UNIVERSITY OF NAIROBI

DEPARTMENT OF COMPUTING & INFORMATICS

FETAL ANOMALIES DETECTION USING CONVOLUTIONAL NEURAL NETWORKS

BY

CHARITY KATHURE

A PROJECT REPORT SUBMITTED TO UNIVERSITY OF NAIROBI IN PARTIAL FULFILMENT OF THE REQUIREMENT FOR THE AWARD OF THE DEGREE OF MASTER OF SCIENCE IN COMPUTATIONAL INTELLIGENCE OF THE UNIVERSITY OF NAIROBI

AUGUST 2021

Declaration

Declaration by the student

This is my original work, and it has not been presented for a degree in any other University. No part of this project may be reproduced without the prior written permission of the author and The University of Nairobi.

Signature: Date: 23rd August 2021

Charity Kathure

P52/35567/2019

Declaration by supervisor

This project report has been submitted for examination with my approval as the University Supervisor.

Signature: Date: 23/08/2021

Prof. Elisha Toyne O. Opiyo

Department of Computing and Informatics

University of Nairobi

Acknowledgement

I thank God for His abundant Grace throughout this journey. My deep appreciation extends to my supervisor, Prof. Elisha Toyne O. Opiyo for the immeasurable guidance and support during the development of this project. Additionally, I thank all the other faculty members at the University of Nairobi’s School of Computing and Informatics, especially the panelists who gave me constructive critique and valuable contribution in this project’s lifecycle. Special thanks to my father, mother, and fiancé for the unconditional and unending support throughout this journey.

Abstract

Fetal anomalies are structural defects in a fetus that can lead to a complicated pregnancy, and disabilities later in life. Early detection and intervention are key in the prevention of later disabilities. Conventionally, specialists detect any anomalies in the fetus by physically analyzing the medical images such as ultrasound scans and MRIs. However, the cost of training a qualified radiologist and the general limitations of human beings such fatigue, lack of speed and experience may lead to delayed or erroneous diagnosis, hence delaying intervention.

In the recent years machine learning has been applied in the detection of conditions such as pneumonia and cancer. This research processes the use of convolutional neural network in the detection of fetal anomalies from ultrasound scans. Due to time limitation, this research Focuses on the detection of only one fetal anomaly, Congenital Talipes Equinovarus (CTEV) which is one of the most common musculoskeletal defects that can be corrected by early detection and intervention. The objective of this study is to develop a deep learning model that can analyze ultrasound scans and detect Congenital Talipes Equinovarus.

200 samples of 2-dimensional ultrasound scans were used in the project, The sample size was split into three main sections: training, validation, and testing data. Three implementations of the model were done and compared: a standard CNN model without augmentation with 67.5% accuracy, a CNN model with augmentation with 77% accuracy and a CNN model with transfer learning with 85%. CNN model with transfer learning was selected to implement the model due to is high accuracy.

Table of Contents

List of Figures:

List of Tables:

Abbreviations

CTEV - Congenital Talipes Equinovarus

CNN - Convolutional Neural Networks

ML - Machine Learning

ReLU - Rectified linear activation function

CT - Computed Tomography

MRI - Magnetic Resonance Imaging

PET - Positron Emission Tomography

CTG - Cardiotocography

FHR - Fetal Heart Rate

CHAPTER ONE: INTRODUCTION

1.1 Background

Fetal anomaly is defined as a structural or functional anomaly in the fetus that occurs during the pregnancy period and is detected prenatally, during or after birth [CITATION Wor20 \l 1033]. These defects can complicate pregnancy and cause negative consequences to the developing infant. Other terms used in referring to fetal anomalies include congenital anomalies, congenital malformations, congenital disorders, and congenital abnormalities.

(World Health Organization, 2015) reports that approximately 276,000 newborns worldwide die due to congenital anomalies. Among the different types of structural congenital anomalies, musculoskeletal defects are the most common type of congenital anomalies in the sub-Saharan African countries [CITATION Fen20 \l 1033]. Congenital musculoskeletal anomalies occur when a fetus' bones, muscle or joints do not develop well or wholly, or some structures are disjointed or not aligned well.

This research will be focusing on the use of convolutional neural networks (CNN) in the detection of Congenital talipes equinovarus (CTEV). Congenital talipes equinovarus, is also known as clubfoot. Club foot is a birth defect that affects the foot, and the foot appears to be bent out of shape. This choice is influenced by the fact that congenital talipes equinovarus is the most reported type of musculoskeletal defect. [CITATION Edw18 \l 1033]

Convolutional Neural Networks, also known as ConvNets or CNNs are a type of Neural Networks that are good at image recognition and classification. Some of the applications that Convolutional Neural Networks have been successful at and known for include face recognition, object recognition and self-driving cars.

This study proposes the use of two-dimensional ultrasound scans to train and test the CNN model. An Ultrasound scan, also known as a sonogram, is a medical test that employs the usage of high frequency sound waves to visualize the internal parts of the body. According to AXA global healthcare, an ultrasound can be charged between 600 to 4000 Kenyan Shillings in Kenyan clinics, making it the most accepted and cost-effective type of imaging modality in Kenya. Additionally, ultrasound scans are the most preferred imaging and monitoring techniques all over the world during the pregnancy period because unlike the other medical imaging techniques, ultrasound does not use radiation.[CITATION Fan17 \l 1033]

The use of deep-learning-based medical image analysis in ultrasound analysis not only promises a great support to doctors and specialists in their decisions, but also gives the first assessment of the probability that the fetus has Congenital talipes equinovarus.

1.2 Problem Statement

Congenital Talipes Equinovarus is a congenital anomaly that affects the foot. If ignored or left untreated, congenital talipes equinovarus can restrain movement by making walking hard, painful, or even impossible. However, early detection and treatment are assumed to be the key to preventing late disabilities.

Although radiologists can interpret ultrasounds and detect anomalies early, it takes several years and a huge financial cost to train a competent radiologist. In addition, a study into the usage of ultrasound in an emergency setup has shown that errors or missed diagnoses occurred in anywhere from eight to 10% of cases.

Ultrasounds are an integral part of pregnancy electronic health records and are presently analyzed and interpreted by human radiologists, who are limited by fatigue, speed, and experience.

1.3 Research Objectives

1.3.0 General Objective

To develop and evaluate a prototype deep learning model that can analyze ultrasound scans and detect Congenital Talipes Equinovarus

1.3.1 Specific Objectives

1. To design a deep learning model to detect Congenital Talipes Equinovarus based on ultrasound images.
2. To train the model with data obtained.
3. To evaluate the model for detection of Congenital Talipes Equinovarus

1.4 Research Questions

To ensure that the objectives of the research are attained, the below research questions are key:

- How can we design a model to detect Congenital Talipes Equinovarus?
- How can we train a model to detect Congenital Talipes Equinovarus?
- How can we test and validate the deep learning model to ascertain accuracy of Congenital Talipes Equinovarus detection?

1.5 Significance of the Research

WHO has estimated that 260,000 deaths worldwide (about 7% of all neonatal deaths) resulted from congenital anomalies in the year 2004. Early detection and treatment and monitoring is key to the reduction of such cases.

Although doctors do a phenomenally good job analyzing findings during prenatal clinics, they are prone to fatigue, lack of experience and errors. Apart from the human limitations faced, the cost of training a qualified medical specialist is high. The cost includes and is not limited to, US$ 48, 169 in tertiary education, US$ 6, 865 in secondary education and US $10, 963 in primary education for a single doctor, all adding up to an average of US$ 65, 997 for each doctor. This being the case, this research will play a part in enhancing accuracy and speed in the detection of Congenital Talipes Equinovarus.

1.6 Scope and Limitation

1. Due to time limitation, this research only covers the detection of one fetal anomaly, namely Congenital Talipes Equinovarus.
2. Data used in this study was obtained from online public repositories of fetus ultrasound scans such as Radiopaedia and https://www.fetalultrasound.com/
3.

CHAPTER TWO: LITERATURE REVIEW

2.0 Introduction

Several measures have been taken to help detect fetal distress, this literature review focuses on understanding machine learning, neural networks, convolutional neural networks, convolution neural networks in medicine as well as the current research and methods applied during pregnancy with a focus on enhancing maternal health and detecting fetal distress.

2.1 Machine Learning

[CITATION Art59 \l 1033] came up with the word machine learning and defined it as an area of study that enables machines to learn just like humans without the need of being explicitly programmed. Machine learning is different from conventional programming in that in conventional programming, we add the input data and program into a computer, that in turn returns an output, while in machine learning input data and expected output are fed into a machine during the learning stage and the machine formulates a program on its own.

