time for each image is only 430 ms per slice. In addition, to restrict the portion of the radio frequency (RF) control statement, a 130-degree refocusing heartbeat was employed.Attallah et al. (2019). Fig. 2 contains sample images from the dataset.

### 2.2. Transfer learning

Transfer learning (TL) is a kind of DL-based model where an architecture implemented for a prediction task is reused on a new problem (Houlsby et al., 2019). A machine utilizes previously trained knowledge to increase the prediction ability for a second task in the TL. TL is widely applied to train deep learning models for diverse natural language processing (Houlsby et al., 2019; Durrani et al., 2021), computer vision (Li et al., 2020; Whatmough et al., 2019), and image processing-related (Hussain et al., 2018; Aslan et al., 2021) tasks. In image processing, neural networks generally apply to find edges in the first layers, build in the central layer, and perform the task in the last layers. The first and middle layers are utilized in TL, and the final layers are only retrained. Using TL in our model aims to improve neural network performance, reduce training time complexity, and manage missing data. The generated model approach and the pre-trained model approach are two popular procedures for the TL model, and both methods were considered for training in this study. We selected multiple pre-trained deep CNN models such as VGG16, VGG19, ResNet50, ResNet152, and DenseNet121 to perform the TL. TL associates the concepts of a domain and a task. A domain $\mathcal{D}$ consists of a feature space $\mathcal{X}$ and a marginal probability distribution $\mathcal{P}(\mathcal{X})$ over the feature space, where $\mathcal{X} = x_1, \ldots, x_n \epsilon \mathcal{X}$. For MRI image classification, $\mathcal{X}$ is the space of all image representations, $x_i$ is the $i$ th term vector corresponding to some image and $\mathcal{X}$ is the sample of images used for training. So the formal definition of transfer learning is, Given a domain, $\mathcal{D} = X, P(X)$, a task $\mathcal{T}$ consists of a label space $\mathcal{Y}$ and a conditional probability distribution $P(Y|X)$ that is typically learned from the training data consisting of pairs $x_i \epsilon \mathcal{X}$ and $y_i \epsilon \mathcal{Y}$. In the MRI image classification example, $\mathcal{Y}$ is the set of all labels, (i.e. Normal, Abnormal) and $y_i$ is either Normal or Abnormal.

### 2.3. Neural architecture search

Neural Architecture Search (NAS) is an automating mechanism for CNN architecture engineering that has outperformed hand-designed architectures on several problems in object detection (Chen et al., 2018), image identification (Zoph et al., 2017; Real et al., 2018), and semantic segmentation (Zoph et al., 2017). NAS method can be categorized into three dimensions: search space which is utilized in StackFBAs, search and performance evaluation techniques. It is a subfield of AutoML (Hutter et al., 2019) shows a lot of similarities with hyperparameter optimization (Feurer and Hutter, 2019) and meta-learning (Vanschoren, 2019). Greedy

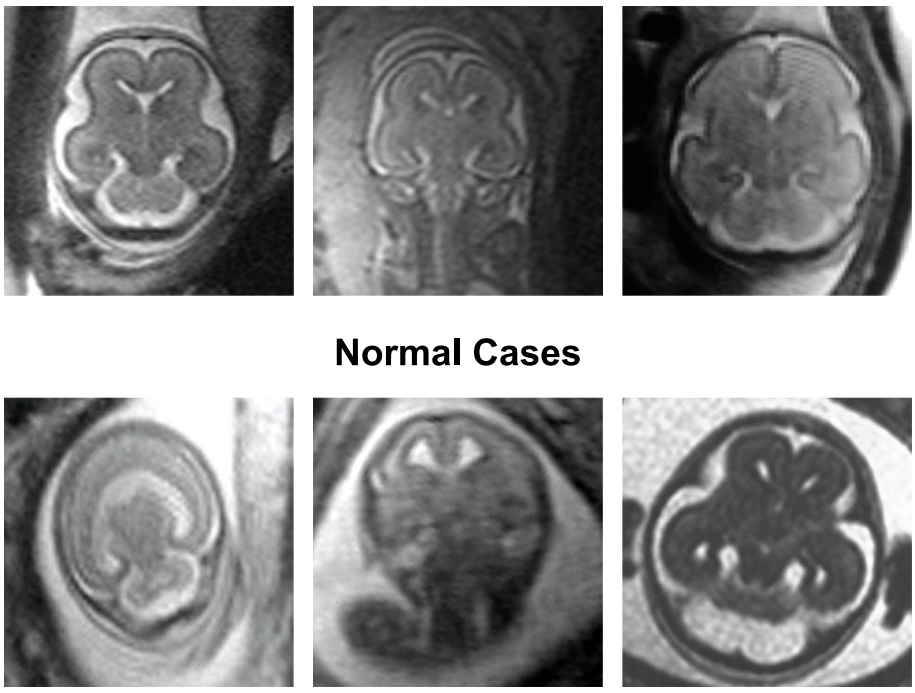


**Fig. 2.** Few samples of the Dataset (Normal and Abnormal cases).

hyperparameter optimization (Chowdhury et al., 2022) has proven to be an effective approach for hyperparameter optimization that can also be applied to NAS. Greedy-based NAS (GNAS) is very effective, compact, and resource-friendly for designing CNNs due to its greedy strategies. We applied Greedy-Based NAS (Chowdhury et al., 2022; Huang et al., 2018) in StackFBAs to automatically find the optimized architectures for the Brain MRI dataset.

In this study, GNAS reduced the range of possible evaluated architectures from exponential to linear complexity of the features using greedy strategies. It can significantly accelerate the back-propagation training by implementing the weight-sharing mechanism of individual possible architectures (Pham et al., 2018; Real et al., 2017). In StackFBAs, the GNAS method handles the automatic design of CNN architectures in a completely different way. It splits the general architecture optimization problem into subproblems, depending on appropriate layer (Inter and Intra) greedy strategies from the aspect of neural architecture optimization and the greedy strategy assures that the architecture search is efficient. Generally, GNAS is a non-parametric technique that avoids the loop of using additional hyper-parameters for reinforcement learning (RL) (Zoph et al., 2018) and Bayesian optimization (BO) (Snoek et al., 2012). Recently DeepQGHO and GNAS have achieved promising results in terms of time consumption and for searching multi-attribute learning, respectively. We have generated the best five CNN architectures from StackFBAs using NAS are shown in Fig. 3.

### 2.4. Stacking strategy

Previous research has shown that ensemble approaches can outperform their baseline models in terms of predictive performance (Manavalan et al., 2020; Wei et al., 2021; Xie et al., 2020; Zhang and Zou, 2020; Charoenkwan et al., 2021). Generally, there are three types of ensemble approaches: stacking, bagging, and boosting strategies (Xie et al., 2020; Qiang et al., 2018). In order to obtain an accurate prediction, we used a feature representation method based on a stacking strategy that may reap the benefits of the individual baseline model. Base-classifier and meta-classifier are the two stages of stacking ensemble learning (Wolpert, 1992; Sun and Trevor, 2018). The training set is used in the base-level classifier to train models and make predictions. The meta-model uses heterogeneous data for training, while the base-output classifier is transferred to the real classification tag. In this work, we have considered different CNN models which were generated from the NAS algorithm. Then we stacked the best five based on the CNN models and then the meta classifier, KNN, is used to predict each base model. Fig. 4 depicts the stages involved in implementing the stacking method. The 5 PFs of a given MRI image were applied as new input features for the meta-model, which can be formulated by the Eq. 1 (Charoenkwan et al., 2022):

$$nFeat(P) = [f(BC_1(P)), f(BC_2(P)), \ldots, f(BC_5(P))]^T \quad (1)$$

where $f(BMi(P))$ represents the $i$th PF attained by the $i$th baseline model $(BC_i)$ of the MRI Image $P$.

### 2.5. Federated learning

Federated learning (FL) is a training statistical model that is constructed using a distributed dataset (Jia et al., 2022; Li et al., 2022; Zhang et al., 2021; Lu et al., 2020). This procedure does not disclose any confidential information between the clients. The trained federated technique can be installed in the systems of each participant or shared with all clients. In terms of data privacy, FL models trained on decentralized data are a new theory. It can ensure privacy while minimizing latency and a central model's replica is distributed into connected devices. The input data of users from each device is employed to train the models and the training results are sent to the cloud server to be compiled and updated to the central model. The working principle of FL is repre-

