Open University of Catalonia (UOC) Master's Degree in Data Science

MASTER'S THESIS

Area: Medicine area (TFM-Med)

# Machine Learning in materno-fetal ultrasound images for early detection of late-onset placental insufficiency

---


Author: Gwendolin Herrera Carballido

Tutor: Xavier Paolo Burgos Artizzu

Professor: Laia Subirats


---

Barcelona, January 9, 2024

# Credits/Copyright

A page with the specification of credits/copyright for the project (either application on one side and documentation on the other, or unified), as well as the use of third-party trademarks, products or services (including source code). If a person other than the author collaborated on the project, their identity and what they did must be explicitly stated.

Below is the most common case, but it can be modified for any other alternative:

# FINAL PROJECT RECORD

| | |
|---|---|
| Title of the project: | Machine Learning in materno-fetal ultrasound images for early detection of late-onset placental insufficiency |
| Author's name: | Gwendolin Herrera Carballido |
| Collaborating teacher's name: | Xavier Paolo Burgos Artizzu |
| PRA's name: | Laia Subirats |
| Delivery date (mm/yyyy): | 01/2024 |
| Degree or program: | Master's Degree in Data Science |
| Final Project area: | Medicine Area (TFM-Med) |
| Language of the project: | English |
| Keywords | Placental insufficiency, Computer Vision, Ultrasound imaging |

"I wasn't brave, I just didn't have time to be scared."

— Amelia Earhart

# Abstract

Late-onset placental insufficiency (LOPI) is a condition in which the placenta does not provide enough oxygen and nutrients to the fetus after 32 weeks of gestation that can lead to severe complications, including preeclampsia, fetal growth restriction (IUGR) and stillbirth. It complicates 10-15% of pregnancies and its non-detection represents 10% of cases of preventable perinatal death, increasing the risk of poor maternal and neonatal outcomes by 4 times.

Early detection and treatment is critical to reduce the risk of these complications. However, LOPI is currently difficult to detect in early pregnancy, and it is usually detected by the presence of its complications when it's already in an advanced clinical stage. Developing better ways to predict which women are at increased risk of LOPI is essential to create opportunities for secondary or tertiary preventive measures.

In recent years, machine learning and deep learning have emerged as new approaches for clinical diagnosis. Algorithms are trained on large datasets of medical images and clinical data to identify patterns and make predictions, for more accurate and efficient diagnosis of diseases and conditions.

Research has shown that 3D MRI placental images can be used to predict placental-related conditions such as fetal growth restriction (FGR). However, MRI scans are expensive, time-consuming, invasive, and not widely available, resulting in long waiting lists. Ultrasound, on the other hand, is a safe and accessible imaging technique that is available in most hospitals and clinics. Therefore, if ultrasound images can be used to predict placental insufficiency, it would be a significant advance.

This study aims to apply machine learning, computer vision, and deep learning techniques to a set of labeled maternal-fetal ultrasound images obtained between weeks 27 and 29 of pregnancy to determine if they can be used to predict which women are at risk of LOPI before the condition develops.

The results obtained with a sample of 354 women, with 121included in the test group to ensure statistical significance, were not optimal. The analysis fell short of achieving a minimum sensitivity of 0.5, set within a specified minimum specificity of 1 - the prevalence of the studied condition. The images encompassed both anterior and posterior planes of the placenta, with the latter demonstrating reduced visibility.

While results for the detection of preeclampsia alone demonstrated satisfactory sensitivity when evaluated on anterior planes, the dataset available considering anterior planes only was insufficient to ensure statistical significance. A reevaluation after collecting more data with additional anterior planes is recommended for a more comprehensive and robust analysis.

**Keywords:** placental insufficiency; deep learning; computer vision; machine learning; ultrasound imaging; clinical diagnosis.

# Resumen

La insuficiencia placentaria tardía es difícil de detectar y puede causar complicaciones graves, como la preeclampsia y la restricción del crecimiento fetal (RCF). Complica un 10-15% de las gestaciones y su no detección representa el 10% de los casos evitables de muerte perinatal y aumenta 4 veces el riesgo de malos resultados maternos y neonatales.