Lately, machine learning (ML) has gained a lot of attention and its algorithms are applied in many fields, such as face detection, pattern recognition, text interpretation as well as other different research areas. Machine learning has given artificial intelligence the capability to understand the world and operate with minimal human intervention.

2.2 Deep Learning

Deep learning is a sub-set of machine learning concerned with algorithms inspired by the structure and function of the brain called artificial neural networks. Unlike in the case of general machine learning algorithms that use structured data, deep learning algorithms use unstructured data. In deep learning, a computer model learns to perform classification tasks directly from images, text, or sound without the need for manual feature extraction. Some of the deep learning algorithms include neural networks, fully connected neural networks, convolutional neural networks, and recurrent neural networks. Some of the existing use cases of deep learning include Speech Recognition for example Amazon Alexa and Apple Siri, Recommendation Systems such as YouTube recommendations, Natural language processing (NLP) and Computer Vision.

2.3 Neural Networks

A Neural Network is a computer program that works in the same mannerism as the human brain with an objective of carrying out those cognitive capabilities that our brain can carry out such as problem-solving and capability to be taught and learn.

A neuron is the basic unit of computation in the neural network. Neurons accept input, process it using several Neurons in multiple hidden layers, and give an output through the output layer.

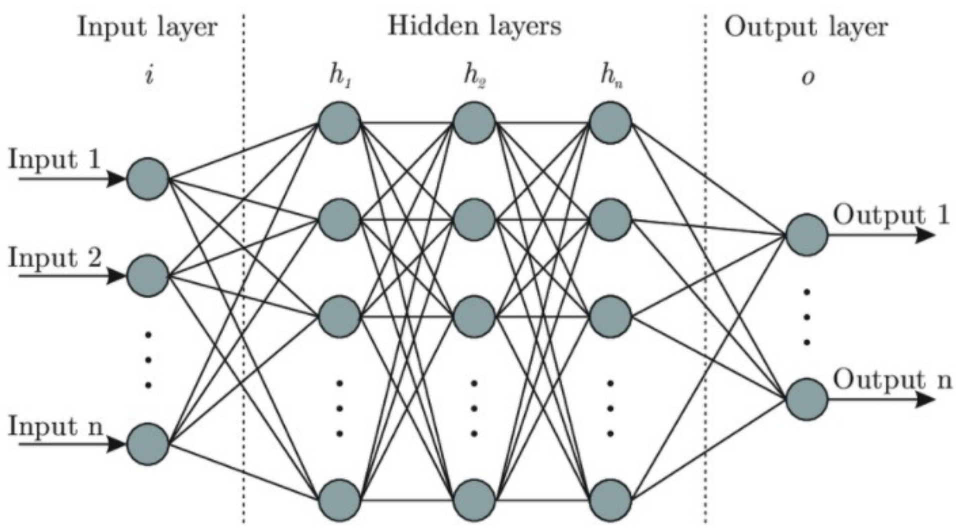


Figure 1: Anatomy of Neural Networks

The artificial neurons, which are also known as units are organized in different layers, each layer connected to subsequent layers on each side. The first layer contains input units that receive different types of information from the real world. This information is used by the network to learn. The layer on the opposite side contains the output units that are used in signaling how the network responds to the learned information. The hidden layer is located between the input and output units. The hidden layer can be single or multiple making up most of the artificial network/brain.

The majority of the neural networks are fully connected. The term fully connected is used to mean that every hidden unit and output unit is connected to every unit in the subsequent layers on both sides. Each connection connecting different units is weighted.

If one unit excites the other, the weight is positive and if one unit represses, restrains, or inhibits the other, the weight is negative. The analogy is quite similar to the working of the brain, whereby the brain cells triggered one another through small gaps known as the synapses.

Deep learning is a form of neural network often referred to as deep neural network that has multiple hidden layers and multiple nodes in each hidden layer.

2.4 Convolutional Neural Networks (CNN)

A convolutional neural network is defined as a deep learning algorithm that can accept an image as an input, assign relevance to different properties and details in the input image as well as differentiate between different images.

Convolutional layer is a key layer in convolutional neural networks that performs an operation known as the "convolution". A convolution refers to the operation of multiplying different weights with inputs. The multiplication occurs between the input data in the form of an array and a two-dimensional array of weights. This operation is known as the kernel or filter. On applying the kernel process to several inputs, it results in a feature map. In an ideal application implementation, given an image, a filter process is methodologically applied to the image providing a feature map as an output. The figure 2 provides a visual representation of this process.

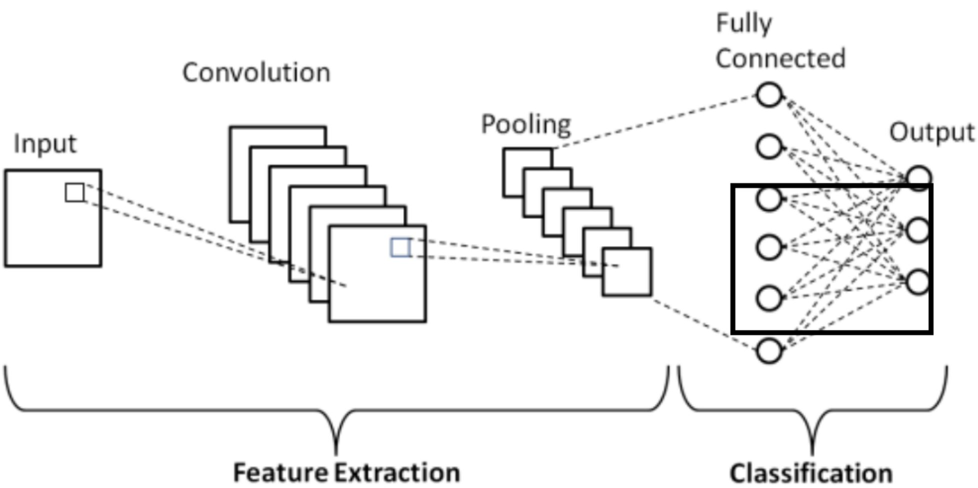


Figure 2: CNN layers

A CNN architecture is comprised of several layers in a defined sequence. The layers convert the image into numerical class scores. On each layer, a given activation function is applied to provide a specific output. The four layers forming a CNN include: convolution, nonlinearity also known as rectified linear unit, fully connected and pooling layers.

A single output matrix of is calculated using the following formula:

$$A_j = f\left(\sum_{i=1}^{N} I_i * K_{i,j} + B_j\right)$$

where ***Ii*** is the input matrix. The input matrix is then convoluted with an n × n filter Ki,j (n < input size).

The pooling layers sample down the input by breaking it into rectangular portions, this reduces the number of outputs within the convolution layer. In this layer, non-linear functions such as rectified linear function (ReLU) are used to enhance both the classification performance and learning speed in the CNN applications.

This study proposes the use of CNN, this choice is influenced by the fact that CNNs do so much better than other neural networks on images due to the convolutional layers taking advantage of inherent properties of the images. This is also because CNN can use convolution on patches of contiguous pixels, also known as local connectivity, to dramatically reduce the number of operations required to train the models.

2.5 Convolutional Neural Networks in Medicine

Medical imaging is very significant in the visualization of internal organs and the detection of any deformities or defects in their anatomy and, or function. Imaging tools such as Computed Tomography (CT), X-ray, Magnetic Resonance Imaging (MRI), ultrasound scanners and Positron Emission Tomography (PET) can capture both the anatomy and showcase the functioning of the internal organs, this is presented in form of videos or images. In order to detect and diagnose anomalies, these videos and images need to be properly understood; a role dedicated to specialized or trained physicians. The diagnosis is usually dependent on the physician's judgement, knowledge, or experience.

However, over time scientific researchers have applied the use of CNN for medical image interpretation applications. Such applications are in detecting tumors and classifying them as either benign or malignant, detection of skin lesions, detection of optical coherence tomography images, detection of colon and blood cancer, abnormalities in the breast tissues, heart, chest, eyes, among many other applications. Additionally, CNN-based models like CheXNet have also been used to classify 14 distinct conditions of the chest and have managed to achieve much better results in comparison to the overall performance of human experts.