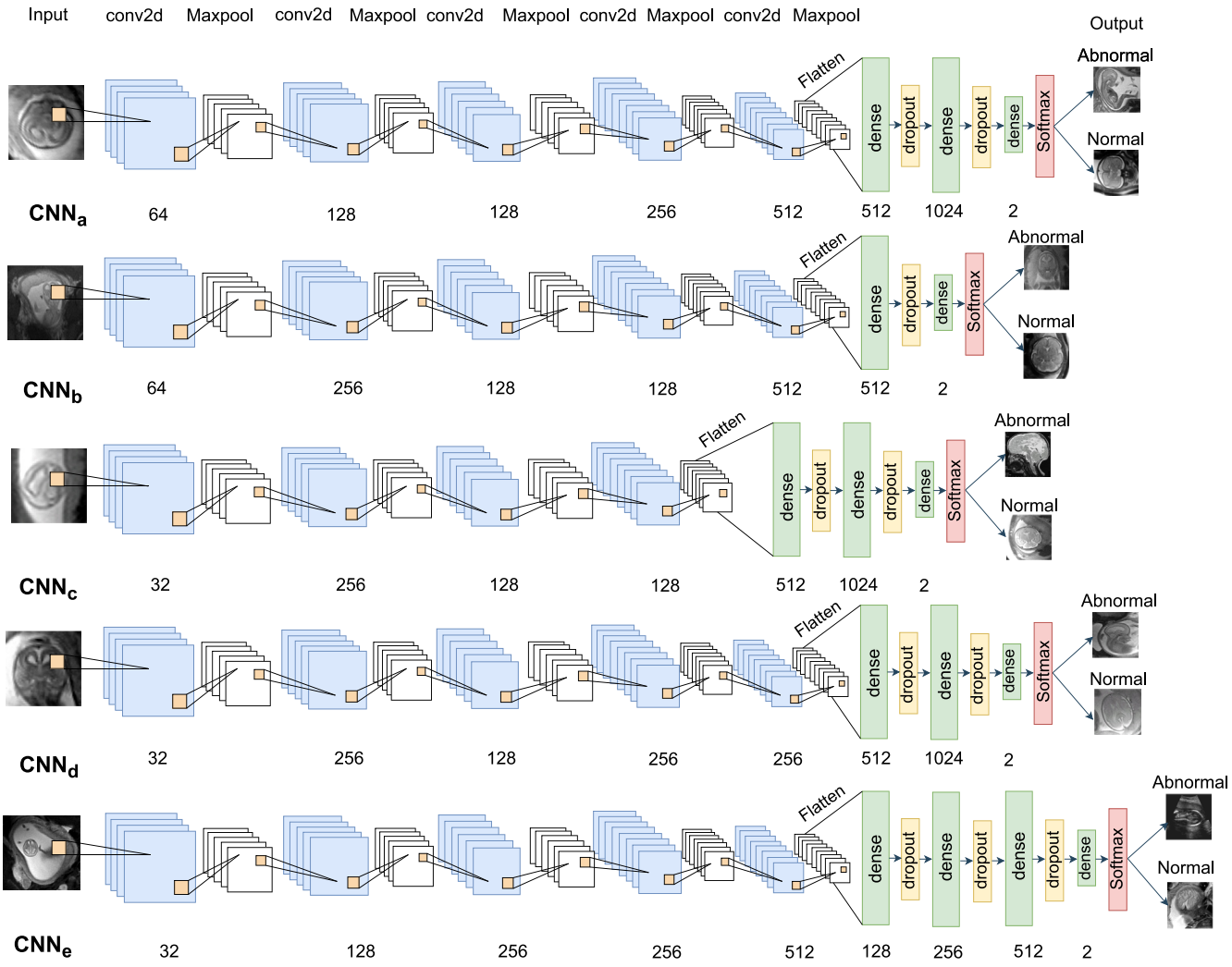


**Fig. 3.** The architecture of the five best CNN models. A and D contain 21 layers with two fully connected layers, and their structure includes five convolutional layers, followed by a MaxPool. The main difference between these two architectures is neuron size. Similarly, B has b has 19 layers with only one fully connected layer, and it also has five sets of convolutional layers with a MaxPool activation. C has 16 layers, and it has four sets of convolutional layers. Lastly, e is the largest architecture because of the layers, where it contains 31 layers, including three fully connected layers. These five architectures are identical in terms of their number of layers, neuron size, and performance.

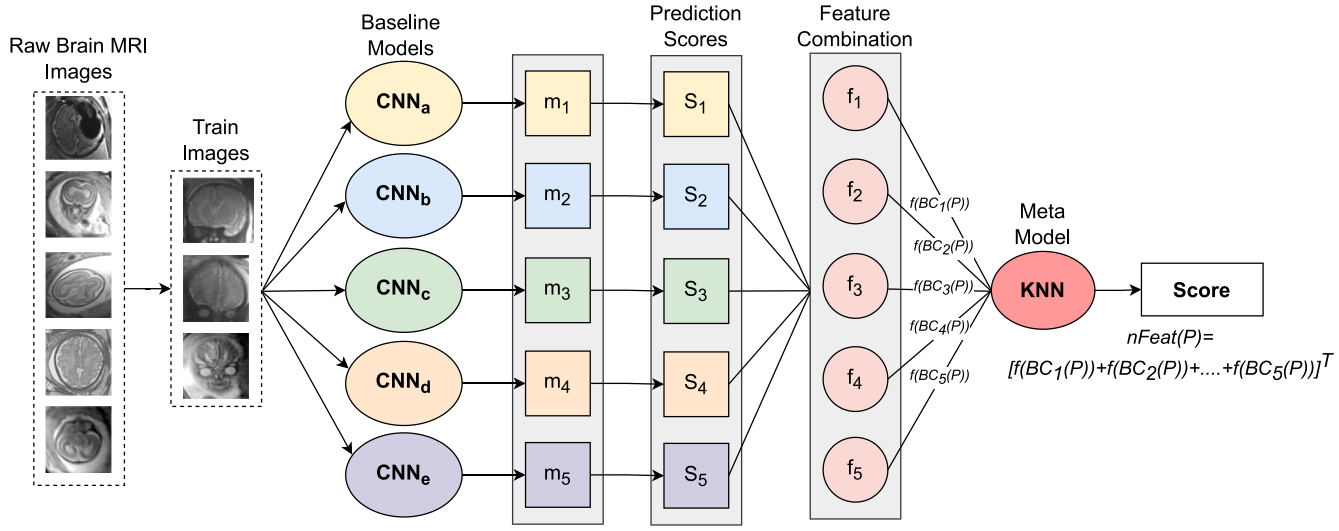


**Fig. 4.** A step-by-step procedure for constructing the stacking models for MRI images. Five different models are generated that are selected by the NAS technique. Second, the prediction probabilities scores were combined by applying the stacking strategies to generate feature sets. Based on the feature set, we train the KNN model with five feature spaces and exploit the integration of individual feature set into the stacking model. Finally, the prediction score is generated from the KNN predicting FBAs with improved accuracy.

sented in 1. An FL technique, particularly for medical diagnosis data, has a substantial impact on data anonymity and protection. Federated learning can be defined as follow: consider a real application scenario such as our application where we developed a CNN-based stacking model which can identify brain abnormalities. It is assumed that N users, $\{N_1, N_2 \ldots, N_n\}$ each have their own database, $\{D_1, D_2 \ldots, D_n\}$ and that none of them can directly access other people's data to extend their own. As we know that the aim of FL is to train a model by combining training data from several devices. We have followed three simple steps to perform FL (McMahan et al., 2016): (1) Each device receives the initial model from the server. (2) The machine $N_i$ does not need to exchange its source data; instead, it can independently train its own model $M_i$ using the local data $D_i$. (3) The server uploads the local models $\{M_1, M_2, \ldots, M_n\}$ to the global model $M$ and then updates each user's local model with the global model. We are dealing with highly sensitive MRI data. Therefore, We need to ensure data confidentiality. We have opted to use the Model aggregation mechanism of the federated learning in the StackFBAs framework to ensure the privacy of the data. We have mentioned the Federated Learning technique in Algorithm 4 as a part of our proposed StackFBAs.

### 2.6. Proposed framework

There are three primary functions in the StackFBAs framework that control the entire procedure: 1) arcSearch() 1 for NAS 2) stacking() for stacking technology 2, and 3) ClientUpdate () 3 for federated learning and the main function controls all of these functions. For finding the optimal architectures for this dataset, each hyperparameter is optimized by the greedy-based NAS algorithm while some hyperparameters are kept constant during the search. Each hyperparameter optimization is repeated until all local hyperparameters have been optimized. Let $\mathcal{T} = (\mathcal{T}_n, \mathcal{T}_h)$ be a DNN configuration, where $\mathcal{T}_n$ denotes a network topology and $\mathcal{T}_h$ denotes the DNN hyperparameters for a decision set $\mathcal{D}$. We suppose that $w$ is the set of all DNN weights. Using the validation accuracy function, $\mathcal{V}$, the equation optimizes the hyperparameters, $\mathcal{T}_h$. With the number of hyperparameters, the cost of computing the optimization in Eq. 2 increases exponentially. This exponential computational cost is reduced using the greedy method. The greedy method, in particular, optimizes one hyperparameter while leaving the others constant. Let $n$ represents the number of hyperparameters in $\mathcal{T}_h = (\mathcal{T}_h^1, \mathcal{T}_h^2, \ldots, \mathcal{T}_h^n)$, with $\mathcal{T}_h^i \epsilon \mathcal{D}^i$ being the $i$th hyperparameter in $\mathcal{T}_h$. Let's say the search space for the ith hyperparameter $\mathcal{T}_h^i$ has $m$ elements and is represented by $Di = (\mathcal{D}_1^i, \mathcal{D}_2^i, \ldots, \mathcal{D}_m^i)$, where $\mathcal{D}_i^j$ is the $j$th element in the $\mathcal{T}_h^i$ search space. The decision set for the hyperparameter $\mathcal{T}_h^i.\mathcal{D}^i \epsilon \mathbb{R}^m$. We can rewrite the equation (Chowdhury et al., 2022) using the greedy-based hyperparameters optimization (GHO) method as follows.

$$\max_{\mathcal{T}_h^i \in \mathcal{D}^i} \quad f_{\mathcal{V}}([\mathcal{T}_n, \mathcal{T}_h], \mathcal{V}, w^*) \qquad \forall i \tag{2}$$

The pseudo-code of the GHO algorithm is mentioned in archSearch() function in Algorithm 1, where, validation accuracy $\mathcal{V}_a = (\mathcal{V}_a^1, \mathcal{V}_a^2, \ldots, \mathcal{V}_a^n)$ and $\mathcal{V}_a^i$ indicates the validation accuracy achieved from the Eq. 2.

**Algorithm 1.** Neural Architecture Search

```
archSearch(i, j, T_h^i, D^i) :
T_h^{i,j} ← D_j^i
Evaluate V_a^{i,j}
if V_a^{i,j} ⩾ V_a^i then
    V_a^i ← V_a^{i,j}
    T_h^i ← T_h^{i,j}
end if
return V_a^i, T_h^i
```

For the stacking process, the data frame $\mathcal{D}_f$ and KNN model $\mathcal{M}_k nn$ are initialized along with a variable that will store the accuracy in $\mathcal{V}_M$. This accuracy is sorted in descending order in a $\mathcal{L}_i$. Then the best five configurations are taken to create another new data frame which is used to train with the KNN model. Finally, it will return the accuracy. The pseudo-code of the stacking technology is mentioned in the stacking() function in Algorithm 2.