La detección temprana y el tratamiento son fundamentales para reducir el riesgo de estas complicaciones. Sin embargo, actualmente es difícil detectar esta condición a tiempo, y suele diagnosticarse por la presencia de sus complicaciones cuando ya se encuentra en una fase clínica avanzada. Desarrollar mejores formas de predecir qué mujeres tienen un mayor riesgo de padecerla es esencial para dar espacio a la prevención secundaria o terciaria.

En los últimos años, machine learning y deep learning han surgido como nuevas herramientas para el diagnóstico clínico. Los algoritmos se entrenan en grandes conjuntos de datos de imágenes médicas y datos clínicos para identificar patrones y realizar predicciones, con el fin de lograr un diagnóstico más preciso y eficiente de diferentes enfermedades y condiciones.

Investigaciones han demostrado que imágenes de placenta en 3D por resonancia magnética (MRI) se pueden utilizar para predecir condiciones relacionadas con la placenta. Sin embargo, las resonancias magnéticas son caras, difíciles de implementar, invasivas y no cuentan con gran disponibilidad, dando lugar a largas listas de espera. La ecografía, por el contrario, es una técnica de imagen segura y accesible que está disponible en la mayoría de hospitales y clínicas. Por lo tanto, supondría un gran avance si las imágenes de ecografía pudieran utilizarse para predecir la insuficiencia placentaria.

El propósito de este trabajo es aplicar técnicas de machine learning, computer vision y deep learning a un conjunto de imágenes de ecografía etiquetadas, obtenidas entre las semanas 27 y 29 de embarazo, para determinar si se pueden utilizar para predecir qué mujeres están en riesgo de sufrir insuficiencia placentaria tardía.

Los resultados obtenidos con una muestra de 354 mujeres, de las cuales 121 fueron incluidas en el grupo de test para garantizar significancia estadística, no fueron óptimos. El análisis no logró alcanzar una sensibilidad mínima del 0.5, establecida dentro de una especificidad mínima deseada del 1 - la prevalencia de la condición estudiada. Las imágenes abarcaron tanto los

planos anterior como posterior de la placenta, siendo reducida la visibilidad de este último.

Mientras que los resultados para la detección de la preeclampsia de forma aislada demostraron una sensibilidad satisfactoria al evaluarse en los planos anteriores, el conjunto de datos disponible considerando solo los planos anteriores resultó insuficiente para garantizar significancia estadística. Se recomienda una reevaluación después de recopilar más datos con planos anteriores adicionales para realizar un análisis más completo y sólido.

**Palabras clave:** insuficiencia placentaria; aprendizaje profundo; visión por computadora; aprendizaje automático; ecografía; diagnóstico clínico. MR

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

## 1.1 Overview and rationale

### 1.1.1 Problem overview

The placenta is the highly specialised organ of pregnancy that supports the normal growth and development of the fetus. It acts to provide oxygen and nutrients to the fetus, whilst removing carbon dioxide and other waste products. It metabolises a number of substances and can release metabolic products into maternal and/or fetal circulations [1].

Placental insufficiency is a condition in which the placenta does not function properly, failing to deliver enough oxygen and nutrients to the fetus. This leads to adverse pregnancy outcomes like fetal growth restriction and preeclampsia. Late-onset placental insufficiency is a type of placental insufficiency that occurs after 32 weeks of pregnancy. Preeclampsia, in particular, is one of the most feared complications of pregnancy. Often presenting as new-onset hypertension and proteinuria during the third trimester, preeclampsia can progress rapidly to serious complications, including death of both mother and fetus [2]. Fetuses with FGR are more likely to have health problems at birth, such as low birth weight, premature birth, and breathing problems.