2.6 Fetal Anomalies

Fetal anomalies refer to unusual, abnormal, or unexpected conditions in a child's growth during pregnancy. According to the [CITATION TJM15 \l 1033], birth defects are the dominating cause of infant mortalities, accounting for roughly 20% of all infant deaths. Although an estimated 50% of these anomalies are not associated with a specific cause, associations of chromosomal disorders, single gene defects, environmental teratogens, micronutrient deficiencies or multifactorial inheritance are some known common causes.

[CITATION Fen20 \l 1033] in their study on prevalence of defects in newborns in sub-Saharan African countries, highlight musculoskeletal systems defect with a prevalence of approximately 4 for every 1000 newborns. Congenital musculoskeletal defects occur before birth when certain muscles, joints or bones fail to develop properly.

Talipes equinovarus, also known as Clubfoot, is a defect of the feet present at birth. This occurs where the foot points downward and inward.

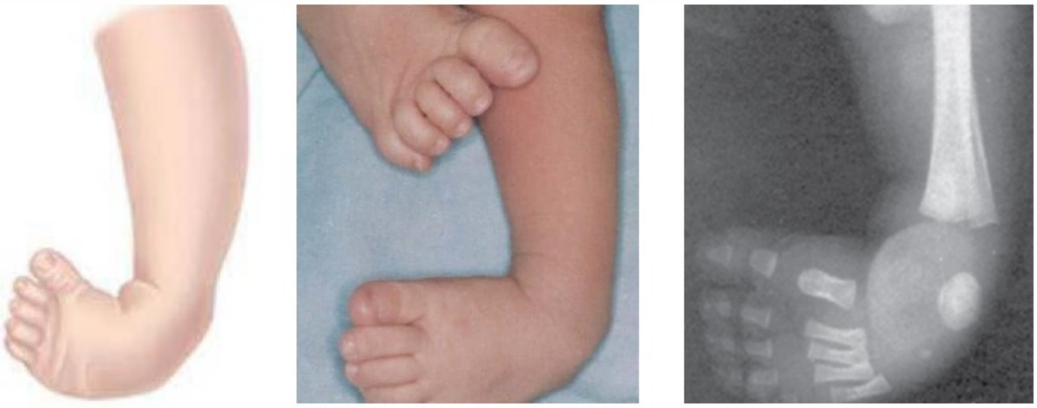


Figure 3: Foot of a child with clubfoot/CTEV

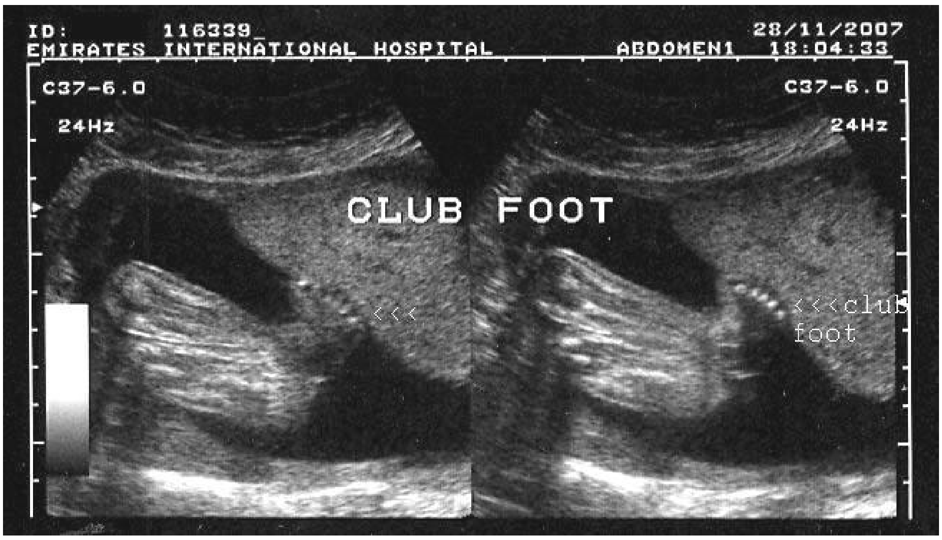


Figure 4: 2D ultrasound scan of a fetus with clubfoot

Clubfoot is a treatable birth anomaly that affects about 150,000 to 200,000 children every single year and it is one of the most prevalent birth defects involving the musculoskeletal system [CITATION Ste16 \l 1033]. Left untreated, this condition can result in severe lifelong disability.

2.7 Related Work

Cardiotocography (CTG) is a monitoring approach applied extensively all over the world to check and determine the distress level of the fetus. CTG entails two major signals: fetal uterine contraction (UC) and heart rate (FHR).

[CITATION Fai21 \l 1033] checked the behaviors and performances of neural network training algorithms comparisons on classification tasks of the CTG traces. In their study, the neural networks were categorized into five groups: Resilient Backpropagation, Gradient Descent, Quasi-Newton, Conjugate Gradient, and Levenberg-Marquardt. Although all the five algorithms provided impressive results, the algorithms that performed the best classifications were obtained with Resilient Backpropagation (RP) and Levenberg-Marquardt backpropagation (LM) algorithms. The outputs of Resilient Backpropagation and Levenberg-Marquardt backpropagation were 89.69% and 86.14%, respectively. Therefore, the study confirmed ANN as a useful machine learning tool to classify fetal heart rate recordings [CITATION ERa20 \l 1033]

[CITATION Phi12 \l 1033] proposed the detection of fetal hypoxia during labor using machine learning by building models of the CTG. The models were structured based on the captured patient information, clinical knowledge, and main physiological stimulants of a fetus' heart rate. In the study, the proposed approach focused on interactions between the fetus uterine pressure as the input and the heart rate as the output signals. The two records were used to train the model without enforcing any prior relationships between the two signals. In their findings, the study correctly detected half of the pathological cases within an accepted false positive rate of 7.5%. This allowed for early clinical interventions.

One of the pre-existing intelligent systems used in reducing the risk during labor is the K2 INFANT medical system. INFANT is used in analyzing the strength and quality of the fetus' heart signal by applying the computerized interpretation of Fetal heart rate during labor. Computerized interpretation of fetal heart rate during labor is an artificial intelligence area that is currently on trial. This is an area that promises to improve the reliability and efficiency of fetal heart rate reading by helping them decide the best management based on the reading and ultimately decreasing the burden of work.

[CITATION Ruo19 \l 1033] proposed use of deep convolutional neural networks (CNN) and CNN-based domain transfer learning to automatically recognize 6 standard planes of fetal brains. The study used two datasets one containing 30,000 2D ultrasound images from a sample size of 155 participants between 16 and 34 weeks and the other containing 1,200 image samples through 40 weeks of the pregnancy period. In their findings, the report highlights that deep convolutional neural network demonstrated better performance than the classical deep learning methods. This demonstrated the huge potential of CNN in recognition of fetal brain standard scales.

[CITATION Jin21 \l 1033] proposed a hybrid prediction model of weight at birth. The model was based on long short-term memory (LSTM) networks, The study demonstrated the establishment of a continuous model of parameters relating to the fetal physical examination and the expectant women. The findings of the study showed that the proposed birth weight model increased the prediction accuracy by 6% as well as the model convergence rate by using a hybrid approach.

2.8 Research Gap

Fetal Heart Rate monitoring does a fairly good job in detecting hypoxia and ischemia when the conditions exist. However, fetal heart rate and weight are not the only factors that could lead to fetal anomaly or death. A technology with the capability to monitor and interpret the status and features of the fetus from medical imaging/videos should be considered to make obstetrics safer. As evident in the existing related work, convolutional neural networks have a great potential in image recognition, however, this has not been greatly applied in addressing fetal anomalies.

2.9 Proposed Model

2.8.1 Conceptual Framework of the proposed model

A conceptual framework is a diagrammatic representation of coherent collection of concepts and propositions with an underlying worldview[CITATION Bri06 \l 1033]. The diagram below is a generalized conceptual framework that shows how this research is geared towards the detection of congenital talipes equinovarus.