**Algorithm 2.** Stacking

```
stacking(L_a) :
D_f ← Initialize data frame
M_knn ← Initialize KNN Model
V_m ← ∅
L_a.sort(reverse = TRUE)
for j ← 1 to 5 do
    D_f.append(predict(L_a[1]))
end for
V_m ← M_knn.fit(D_f)
return V_m
```

Although one of FL's primary goals is to increase data confidentiality, the attacks threaten to undermine the advantage. As a result, several well-known techniques have been incorporated into FL algorithms to optimize the systems' resistance to them. Differential privacy is a term that arose in machine learning and is used to characterize how resistant data and its analysis are to membership inference attacks. A system that is randomized $\mathcal{S} : \mathcal{D} \rightarrow \mathcal{R}$ with range $\mathcal{R}$ and domain $\mathcal{D}$ fulfills $(\epsilon, \delta)$ differential privacy if it holds for any two consecutive inputs $d, d_1 \in \mathcal{D}$ and any subset of outputs $\mathcal{M} \subseteq \mathcal{R}$ then Xu et al. (2021),

$$Pr[\mathcal{S}(d) \in \mathcal{M}] \leqslant e^{\epsilon} Pr[\mathcal{S}(d_1) \in \mathcal{M}] + \delta \tag{3}$$

In the domain of machine learning, adjacent inputs are two datasets $Y, Y\prime$ that differ in a discrete training sample, so that $Y = Y\prime \{y_n\}$. The randomized process is intended to ensure that the output of the Machine learning model cannot be followed back to the consequences of a single sample. Using a Gaussian method is one technique to provide differential privacy. Assume we want to encrypt $f : \mathcal{D} \rightarrow \mathbb{R}$, a deterministic real-valued function. Then, the Gaussian mechanism can be used as in Eq. 4 (Xu et al., 2021):

$$\mathcal{S}(d) = f(d) + \mathcal{N}(0, M_f^2 \sigma^2) \tag{4}$$

$S_f$ denotes the sensitivity of $f$, which is described as the greatest absolute distance $|f(d) - f\prime(d)|$ between two consecutive inputs $d$ and $d\prime$. When this additional Gaussian noise is used once with $f$, it fulfills $((\epsilon, \delta))$ differential privacy if $\delta \geqslant \frac{4}{5}\exp(-(\sigma\epsilon)^2/2)$ and $\epsilon < 1$. Abadi et al. provide a more extensive discussion of differentially private SGD and differential privacy (Abadi et al., 2016).

**Algorithm 3.** Client Update

**ClientUpdate**$(k_c, w_c)$ :
$\mathcal{B}_c \leftarrow$ (divide $\mathcal{P}_k$ into batches of size $\mathcal{B}_c$)
**for** every local iterations i from 0 to E − 1 **do**
  **for** batch $b_c \in \mathcal{B}_c$ **do**
    $w_c \leftarrow w_c - n_c \Delta l_c(w_c; b_c)$
  **end for**
**end for**
**return** $w_c$ to server

A client–server architecture is used in an FL system (pseudo-code of the client–server architecture is mentioned in Algorithm 3 & 4), with one server in charge of enabling training, constructing the model, and making it for training the model to all clients with the local data (Pfitzner et al., 2021). A formal algorithm in McMahan et al. (2016) implies that the weights of the model are distributed among clients and servers. Most of the techniques just share parameter information to limit the amount of data communication. When employing local mini-batches, McMahan et al. (2016) discovered that selecting a proportion of $C_k$ = 0.1 for each iteration ($K$ customers) is the best choice, while smaller values are rarely good. Furthermore, if the clients have sufficiently powerful computing equipment, setting the maximum epochs $E$ or minimizing the batch size $B$ might optimize the communication cost and accelerate the model convergence. We have utilized this FL algorithm in StackFBAs as shown in Algorithm 4 where $\eta$ represents the learning rate, $\mathcal{K}$ are indexed by $k; E$ indicates the total epochs, and $\mathcal{B}$ is the local minibatch size. The pseudo-code of the StackFBAs is mentioned in Algorithm 4.

**Algorithm 4.** StackFBAs

$\mathcal{V}_a \leftarrow \varnothing$
$\mathcal{L}_i \leftarrow \varnothing$
$\mathcal{T}_h^i \leftarrow$ Initialize hyperparamters
$\mathcal{D}^i \leftarrow$ Initialize search space
**for** $i \leftarrow 0$ to $n - 1$ **do**
  **for** $j \leftarrow 0$ to $m - 1$ **do**
    $\mathcal{L}_{i_0}, \mathcal{L}_{i_1} = \text{archSearch}(i, j, \mathcal{T}_h^i, \mathcal{D}^i)$
  **end for**
**end for**
$\mathcal{P}_s \leftarrow \varnothing$
$\mathcal{P}_s = \text{stacking}(\mathcal{L}_i)$
$\mathcal{N} \leftarrow \text{maximum}(\text{C} \times \text{K}, 1)$
**for** $t \leftarrow 0$ to $T - 1$ **do**
  $\mathcal{S}^t \leftarrow$ (arbitrary set of N clients)
  **for every clients** k $\in \mathcal{S}_t$ **in parallel do**
    $w_t^k \leftarrow \text{ClientUpadte}(k, w_t)$
  **end for**
  $w_t + 1 \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} w_t^k + 1$
**end for**

### 2.7. Evaluation metrics

To evaluate the performance of the generated five best models from NAS, five performance metrics are used, including sensitivity (SN), specificity (SP), accuracy (ACC), F-Score, and Matthews correlation coefficient (MCC). These metrics are formulated as follows:

$$SN = \frac{TP}{TP + FN} \tag{5}$$

$$SP = \frac{TN}{FP + TN} \tag{6}$$

$$ACC = \frac{TP + TN}{TP + FN + FP + TN} \tag{7}$$

$$F - Score = \frac{2TP}{2TP + FP + FN} \tag{8}$$

$$MCC = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{9}$$

## 3. Results

### 3.1. Evaluation of Five CNN models with transfer learning

All parameters of CNN models were started with random Gaussian distributions and trained for a maximum of 100 epochs with a batch size of 64 images. The training convergence can be noticed within the 15 epochs. The remaining hyperparameters are weight decay (0.0005), learning rate (0.001), and momentum (0.9) which was reduced by a factor of 10 every 15 epochs. Its learning rate remains unchanged at 0.001. Determining the ideal learning rate for each layer, particularly for very deep networks like DenseNet121, is difficult. For this experiment, we have used the Tensorflow framework and Nvidia GTX 1060 6 GB GPU to train CNNs. The experiments were started by splitting the dataset into two parts for training and testing, with an exact ratio of 80 and 20. We determined to apply the TL technique to the Five CNNs (VGG16, VGG19, ResNet50, ResNet152, and densenet121). These CNNs can be either fine-tuned from the pre-trained models or learned from scratch. Both approaches were performed in this work. We utilized the training dataset to execute, evaluate, and compare the results of the 5 pre-trained models. To observe how these models are working on the training dataset, we applied the K-fold cross-validation (CV) technique. The k-fold CV technique is a resampling technique that is splitting the dataset into two portions- training and testing data and training data are split into $k$ parts. In each iteration, $k - 1$ parts are participating in training and a single portion is for validation. To evaluate the pre-trained models, we used validation accuracy as the evaluation metric. The results of the CNN models utilized, together with their 5-fold accuracy and validation accuracy, are shown in Table 1. We have achieved the best results with ResNet50 with or without using transfer learning. Without using TL the test result of ResNet 50 is 0.68 and VGG16 and RestNet152 give the same result and hold the second position with a difference of 0.02 and a test result of 0.66. From Table 1 it can be seen that VGG19 gave the lowest accuracy at 0.63. Implementing TL has improved the results slightly. From the table, it can be seen that ResNet50's accuracy has increased slightly by 0.01 with an accuracy of 0.69. Here, VGG19 had an increase in accuracy by 0.05 and also gives the second-best result along with VGG16 with an accuracy of 0.68. DenseNet121 gave the least accuracy with

**Table 1**
K-Folds Accuracy and Validation Accuracy of CNN Models with and without Transfer Learning (TL).

| CNN Models | $CV_1$ | $CV_2$ | $CV_3$ | $CV_4$ | $CV_5$ | Test |
|---|---|---|---|---|---|---|
| VGG16 | 0.66 | 0.67 | 0.67 | 0.66 | 0.66 | 0.66 |
| VGG19 | 0.63 | 0.63 | 0.65 | 0.63 | 0.65 | 0.63 |
| ResNet50 | 0.68 | 0.67 | 0.67 | 0.67 | 0.68 | 0.68 |
| DenseNet121 | 0.64 | 0.64 | 0.65 | 0.64 | 0.65 | 0.64 |
| ResNet152 | 0.66 | 0.66 | 0.63 | 0.66 5 | 0.66 | 0.66 |
| VGG16-TL | 0.68 | 0.69 | 0.67 | 0.67 | 0.68 | 0.68 |
| VGG19-TL | 0.65 | 0.69 | 0.68 | 0.68 | 0.68 | 0.68 |
| ResNet50-TL | 0.70 | 0.67 | 0.68 | 0.67 | 0.69 | 0.69 |
| DenseNet121-TL | 0.66 | 0.66 | 0.66 | 0.68 | 0.67 | 0.66 |
| ResNet152-TL | 0.68 | 0.67 | 0.67 | 0.66 | 0.67 | 0.67 |

