**Implementation of Convolutional Neural Networks for Classifying Fetal Heart Rate (FHR) Tracings**

Sarah Ninan

**Implementation of Convolutional Neural Networks for Classifying Fetal Heart Rate (FHR) Tracings**

**Abstract**

During labor, obstetricians visually examine fetal heart rate (FHR) signals to predict fetal health complications through signal features. However, studies have determined the presence of inter- and intra-observer disagreements regarding classification of FHR patterns between many obstetricians. Machine learning (ML) algorithms can potentially assess these signals with greater accuracy and uniformity. I designed a convolutional neural network (CNN) for FHR classification with 5520 images and examined the effectiveness of my CNN with this low image input. 552 FHR signals from an open-source CTU-CHB database were preprocessed. Then, with the pH threshold set at 7.15, the signals were labeled as 'abnormal' or ‘normal,’ based on proximity to the threshold. Each signal was segmented to create 5520 images. Finally, the signals were converted into 2D images and inputted into a five-layer CNN. I evaluated various k-fold validation tests for cross entropy loss and classification accuracy. During the first five-fold validation test, highest mean accuracy percent returned was 81.108%, with the lowest cross entropy loss range (≤150 nats) observed out of all k-fold tests. While the CNN was functional despite the small input size, it had significantly less classification accuracy than four other ML models, the highest of which scored up to 93.24% accuracy. Findings suggest that smaller input size do not allow for high levels of classification accuracy. Additional CNN layers or signal segmentation should be utilized in future research to further examine the use of deep learning in FHR signal analysis.

**Implementation of Convolutional Neural Networks for Classifying Fetal Heart Rate (FHR) Tracings**

**1.Introduction**

**1.1 Rationale**

Within recent years, the advancement of obstetrics technology and electronic fetal monitoring (EFM) has brought about a demand for machine learning (ML) use in the delivery room (Cömert & Kocamaz, 2017; Yu et al., 2017; Jehkonen, 2021; Tomlinson et al., 2020). Fetuses are most frequently screened intrapartum using signals via a cardiotocographic machine (CTG) to help determine fetal heart rate (FHR) throughout the course of pregnancy and delivery (German Society of Gynecology and Obstetrics (DGGG) et al., 2014). Visual readings of FHR through CTG signal outputs performed by obstetricians help predict complications such as fetal hypoxia through signal features (Tarvonen et al., 2020a). However, previous studies have determined the presence of frequent discrepancies within CTG signal readings done by various obstetricians, including inter- and intra-observer disagreements regarding classification of abnormalities in FHR patterns (Chen, 2016; Jehkonen, 2021). Misinterpretations or disagreements of abnormal FHR patterns can prevent the necessary early detection of fetal hypoxia, and, subsequently, result in complications during labor or with the newborn’s health post-delivery (Jehkonen, 2021).

The use of ML algorithms while reading EFM data such as FHR signals, have been shown to improve signal analysis accuracy and interpretation, particularly ML’s contribution to lower error rates with signal readings (Chen, 2016; Yu et al., 2017; Zhao et al., 2019b). However, due to disruptions in the internal and external environment of the womb, artifacts, or portions of the signal with no data or inaccurately recorded data, may occur during both invasive and non-invasive data collection (Yu et al, 2017). This could obscure true signals and possibly create more difficult circumstances for accurate AI readings (Polnaszek et al., 2020). For example, zigzag and saltatory patterns often have difficulty being identified by most ML algorithms due to abnormally high FHR variability, which is a defining trait for these patterns but often interpreted as artifacts by the machine (Jehkonen, 2021). However, deep learning has been able to identify preprocessed signals and patterns with higher levels of accuracy compared to traditional ML (Chen, 2016). Deep learning, specifically neural networks, utilize pattern recognition software to aid in the detection of pattern breaks and irregularities (Sadiq et al., 2019). Neural networks can more accurately classify small differences between similar looking images without necessarily evaluating irrelevant inputs by utilizing many layers, which contributes to their high levels of accuracy (Chen, 2016; Dargan et al., 2019). For my research, I designed a convolutional neural network (CNN) for FHR classification with 5520 images and examined the effectiveness of my CNN with this low image input.

In this study, I present the protocol for: preprocessing FHR signals to remove given artifacts, designing and testing my CNN and accuracy analysis. This study is necessary for the advancement of automated EFM technology, as it provides a means for more accurately analyzing signals, such as FHR or uterine activity signals, to reduce post-delivery complications and conditions of newborns and their mothers.

**1.2 Background**

*1.2.1 FHR Signals*

Modern day obstetricians and gynecologists visually assess FHR signals based on their expertise and published guidelines from the 2008 National Institute of Child Health and Human Development workshop report on EFM (Macones et al., 2008). Visual assessments of an FHR signal’s amplitude and variability provide a fetal health reading that can determine whether the baby will be born with any underlying cardiac or pulmonary conditions such as fetal asphyxia or hypoxia (Yu et. al, 2017).

In order to accurately assess fetal health during pregnancy, obstetricians often turn to cardiotocographic readings (CTG) or other forms of electronic fetal monitoring (EFM) such as using sensory technology to examine the biophysical profile of the fetus (Matonia et al., 2020). CTG is currently one of the most accepted methods of recording FHR signals due to their cost efficient, non-invasive nature (Elmansouri et al., 2014). However, a variety of factors including maternal obesity, sudden movement by the mother, or actions such as contractions, sniffs, swallows, or movement of the machine or fetus, can interfere with both internal and external FHR monitoring (Macones et al., 2008; Nunes et al., 2014). For these reasons, use of CTG machines often produce FHR signals that are more susceptible to signal loss and signal artifacts, making them harder to read and accurately examine (Nunes et al., 2014).

Due to these discrepancies, a recent study has shown there have been high levels of inter- and intra-level variability between obstetricians observing FHR samples, with the overall proportion of agreement between clinician only reaching 48% (Hruban et al., 2015). This signifies high amounts of variability of judgement amongst said clinicians. Misjudgment of the fetal state due to inaccurate assessments of FHR signals can have fatal effects on both the mother and the fetus (Jehkonen, 2021). For the mother, these assessments could lead to unnecessary cesarean deliveries, which is known to have more long-term complications compared to vaginal delivery (Chen, 2016, Zhao et al., 2018, Jehkonen, 2021). For fetuses, certain patterns in FHR signals that go undetected by clinicians can lead to health complications such as fetal hypoxia, or fetal asphyxia, which can result in irreversible side effects, including death, if left undetected (Chen, 2016). These findings demonstrate a need for uniform assessments from FHR signals to prevent further disagreements and provide increased accuracy in the reading of FHR signals.

*1.2.2 Machine Learning*

In the past two decades, studies have examined the benefits of using ML techniques to more accurately analyze FHR signals (Love, 2002; Cömert & Kocamaz, 2017; Jehkonen, 2021). Typically, supervised ML is used to classify data as "abnormal,” "healthy," or, sometimes, "indeterminate" based on thresholds established from the clinical information regarding newborn health (Love, 2002). Commonly supervised ML algorithms include Support Vector Machine (SVM), Extreme Learning Machine (ELM), and Random Forest (RF) (Cömert & Kocamaz, 2017). Out of the three algorithms mentioned, SVM, a kernel-based ML method, tends to produce the most accurate assessments of FHR signals, with one study finding that SVM had a 99.21 sensitivity and 97.02 specificity levels on the confusion matrix, which were slightly higher compared to ELM and RF levels (Cömert & Kocamaz, 2017). However, it is hard to determine the validity of these results, as they have yet to be reproduced with such a high classification accuracy. Additionally, while typical supervised ML methods use linear classifiers when analyzing data, linear classifiers have limitations when it comes to distinguishing labeled data that could present similar features, particularly at the pixel level, when it comes to the examination of 2-D images (Bengio & Lecun, 2007).