The assessment of placental insufficiency is currently based on repetitive ultrasonography of the placenta and the fetus and Doppler ultrasonography of the umbilical vessels. However, the positive predictive value of ultrasonography in placental insufficiency barely exceeds 50% [3]. In many cases, placental disease may not be suspected until well after complications of fetal growth have occurred [4].

Early detection of placental insufficiency is crucial, as it allows for close monitoring of the fetus and mother, which can help to prevent or minimize the serious complications associated with this condition.

### 1.1.2 Rationale

In recent years, data science has been successfully applied to the diagnosis of patients through medical imaging, with deep learning and computer vision techniques. Computers are learning to interpret MRI's, X-Rays, ultrasound images and other types of images more accurately, and they can detect microscopic deformities that lead to identify anomalies that could easily go undetected.

Ultrasound imaging is a non-invasive method that can be used to assess placental function. However, changes might not be appreciable when it comes to detection of late-onset placental insufficiency. Data science techniques such as machine learning, computer vision, and deep learning have the potential to improve the accuracy of ultrasound-based LOPI detection. By applying these techniques, we hope to identify patterns and features associated with LOPI as early as within 27-29 weeks of pregnancy, allowing for timely preventive and therapeutic measures to improve outcomes for mothers and fetuses. Despite the potential benefits of ML for LOPI detection, research on this specific problem is lacking. Placental textures obtained by MRI have been shown to be useful in detecting placental insufficiency. However, MRI scans are expensive, time-consuming, invasive, and not widely available, resulting in long waiting lists. Ultrasound, on the other hand, is a safe and accessible imaging technique that is available in most hospitals and clinics. Therefore, if ultrasound images can be used to predict placental insufficiency, it would be a significant advance.

Therefore, this project aims to determine whether ultrasound imaging can be useful to predict LOPI. Overall, the development of accurate and reliable ML-based LOPI detection algorithms based on ultrasound images, due to their accessibility, would have a significant impact on clinical practice and improve outcomes for women and their babies.

## 1.2 Objectives and methodology

### 1.2.1 Objective

The main objective of this project is to use machine learning techniques to identify patterns in placental textures from ultrasound images obtained between weeks 27 and 29 of pregnancy that are associated with late-onset placental insufficiency, in order to distinguish if a given placental texture is indicative of placental insufficiency or not.

### 1.2.2 Methodology

This project involved collaboration with the Preeclampsia and Intrauterine Growth Restriction Research Unit at the Sant Joan de Déu Hospital in Barcelona, where patient recruitment and ultrasound image acquisition were conducted. We used a dataset of ultrasound materno-fetal images of patients between 27 and 29 weeks of pregnancy. The images were manually segmented by the medical professionals who are part of the research team and labeled according to whether or not the patient developed placental insufficiency, as determined during childbirth. All patients included in the dataset were of legal age and were assessed as high risk of intrauterine growth restriction (IUGR) on routine ultrasound during the second trimester. At the time of inclusion, there were no congenital malformations detected and the fetus was alive. We have processed the images using classic computer vision algorithms to extract features, followed by a variety of machine learning algorithms, including deep learning techniques. We have trained several different models and selected the best performer for further optimization and parameter hyper-tuning. Since we were trying to determine whether or not there is a risk of developing late-onset placental insufficiency, we focused on optimizing sensitivity given a minimum of specificity that must be met, which equals 1 - the prevalence of the condition. For the results to have clinical relevance, it was agreed that a minimum sensitivity of 0.50 was required.

### 1.2.3 Planning

In general terms, this is the planning for the different phases of the project:

A. Research about the condition and defining approach. (13 days)

B. Research on the approach: 1) technical: binary classification with images, computer vision and deep learning algorithms; 2) clinical: what has been done before and what tools are available. (13 days)

C. Implementation: processing of images, data preparation, running different algorithms, evaluation, analysis of results and final conclusions. (55 days)

D. Preparation of final report and presentation. (27 days)

### 1.2.4 Personal motivation

There are several reasons why I am motivated to undertake this project. First, I am excited by the potential of Machine Learning to revolutionize many industries, including healthcare.