**Table 2**
Performance of the four NAS algorithms with several parameter settings.

| Experiments | Models | Algorithm | Computational Time | Validation Accuracy |
|---|---|---|---|---|
| $1^{st}$ Experimental Setup | $1^{st}$ Best | GHO | 2hr 22 min | 0.76 |
| | $1^{st}$ Best | BO | 5hr 14 min | 0.76 |
| | $1^{st}$ Best | RS | 5hr 01 min | 0.75 |
| | $1^{st}$ Best | HB | 3hr 12 min | 0.74 |
| $2^{nd}$ Experimental Setup | $2^{nd}$ Best | GHO | 1hr 51 min | 0.75 |
| | $2^{nd}$ Best | BO | 4hr 49 min | 0.74 |
| | $2^{nd}$ Best | RS | 4hr 41 min | 0.74 |
| | $2^{nd}$ Best | HB | 2hr 58 min | 0.75 |
| $3^{rd}$ Experimental Setup | $3^{rd}$ Best | GHO | 1hr 40 min | 0.74 |
| | $3^{rd}$ Best | BO | 4hr 38 min | 0.74 |
| | $3^{rd}$ Best | RS | 4hr 31 min | 0.74 |
| | $3^{rd}$ Best | HB | 2hr 17 min | 0.74 |
| $4^{th}$ Experimental Setup | $4^{th}$ Best | GHO | 1hr 22 min | 0.73 |
| | $4^{th}$ Best | BO | 4hr 10 min | 0.72 |
| | $4^{th}$ Best | RS | 3hr 55 min | 0.73 |
| | $4^{th}$ Best | HB | 2hr 02 min | 0.71 |
| $5^{th}$ Experimental Setup | $5^{th}$ Best | GHO | 1hr 08 min | 0.70 |
| | $5^{th}$ Best | BO | 3hr 48 min | 0.71 |
| | $5^{th}$ Best | RS | 3hr 38 min | 0.70 |
| | $5^{th}$ Best | HB | 1hr 37 min | 0.71 |

0.66. So, it can be observed that accuracy has improved marginally as a result of transfer learning.

### 3.2. Effectiveness of NAS algorithms

In this section, we have selected four algorithms, such as GHO (Chowdhury et al., 2022), Bayesian Optimization (BO) (Snoek et al., 2012), Random Search (RS) (Bergstra and Bengio, 2012), and Hyper Band (HB) (Li et al., 2017), to search for the optimal architectures from the search space (see Table 4). Table 2 shows the performance accuracies and computational times of the different search algorithms with different parameter settings. Here, we have only considered the best five architectures and their performances from each algorithm. In the $1^{st}$ experimental setup, GHO obtained the highest accuracy of 0.76 with 2 h 22 min computational time, and HB received the second-best accuracy of 0.75 with 3 h 12 min. The other variant of the search algorithm, RS, has the third best performer achieving a 0.75 accuracy score with 5hr 01 min. The BO algorithm obtained a 0.74 accuracy score with 5hr 14 min, which attained the worst prediction performance of the other three algorithms.

Over the four NAS algorithms, GHO and HB have consistently achieved the best results and remained the best algorithms in terms of validation accuracy and computational times. For the $2^{nd}$, $3^{rd}$, and $4^{th}$ experimental setup, we can also see that GHO and HB algorithms took much less computational time and showed high accuracy. The best performance (Accuracy: 0.76) we have obtained in $1^{st}$ experimental setup and less computational time (1 h 08 Mins) was taken by also GHO in the 5th experimental setup. In contrast, the GHO algorithm showed the highest prediction performance because of its fastest optimization (Greedy Algorithms (Zhang et al., 2000)), whereas Greedy Algorithms perform recursively based on the parameter set by a scanning list of objects. In Fig. 5 (a–e), the line graphs (epoch vs. accuracy) display different coverage areas, the line from GHO covers the highest spot, which is larger than the other three techniques. Besides, the computational time comparison graphs of different algorithms are shown in Fig. 6. The main aim of this experiment was to compare and analyze the performance of four NAS search algorithms and select the most effective search algorithm. This experiment discloses that GHO is more powerful and plays a significant role in choosing the best five CNN models in terms of accuracy and computational time.

### 3.3. Performance evaluation of baseline models

The accuracy of classification has improved significantly because of automated the design of CNN architectures. We applied a greedy-based NAS algorithm in StackFBAs. It generated 94 architectures among them best 10 are visually represented in Fig. 7. Later, the best 5 architectures were used to combine together using the stacking strategy. Then, based on the highest validation accuracy, each of the HPO algorithms chose the optimal hyperparameter configuration. We have set the activation function to relu, the

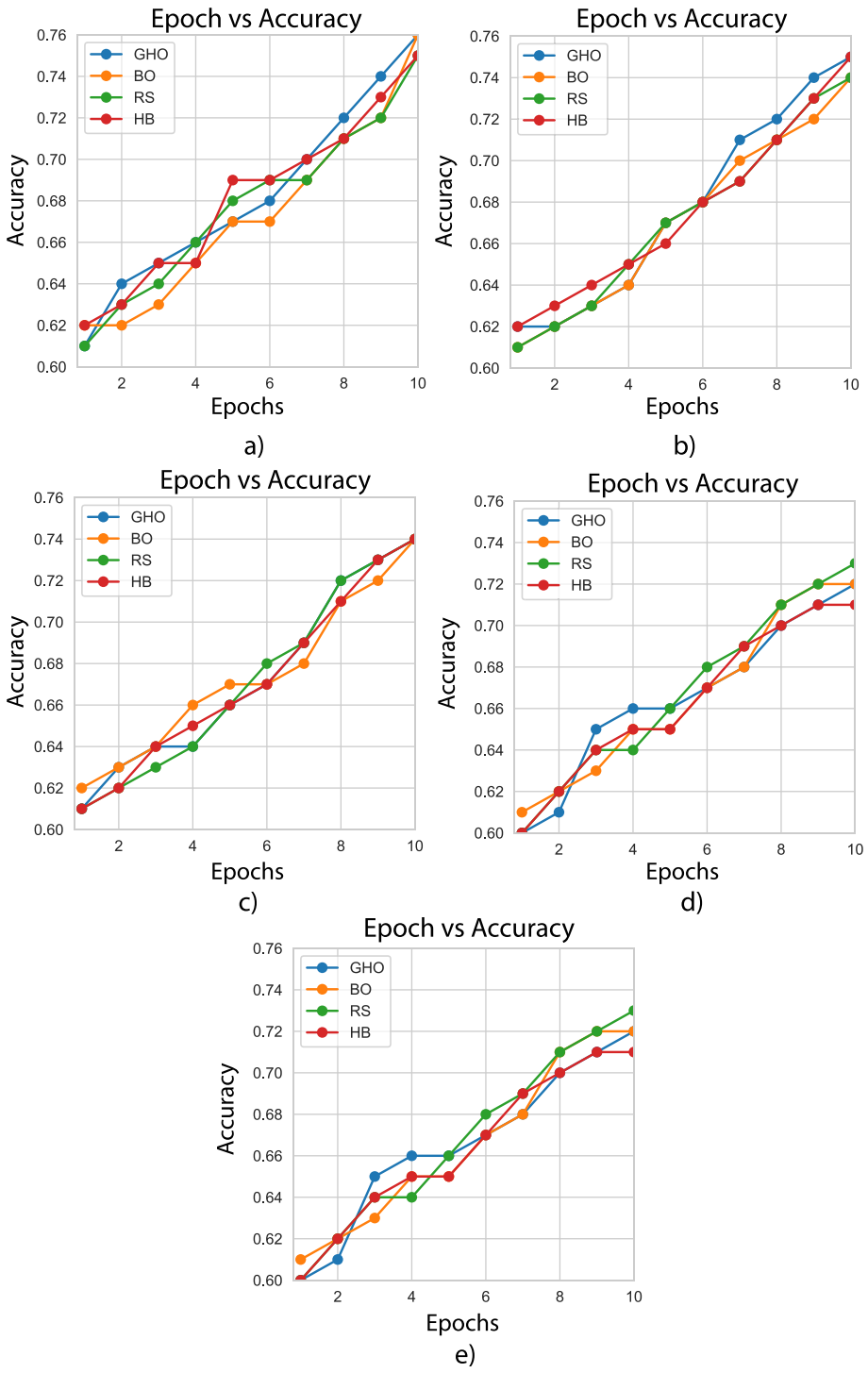


**Fig. 5.** Epoch Vs Accuracy curve a) 1$^{st}$ Experimental Setup b) 2$^{nd}$ Experimental Setup c) 3$^{rd}$ Experimental Setup d) 4$^{th}$ Experimental Setup e) 5$^{th}$ Experimental Setup.

optimizer to adam, the epochs to 100, and the kernel size to (3,3) to prevent time complexity. stride size: (2,2), and Pool size: (2,2).To prevent the model from overfitting, we used the early stopping callback function, as well as the monitor: 'val loss' and patience: 3. The hyperparameter configuration space is summarized in Table 4. The CNNs were tuned on a computer equipped with an Intel Core i5-4460 Processor, NVIDIA GTX 1060 GPU card, 16 GB of DDR4 CPU memory, and 6 GB of GPU memory. We used several open-source Python libraries and frameworks, including TensorFlow and Keras.