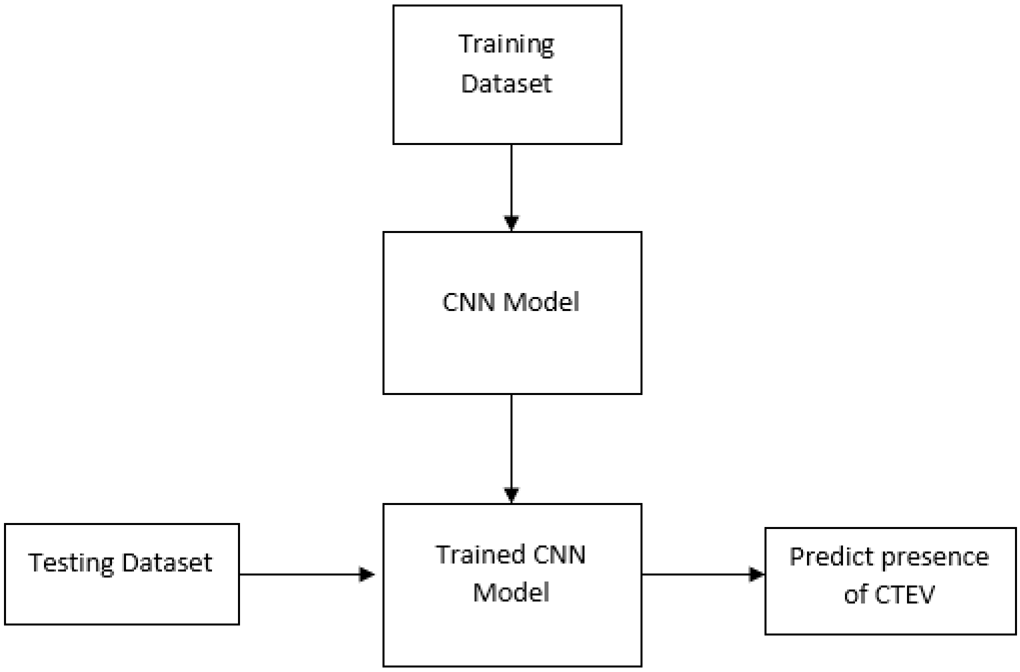


Figure 5: Conceptual Framework

2.8.2 Architectural Diagram

The figure below shows the high-level architecture of the convolutional neural network model:

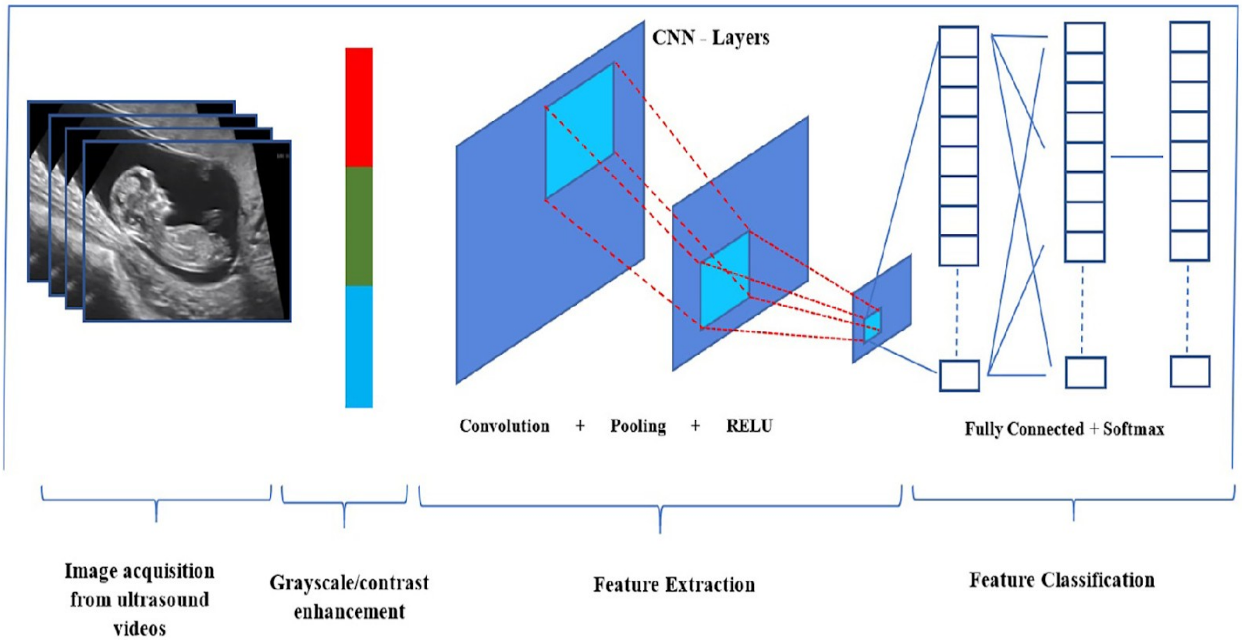


Figure 6: CNN architecture

CHAPTER THREE: METHODOLOGY

3.0 Introduction

This section highlights the concepts, guiding principles, set rules, and set procedures in this study. The research methodology is presented in the following sequence: (i) Research design, (ii). Research approach, (iii). Target population, (iv). Data collection procedures and instruments, (v). Data analysis and presentation. This study aims at detecting fetal anomalies, specifically Congenital talipes equinovarus from ultrasound images. The whole concept entails designing a model based on machine learning algorithms capable of detecting the fetal anomaly based on supplied data features. The data used in this research is acquired from fetal ultrasound online public repositories.

3.1 Research Design

Research design is the conceptual structure within which research is conducted. It constitutes the blueprint for collecting, measuring, and analyzing data (Research Methodology Method and Techniques, C.R. Kothari). Research design is relevant in facilitating the smooth sailing of the various research operations, making research as efficient as possible yielding maximal information with minimal expenditure of effort, time, and money.

3.1.1 Target Population

In this study, the target population was ultrasound scan image samples from various online public repositories.

3.1.2 Sample Selection and Sample Size

Sample size refers to the number of items to be selected from the universe to constitute a sample. 200 two-dimensional ultrasound image samples were used in this study for both training, validation, and testing. The sample size was selected using convenience sampling. Convenience sampling (also known as availability sampling) is a specific type of non-probability sampling method that relies on data collection from population members who are conveniently available to participate in study. 71% of the total sample constitutes scans with congenital talipes equinovarus while 29% constitutes scans without congenital talipes equinovarus.

Table 1: Sample size

Category	**Sample Type**	**Samples Size**
Training	Scans with CTEV	100
	Scans without CTEV	30
Validation	Scans with CTEV	25
	Scans without CTEV	15
Testing	Scans with CTEV	25
	Scans without CTEV	15
Totals		**210**

3.1.3 Study Ethics

No patient identifier features were mined from the source and the data collected was solely used for research purposes only.

3.2 Data Acquisition and Analysis

Data acquisition is the process of preparing and collecting data. Data was obtained from online public repositories of fetal ultrasounds and datasets were labeled and divided based on the publisher's categorization. The data was divided into three categories: training data, validation data and testing data. Each of the three categories was then categorized into two specific categories: scans with CTEV and scans without CTEV (Normal).

Confusion matrix was used to analyze the performance of the model. A Confusion matrix is an N x N matrix used to evaluate the performance of a classification model, where N is the number of target classes. [CITATION Ani20 \l 1033]

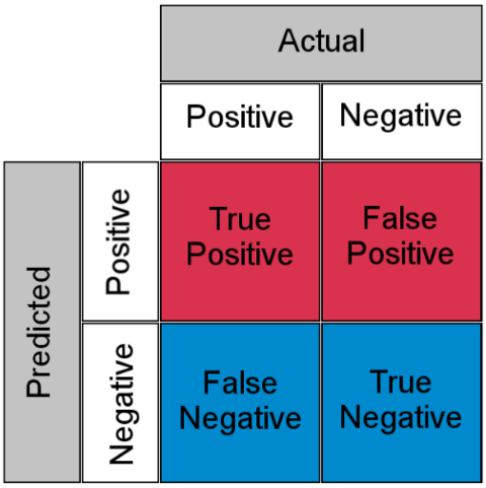


Figure 7: Confusion Matrix

Confusion matrix compares the expected values or outcome with the values/ outcome predicted by the machine learning model. This gives a general view of how well the model performed as well as highlights the errors made by the model.

3.3 Developing Convolutional Neural Network

3.3.1 Development Methodology

Waterfall development methodology was applied in the development of the model. Waterfall methodology is one of the traditional software development methods. It is a rigid linear model that consists of sequential phases (requirements, design, implementation, verification, maintenance) focusing on distinct goals and making the entire development process easy to manage.[CITATION syn17 \l 1033]

3.3.2 Model Development

The model is developed in python, using tensorflow. Tensorflow is an end-to-end machine learning platform that can be used to build deep learning architecture. Apart from Tensorflow, other python libraries that will be used include: numpy, pandas, PIL, seaborn and sklearn.