Personally, I have experience working with supervised and unsupervised learning on tabular datasets, but I am eager to expand my skills into image processing, computer vision, and deep learning. Furthermore, I believe that clinical diagnosis is a field where machine learning, computer vision, and deep learning can have a profound and positive impact on society. These technologies have already made significant advances in the speed and accuracy of medical diagnoses and they have the potential to keep evolving. Finally, I am motivated by the fact that women's health issues have been historically under-researched and have received inadequate attention and funding. Because conditions that primarily affect women have often been overlooked, it was important for me to put my effort and focus on a condition that affects women, such as late-onset placental insufficiency, object of this study. The research aspect of this project and lack of similar implementations to this date is also something that I find highly motivating.

# Chapter 2

# State of the art

## 2.1 Overview

Computer vision and machine learning have developed in tandem over the past few decades, and are two of the most rapidly developing fields in technology today. Their evolution has led to significant advances in a wide range of industries, from healthcare to transportation. As these fields continue to evolve, we can expect to see even more groundbreaking innovations in the years to come.

The origin of both fields can be traced back to the 1950s, when scientists discovered that image processing begins with simple shapes like straight edges, and when the "Turing test" was created to test a machine's ability to exhibit intelligent behavior indistinguishable from that of a human[1].

AI emerged as an academic field in the 1960s, with advances in computer vision and machine learning in the 1970s and 1980s. In the 2000s, computer vision focused on object recognition, aided by datasets like ImageNet, leading to the rise of Convolutional Neural Networks (CNNs[5]) and deep learning. Machine learning saw a resurgence with the growth of distributed processing. In 2012, AlexNet[6] marked a pivotal moment in computer vision, significantly reducing image recognition errors and showcasing deep learning's effectiveness[2].

During the 2000s and 2010s, deep learning, especially CNNs[7][8][9], significantly enhanced object recognition accuracy in computer vision, while machine learning applications expanded due to Big Data and distributed processing.

Today, computer vision is a fast-growing field with diverse industry applications, poised to advance further alongside artificial intelligence progress. Some of the major advances in computer vision that have occurred in the past decade are:

[1] https://www.futurespace.es/en/machine-learning-los-origenes-y-la-evolucion/
[2] https://www.ibm.com/topics/computer-vision

• **Convolutional Neural Networks (CNNs)**[5][7][8][9]: CNNs have evolved and have revolutionized the accuracy and performance of computer vision systems. They enable tasks such as object recognition, image classification, and segmentation to be performed with unprecedented accuracy and speed. This technology is now being used in a wide range of applications, such as self-driving cars, medical imaging, and social media.

• **Generative Adversarial Networks (GANs)**[10]: GANs have the ability to generate realistic images by training a generator network to produce images that are indistinguishable from real images, while simultaneously training a discriminator network to differentiate between real and generated images. This technology has applications in various domains, including image synthesis[11], image editing[12], and even deepfake detection[13].

• **Natural Language Processing (NLP[14]) and reinforcement learning[15]**: the fusion of computer vision with other technologies like NLP and reinforcement learning allows computer vision systems to understand and interpret textual information related to images, enabling tasks such as image captioning[16] and visual question answering[17].

• **Hardware**: the deployment of computer vision technology has been greatly facilitated by the availability of powerful hardware, such as graphics processing units (GPUs) and specialized vision processing units (VPUs). These hardware advancements have significantly accelerated the computation required for complex computer vision algorithms, making real-time applications feasible.

## 2.2 Computer vision and machine learning applied to medicine

### 2.2.1 Types of medical imaging

There are different types of medical images to help professionals with diagnosis and treatment of different health conditions, chosen according to the type of injury or disease. Out of those, three types commonly used in machine learning are X-rays, MRIs, and ultrasound images.

#### X-ray

X-rays are 2D images obtained using radiation of the body's dense tissues, including bones.

The most familiar use of X-rays is checking for bone fractures[18], but X-rays are also used to check for dislocated joints or lung diseases such as pneumonia. Mammograms also use X-rays to look for breast cancer[19].