The performance of the best five models is shown in Table 3. These models are represented as $CNN_a, CNN_b, CNN_c, CNN_d$, and $CNN_e$. The $CNN_a$ has the highest accuracy of 0.76 and the $CNN_b$ comes up next with a 0.75 accuracy, while $CNN_e$ has the least accuracy with a 0.70. As can be seen, the accuracy rate has increased dramatically while employing NAS rather than transfer learning. Among the 5 architectures, $CNN_a$ has the highest f-score, SP, and MCC among the rest of the architectures with 0.76, 0.75, and 0.51 respectively. Therefore, we can say that the $CNN_a$ architecture is the best for our experiment. Fig. 8 represents ROC and PR curve

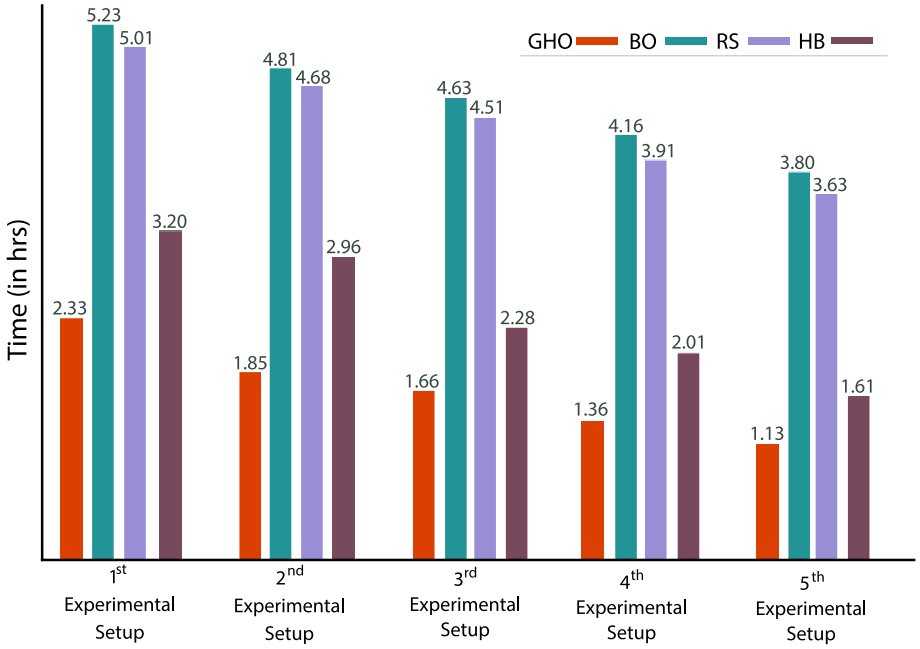


**Fig. 6.** Computational time comparison among several experimental setups.

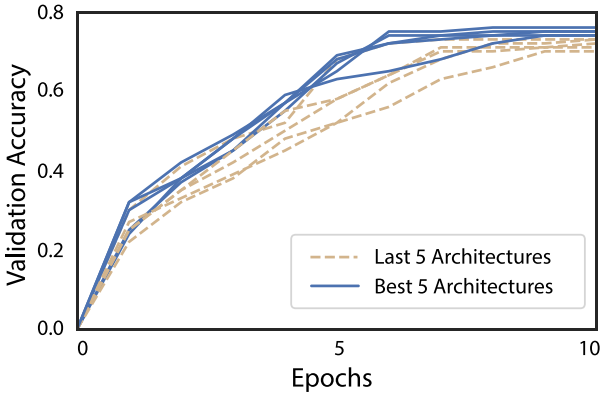


**Fig. 7.** Best ten CNN architectures generated from NAS.

for each experiment. From Fig. 8, we can observe that $CNN_a$ has the best performance than other CNNs that are generated from StackFBAs using NAS. The second and third (Fig. 8) ROC and PR curves are for pre-trained models with and without TL. In both cases, ResNet50 shows promising performances. However, the best five architectures ($CNN_a, CNN_b, CNN_c, CNN_d$, and $CNN_e$) of StackFBAs outperformed the pre-trained models in terms of ROC and PR curves in every scenario. Because of this, we fused the strength of these best five architectures using the stacking strategy.

### 3.4. Establishment of a meta-model using KNN

KNN was used to create a meta-model using the new 5-dimensional feature vectors. Unlike models created using simple ensemble procedures (such as average scoring and majority voting), our stacking techniques allow for the automatic exploration of several baseline architectures. The final meta-model might potentially give better and more steady performance by intelligently integrating their distinct strengths without human intervention (Wei et al., 2018; Xiong et al., 2018; Zhang et al., 2015). Table 5 shows the impact of each meta-model. As a result, we will be able to understand the performance of various ML algorithms as meta-models.

Table 5 shows the prediction results of different ML algorithms to find the suitable meta-model. For the meta-model, the KNN algorithm is considered the best prediction classifier. Whereas the SN, SP, F-Score and ACC of the KNN model reach 0.76, 0.78, 0.78, and 0.80. These results support the superiority of KNN over the other four classifiers. However, applying an LR algorithm as a meta-model, the SN, SP, F-Score and ACC are 0.74, 0.75, 0.75, and 0.76, which are 2%, 3%, 3%, and 4% lower than the KNN classifier. The XGBoost and NB gave the same and the least accuracy with an accuracy of 0.72 which is 8% lower than KNN and 4% lower than LR.

### 3.5. Performance of ensemble models

We developed and ran an experiment to test the performance of ensemble models using the stacking strategy: five CNN baseline models ($CNN_a, CNN_b, CNN_c, CNN_d$, and $CNN_e$) generated from StackFBAs using NAS were combined to form an ensemble model. When compared to pre-trained models with or without TL, it is clear that CNNs with stacking strategy consistently outperformed pre-trained CNN models on all metrics (i.e., SN, SP, ACC, and F-Score). These findings suggested that utilizing an ensemble model

**Table 3**
Performance of the best five architectures.

| Architectures | SN | SP | ACC | F-Score | MCC |
|---|---|---|---|---|---|
| $CNN_a$ | 0.74 | 0.75 | 0.76 | 0.76 | 0.51 |
| $CNN_b$ | 0.73 | 0.74 | 0.75 | 0.75 | 0.50 |
| $CNN_c$ | 0.74 | 0.74 | 0.74 | 0.75 | 0.50 |
| $CNN_d$ | 0.72 | 0.73 | 0.72 | 0.74 | 0.48 |
| $CNN_e$ | 0.73 | 0.72 | 0.70 | 0.74 | 0.47 |

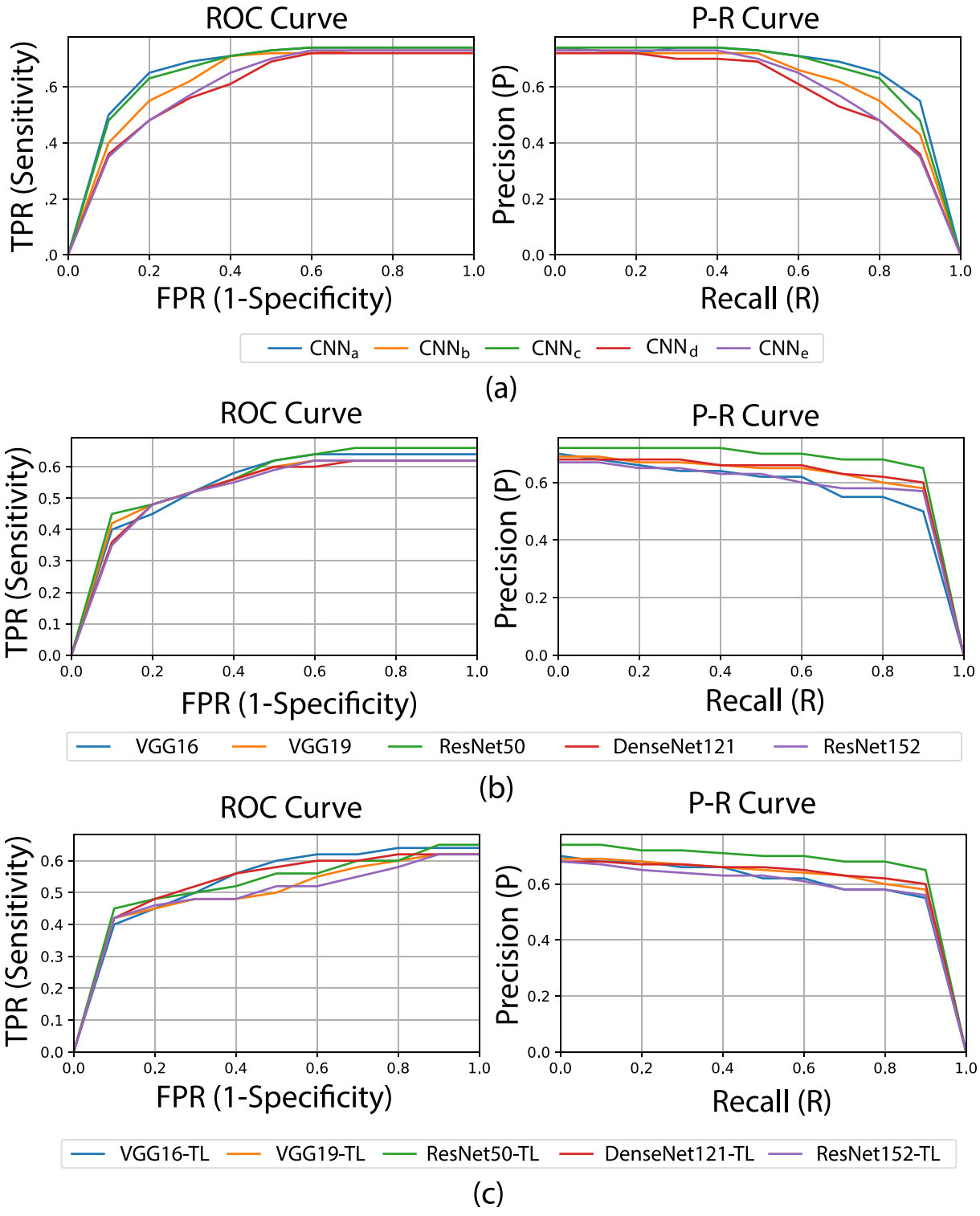


**Fig. 8.** ROC and P-R Curve of a) CNN Models generated from NAS. b) CNN Models without Transfer Learning. c) CNN Models with Transfer Learning.

can improve the prediction performance of individual models, as has been demonstrated in several earlier research (Garg and Gupta, 2008; Wei et al., 2018; Zhang et al., 2015). We also compared our stacking strategy with average scoring (AS) (Zhang et al., 2018; Wang et al., 2018; Zhang et al., 2020b) and majority voting (MV) (Wang et al., 2017; Chen and Jeong, 2009). Table 6 illustrates the performance of different ensemble methods. As demonstrated in Table 6, these two ensemble models did not significantly enhance prediction performance over the stacking strategy where the stacking strategy achieved the best performance, showing its ability for the prediction tasks. We can see that the stacking strategy has the best ACC 0.80 where average scoring and majority voting has 0.75 and 0.74 which is 5% and 4% lower than the stacking strategy respectively. Compared with the stacking technique, both the models (AS, MV) were decreased by 2%, 4%, 2% and 1%, 4%, 3% on SN, SP, and F-Score. The findings proved that the stacking strategy is the best suitable ensemble model for StackFBAs to predict brain abnormalities.