The model is constructed with a layer of Conv2D followed by a layer of MaxPooling. Max pooling is used to down sample the input representation, helping the model in dealing with overfitting as well as reducing the computational cost. Rectified linear activation function (ReLU) is applied as the activation function after every convolution to transform the output values between the range of 0 and 1. ReLU can be described as a piecewise linear function that gives an output of the input directly if it is positive and zero if it's not positive. This activation function is preferred because it overcomes the vanishing gradients problem, allowing models to learn faster and perform better.

```
cnn = Sequential()
cnn.add(Conv2D(32, (3, 3), activation="relu", input_shape=(img_width, img_height, 1)))
cnn.add(MaxPooling2D(pool_size=(2, 2)))
cnn.add(Conv2D(32, (3, 3), activation="relu", input_shape=(img_width, img_height, 1)))
cnn.add(MaxPooling2D(pool_size=(2, 2)))
cnn.add(Conv2D(32, (3, 3), activation="relu", input_shape=(img_width, img_height, 1)))
cnn.add(MaxPooling2D(pool_size=(2, 2)))
cnn.add(Conv2D(64, (3, 3), activation="relu", input_shape=(img_width, img_height, 1)))
cnn.add(MaxPooling2D(pool_size=(2, 2)))
cnn.add(Conv2D(64, (3, 3), activation="relu", input_shape=(img_width, img_height, 1)))
cnn.add(MaxPooling2D(pool_size=(2, 2)))
cnn.add(Flatten())
cnn.add(Dense(activation='relu', units=128))
cnn.add(Dense(activation='relu', units=64))
cnn.add(Dense(activation='sigmoid', units=1))
```

Figure 7: CNN Model design

The model was trained and validated, and the accuracy and loss graph were plotted using Matplotlib. The model was then be tested and saved for future use.

Apart from the trained model, the project entails and API interface developed on Django rest framework that the users can use to interact with the prediction model.

CHAPTER FOUR: RESULTS AND DISCUSSIONS

4.0 Introduction

This chapter highlights the data collected during the study, a detailed interpretation of the data and a discussion of the results, indicating the significance of the results and analyzing how the results help to answer the research question, all while considering all the limitations of the study.

4.1 Results

The model was exposed to 60 epochs and the accuracy of the model as well as other metrics were tested and monitored in three different instances: standard CNN without using augmentation, CNN with augmentation and CNN with transfer learning. Following the testing and monitoring, following findings noted:

4.1.1 Standard CNN (Without Augmentation)

```
test_accu = cnn.evaluate(test)
train_accu = cnn.evaluate(train)
print('The testing accuracy is :',test_accu[1]*100, '%')

print('The training accuracy is :',train_accu[1]*100, '%')
```

```
3/3 [==============================] - 1s 203ms/step - loss: 2.3294 - accuracy: 0.6750
9/9 [==============================] - 4s 401ms/step - loss: 5.2416e-06 - accuracy: 1.0000
The testing accuracy is : 67.5000011920929 %
The training accuracy is : 100.0 %
```

In the case of standard CNN without augmentation, the accuracy of 67.5% was attained, the precision of 70% and recall of 84% was observed.

Table 2: Standard CNN Metrics

Metric	**Value (%)**

Testing Accuracy	67.5%
Training Accuracy	100%
Precision	70.0%
Recall	84.0%

The training accuracy was noted to increase with increase of epochs; however, the testing accuracy did not increase with increase in epochs.

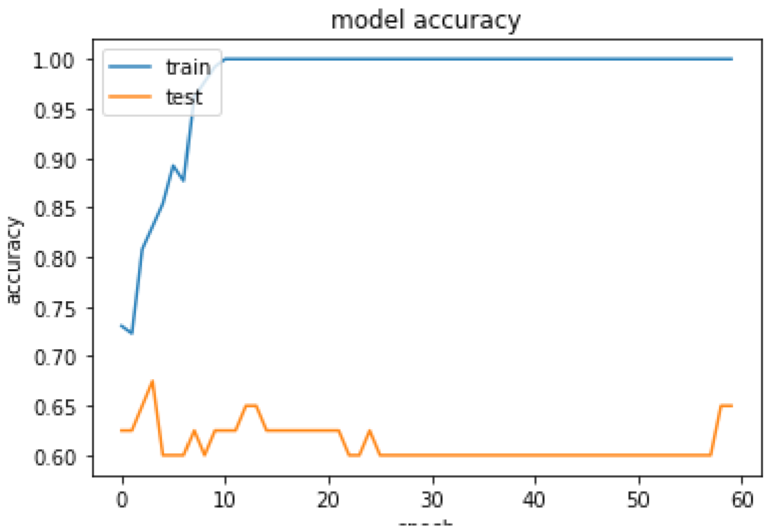


Figure 8: Model Accuracy (Standard CNN without Augmentation)

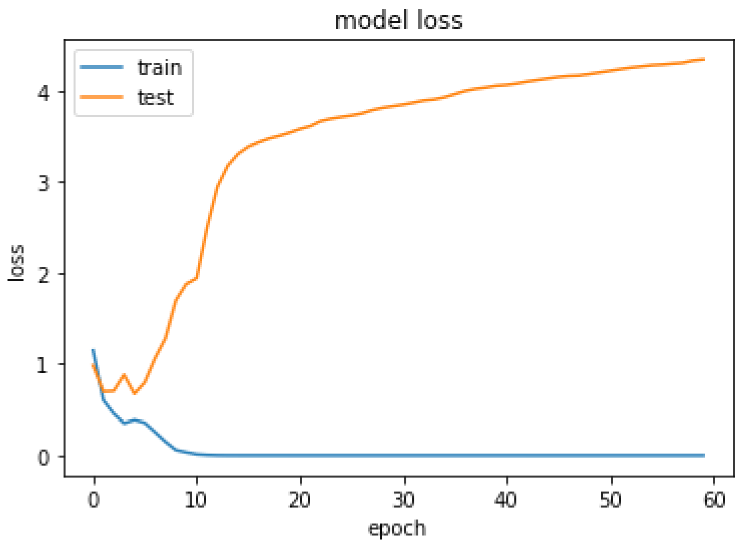


Figure 9: Model Loss (Standard CNN without Augmentation)

4.1.2 CNN With Augmentation

```
test_accu = cnn.evaluate(test)
train_accu = cnn.evaluate(train)
print('The testing accuracy is :',test_accu[1]*100, '%')

print('The training accuracy is :',train_accu[1]*100, '%')
```

```
3/3 [==============================] - 1s 260ms/step - loss: 0.6718 - accuracy: 0.7750
9/9 [==============================] - 4s 444ms/step - loss: 0.2250 - accuracy: 0.9077
The testing accuracy is : 77.49999761581421 %
The training accuracy is : 90.76923131942749 %
```

In the case of standard CNN with augmentation, the accuracy of 77.49% was attained, the precision of 72.72% and recall of 96.0% were observed.

Table 3: CNN with Augmentation Metrics

Metric	**Value (%)**
Testing Accuracy	77.49%
Training Accuracy	90.76%
Precision	72.72%
Recall	96.0%

This showed an increase in accuracy, precision and recall in comparison to the standard CNN without augmentation. The CNN accuracy against epoch for the train and test is plotted in the figure below.

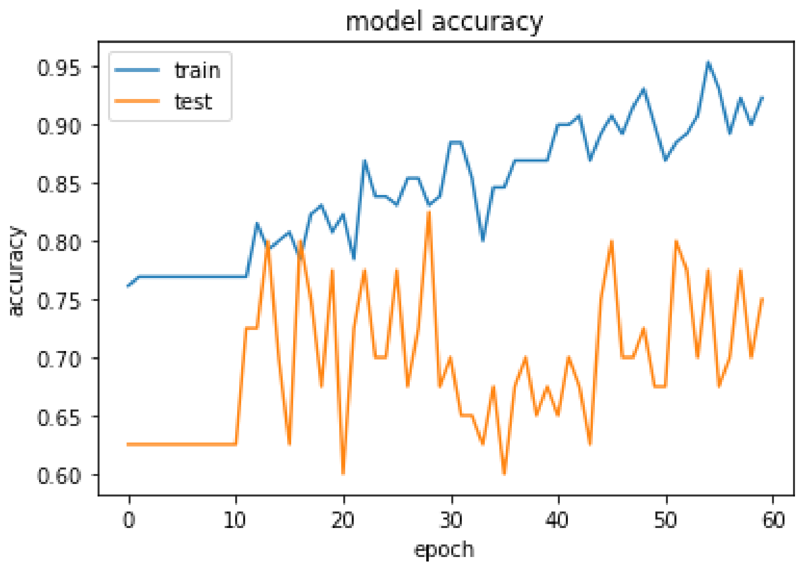


Figure 10: Model Accuracy (Standard CNN with Augmentation)

Based on the visualization, it is evident that accuracy for both the train and test samples increases with an increase in number of epochs. As accuracy increases, the model is portrayed as performing better based on the given data.