X-ray imaging is quick, non-invasive, and painless. However, the ionising radiation exposure

Figure 2.1: Example of X-ray image. Credit: iStock[20]

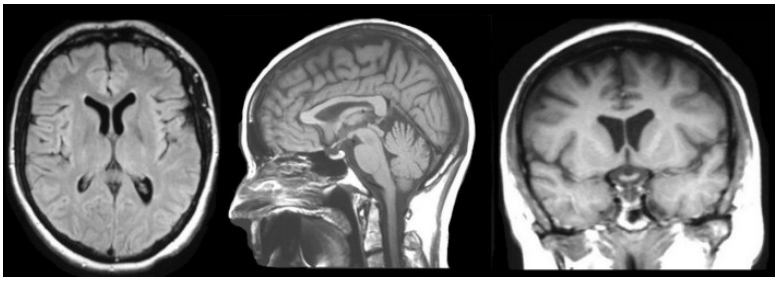

Figure 2.2: Example of MRI image. Credit: Case Western Reserve University[22]

has the potential to cause harm and increases the risk of cancer. As well, they offer limited ability to differentiate between different types of soft tissue, and X-rays are advised against pregnant women due to the potential harm they may cause to the fetus.

### MRI

MRI images are typically two-dimensional (2D) or three-dimensional (3D) cross-sectional slices of the body, obtained by powerful magnetic field and radio waves, that provide precise and detailed images of the body's soft tissues and almost every internal structure in the human body.

MRI is generally used to identify tumors, multiple sclerosis, and stroke-related issues with the brain and spinal cord. It can also be used to diagnose issues in the muscles, joints, and other soft tissues[21]. MRI technique is non-invasive for the patient and is considered safe, since it does not use ionizing radiation. However, it is more expensive and time-consuming than X-ray imaging, and not all patients can have an MRI due to metal implants or other factors.

### Ultrasound

An ultrasound image is a (usually 2D) visual representation of the internal organs and soft tissues of the body created using ultrasound technology. The quality may be lower than MRI, and it is often used for more superficial structures.

It is most commonly used to track the growth and development of foetus during pregnancy. Additionally, it can be used to detect issues related to the heart, blood vessels, abdomen, and pelvis.

It is a non-invasive and painless procedure, and safe for the patient as it does not use ionizing radiation. It is also a real-time imaging technique, allowing the healthcare provider to see the images immediately as they are being captured. One of the main limitations is that it is highly operator-dependent. The quality of the ultrasound images can vary depending on the skills and experience of the technician performing the procedure. This means that the accuracy and reliability of the results may be influenced by the operator's expertise. Another disadvantage of ultrasound is that it has limited penetration through bone and air. This makes it less effective for imaging certain areas of the body, such as the lungs or bones. Additionally, ultrasound may not provide as detailed images as other imaging techniques like MRI or CT scans. It may not be able to detect small abnormalities or provide precise anatomical details in some cases.