*1.2.3 Deep Learning*

Deep learning is a subset of ML that concerns itself specifically with the use of neural networks to classify 2-D images (Dargan et al., 2019). A neural network is one form of deep learning algorithms that has become more popular in recent years in the classification of FHR signals (Cömert & Kocamaz, 2016; Zhao et al., 2019b). Neural networks are composed of several layers of ML modules which contain non-linear classifiers (Bengio & Lecun, 2007). Due to their many layers and non-linear elements, neural networks can more accurately distinguish small differences between similar images without necessarily processing irrelevant inputs, oftentimes resulting in more accurate classifications compared to traditional ML methods (Chen, 2016; Dargan et al., 2019). CNNs use several convolutional and pooling layers to identify and recombine features for classification, which is then followed by an output layer that uses back-propagation algorithms for feature recognition (Dargan et al., 2019). Previous studies have examined the effectiveness of utilizing an 8-layer CNN in the classification of FHR signals as abnormal or normal, and found a 98.4% level of accuracy on the fourth set tested (Chen, 2016; Zhao et al., 2019b). However, these results, similar to the aforementioned Cömert & Kocamaz (2017) study which presented the classification of signals using EFM and RF, have not been reproduced with such high achieving results. Also, 8-layers are inefficient to execute on a PC with only 5520 samples, as many of the layers are dropout layers which randomly remove some images for increased randomization. This is necessary for high volumes of images but not for only 5520. Therefore, this study will test the use of a 5-layer CNN in the classification of FHR signals.

**1.3 Objectives/Hypothesis**

*1.3.1 Objectives*

The primary objective of this study was to design a CNN for FHR classification with 5520 images and examine the effectiveness of my CNN with this low image input.

*1.3.2 Hypotheses*

The primary expected outcomes for this study include the following: 1) the 5-layer CNN will produce classification of FHR signals with over 75% accuracy, and 2) the 5-layer CNN will rank significantly higher in performance compared to other ML methods used to classify FHR signals.

**2. Methodology**

*\*All methods were conducted by Author unless otherwise stated*

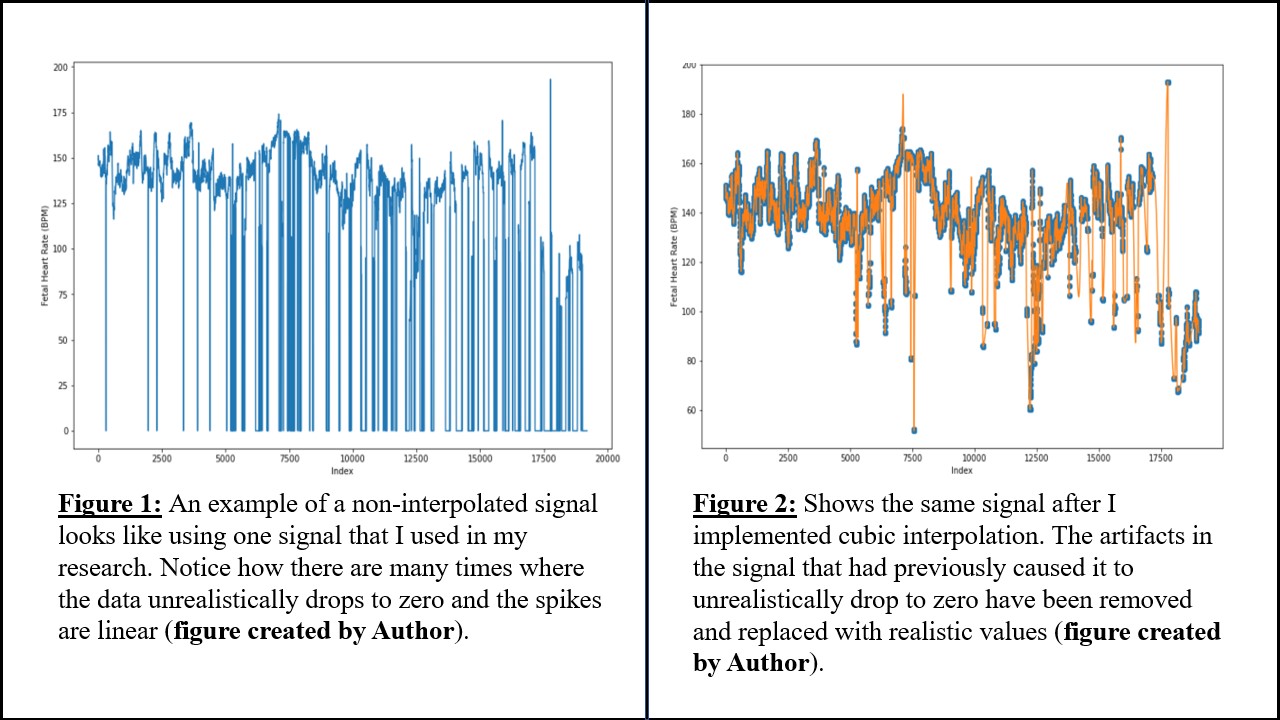
**2.1 CTU-CHB Database**

I used a CTG database called CTU-CHB provided by the Czech Technical University which supplied the FHR tracings for 552 different patients and the accompanying clinical information (Chudáček et al., 2014). The database contained the 552 CTG recordings of both the FHR and uterine contractions (UC) signal data taken at a maximum of 90 minutes before delivery time at the Czech Technical University hospital, but this study only utilized the FHR signals (Chudáček et al., 2014). All signals from this database were sampled at 4 Hz (Chudáček et al., 2014).. CTU-CHB is publicly available, de-identified and anonymous. Clinical information regarding the Apgar scores at one minute and five minutes from delivery, blood base deficit (BDecf), and pH were provided by the database (Chudáček et al., 2014). I later used the pH values later in the methodology to label the data as abnormal or normal.

**2.2 Preprocessing**

Preprocessing the FHR signal is a necessary prerequisite to use any sort of ML algorithm to classify FHR signals. When a CTG machine records FHR signals, portions of the signal can be warped or disrupted due to external factors like the mother’s movement, or internal factors like maternal obesity or the probing device position, which forms noise within the signal (Nunes et al., 2014). Noise is when portions of the raw FHR signal appear as sudden jumps in the data or non-existent data plot points listed as "nans,” known as artifacts, and do not correctly reflect the condition of the true FHR signal (Yu et al., 2017). In order to remove this noise, I conducted preprocessing.

In a continuous function, when new samples are created between existing samples or nodes, interpolation allows the computer to consider nearby points to estimate an accurate value to create a new node (Ahmad & Deeba, 2017). The use of cubic spline interpolation in preprocessing ensures that a new sample is created with limited error and preserves the values of the first and second derivative of the function in the process (Ahmad & Deeba, 2017). The final plot shows curved edges between points rather than straight lines which allows for a more accurate representation of the FHR signal (Ahmad & Deeba, 2017). Therefore, in my preprocessing procedure, I removed those 'nan' values by using code that implemented cubic spline interpolation to interpolate the missing points. An example of what an FHR signal looks like before and after interpolation can be seen in **Figure 1** and **Figure 2**, which were images I printed from the first FHR signal provided by the database that I preprocessed. **Figure 1** shows what the raw signal had looked like, and **Figure 2** demonstrates the preprocessed signal, or the signal after interpolation.