**Table 4**
Search space for the Neural Architecture Search.

| Hyperparameter | Range |
|---|---|
| number of Conv layers | ∈ [1 to 5] |
| units per layer | ∈ [16, 32, 64, 128, 256, 512] |
| dense units per layer | ∈ [64, 128, 256, 512, 1024] |
| number of dense layers | ∈ [1, 2, 3] |
| dropout | ∈ [0.0 to 0.8] |
| weight decay | ∈ [0.0, 1e-01, 1e-03, 1e-04] |
| learning rate | ∈ [1e-01, to 1e-05] |

**Table 5**
Performance measurements on different algorithms for level-1 model (meta-model).

| Algorithms | SN | SP | F-Score | ACC |
|---|---|---|---|---|
| LR | 0.74 | 0.75 | 0.75 | 0.76 |
| KNN | 0.76 | 0.78 | 0.78 | 0.80 |
| SVM | 0.75 | 0.74 | 0.74 | 0.74 |
| XGBoost | 0.72 | 0.71 | 0.71 | 0.72 |
| NB | 0.72 | 0.71 | 0.71 | 0.72 |

**Table 6**
Performance comparison of different ensemble strategies.

| Ensemble strategy | SN | SP | F-Score | ACC |
|---|---|---|---|---|
| Stacking | 0.76 | 0.78 | 0.78 | 0.80 |
| Average Scoring | 0.74 | 0.74 | 0.76 | 0.75 |
| Majority Voitng | 0.75 | 0.74 | 0.75 | 0.74 |

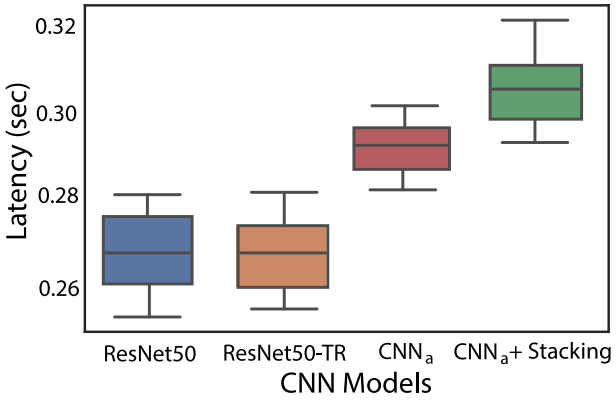


**Fig. 9.** Comparing inference latency for different models.

Moreover, we have considered four models (i.e., ResNet50 with and without transfer Learning, $CNN_a$, and $CNN_a$ + Stacking model) to compare the inference time. The Inference time of all models is shown in Fig. 9. We can see that the stacking model has higher latency due to its multiple levels.

## 4. Conclusion

We proposed an efficient algorithm named StackFBAs for diagnosing abnormalities in fetal brains. This study aimed to develop a new stacking framework with greedy-based neural architecture search (NAS) and federated learning techniques to classify brain abnormalities at an early stage. This approach was developed based on the different architectures of the baseline CNN classifiers, where the probabilistic score of these baseline models was combined to construct the final meta-model (KNN) using the stacking strategy. We considered the greedy-based NAS method to identify the best CNN architectures, which attained higher prediction results than popular ML classifiers. Moreover, the federated learning technique was utilized to protect sensitive fetal MRI data, combine results, and find common patterns from many users, making the model more robust for the privacy and security of user-sensitive data. Our framework has successfully diagnosed fetal brain abnormalities in MRIs of various planes and ages (from 16 weeks to 39 weeks). Furthermore, the proposed algorithms outperformed the individual models in most situations. To our knowledge, most of the past research has focused on segmenting preterm and neonatal MRI brain pictures. Few studies that looked at fetal MRI scans for classification used fetal scans in conjunction with newborn scans to quantify the deficit, known as SGA, in newborns rather than fetuses. The experimental results disclosed that our StackFBAs model beats the transfer learning model in categorization. This discovery will assist in developing guidance for future research by encouraging academics to start categorizing the prenatal brain. It will also assist physicians in making an accurate diagnosis, determining a suitable treatment schedule, and appropriately advising parents on how to cope with the abnormalities before the birth of the child.

## Funding

This research work was funded in part by the Natural Sciences and Engineering Research Council of Canada (NSERC).

## Author Contributions

This work was done in close collaboration among all the authors. The original idea was conceived by AAC, SMHM, and KKSH. The methodology was implemented by AAC, SMHM, and KA. Results were interpreted by AAC, SMHM, KKSH, KA, FAA, FMB, PL, and MAM. The manuscript was written and edited by AAC, SMHM, KKSH and KA. All authors have seen and approved the final version of the manuscript.

## Data Availability

The corresponding author can provide the data that were utilized to support the study upon request.

## Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgement

The authors extend their appreciation to the Deputyship for Research & Innovation, Ministry of Education in Saudi Arabia for funding this research work through the project number: IFP22UQU4170008DSR265.

## References

Abadi, M., Chu, A., Goodfellow, I., McMahan, H.B., Mironov, I., Talwar, K., Zhang, L., 2016. Deep learning with differential privacy. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, CCS '16, Association for Computing Machinery, New York, NY, USA, pp. 308–318. https://doi.org/10.1145/2976749.2978318.

Alansary, A., Lee, M., Keraudren, K., Kainz, B., Malamateniou, C., Rutherford, M., Hajnal, J.V., Glocker, B., Rueckert, D., 2015. Automatic brain localization in fetal mri using superpixel graphs. Lect. Notes Comput. Sci., 13–22 https://doi.org/10.1007/978-3-319-27929-9_2.