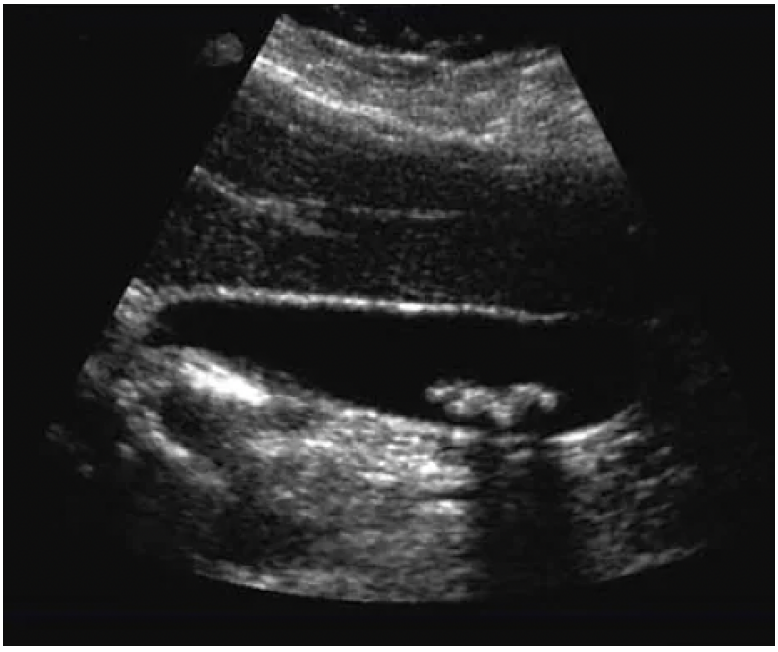

Figure 2.3: Example of ultrasound image. Credit: mayoclinic[23]

### 2.2.2 Classical image processing techniques

#### Histogram of Oriented Gradients (HOG)

The Histogram of Oriented Gradients (HOG)[24] is a computer vision technique that captures the gradient orientations within a defined area through the construction of a histogram. The concatenation of these histograms, derived from distinct areas or cells within an image, forms a representation of the entire image. Additionally, each cell undergoes a normalization process, taking into account its surrounding neighborhood, referred to as a block. Finally, a feature vector that captures the shape and texture of the image based on gradient information is obtained. The feature vector is flattened to transform it into a 1D array suitable for machine learning algorithms.

In the original work from Dalal et al. that introduced the HOG method, the cells were originally configured with dimensions of 8x8 pixels, and these cells were organized into blocks with a size of 2x2 cells.

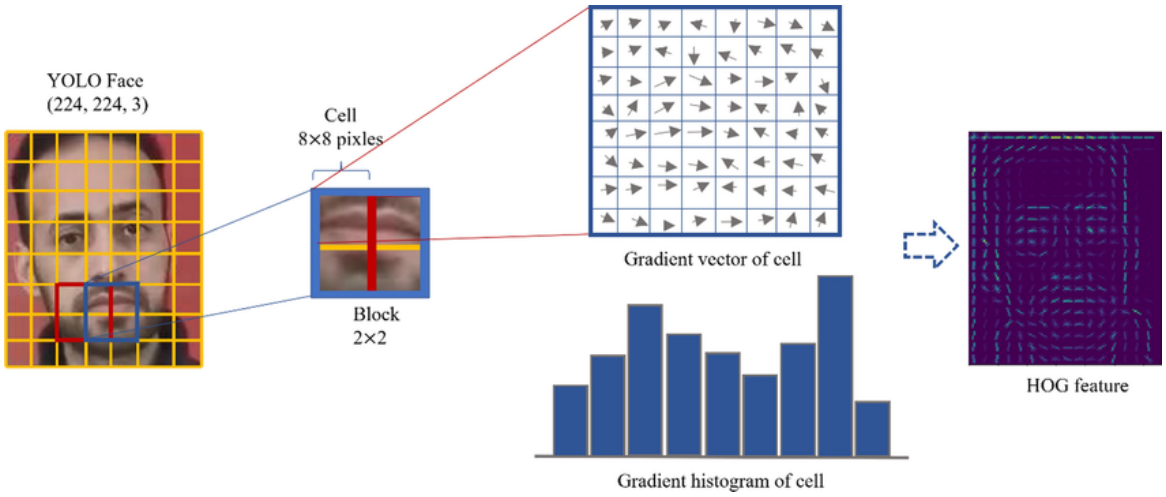


Figure 2.4: HOG illustration. Credit: Neural Computing and Applications[25].

#### Gray-Level Co-occurrence Matrix (GLCM)

The Gray-Level Co-occurrence Matrix (GLCM)[26] is a statistical tool for analyzing the relationships between pixels in images. It was first introduced by Haralick et al. in 1973. It calculates how pixel intensities occur together within specific spatial regions, to produce a matrix that serves as a basis for characterizing different texture properties, like homogeneity, contrast, and roughness.