**2.3 Segmentation**

After the data was interpolated, I prepared to segment 552 interpolated signals. Each signal starts, at most, 90 minutes prior to delivery (Chudáček et al., 2014). However, CNNs are typically designed to input tens of thousands of images for further cross validation and accuracy (Chen, 2016). The present study examined if a CNN would still successfully classify FHR signals with only a few thousand images. Therefore, in order to later extract thousands of images from 552 samples, I segmented the data by isolating the last ten minutes from each signal and then separated each ten-minute segment by minute. I selected the last ten minutes because I used the umbilical artery (UA) pH value provided by the clinical information of each patient given by the database as the main identifying factor when labeling the data. pH value was used as an assessment of fetal health as it is a biomarker of the biophysical state of the neonate directly following delivery. Since pH is recorded immediately after delivery and is used to assess the health of the baby, the most accurate indicator of fetal health would be during the last ten minutes, as pH values may vary prior to the last ten minutes of labor (Chudáček et al., 2014). I had 5520 segmented pieces of the original 552 signals that I later converted into images.

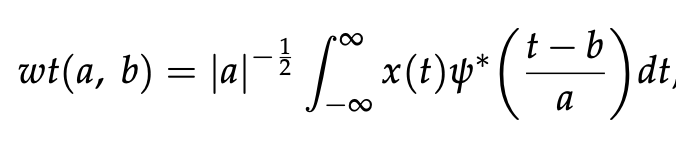
**2.4 Labeling Data**

Since a CNN is a type of supervised ML, data needs to be labeled prior to testing to analyze if the neural network is capable of classifying it as 'abnormal' or 'normal' (Dargan et al., 2019). As mentioned prior, I utilized pH values provided by the CTU-CHB database to label the data as 'normal' or 'abnormal' (unhealthy). After careful consideration of pH thresholds used in similar studies and examining the pH values of unhealthy and healthy fetuses as recorded in the clinical info provided by the database, I decided to use the pH value 7.15 as my threshold (Zhao et al., 2019b; Jehkonen, 2021). I created looped code that labeled signals with pH values of greater than or equal to 7.15 as 'normal', whilst signals with a pH value of less than 7.15 were labeled as 'abnormal' (Vayssiere et al., 2007). Later, the CNN I designed was tested to examine if it could correctly classify the "unhealthy" and "healthy" images.

**2.5 1-D Signal to 2-D image Conversion**

Though some studies exist where CNN was able to use 1-D signals as inputs, CNN typically requires an input of tens of thousands of 2D image inputs for accurate analysis of data (Zhao et al., 2019a). The process of converting 1D signals to 2D images is time-consuming and taxing on most accessible computers (Chen, 2016). A substantial number of images is required as CNNs identify edges of 2D images during the training process (Fotiadou & Vullings, 2020). For this reason, I found it necessary to convert the 5520 1D signal segments into 2D images.

Previous research has demonstrated that the continuous wavelet transform (CWT) can be used to undergo the conversion of 1D signals to 2D images (Aguiar-Conraria & Soares, 2014). CWTs are a form of 2D signal representations via wavelets, where the CWT occurs as a function of the scaling and shifting parameters (Yoo & Baek, 2018). More specifically, the CWT of a function can be obtained by using **Equation 1** to rescale and shift a mother function (Yoo & Baek, 2018). In many cases, in order to convert a time series signal into its time frequency counterpart, the use of a Fourier transform can be used (Aguiar-Conraria & Soares, 2014). However, the utilization of Fourier transform is not applicable to signals that are not "time stationary", i.e., when the signals’ features demonstrated are not affected by changes in time (Aguiar-Conraria & Soares, 2014). Though a CWT tends to have greater computational time, it provides the opportunity to establish exact parameters for scaling and position, thereby, providing accurate representation of signal features in the time-frequency plot (Reddy et al., 2014). For those reasons, I used a CWT to produce 5520 images that were later put into the CNN. After the data was labeled, I later programmed code in my CNN that would randomly choose of the data to be a part of the training set and isolate the remaining data to be a part of the testing set. The proportion of training to testing would change depending on the k-fold tested in each trial.

**Equation 1**: Continuous Wavelet Transformation can be utilized on a 1D signal using the above equation (Yoo & Baek, 2018). The *wt(a,b)* represents the wavelet coefficient, *a* is the scaling parameter, *b*is the shifting parameter, *x(t)* represents the given signal and *ψ* represents the mother function.

**2.6 CNN Architecture**

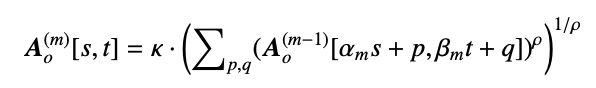
In order to determine if a CNN can accurately classify FHR signals given limited input, I designed a CNN using Python on Jupyter Notebook. A CNN requires the use of multiple layers with specific functions in order to carry out its classification.

*2.5.1 Layer 1: Input Convolutional Layer*

The first layer I used was a convolutional layer that allowed for the input of images. In a convolutional layer, the necessary parameters are given to the machine that pertains to what the computer is to learn (Ke et al., 2018). In the present study, the 2D convolutional layer has 32 filters. Filters are used to find the scalar product of the input data by node and the 2D weights that compose the filter (Ke et al., 2018). This scalar product provides information regarding key features presented in the input images, such as position of certain pixels relative to other pixels necessary to identify certain patterns in the image, which could help with identification (Sinam et al., 2020). The filter allows the CNN to first obtain a single output known as a feature map through its linearity, which can then be run through a nonlinear layer to obtain more complex classifications (Ke et al., 2018). 32 layers were chosen due to time efficiency and each filter was 3x3 in size to analyze rows, columns and the single channel of the 2D images given as input.

*2.5.2 Layer 2: Max Pooling Layer*

The second layer I coded into my CNN is a max pooling layer. Pooling layers are typically used in CNNs in order to bind convolutional layers together by reducing the parameters needed to complete computation and allowing the CNN to focus on larger areas of the input image (Balocco et al., 2020). Different kinds of pooling layers demonstrate the varying values as shown in **Equation 1** (Balocco et al., 2020). I chose to use a max pooling layer over other types of pooling layers because in max pooling layers the maximum number of values in regions of the feature map represent the value, allowing for the identification of the most prominent features presented in the image rather than a less precise estimate of the average presence of the feature. After understanding what pooling layers are used for, I decided to add a max pooling layer into my CNN code. I chose filter sizes that were assigned as 2x2, as they needed to be smaller than the size of the filters in the first layer in order to properly assess the feature map.

**Equation 2:** The equation shown above represents c1qhannel-wise operation performed by a typical pooling layer. represents the image inputted. In max pooling layers, (ρ = ∞, κ = 1), and (Balocco et al., 2020).

*2.5.3 Layer 3: Flattening Layer*

Previously established research demonstrated that reducing the input down to a single vector allowed the following activation layer to move the inputs into the last layer for final classification (Balocco et al., 2020). Therefore, in order to flatten the inputs into a single dimensional channel without reducing the batch size of my input, I programmed a flattening layer in between the max pooling layer and the dense layer.

*2.5.4 Layer 4: Activation Layer*

All layers utilize the rectification linear activation function (ReLu). ReLu is a linear activation function which works to account for all values > 0 as linear values while setting negative values as 0 (Agarap, 2019). However, I still required one layer to gather information of the Bernoulli probability distribution of the inputted data, with which the computer can grasp the likelihood of each node belonging to one of the two classification categories (in the case of my study, either normal or abnormal) (Freund et al., 2010). For this, I coded a layer that activated the Softmax activation function, which carries out the Bernoulli probability distribution by utilizing the ReLu linear activation function.

*2.5.5 Layer 5: Fully Connected Layer*

The fully connected layer is the final layer I coded into my CNN in order to undergo the final classification of images at the end of the neural network. Smaller, more detailed features can be identified in this layer as information from previous layers accumulate here and helps the computer finalize how to identify the images (as either normal or abnormal) (Zhao et al., 2019a).