Plotting loss against epoch on the other hand shows a slight progressive decrease in loss with increase in the number of Epochs. Losses occur with bad prediction

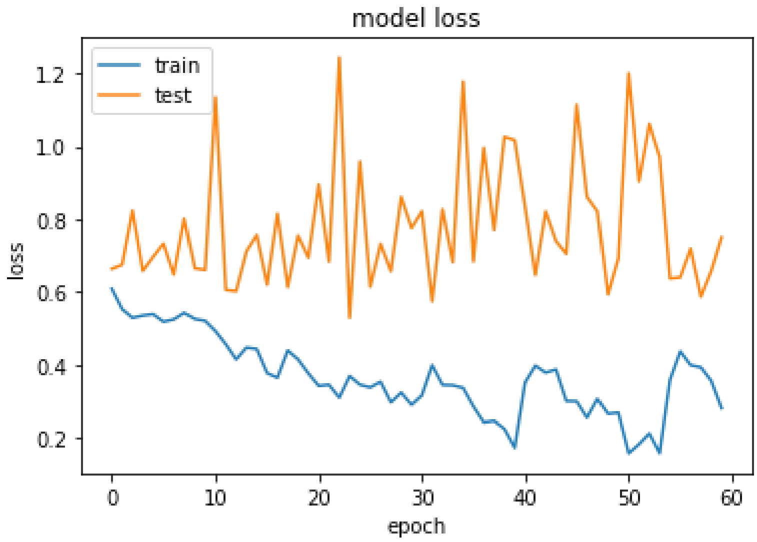


Figure 11: Model Accuracy (Standard CNN with Augmentation)

4.1.3 CNN with Transfer Learning

```
In [106]:  train_accu = model.evaluate(train_generator)
           test_accu = model.evaluate(test_generator)

           print('The training accuracy is :',train_accu[1]*100, '%')
           print('The testing accuracy is :',test_accu[1]*100, '%')

           5/5 [==============================] - 7s 1s/step - loss: 3.3929e-08 - accuracy: 1.0000
           2/2 [==============================] - 1s 309ms/step - loss: 2.4644 - accuracy: 0.8500
           The training accuracy is : 100.0 %
           The testing accuracy is : 85.00000238418579 %
```

In the case of standard CNN with transfer learning, an accuracy of 85% was attained, a precision of 96.18% and recall of 96.78% was observed.

Table 4: CNN with Transfer Learning Metrics

Metric	Value (%)
Testing Accuracy	85.00%

Training Accuracy	100%
Precision	96.18%
Recall	96.78%

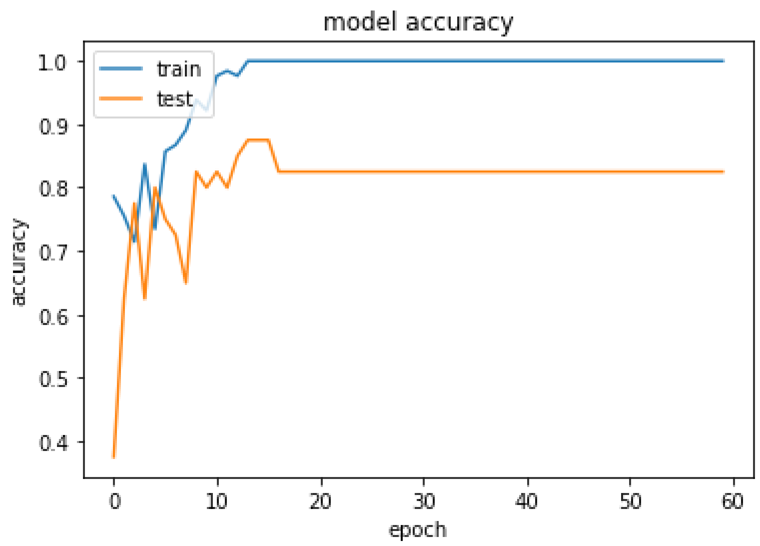


Figure 12: Model Accuracy (Standard CNN with Transfer Learning)

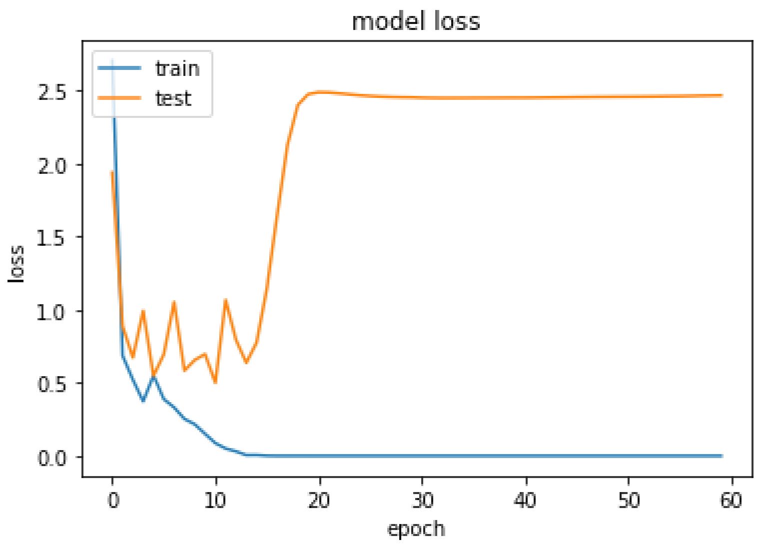


Figure 13: Model Loss (Standard CNN with Transfer Learning)

4.2 Discussion

The study's main objective was to develop a deep learning model to detect congenital talipes equinovarus in fetal ultrasound scans. A CNN classification model was used for the experiment. The different model metrics were monitored on alterations of the training data and model itself. Among the several alterations that were applied to help improve the performance of the model were: the use of data augmentation, as well as the use of transfer learning. ResNet, an artificial neural network (ANN) of that builds on constructs from pyramidal cells in the cerebral cortex was used in transfer learning. CNN model with transfer learning was selected for implementation because of its high accuracy as compared to the standard model.

The CNN model with transfer learning demonstrated an increasing accuracy with a reducing loss. It demonstrated considerable loss reduction as the number of epochs increased during training.

In addition to fetal heart rate and weight monitoring to ensure fetal wellbeing, the use of the proposed CNN model to detect fetal anomalies with an accuracy of 85% should be considered to make obstetrics safer. This gives the capability to interpret the status and features of the fetus from medical images.

CHAPTER FIVE: CONCLUSION AND RECOMMENDATIONS

5.1 Achievements

The overall objective was to develop a model that would detect congenital talipes equinovarus from fetal ultrasound scans. The research has generated a working model that is able to classify ultrasound scan as either with CTEV or Normal using CNN with transfer learning and data augmentation with a high-performance accuracy of 85.00%.

Additionally, an API has been created for future interaction with the model. On receiving a POST request, the API responds with the predicted label and accuracy percentage.

5.2 Limitations

One major limitation of the study was the hardship in getting a lot of data. CNN and all deep learning models require a large amount of data to train.

Another limitation of the study was in getting clean ultrasound scans from the online public repositories, most of the available scans had labels and wordings. It was challenging to remove the labels to avoid the model capturing them during the training process.

5.3 Further Work

There is a need to collect more data for effective classification. A mobile or web platform needs to be created in future to interact with the created API to help non-technical users interact with the prediction model and API. The model and API also need to be hosted on a cloud platform to facilitate faster processing.

References

Amy MiHyun Jang. (2020, 07 28). *Keras*. Retrieved from https://keras.io/examples/vision/xray_classification_with_tpus/

Arthur, L. S. (1959). Some Studies in Machine Learning Using the Game Checkers. *IBM Journal of Research and Development 3*.

Bhandari, A. (2020). Everything you Should Know about Confusion Matrix for Machine Learning. *Analytics Vidhya*.

Briony, J. O. (2006). *Researching Information Systems and Computing.* SAGE Publications Ltd.