Aslan, M.F., Unlersen, M.F., Sabanci, K., Durdu, A., 2021. Cnn-based transfer learning–bilstm network: A novel approach for covid-19 infection detection. Appl. Soft Comput. 98, 106912.
Attallah, O., Gadelkarim, H., Sharkas, M.A., 2018. Detecting and classifying fetal brain abnormalities using machine learning techniques. In: 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), pp. 1371–1376. https://doi.org/10.1109/ICMLA.2018.00223.
Attallah, O., Sharkas, M.A., Gadelkarim, H., 2019. Fetal brain abnormality classification from mri images of different gestational age, Brain Sci. 9 (9). https://doi.org/10.3390/brainsci9090231. https://www.mdpi.com/2076-3425/9/9/231.
Bergstra, J., Bengio, Y., 2012. Random search for hyper-parameter optimization. J. Mach. Learn. Res. 13 (null), 281–305.
Charoenkwan, P., Chiangjong, W., Nantasenamat, C., Hasan, M.M., Manavalan, B., Shoombuatong, W., 2021. StackIL6: a stacking ensemble model for improving the prediction of IL-6 inducing peptides, Briefings Bioinformat. 22(6), bbab172. arXiv:https://academic.oup.com/bib/article-pdf/22/6/bbab172/41087892/bbab172.pdf, https://doi.org/10.1093/bib/bbab172.
Charoenkwan, P., Nantasenamat, C., Hasan, M.M., Moni, M.A., Lio', P., Manavalan, B., Shoombuatong, W., 2022. Stackdppiv: A novel computational approach for accurate prediction of dipeptidyl peptidase iv (dpp-iv) inhibitory peptides. Methods 204, 189–198. https://doi.org/10.1016/j.ymeth.2021.12.001. URL https://www.sciencedirect.com/science/article/pii/S1046202321002711.
Chen, X.-W., Jeong, J.C., 2009. Sequence-based prediction of protein interaction sites with an integrative method. Bioinformatics 25 (5), 585–591. https://doi.org/10.1093/bioinformatics/btp039. arXiv:https://academic.oup.com/bioinformatics/article-pdf/25/5/585/16892076/btp039.pdf.
Chen, L.-C., Collins, M.D., Zhu, Y., Papandreou, G., Zoph, B., Schroff, F., Adam, H., Shlens, J., 2018. Searching for efficient multi-scale architectures for dense image prediction. In: Proceedings of the 32nd International Conference on Neural Information Processing Systems, NIPS'18, Curran Associates Inc., Red Hook, NY, USA, , pp. 8713–8724.
Chowdhury, A.A., Hossen, M.A., Azam, M.A., Rahman, M.H., 2022. Deepqgho: Quantized greedy hyperparameter optimization in deep neural networks for on-the-fly learning. IEEE Access 10, 6407–6416. https://doi.org/10.1109/ACCESS.2022.3141781.
Durrani, N., Sajjad, H., Dalvi, F., 2021. How transfer learning impacts linguistic knowledge in deep nlp models?, arXiv preprint arXiv:2105.15179.
Feurer, M., Hutter, F., 2019. Hyperparameter optimization. In: Hutter2019, pp. 3–38.
Fiorentino, M.C., Villani, F.P., Cosmo, M.D., Frontoni, E., Moccia, S., 2022. A review on deep-learning algorithms for fetal ultrasound-image analysis, ArXiv abs/2201.12260.
Garg, A., Gupta, D., 2008. Virulentpred: A svm based prediction method for virulent proteins in bacterial pathogens. BMC Bioinformat. 9 (1). https://doi.org/10.1186/1471-2105-9-62.
Griffiths, P.D., Bradburn, M., Campbell, M.J., Cooper, C.L., Graham, R., Jarvis, D., Kilby, M.D., Mason, G., Mooney, C., Robson, S.C., et al., 2017. Use of mri in the diagnosis of fetal brain abnormalities in utero (meridian): A multicentre, prospective cohort study. The Lancet 389 (10068), 538–546. https://doi.org/10.1016/s0140-6736(16)31723-8.
Hosseini, M., Zekri, M., 2012. Review of medical image classification using the adaptive neuro-fuzzy inference system. J. Medical Sig. Sens. 2 (1), 49. https://doi.org/10.4103/2228-7477.108171.
Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., Attariyan, M., Gelly, S., 2019. Parameter-efficient transfer learning for nlp. In: International Conference on Machine Learning, PMLR, pp. 2790–2799.
Huang, R., Xie, W., Alison Noble, J., 2018. Vp-nets: Efficient automatic localization of key brain structures in 3d fetal neurosonography. Med. Image Anal. 47, 127–139. https://doi.org/10.1016/j.media.2018.04.004. URL https://www.sciencedirect.com/science/article/pii/S1361841518301920.
Huang, S., Li, X., Cheng, Z.-Q., Zhang, Z., 2018. A. Hauptmann, Gnas: A greedy neural architecture search method for multi-attribute learning. In: Proceedings of the 26th ACM international conference on Multimedia, pp. 2049–2057.
Hussain, M., Bird, J.J., Faria, D.R., 2018. A study on cnn transfer learning for image classification. In: UK Workshop on computational Intelligence. Springer, pp. 191–202.
Hutter, F., Kotthoff, L., Vanschoren, J. (Eds.), 2019. Automatic Machine Learning: Methods, Systems, Challenges. Springer.
Jia, B., Zhang, X., Liu, J., Zhang, Y., Huang, K., Liang, Y., 2022. Blockchain-enabled federated learning data protection aggregation scheme with differential privacy and homomorphic encryption in iiot. IEEE Trans. Industr. Inf. 18 (6), 4049–4058. https://doi.org/10.1109/TII.2021.3085960.
Kainz, B., Keraudren, K., Kyriakopoulou, V., Rutherford, M., Hajnal, J.V., Rueckert, D., 2014. Fast fully automatic brain detection in fetal mri using dense rotation invariant image descriptors. In: 2014 IEEE 11th International Symposium on Biomedical Imaging (ISBI), pp. 1230–1233. https://doi.org/10.1109/ISBI.2014.6868098.
Katorza, E., Gat, I., Duvdevani, N., Meller, N., Pardo, N., Barzilay, E., Achiron, R., 2018. Fetal brain anomalies detection during the first trimester: expanding the scope of antenatal sonography. J. Maternal-Fetal Neonatal Med. 31 (4), 506–512. https://doi.org/10.1080/14767058.2017.1289165, pMID: 28282781. arXiv: https://doi.org/10.1080/14767058.2017.1289165.
Lee, L.H., Bradburn, E., Papageorghiou, A.T., Noble, J.A., 2020. Calibrated bayesian neural networks to estimate gestational age and its uncertainty on fetal brain ultrasound images. In: Medical Ultrasound, and Preterm, Perinatal and Paediatric Image Analysis: First International Workshop, ASMUS 2020, and 5th International Workshop, PIPPI 2020, Held in Conjunction with MICCAI 2020, Lima, Peru, October 4–8, 2020, Proceedings, Springer-Verlag, Berlin, Heidelberg, pp. 13–22. https://doi.org/10.1007/978-3-030-60334-2_2.
Li, L., Jamieson, K., DeSalvo, G., Rostamizadeh, A., Talwalkar, A., 2017. Hyperband: A novel bandit-based approach to hyperparameter optimization. J. Mach. Learn. Res. 18 (1), 6765–6816.
Li, X., Grandvalet, Y., Davoine, F., Cheng, J., Cui, Y., Zhang, H., Belongie, S., Tsai, Y.-H., Yang, M.-H., 2020. Transfer learning in computer vision tasks: Remember where you come from. Image Vis. Comput. 93, 103853.
Li, Y., Li, J., Wang, Y., 2022. Privacy-preserving spatiotemporal scenario generation of renewable energies: A federated deep generative learning approach. IEEE Trans. Industr. Inf. 18 (4), 2310–2320. https://doi.org/10.1109/TII.2021.3098259.
[link]. http://radnet.bidmc.harvard.edu/fetalatlas/brain/brain.html.
Liu, Y., Sun, Y., Xue, B., Zhang, M., Yen, G.G., 2020. A survey on evolutionary neural architecture search, CoRR abs/2008.10937. arXiv:2008.10937. URL https://arxiv.org/abs/2008.10937.
Lu, Y., Huang, X., Dai, Y., Maharjan, S., Zhang, Y., 2020. Blockchain and federated learning for privacy-preserved data sharing in industrial iot. IEEE Trans. Industr. Inf. 16 (6), 4177–4186. https://doi.org/10.1109/TII.2019.2942190.
Makropoulos, A., Counsell, S.J., Rueckert, D., 2018. A review on automatic fetal and neonatal brain mri segmentation. NeuroImage 170, 231–248. https://doi.org/10.1016/j.neuroimage.2017.06.074.
Manavalan, B., Hasan, M.M., Basith, S., Gosu, V., Shin, T.-H., Lee, G., 2020. Empirical comparison and analysis of web-based dna n4-methylcytosine site prediction tools. Molecular Therapy - Nucleic Acids 22, 406–420. https://doi.org/10.1016/j.omtn.2020.09.010.
McMahan, H.B., Moore, E., Ramage, D., y Arcas, B.A., 2016. Federated learning of deep networks using model averaging, CoRR abs/1602.05629. arXiv:1602.05629. http://arxiv.org/abs/1602.05629.
Moser, F., Huang, R., Papageorghiou, A.T., Papiez, B.W., Namburete, A.I.L., 2019. Automated fetal brain extraction from clinical ultrasound volumes using 3d convolutional neural networks. MIUA, pp. 151–163. https://doi.org/10.1007/978-3-030-39343-4_13.
Namburete, A.I., Xie, W., Noble, J.A., 2017. Robust regression of brain maturation from 3d fetal neurosonography using crns. In: Fetal, Infant and Ophthalmic Medical Image Analysis. Springer, pp. 73–80.
Namburete, A.I., Xie, W., Yaqub, M., Zisserman, A., Noble, J.A., 2018. Fully-automated alignment of 3d fetal brain ultrasound to a canonical reference space using multi-task learning. Med. Image Anal. 46, 1–14. https://doi.org/10.1016/j.media.2018.02.006. URL https://www.sciencedirect.com/science/article/pii/S1361841518300306.
Ouahabi, A., Taleb-Ahmed, A., 2021. Deep learning for real-time semantic segmentation: Application in ultrasound imaging. Pattern Recogn. Lett. 144, 27–34. https://doi.org/10.1016/j.patrec.2021.01.010. URL https://www.sciencedirect.com/science/article/pii/S0167865521000234.
Pfitzner, B., Steckhan, N., Arnrich, B., 2021. Federated learning in a medical context: A systematic literature review. ACM Trans. Internet Technol. 21 (2). https://doi.org/10.1145/3412357.
Pham, H., Guan, M., Zoph, B., Le, Q., Dean, J., 2018. Efficient neural architecture search via parameters sharing. In: Dy, J., Krause A. (Eds.), Proceedings of the 35th International Conference on Machine Learning, Vol. 80 of Proceedings of Machine Learning Research, PMLR, pp. 4095–4104. https://proceedings.mlr.press/v80/pham18a.html.
Qiang, X., Zhou, C., Ye, X., Du, P.-F., Su, R., Wei, L., 2018. Cppred-fl: A sequence-based predictor for large-scale identification of cell-penetrating peptides by feature representation learning, Briefings Bioinformat. https://doi.org/10.1093/bib/bby091.
Real, E., Moore, S., Selle, A., Saxena, S., Suematsu, Y.L., Tan, J., Le, Q.V., Kurakin, A., 2017. Large-scale evolution of image classifiers. In: Precup, D., Teh, Y.W. (Eds.), Proceedings of the 34th International Conference on Machine Learning, Vol. 70 of Proceedings of Machine Learning Research, PMLR, pp. 2902–2911. https://proceedings.mlr.press/v70/real17a.html.
Real, E., Aggarwal, A., Huang, Y., Le, Q.V., 2018. Regularized evolution for image classifier architecture search, CoRR abs/1802.01548. arXiv:1802.01548. URL http://arxiv.org/abs/1802.01548.
Sanz-Cortes, M., Ratta, G.A., Figueras, F., Bonet-Carne, E., Padilla, N., Arranz, A., Bargallo, N., Gratacos, E., 2013. Automatic quantitative mri texture analysis in small-for-gestational-age fetuses discriminates abnormal neonatal neurobehavior. PLOS ONE 8 (7), 1–7. https://doi.org/10.1371/journal.pone.0069595.
Sanz-Cortés, M., Figueras, F., Bonet-Carne, E., Padilla, N., Tenorio, V., Bargalló, N., Amat-Roldan, I., Gratacós, E., 2013. Fetal brain mri texture analysis identifies different microstructural patterns in adequate and small for gestational age fetuses at term. Fetal Diagn. Ther. 33 (2), 122–129. https://doi.org/10.1159/000346566.
Shen, Y.-T., Chen, L., Yue, W.-W., Xu, H.-X., 2021. Artificial intelligence in ultrasound. Eur. J. Radiol. 139, 109717. https://doi.org/10.1016/j.ejrad.2021.109717. URL https://www.sciencedirect.com/science/article/pii/S0720048X21001972.
Singh, V., Sridar, P., Kim, J., Nanan, R., Poornima, N., Priya, S., Reddy, G.S., Chandrasekaran, S., Krishnakumar, R., 2021. Semantic segmentation of cerebellum in 2d fetal ultrasound brain images using convolutional neural networks. IEEE Access 9, 85864–85873. https://doi.org/10.1109/ACCESS.2021.3088946.
Snoek, J., Larochelle, H., Adams, R.P., 2012. Practical bayesian optimization of machine learning algorithms. In: Pereira, F., Burges, C., Bottou, L., Weinberger, K.