Some studies choose to include a dropout layer as well, which can prevent a model from overfitting in the face of large amounts of input images. However, since my study utilized 5520 images, a dropout layer was omitted.

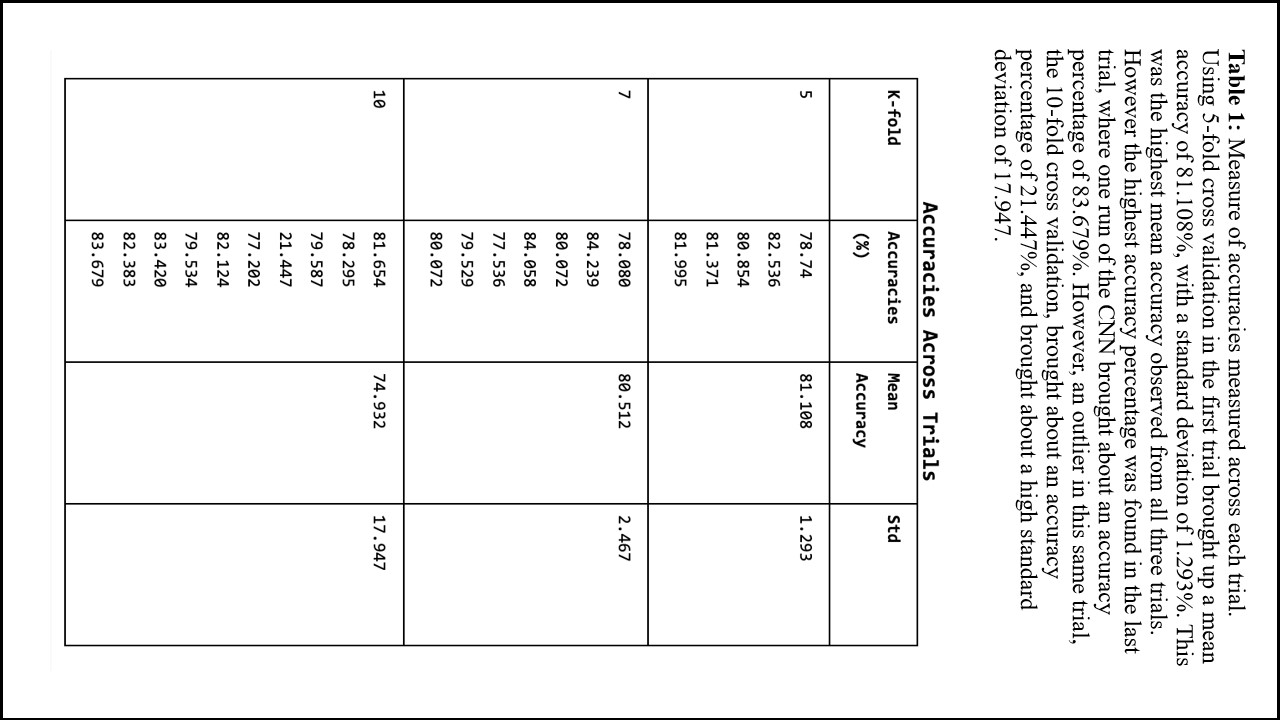
**2.7 Performance Evaluation/Data Analysis**

I evaluated the CNN performance by programming my CNN to undergo k-fold evaluation. This means that in each trial, I programmed the CNN to run 'k' number of times, with the data shuffled within their designated sets each time, and the CNN was programmed to evaluate the performance for classification accuracy and entropy loss each time. Cross entropy measures essentially the degree to which the neural network made incorrect predictions. A higher cross entropy loss demonstrates a high degree of incorrect classification (Zhao et al., 2018). The average accuracy measures the opposite, demonstrating the extent to which the CNN was able to correctly classify the given data. The CNN was run with 5-fold, 7-fold, and 10-fold cross validation and the results are demonstrated below. The mean accuracy and the standard deviation were calculated and plotted below. Plots were created on Jupyter Notebook.