First, a spatial neighborhood is defined, which is the area around the pixel that will be considered when calculating the co-occurrences of gray levels. The authors stablished this

area should be small enough to be representative of local texture, but large enough to capture significant spatial relationships. A common choice is a square neighborhood with a side length of 3 or 5 pixels.

Then, for each pair of gray levels in the spatial neighborhood, the number of times that two pixels with those gray levels occur adjacent to each other are counted and stored in the GLCM matrix.

Angles are used to determine the direction of the spatial relationship between pixels. For example, a GLCM angle of 0 degrees indicates that two pixels are adjacent to each other horizontally, while a GLCM angle of 90 degrees indicates that they are adjacent to each other vertically.

Once the GLCM matrix is calculated, an array of texture features can be extracted from it. These texture features can be used to describe the texture characteristics of the image. Some common texture features extracted from GLCMs include homogeneity (a measure of the degree to which the gray levels in the image are uniform, contrast (a measure of the difference in gray levels between adjacent pixels), energy (a measure of the overall intensity of the texture in the image) or correlation (a measure of the linear relationship between the gray levels of adjacent pixels).

This method is used in applications such as image segmentation (e.g., dividing a human body image into its components), texture classification (e.g., distinguishing between materials like wood), and image quality assessment.

### 2.2.3 CNN for image processing

In the realm of deep learning for image processing, several widely adopted network architectures stand out. Here, we'll introduce five of the most prevalent ones.

#### AlexNet

This architecture was developed by Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton in 2012[6]. It gained widespread attention by winning the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2012, significantly advancing the field of deep learning.

Its architecture, although inspired by LeNet[5], featured significant improvements: increased depth, more filters per layer, and stacked convolutional layers. Key components included 11x11, 5x5, and 3x3 convolutions, max pooling, dropout[27], data augmentation, ReLU activations[28], and the use of SGD[29] with momentum. Moreover, it incorporated ReLU activations after each layer.

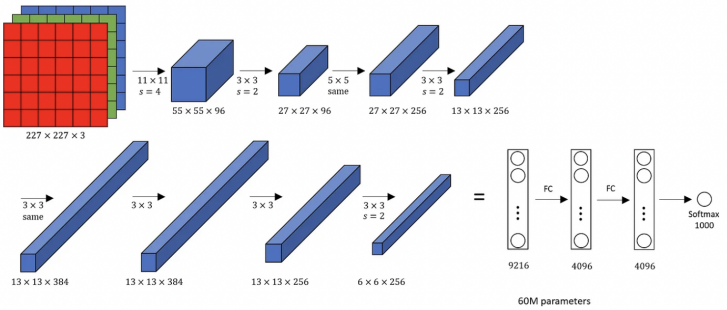


Figure 2.5: AlexNet architechture. Credit: datahacker.rs[30]

**VGGNet**

This architecture that made significant contributions to the field was developed by Simonyan and Zisserman[7] and got the second place in the ILSVRC 2014 competition[31].

VGGNet stands out due to its remarkably uniform architecture, consisting of 16 convolutional layers. It shares similarities with AlexNet, employing 3x3 convolutions but with a more substantial number of filters. The network was trained on 4 GPUs for 2-3 weeks and has become a top choice within the community for image feature extraction. Notably, the weight configuration of VGGNet is publicly available and serves as a baseline feature extractor in various applications and challenges. However, it's important to note that VGGNet's 138 million parameters can pose challenges in terms of computational demands.