(Eds.), Advances in Neural Information Processing Systems, vol. 25. Curran Associates, Inc..

Sun, W., Trevor, B., 2018. A stacking ensemble learning framework for annual river ice breakup dates. J. Hydrol. 561, 636–650. https://doi.org/10.1016/j.jhydrol.2018.04.008. URL https://www.sciencedirect.com/science/article/pii/S0022169418302506.

Vanschoren, J., 2019. Meta-learning. In: Hutter2019, pp. 39–68.

van Sloun, R.J.G., Cohen, R., Eldar, Y.C., 2020. Deep learning in ultrasound imaging. Proc. IEEE 108 (1), 11–29. https://doi.org/10.1109/JPROC.2019.2932116.

Venturini, L., Papageorghiou, A.T., Noble, J.A., Namburete, A.I.L., 2020. Multi-task cnn for structural semantic segmentation in 3d fetal brain ultrasound. In: Zheng, Y., Williams, B.M., Chen, K. (Eds.), Medical Image Understanding and Analysis. Springer International Publishing, Cham, pp. 164–173.

Wang, J., Yang, B., An, Y., Marquez-Lago, T., Leier, A., Wilksch, J., Hong, Q., Zhang, Y., Hayashida, M., Akutsu, T., Webb, G.I., Strugnell, R.A., Song, J., Lithgow, T., 2017. Systematic analysis and prediction of type IV secreted effector proteins by machine learning approaches. Brief. Bioinformat. 20 (3), 931–951. https://doi.org/10.1093/bib/bbx164. arXiv:https://academic.oup.com/bib/article-pdf/20/3/931/28847321/bbx164.pdf.

Wang, S., Hua, Y., Cao, Y., Song, T., Xue, Z., Gong, X., Wang, G., Ma, R., Guan, H., 2018. Deep learning based fetal middle cerebral artery segmentation in large-scale ultrasound images. 2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), pp. 532–539. https://doi.org/10.1109/BIBM.2018.8621510.

Wang, J., Yang, B., Leier, A., Marquez-Lago, T.T., Hayashida, M., Rocker, A., Zhang, Y., Akutsu, T., Chou, K.-C., Strugnell, R.A., Song, J., Lithgow, T., 2018. Bastion6: a bioinformatics approach for accurate prediction of type VI secreted effectors. Bioinformatics 34 (15), 2546–2555. https://doi.org/10.1093/bioinformatics/bty155. arXiv:https://academic.oup.com/bioinformatics/article-pdf/34/15/2546/25230727/bty155.pdf.

Wei, L., Zhou, C., Chen, H., Song, J., Su, R., 2018. Acpred-fl: A sequence-based predictor using effective feature representation to improve the prediction of anti-cancer peptides. Bioinformatics. https://doi.org/10.1093/bioinformatics/bty451.

Wei, L., Ye, X., Xue, Y., Sakurai, T., Wei, L., 2021. ATSE: a peptide toxicity predictor by exploiting structural and evolutionary information based on graph neural network and attention mechanism. Briefings Bioinformat. 22 (5), bbab041. arXiv:https://academic.oup.com/bib/article-pdf/22/5/bbab041/40260952/bbab041.pdf, https://doi.org/10.1093/bib/bbab041.

Whatmough, P.N., Zhou, C., Hansen, P., Venkataramanaiah, S.K., Seo, J.-S., Mattina, M., 2019. Fixynn: Efficient hardware for mobile computer vision via transfer learning, arXiv preprint arXiv:1902.11128.

Wolpert, D.H., 1992. Stacked generalization. Neural Networks 5 (2), 241–259. https://doi.org/10.1016/S0893-6080(05)80023-1. URL https://www.sciencedirect.com/science/article/pii/S0893608005800231.

Wu, Y., Shen, K., Chen, Z., Wu, J., 2020. Automatic measurement of fetal cavum septum pellucidum from ultrasound images using deep attention network. In: 2020 IEEE International Conference on Image Processing (ICIP), pp. 2511–2515. https://doi.org/10.1109/ICIP40778.2020.9191002.

Wyburd, M., Jenkinson, M., Namburete, A., 2020. Cortical plate segmentation using cnns in 3d fetal ultrasound. In: Papiez, B., Namburete, A., Yaqub, M., Noble, J., Yaqub, M. (Eds.), Medical Image Understanding and Analysis - 24th Annual Conference, MIUA 2020, Proceedings, Communications in Computer and Information Science. Springer, pp. 56–68. https://doi.org/10.1007/978-3-030-52791-4_5.

Wyburd, M.K., Hesse, L.S., Aliasi, M., Jenkinson, M., Papageorghiou, A.T., Haak, M.C., Namburete, A.I.L., 2021. Assessment of regional cortical development through fissure based gestational age estimation in 3d fetal ultrasound. In: Sudre, C.H., Licandro, R., Baumgartner, C., Melbourne, A., Dalca, A., Hutter, J., Tanno, R., Abaci Turk, E., Van Leemput, K., Torrents Barrena, J., Wells, W.M., Macgowan, C. (Eds.), Uncertainty for Safe Utilization of Machine Learning in Medical Imaging, and Perinatal Imaging, Placental and Preterm Image Analysis, Springer International Publishing, Cham, pp. 242–252.

Xie, R., Li, J., Wang, J., Dai, W., Leier, A., Marquez-Lago, T.T., Akutsu, T., Lithgow, T., Song, J., Zhang, Y., et al., 2020. Deepvf: A deep learning-based hybrid framework for identifying virulence factors using the stacking strategy. Brief. Bioinformat. 22 (3). https://doi.org/10.1093/bib/bbaa125.

Xiong, Y., Wang, Q., Yang, J., Zhu, X., Wei, D.-Q., 2018. Predt4se-stack: Prediction of bacterial type iv secreted effectors from protein sequences using a stacked ensemble method. Front. Microbiol. 9. https://doi.org/10.3389/fmicb.2018.02571. https://www.frontiersin.org/article/10.3389/fmicb.2018.02571.

Xu, R., Baracaldo, N., Joshi, J., 2021. Privacy-preserving machine learning: Methods, challenges and directions, CoRR abs/2108.04417. arXiv:2108.04417. https://arxiv.org/abs/2108.04417.

Yang, Q., Liu, Y., Cheng, Y., Kang, Y., Chen, T., Yu, H., 2019. Federated learning, synthesis lectures on artificial intelligence and machine. Learning 13 (3), 1–207. https://doi.org/10.2200/s00960ed2v01y201910aim043.

Zaffino, P., Moccia, S., De Momi, E., Spadea, M.F., 2020. A review on advances in intra-operative imaging for surgery and therapy: Imagining the operating room of the future. Ann. Biomed. Eng. 48 (8), 2171–2191. https://doi.org/10.1007/s10439-020-02553-6.

Zhang, Y.P., Zou, Q., 2020. PPTPP: a novel therapeutic peptide prediction method using physicochemical property encoding and adaptive feature representation learning. Bioinformatics 36 (13), 3982–3987. https://doi.org/10.1093/bioinformatics/btaa275. arXiv:https://academic.oup.com/bioinformatics/article-pdf/36/13/3982/33459034/btaa275.pdf.

Zhang, Z., Schwartz, S., Wagner, L., Miller, W., 2000. A greedy algorithm for aligning dna sequences. J. Comput. Biol. 7 (1–2), 203–214, pMID: 10890397.

Zhang, L., Zhang, C., Gao, R., Yang, R., 2015. An ensemble method to distinguish bacteriophage virion from non-virion proteins based on protein sequence characteristics. Int. J. Mol. Sci. 16 (9), 21734–21758. https://doi.org/10.3390/ijms160921734. URL https://www.mdpi.com/1422-0067/16/9/21734.

Zhang, Y., Xie, R., Wang, J., Leier, A., Marquez-Lago, T.T., Akutsu, T., Webb, G.I., Chou, K.-C., Song, J., 2018. Computational analysis and prediction of lysine malonylation sites by exploiting informative features in an integrative machine-learning framework. Brief. Bioinformat. 20 (6), 2185–2199. https://doi.org/10.1093/bib/bby079. arXiv:https://academic.oup.com/bib/article-pdf/20/6/2185/31789340/bby079.pdf.

Zhang, L., Zhang, J., Li, Z., Song, Y., 2020a. A multiple-channel and atrous convolution network for ultrasound image segmentation. Med. Phys. 47 (12), 6270–6285. https://doi.org/10.1002/mp.14512.

Zhang, Y., Yu, S., Xie, R., Li, J., Leier, A., Marquez-Lago, T.T., Akutsu, T., Smith, A.I., Ge, Z., Wang, J., et al., 2020b. Pengaroo, a combined gradient boosting and ensemble learning framework for predicting non-classical secreted proteins. Bioinformatics 36 (3), 704–712.

Zhang, X., Fang, F., Wang, J., 2021. Probabilistic solar irradiation forecasting based on variational bayesian inference with secure federated learning. IEEE Trans. Industr. Inf. 17 (11), 7849–7859. https://doi.org/10.1109/TII.2020.3035807.

Zoph, B., Vasudevan, V., Shlens, J., Le, Q.V., 2017. Learning transferable architectures for scalable image recognition, CoRR abs/1707.07012. arXiv:1707.07012. http://arxiv.org/abs/1707.07012.

Zoph, B., Vasudevan, V., Shlens, J., Le, Q.V., 2018. Learning transferable architectures for scalable image recognition. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8697–8710. https://doi.org/10.1109/CVPR.2018.00907.