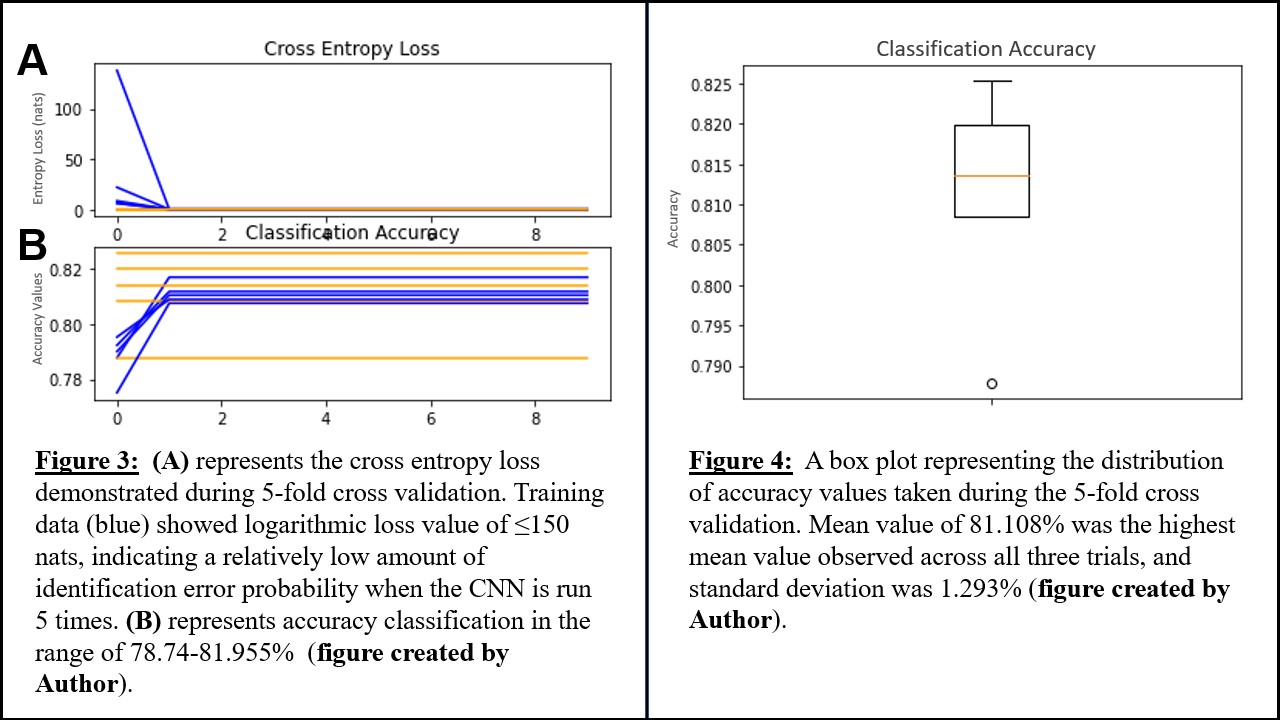
**3. Results**

*\*All results were obtained by the Author unless otherwise stated.*

**3.1 Five-Fold Cross Validation**

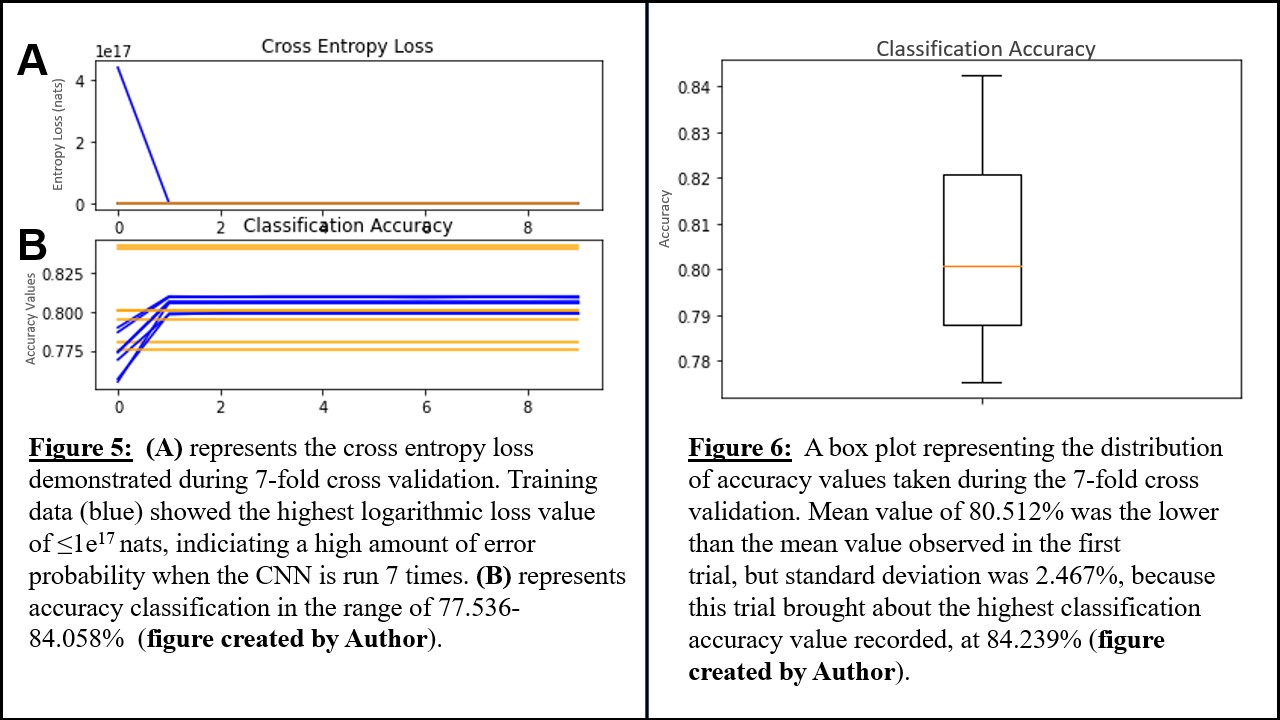
 In the five-fold cross validation, accuracy values were given as 78.74%, 82.536%, 80.854%, 81.371%, 81.995%, with the mean accuracy presented as 81.108 **(Table 1)**. In this trial, plotted entropy loss primarily converged towards 0 at classification of 1, indicating a decreasing logarithmic loss function as shown in **Figure 3a.** However, the training data (represented by the blue in **Figure 3a**) shows higher levels of loss prior to convergence, primarily with the first fold, compared to the testing dataset (represented by the orange in **Figure 3a**). The range of loss demonstrated with the five-fold CNN is ≤150 nats, which is significantly smaller compared to the following trials.

The plot generated demonstrating the classification accuracy (**Figure 3b**) shows initial convergence towards values ranging from 78% to approximately 81% for the training data, however, the testing data (orange) showed no signs of convergence, instead demonstrating a constant straight-line pattern at the respective accuracy value printed. Out of the three different k-fold values tested, five-fold showed the smallest standard deviation value, at 1.293% variation from the mean of 81.108% (**Figure 4**). This was the highest mean accuracy retrieved from all three CNNs.



**3.2 Seven-Fold Cross Validation**

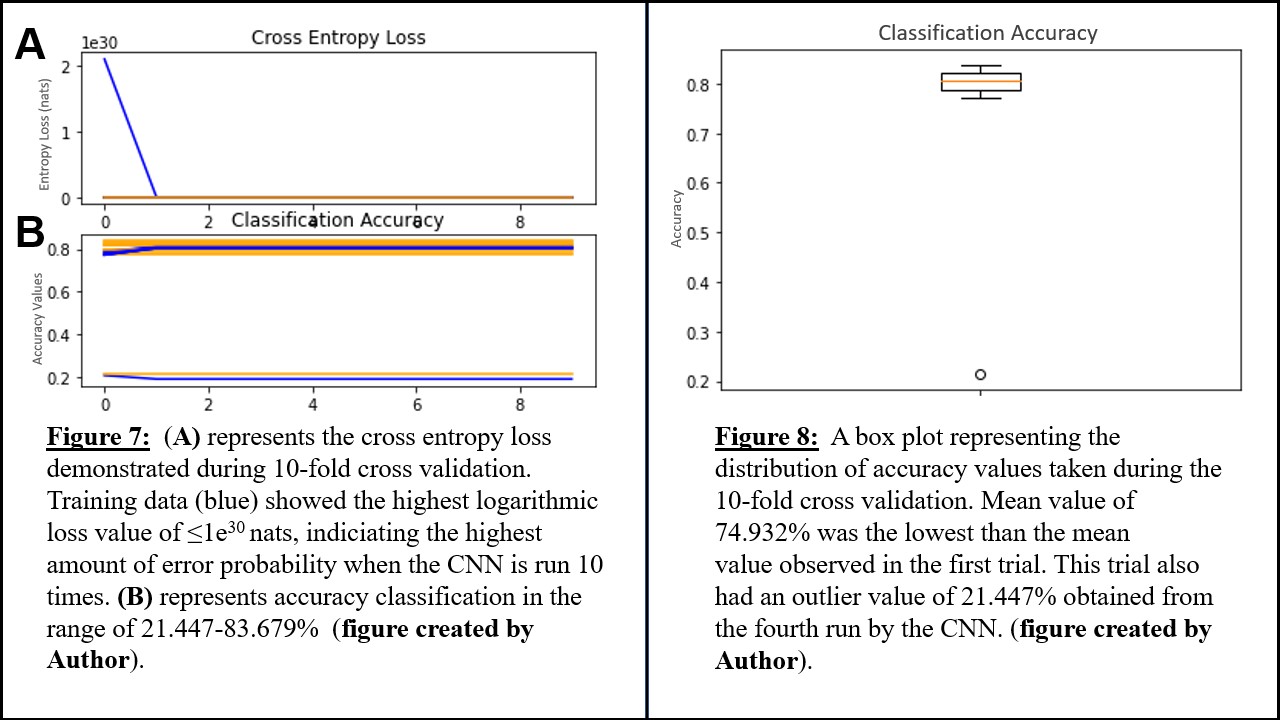
For the seven-fold cross validation, accuracy values were given as 78.080%, 84.239%, 80.072%, 84.058%, 77.536%, 79.529%, 80.072% with the mean accuracy presented as 80.512 (**Table 1)**. Despite retrieving accuracy values that reflected the values and average output by the five-fold CNN, this trial showed a much more extreme range of cross entropy loss values, with the highest value obtained being retrieved from the training dataset with a value of approximately 1017, as shown in **Figure 5a**. However, this value seems to be a major outlier, as the remaining testing and training dataset trials all converge at 0, suggesting a decreasing logarithmic loss. While the range of loss demonstrated in the five-fold CNN was <100, due to this extreme outlier, the range of cross entropy loss, which was closer to 1017 nats, is much larger for the seven-fold CNN.

The plot created shows the classification accuracy (**Figure 5b**) displaying initial convergence towards values ranging from 77.5% to a maximum of approximately 84% for the training data. However, like the five-fold CNN, the testing data (orange) showed no signs of divergence, instead demonstrating a constant straight-line pattern at the respective accuracy value printed. The maximum accuracy value obtained from the seven-fold CNN was 84.239%, which was the highest overall maximum accuracy retrieved from all the three different k-fold CNNs (**Figure 6**). 