Callen , K. O., & Norah, M. T. (2018). Screening for congenital fetal anomalies in low risk pregnancy: the Kenyatta National Hospital experience. *BMC Pregnancy and Childbirth*.

E, R., T, C., K, N., & N, T. L. (2020). Prediction of Fetal Distress Using Linear and Non-linear Features of CTG Signals. *Springer*.

EdwinOmo, E., Enemudo, R., Eghosa, M., & TopeJohnson, O. (2018). Pattern of congenital musculoskeletal anomalies in a suburban Nigerian tertiary hospital. *ResearchGate*.

Fairoz, Q. K., & Adnan, M. A. (2021). Ultrasound Medical Images Classification Based on Deep Learning Algorithms: A Review. *ResearchGate*.

Fanni, R. E., & Artúr , B. (2017). Congenital Fetal Anomalies and the Role of Prenatal Ultrasound. *IntechOpen*.

Feng, V. (2017, July). *Towards Data Science*. Retrieved from An Overview of ResNet and its Variants: https://towardsdatascience.com/an-overview-of-resnet-and-its-variants-5281e2f56035

Fentahun, A., Mekbeb, A., Girma, S., & Alemu, G. (2020). Prevalence and associated factors of birth defects among newborns in sub-saharan African counties: a systematic review and meta-analysis. *Pan African Medical Journal*.

Jing, T., Zhenming, Y., Li, S., Kai, Y., & Zhifen, Z. (2021). Fetal birthweight prediction with measured data by a temporal machine learning method. *BMC Medical Informatics and Decision Making*.

Kabir, S. M. (2016). METHODS OF DATA COLLECTION. *ResearchGate*.

Phillip, A. W., Emily, F. H., Robert, E. K., & Doina, P. (2012). A machine learning approach to the detection of Fetal Hypoxia during labor and delivery.

Ramírez, E. (2020, July 24). *ecode.Dev*. Retrieved from https://ecode.dev/cnn-for-medical-imaging-using-tensorflow-2/

Ruowei, Q., Guizhi, X., Chunxia, D., & Wenyan, J. (2019). Deep learning-bssed methodology for recognition of fetal brain standard scan planes in 2D ultrasound images. *ReserchGate*.

Stephen , T. C., & Daniel , W. S. (2017, July). *Contemporary OB/GYN*. Retrieved from https://www.contemporaryobgyn.net/view/using-ultrasound-recognize-fetal-anomalies-part-1

Steps Clubfoot Care. (2016). Retrieved from Steps Clubfoot Care: steps.org.za/about-clubfoot/

synopsys. (2017, March 28). Retrieved from https://www.synopsys.com/blogs/software-security/top-4-software-development-methodologies/

T, J. M., Marian, F. M., & Marie, E. T. (2015). *Infant Mortality Statistics from the 2013 Period Linked Birth/ Infant Death Data Set.* National Vital Statistics Reports.

World Health Organisation. (2020).

Appendices

Appendix 1: Code Listings

Code Listing 1: Define the directory path, import some needed libraries, and define some common constant parameters

```
import numpy as np
import pandas as pd
import seaborn as sns
from matplotlib import pyplot as plt
from sklearn.utils import compute_class_weight
from sklearn.metrics import classification_report, confusion_matrix
from tensorflow.python.keras.callbacks import EarlyStopping,
ReduceLROnPlateau
from tensorflow.python.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.layers import MaxPooling2D
from tensorflow.keras import Sequential
from tensorflow.keras.layers import Conv2D
from tensorflow.keras.layers import Flatten, Dense


def __init__(self):
    self.image_gen = ImageDataGenerator(
        rescale=1. / 255,
        shear_range=0.2,
        zoom_range=0.2,
        horizontal_flip=True,
    )
```

```
    self.test_data_gen = ImageDataGenerator(rescale=1. / 255)


    self.train_path = 'assets/model_images/train'

    self.test_path = 'assets/model_images/test'

    self.valid_path = 'assets/model_images/validation'

    self.batch_size = 16

    self.img_width = 500

    self.img_height = 500

    self.img_gen = None
```

Code Listing 2: Loading and converting images to grayscale

```
def load_train_data(self):

    train_data = self.image_gen.flow_from_directory(

        self.train_path,

        target_size=(self.img_height, self.img_width),

        color_mode='grayscale',

        class_mode='binary',

        batch_size=self.batch_size

    )

    return train_data


def load_test_data(self):

    test_data = self.test_data_gen.flow_from_directory(

        self.test_path,

        target_size=(self.img_height, self.img_width),

        color_mode='grayscale',

        shuffle=False,

        class_mode='binary',
```

```
        batch_size=self.batch_size

    )

    return test_data


def load_validation_data(self):

    valid_data = self.test_data_gen.flow_from_directory(

        self.valid_path,

        target_size=(self.img_height, self.img_width),

        color_mode='grayscale',

        class_mode='binary',

        batch_size=self.batch_size

    )

    return valid_data
```

Code List 3: Standard CNN Model

```
def design_model(self):

    cnn = Sequential()

    cnn.add(Conv2D(32, (3, 3), activation="relu",

input_shape=(self.img_width, self.img_height, 1)))

    cnn.add(MaxPooling2D(pool_size=(2, 2)))

    cnn.add(Conv2D(32, (3, 3), activation="relu",

input_shape=(self.img_width, self.img_height, 1)))

    cnn.add(MaxPooling2D(pool_size=(2, 2)))

    cnn.add(Conv2D(32, (3, 3), activation="relu",

input_shape=(self.img_width, self.img_height, 1)))

    cnn.add(MaxPooling2D(pool_size=(2, 2)))

    cnn.add(Conv2D(64, (3, 3), activation="relu",

input_shape=(self.img_width, self.img_height, 1)))
```

```
    cnn.add(MaxPooling2D(pool_size=(2, 2)))
    cnn.add(Conv2D(64, (3, 3), activation="relu",
input_shape=(self.img_width, self.img_height, 1)))
    cnn.add(MaxPooling2D(pool_size=(2, 2)))
    cnn.add(Flatten())
    cnn.add(Dense(activation='relu', units=128))
    cnn.add(Dense(activation='relu', units=64))
    cnn.add(Dense(activation='sigmoid', units=1))
    return cnn
```

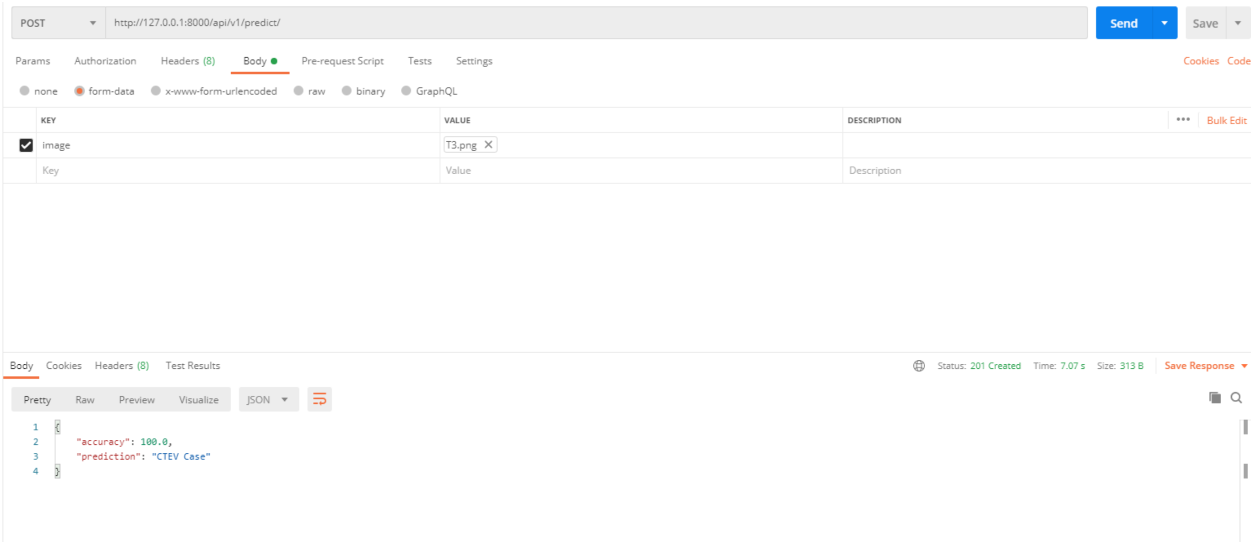
Code List 4: ResNet

```
base_model = tf.keras.applications.ResNet50(weights='imagenet',
include_top=False)
print(base_model.summary())

x = base_model.output
x = tf.keras.layers.GlobalAveragePooling2D()(x)
x = tf.keras.layers.Dense(1024, activation='relu')(x)
x = tf.keras.layers.Dense(1024, activation='relu')(x)
x = tf.keras.layers.Dense(1024, activation='relu')(x)
x = tf.keras.layers.Dense(512, activation='relu')(x)
preds = tf.keras.layers.Dense(2, activation='softmax')(x)
model = tf.keras.models.Model(inputs=base_model.input, outputs=preds)
```