**3.3 Ten-Fold Cross Validation**

With the final ten-fold cross validation CNN, accuracy values in percent were given as 81.654%, 78.295%, 79.587%, 21.447%, 77.202%, 82.124%, 79.534%, 83.420%, 82.383%, 83.679% with the mean accuracy presented as 74.932% **(Table 1)**. The cross entropy loss in this trial more so reflected the extreme range for cross entropy loss values in the seven-fold CNN (which was 1017  nats) than the smaller range presented in the five-fold CNN (which was approximately 150 nats), with the highest value obtained being from the training dataset with a value of approximately 1030, as shown in **Figure 7a**. Like the seven-fold CNN, this value seems to be a major outlier, as the remaining testing and training dataset trials all converge at 0, indicating a decreasing logarithmic loss. However, this cross entropy value of 1030 was the highest loss output out of all three k-fold CNNs.

For the classification accuracy (**Figure 7b**), the plot displayed was much more extreme convergence towards approximately .80 was demonstrated across all training and testing datasets tested, with the exception of the two outliers, both from the same fold as the respective training and testing data from that fold, which converges at a significantly lower accuracy value of approximately 0.21. Similarly, the five-fold and seven-fold CNN, a constant straight-line pattern at the respective accuracy value printed, can be noted across all testing and training datasets assessed, including the outliers. Due to the extreme outlier in the classification, the mean of the accuracy in percent for the ten-fold CNN was 74.932, significantly lower compared to the previous k-fold tests, and the standard deviation was much greater, with the value recorded as 17.947 (**Figure 8**).

****

**4. Discussion**

**4.1 Objectives**

The objective of my study was to design a CNN for FHR classification with 5520 images and examine the effectiveness of my CNN with this low image input. This study successfully established a CNN with relatively high mean accuracy in all trials after carefully designing and implementing the specific layers to run efficiently (**Table 1**). This study showed that it was also possible to utilize a small number of images as input and gain relatively high accuracy and low cross entropy loss (**Figure 3a, 3b, 5a, 5b, 7a,7b**)

**4.1 K-fold and Accuracy**

Utilizing lower 'k' values in K-fold validation may provide analysis of the CNN accuracy with lower processing time (Yasar & Ceylan, 2020). Despite this, there has been no guarantee, as of yet, to suggest that higher or lower k-values can vary the accuracies recorded. The five-fold presented the lowest cross entropy loss range (**Figure 4**) and the smallest standard deviation of mean as 1.293% (**Figure 5a**). As explained in Zhao et al., (2018), a small cross entropy range shows that with each fold that was run, the CNN applied a logarithmic function that was able to assess to what degree the machine was certain that the classification it had assigned for a specific FHR signal tracing, and then to what level that certainty value strayed from the actual classification. For example, if from a scale of 0 to 1, the machine was .75 certain that the signal it was examining was a healthy signal, but the signal was, in fact, labeled as unhealthy, the cross entropy loss value for that signal classification would be higher (Zhao et al., 2018). Because of this, cross entropy loss graphs that converge towards 0 and have lower values overall are favored, as it demonstrates the machine was more accurate with its classifications. For my study, despite running the CNN an increased number of times for the ten-fold and seven-fold cross validation runs, during the five-fold trial, the lowest initial cross entropy loss value indicated that there was the lowest amount of classification uncertainty detected (**Figure 4**). The greatest cross entropy loss occurred in the ten-fold cross validation, with a value of after the logarithmic function was applied, (**Figure 7a**) demonstrates that the machine had the greatest degree of uncertainty with its answers in the ten-fold validation (**Figure 7a**). However, an inverse relationship between loss and time run was demonstrated in all cross entropy loss graphs, which showed as training data was processed, the amount of incorrectly classified FHR tracings was going down, which was ideal (**Figures 3a, 5a, and 7a)**. This suggests that consistently throughout all trials, as more training data was processed by the CNN, the degree to which the CNN was incorrectly identifying signal tracings was decreasing and approaching 0, which improves classification accuracy (**Figures 3a, 5a, and 7a)**.

The mean value of the accuracy also decreased as the number of k-folds increased. The five-fold validation, despite running the CNN less times, were able to correctly identify the images according to their labels at an average of 81.108% accuracy (**Figure 4**). While the seven-fold validation was able to retrieve the highest accuracy value during its fourth fold, at a value of 84.058% classification accuracy, its overall mean was slightly lower than the five-fold validation, at 80.052% (**Figure 5b**). Considering both the five-fold and seven-fold trials were able to obtain accuracies more than 2 standard deviations over 75%, the hypothesis was therefore supported that the CNN I created would be able to classify the given FHR tracings with over 75% accuracy. Though the third trial obtained a mean accuracy of 74.932%, this lower mean can be attributed to the influence of the outlier obtained in the fourth run (**Table 1**). Mean of all three averages was approximately 77.851%. Since this accuracy is relatively high, this suggests that the CNN I designed had been able to successfully classify the FHR signals with overall accuracy at least 75% or over, as I had originally predicted in my hypothesis.

Both the cross entropy loss values and the accuracy values show that increasing the n-value in k-fold cross validation does not necessarily improve the accuracy of the classification that the CNN outputs (**Table 1**). It can be inferred that due to the random selection of values and shuffling done by the CNN during each run may not have prevented the increased k-fold cross validation from improving the accuracy received. Additionally, though this study purposely examines a smaller dataset with 5520 images and identifies how accurately it may be classified by the CNN with fewer than 10,000 images in the full set, having a small data set could have prohibited the increased k-fold from playing a part in increasing accuracy, as the subset of images used to train the machine are therefore much smaller than they'd usually be (Jehkonen, 2021). Though other studies that have constructed other types of neural networks with different structures or greater input image amounts have been able to obtain results of over 85% classification accuracy, those studies have yet to have their results reproduced, and the validity of those stated classification accuracies can therefore not be confirmed (Zhao et al., 2019b).

**4.2 The Role of Limited Input**

Previous research has shown that higher numbers of input images could result in a CNN with greater accuracy in classification (Heidari et al., 2020). However, utilizing a smaller number of images is more time efficient (Chen, 2016). Hence, one of the primary objectives of this study examined whether a CNN would be able to effectively classify FHR signals as abnormal or normal if the given dataset was much smaller than the normal amount of input images normally provided (Chen, 2016). In my study, the CNN itself was successful after retrieving a maximum accuracy mean of 81.108%, which supported my overall hypothesis **(Figure 3a)**. However, when compared to other studies that used various ML methods, or used other types of neural networks with various structures and input levels, it becomes clear that the highest accuracy mean retrieved in this study by this low input CNN does not compare to the accuracy compared to those retrieved by other neural networks and ML algorithms that were able to receive a greater volume of images or input data (Cömert & Fatih, 2016; Zhao et al., 2019b). For example, researchers in a previous study were able to achieve an accuracy of 92.4% despite using the same signals from the open database as the ones used in this study (Cömert & Fatih, 2016). However, it is hard to confirm the validity of these results, as they have yet to be reproduced by other scientists. In the case those results can be reproduced by other researchers, one of the possible explanations for the significant difference between my accuracy values and the accuracy values in their studies is the size of each FHR tracing they used, as they used the entire signal for their input, and my study only examined the last ten minutes of each signal divided into individual one minute segments. We did this in order to maintain a 2D signal image with a relatively small image size, which could have contributed to the time efficiency, but could have possibly played a part in the discrepancies in my accuracy levels compared to other similar studies. Despite this, I can conclude that the CNN was successful with significantly less images used as input, and that my original hypothesis, which states that my CNN will be able to produce an accuracy higher than 75% and perform better than other ML methods, can be supported.