Code List 5: Prediction

```python
def test(self, test_image_path=None):
    image_path = f"media/{str(test_image_path)}"
    img = image.load_img(image_path, target_size=(224, 224))
    x = image.img_to_array(img)
    x = np.expand_dims(x, axis=0)
    # x = tf.keras.applications.resnet50.preprocess_input(x)

    model = load_model('model/fetal_anomaly_model.h5py')

    predic = model.predict(x)[0].tolist()
    max_value = max(predic)
    max_index = predic.index(max_value)

    if max_index == 0:
        predicted_label = 'CTEV Case'
    elif max_index == 1:
        predicted_label = 'Normal Case'

    return predicted_label, max_value * 100
```

Code List 5: API call


POST
http://127.0.0.1:8000/api/v1/predict/
Send
Save
Params
Authorization
Headers (8)
Body
Pre-request Script
Tests
Settings
Cookies
Code
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
KEY
VALUE
DESCRIPTION
Bulk Edit
image
T3.png
Key
Value
Description
Body
Cookies
Headers (8)
Test Results
Status: 201 Created
Time: 7.07 s
Size: 313 B
Save Response
Pretty
Raw
Preview
Visualize
JSON
{
"accuracy": 100.0,
"prediction": "CTEV Case"
}

Appendix 2: Budget

Table 5: Project Budget

#	Line Item	Description	Line Cost	Budget (KES)
Travel Costs / Airtime & Bundles				
1	Travel Costs	2 trips for expert interviews,1 trip for de-identifying and picking of data,6 trips for installation and testing	@2,000	16,000
2	Bundles & Airtime	Communication with expertise (6 times)	6,000	6,000
Model / Prototype Development				
3	Laptop	1 laptop for development and testing	50, 000	50,000
4	Clinical Validations (x2)	Testing and feedback on models programmed (3 times)	2,000	12,000
5	Thumb Drive (32 GB)	Collection of data samples.	3,000	3,000
6	Hard Drive (500GB)	Storage of data samples	6,000	6,000
Dissemination				
7	Preparation of dissemination materials		5,000	5,000
			TOTAL	**97,000**

Table 6: Project Schedule

Task	**Start Date**	**End Date**	**Duration (days)**
Preliminary studies and project proposal	15 February 2021	6 Apr 2021	50
Data acquisition and preparation	6 April 2021	15 May 2021	39
Modelling	5 May 2021	5 Jun 2021	31
Training the model	5 June 2021	27 Jun 2021	22
Testing the model	27 June 2021	10 Jul 2021	13
Documentation	6 July 2021	25 Jul 2021	19

Appendix 3: Schedule

The figure below represents a Gantt chart of how this project is planned to be tackled.

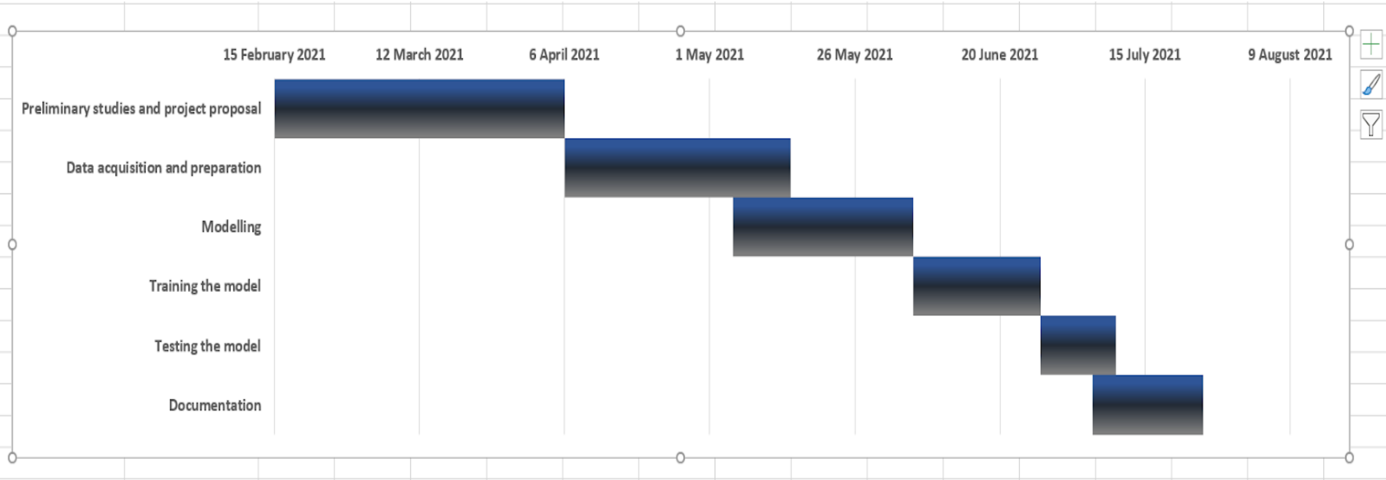


Figure 14: Gantt Chart

Appendix 4: Resources

The following resources will be required in the project:

1. Internet Access
2. Airtime
3. Laptop
4. Thumb drive
5. Hard drive

Appendix 5: Plagiarism Check Report

FETAL ANOMALIES DETECTION USING CONVOLUTIONAL NEURAL NETWORKS

ORIGINALITY REPORT

10%	5%	5%	6%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Institute of Management Technology Student Paper	1%
2	repository.out.ac.tz Internet Source	1%
3	Ruowei Qu, Guizhi Xu, Chunxia Ding, Wenyan Jia, Mingui Sun. "Deep Learning-Based Methodology for Recognition of Fetal Brain Standard Scan Planes in 2D Ultrasound Images", IEEE Access, 2020 Publication	1%
4	Submitted to Eiffel Corporation Student Paper	1%
5	www.investopedia.com Internet Source	1%
6	Petros Kalantonis, Andreas Errikos Delegkos, Emmanouela Sotirchou, Aristidis Papagrigoriou. "Modern business development and financial reporting:	1%

	exploring the effect of corporate governance on the value relevance of accounting information—evidence from the Greek listed firms", Operational Research, 2021 Publication	
7	www.tzonline.org Internet Source	1 %
8	Kathleen Rice Simpson. "Emerging Trends in Perinatal Quality and Risk With Recommendations for Patient Safety", Journal of Perinatal & Neonatal Nursing, 2018 Publication	<1 %
9	Submitted to University of Birmingham Student Paper	<1 %
10	Submitted to University of Greenwich Student Paper	<1 %
11	docplayer.net Internet Source	<1 %
12	Submitted to New England Institute of Technology Student Paper	<1 %
13	Submitted to San Joaquin Valley College Student Paper	<1 %
14	web.wpi.edu Internet Source	<1 %

15	Mehdi Fasihi, Mohammad H. Nadimi-Shahraki, Ali Jannesari. "A Shallow 1-D Convolution Neural Network for Fetal State Assessment Based on Cardiotocogram", SN Computer Science, 2021 Publication	<1 %
16	Laith Alzubaidi, Jinglan Zhang, Amjad J. Humaidi, Ayad Al-Dujaili et al. "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions", Journal of Big Data, 2021 Publication	<1 %
17	baadalsg.inflibnet.ac.in Internet Source	<1 %
18	scholar.mzumbe.ac.tz Internet Source	<1 %
19	www.mdpi.com Internet Source	<1 %
20	Antonio Guadix, Jose E. Zapata, M. Carmen Almecija, Emilia M. Guadix. "Predicting the flux decline in milk cross-flow ceramic ultrafiltration by artificial neural networks", Desalination, 2010 Publication	<1 %
21	J. F. Mas. "The application of artificial neural networks to the analysis of remotely sensed	<1 %

data", International Journal of Remote Sensing, 2/2008
Publication

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		