**4.3 Limitations**

Data used in my research was derived from the last ten minutes prior to the delivery time. The assigned pH values used as labels were likely most accurate during this period, since the pH is recorded after birth (and this time frame is closest to that moment). However, the last five minutes of data propose some risk of being inaccurately labeled since external factors affecting the baby and the mother during this period (for example, extreme pressure applied on the baby during vaginal delivery) may interfere with the recording of the FHR signal. However, since pH threshold was the only factor considered for classification in this study, using the last ten minutes was most sensible. Additionally, though I conducted much of the research virtually from my personal laptop, when it came time to convert large amounts of signal tracings into 2D images and test my CNN, my personal computer often ran into technical issues while attempting to process such large amounts of data at once. Therefore, it became necessary to travel to StonyBrook laboratory for the completion of my research, which was not the most time efficient. This prevented me from gathering further results and running more tests that may have helped further validate my results.

**5. Conclusions and Future Studies**

My study successfully designed and created a CNN using Python and determined that CNN would provide the relatively high classification accuracy of given FHR signals with a limited amount of training data. This study supports the original hypothesis that a CNN will be able to achieve over 75% accuracy and receive better accuracy results in comparison to past studies containing other ML methods. The highest mean accuracy was obtained in the five-fold validation test, containing 81.108% as the mean of all accuracy values calculated (**Figure 3a**). The highest accuracy obtained from all tests run was from one of the folds in the seven-fold validation, which came out as 84.058% (**Table 1)**. Additionally, increasing the k value in k-fold cross validation demonstrated little to no significant increase in accuracy or cross entropy loss levels gathered, most likely attributed to having a small training dataset (Jehkonen, 2021). For these reasons, the original hypothesis was supported.

Future studies should change the CNN architecture to add layers that will further examine non-linear features and produce more accurate analysis or increase efficiency. For example, a batch normalization layer or dropout layer could be added, if more images are used, as these two layers can prevent overfitting and decrease computational time (Zhao et al., 2019b). Given more time, I would have repeated my study, except instead extending time segments used to beyond the last ten minutes. For example, segmentation can begin fifteen minutes prior to delivery and end ten minutes prior to delivery, and then each segment can be divided into smaller than one-minute segments to provide a more diverse array of images. This could prevent inaccurate labeling that may occur due to the external factors that affect a fetus during the last five minutes before birth. Additionally, the CNN could be programmed to assess its performance evaluation utilizing a confusion matrix, instead of only utilizing classification accuracy as well as cross entropy loss, which could have further validated the results I was able to obtain as well as provide an additional metrics to evaluate the effectiveness of my CNN in FHR signal tracing classification. It can be stated that this study successfully accomplished its objectives to design and implement a CNN with over 75% mean classification accuracy of given FHR signals tracings with a limited number of tracings. However, recreating the study with the suggestions listed above could allow for the progression of CTG technology and possibly help contribute to the improved assessment and classification of FHR signals more accurately and efficiently. Furthermore, improvements in electronic fetal monitoring using deep machine learning could further prevent unnecessary cesarean deliveries and make it possible to increase the safety of both a mother and the fetus during delivery.

**References**

Agarap, A. F. (2019). Deep Learning using Rectified Linear Units (ReLU). *ArXiv:1803.08375 [Cs, Stat]*. http://arxiv.org/abs/1803.08375

Aguiar-Conraria, L., & Soares, M. J. (2014). The Continuous Wavelet Transform: Moving Beyond Uni- and Bivariate Analysis. *Journal of Economic Surveys*, *28*(2), 344–375. https://doi.org/10.1111/joes.12012

Ahmad, N., & Deeba, khan. (2017). The Study of New Approaches in Cubic Spline Interpolartion for Auto Mobile Data. *Journal of Science and Arts*, *17*, 401–406.

Balocco, S., Nanculef, R., & Radeva, P. (2020). *Training Convolutional Nets to Detect Calcified Plaque in IVUS Sequences*. https://doi.org/10.1016/B978-0-12-818833-0.00009-6

Bengio, Y., & Lecun, Y. (2007). Scaling learning algorithms towards AI. In L. Bottou, O. Chapelle, D. DeCoste, & J. Weston (Eds.), *Large-scale kernel machines*. MIT Press.

Chen, X. (2016). *Classification of Fetal Heart Rate Signals by Deep Learning* [Thesis, The Graduate School, Stony Brook University: Stony Brook, NY.]. https://ir.stonybrook.edu/xmlui/handle/11401/77438

Chudáček, V., Spilka, J., Burša, M., Janků, P., Hruban, L., Huptych, M., & Lhotská, L. (2014). Open access intrapartum CTG database. *BMC Pregnancy and Childbirth*, *14*(1), 16. https://doi.org/10.1186/1471-2393-14-16

Cömert, Z., & Kocamaz, A. (2016). Evaluation of Fetal Distress Diagnosis during Delivery Stages based on Linear and Nonlinear Features of Fetal Heart Rate for Neural Network Community. *International Journal of Computer Applications*, *156*, 26–31. https://doi.org/10.5120/ijca2016912417

Cömert, Z., & Kocamaz, A. (2017). Comparison of Machine Learning Techniques for Fetal Heart Rate Classification. *Acta Physica Polonica A*, *132*(3), 451–454. https://doi.org/10.12693/APhysPolA.132.451

Czabanski, R., Jezewski, J., Matonia, A., & Jezewski, M. (2012, November). *Computerized analysis of fetal heart rate signals as the predictor of neonatal acidemia | Elsevier Enhanced Reader*. Science Direct. https://doi.org/10.1016/j.eswa.2012.01.196

Dargan, S., Kumar, M., Ayyagari, M., & Kumar, G. (2019, June 1). *A Survey of Deep Learning and Its Applications: A New Paradigm to Machine Learning | SpringerLink*. SpringerLink. https://link.springer.com/article/10.1007/s11831-019-09344-w

Elmansouri, K., Latif, R., & Maoulainine, F. (2014). Efficient Fetal Heart Rate extraction using Undecimated Wavelet Transform. *2014 Second World Conference on Complex Systems (WCCS)*, 696–701. https://doi.org/10.1109/ICoCS.2014.7060987

Fotiadou, E., & Vullings, R. (2020). Multi-Channel Fetal ECG Denoising With Deep Convolutional Neural Networks. *Frontiers in Pediatrics*, *8*, 508. https://doi.org/10.3389/fped.2020.00508

Freund, R. J., Wilson, W. J., & Mohr, D. L. (2010). CHAPTER 5—Inferences for Two Populations. In R. J. Freund, W. J. Wilson, & D. L. Mohr (Eds.), *Statistical Methods (Third Edition)* (pp. 201–243). Academic Press. https://doi.org/10.1016/B978-0-12-374970-3.00005-6

German Society of Gynecology and Obstetrics (DGGG), Maternal Fetal Medicine Study Group (AGMFM), German Society of Prenatal Medicine and Obstetrics (DGPGM), & German Society of Perinatal Medicine (DGPM). (2014). S1-Guideline on the Use of CTG During Pregnancy and Labor. *Geburtshilfe Und Frauenheilkunde*, *74*(8), 721–732. https://doi.org/10.1055/s-0034-1382874

Heidari, M., Mirniaharikandehei, S., Khuzani, A. Z., Danala, G., Qiu, Y., & Zheng, B. (2020). Improving the performance of CNN to predict the likelihood of COVID-19 using chest X-ray images with preprocessing algorithms. *International Journal of Medical Informatics*, *144*, 104284. https://doi.org/10.1016/j.ijmedinf.2020.104284

Hruban, L., Spilka, J., Chudáček, V., Janků, P., Huptych, M., Burša, M., Hudec, A., Kacerovský, M., Koucký, M., Procházka, M., Korečko, V., Seget’a, J., Šimetka, O., Měchurová, A., & Lhotská, L. (2015). Agreement on intrapartum cardiotocogram recordings between expert obstetricians. *Journal of Evaluation in Clinical Practice*, *21*(4), 694–702. https://doi.org/10.1111/jep.12368

Jehkonen, P. (2021). *Cardiotocograph dataset engineering and detection of an increased fetal heart rate variability pattern*. https://aaltodoc.aalto.fi:443/handle/123456789/103056

Ke, Q., Liu, J., Bennamoun, M., An, S., Sohel, F., & Boussaid, F. (2018). Chapter 5—Computer Vision for Human–Machine Interaction. In M. Leo & G. M. Farinella (Eds.), *Computer Vision for Assistive Healthcare* (pp. 127–145). Academic Press. https://doi.org/10.1016/B978-0-12-813445-0.00005-8

Kiely, D. J., Oppenheimer, L. W., & Dornan, J. C. (2019). Unrecognized maternal heart rate artefact in cases of perinatal mortality reported to the United States Food and Drug Administration from 2009 to 2019: A critical patient safety issue. *BMC Pregnancy and Childbirth*, *19*(1), 501. https://doi.org/10.1186/s12884-019-2660-5

Krupa, N., MA, M. A., Zahedi, E., Ahmed, S., & Hassan, F. M. (2011). Antepartum fetal heart rate feature extraction and classification using empirical mode decomposition and support vector machine. *BioMedical Engineering OnLine*, *10*, 6. https://doi.org/10.1186/1475-925X-10-6

Li, J., Chen, Z.-Z., Huang, L., Fang, M., Li, B., Fu, X., Wang, H., & Zhao, Q. (2019). Automatic Classification of Fetal Heart Rate Based on Convolutional Neural Network. *IEEE Internet of Things Journal*, *6*(2), 1394–1401. https://doi.org/10.1109/JIOT.2018.2845128

Love, B. C. (2002). Comparing supervised and unsupervised category learning. *Psychonomic Bulletin & Review*, *9*(4), 829–835. https://doi.org/10.3758/BF03196342

Macones, G. A., Hankins, G. D. V., Spong, C. Y., Hauth, J., & Moore, T. (2008). The 2008 National Institute of Child Health and Human Development workshop report on electronic fetal monitoring: Update on definitions, interpretation, and research guidelines. *Obstetrics and Gynecology*, *112*(3), 661–666. https://doi.org/10.1097/AOG.0b013e3181841395

Matonia, A., Jezewski, J., Kupka, T., Jezewski, M., Horoba, K., Wrobel, J., Czabanski, R., & Kahankow, R. (2020, June 25). *Fetal electrocardiograms, direct and abdominal with reference heartbeat annotations | Scientific Data*. Nature. https://www.nature.com/articles/s41597-020-0538-z

Nunes, I., Ayres-de-Campos, D., Costa-Santos, C., & Bernardes, J. (2014). Differences between external and internal fetal heart rate monitoring during the second stage of labor: A prospective observational study. *Journal of Perinatal Medicine*, *42*(4), 493–498. https://doi.org/10.1515/jpm-2013-0281

Polnaszek, B., López, J. D., Clark, R., Raghuraman, N., Macones, G. A., & Cahill, A. G. (2020). Marked variability in intrapartum electronic fetal heart rate patterns: Association with neonatal morbidity and abnormal arterial cord gas. *Journal of Perinatology*, *40*(1), 56–62. https://doi.org/10.1038/s41372-019-0520-9

Reddy, V. K., Siramoju, K. K., & Sircar, P. (2014). Object Detection by 2-D Continuous Wavelet Transform. *2014 International Conference on Computational Science and Computational Intelligence*, *1*, 162–167. https://doi.org/10.1109/CSCI.2014.34

Sadiq, A., Faris, H., Al-Zoubi, A. M., Mirjalili, S., & Ghafoor, K. (2019). Fraud Detection Model Based on Multi-Verse Features Extraction Approach for Smart City Applications. *Smart Cities Cybersecurity and Privacy*. https://doi.org/10.1016/B978-0-12-815032-0.00017-2

Sinam, A., Meitei, T., & Majumder, S. (2020). *Short PCG classification based on deep learning* (pp. 141–164). https://doi.org/10.1016/B978-0-12-819061-6.00006-9

Tarvonen, M., Hovi, P., Sainio, S., Vuorela, P., Andersson, S., & Teramo, K. (2020, September 27). *Intrapartum zigzag pattern of fetal heart rate is an early sign of fetal hypoxia: A large obstetric retrospective cohort study—Tarvonen—2021—Acta Obstetricia et Gynecologica Scandinavica—Wiley Online Library*. Wileys Obstretics and Gynecology Online Library. https://obgyn.onlinelibrary.wiley.com/doi/full/10.1111/aogs.14007?af=R

Tarvonen, M., Sainio, S., Hämäläinen, E., Hiilesmaa, V., Andersson, S., & Teramo, K. (2020). Saltatory Pattern of Fetal Heart Rate during Labor Is a Sign of Fetal Hypoxia. *Neonatology*, *117*(1), 111–117. https://doi.org/10.1159/000504941

Tomlinson, M. W., Brumbaugh, S. A., O’Keeffe, M., Berkowitz, R. L., D’Alton, M., Nageotte, M., & Perinatal Quality Foundation. (2020). Electronic Fetal Monitoring Credentialing Examination: The First 4000. *AJP Reports*, *10*(1), e93–e100. https://doi.org/10.1055/s-0040-1705141

Vayssiere, C., Haberstich, R., Sebahoun, V., David, E., Roth, E., & Langer, B. (2007). Fetal electrocardiogram ST-segment analysis and prediction of neonatal acidosis. *International Journal of Gynaecology and Obstetrics: The Official Organ of the International Federation of Gynaecology and Obstetrics*, *97*(2), 110–114. https://doi.org/10.1016/j.ijgo.2007.01.003

Yasar, H., & Ceylan, M. (2020). A novel comparative study for detection of Covid-19 on CT lung images using texture analysis, machine learning, and deep learning methods. *Multimedia Tools and Applications*, 1–25. https://doi.org/10.1007/s11042-020-09894-3

Yoo, Y., & Baek, J.-G. (2018). A Novel Image Feature for the Remaining Useful Lifetime Prediction of Bearings Based on Continuous Wavelet Transform and Convolutional Neural Network. *Applied Sciences*, *8*(7), 1102. https://doi.org/10.3390/app8071102

Yu, K., Quirk, J. G., & Djurić, P. M. (2017). Dynamic classification of fetal heart rates by hierarchical Dirichlet process mixture models. *PLoS ONE*, *12*(9), e0185417. https://doi.org/10.1371/journal.pone.0185417

Zhao, J., Mao, X., & Chen, L. (2019). Speech emotion recognition using deep 1D & 2D CNN LSTM networks. *Biomedical Signal Processing and Control*, *47*, 312–323. https://doi.org/10.1016/j.bspc.2018.08.035

Zhao, Z., Yanjun, D., Zhang, Y., Zhang, Y., Zhang, X., & Shao, L. (2019, December 30). *DeepFHR: intelligent prediction of fetal Acidemia using fetal heart rate signals based on convolutional neural network | BMC Medical Informatics and Decision Making | Full Text*. BioMed Central. https://bmcmedinformdecismak.biomedcentral.com/articles/10.1186/s12911-019-1007-5

Zhao, Z., Zhang, Y., & Deng, Y. (2018, August). *A Comprehensive Feature Analysis of the Fetal Heart Rate Signal for the Intelligent Assessment of Fetal State*. PubMed Central. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6111566/