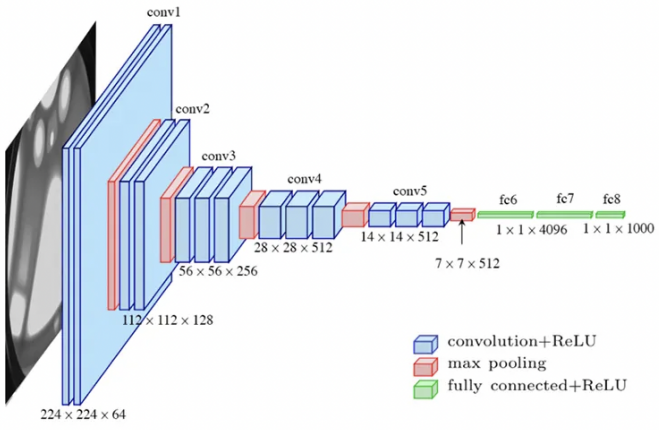


Figure 2.6: VGG-16 architechture. Credit: Max Ferguson in ResearchGate[32]

**GoogLeNet (Inception)**

GoogLeNet, also recognized as Inception V1[9], emerged as the champion of the ILSVRC 2014 competition, boasting an impressive top-5 error rate of merely 6.67%. This level of performance approached human-like accuracy, and surpassing it required the intervention of human

expertise. Within a few days of expert Andrej Karpathy's training efforts, he achieved a remarkable top-5 error rate of 5.1% for a single model and 3.6% for an ensemble.

What sets GoogLeNet apart is its architectural innovation, drawing inspiration from LeNet but introducing a revolutionary element called the "inception module." This module leveraged techniques like batch normalization, image distortions, and RMSprop. Most notably, it adopted a strategy of employing numerous small convolutions, resulting in a substantial reduction of parameters. Despite sporting a deep 22-layer CNN architecture, GoogLeNet managed to shrink the parameter count from 60 million in AlexNet to a mere 4 million, marking a significant leap in both efficiency and performance.

**ResNet (Residual Networks)**

In the ILSVRC 2015 competition, Kaiming He and his team introduced a groundbreaking innovation: the Residual Neural Network (ResNet)[8]. This novel architecture incorporated a technique known as "skip connections," which also goes by the names "gated units" or "gated recurrent units." These skip connections bore a striking resemblance to successful components found in Recurrent Neural Networks (RNNs)[33][34].

By integrating these skip connections with robust batch normalization, they achieved the remarkable feat of training a neural network with an unprecedented depth of 152 layers. This depth surpassed previous models like VGGNet while maintaining lower model complexity.

The results were equally remarkable. ResNet achieved an exceptional top-5 error rate of merely 3.57%. In fact, this performance exceeded human-level accuracy on the dataset, marking a significant milestone in the field of deep learning.

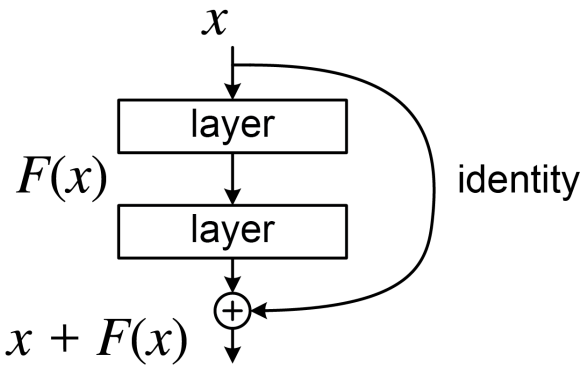


Figure 2.7: A Residual Block in a deep Residual Network. Here the Residual Connection skips two layers. Credit: Wikipedia[35]

**DenseNet (Densely Connected Convolutional Networks)**

DenseNet[36] stands out as another remarkable architectural advancement that addressed the challenges posed by extremely deep networks and the vanishing gradient problem.

The distinguishing feature of DenseNet's innovative architecture is its dense connectivity pattern. Unlike conventional convolutional neural networks (CNNs), where information typically travels through layers in a sequential manner, in DenseNet, every layer directly receives input from all preceding layers. This dense interconnection facilitates an efficient information flow, encourages the reuse of features, and simplifies the training of exceptionally deep networks.

Thanks to this pioneering approach, DenseNet delivered impressive outcomes. It achieved a significant reduction in the number of parameters while maintaining high-performance levels. In essence, it managed to achieve more with fewer parameters, positioning it as a valuable addition to the landscape of deep learning.

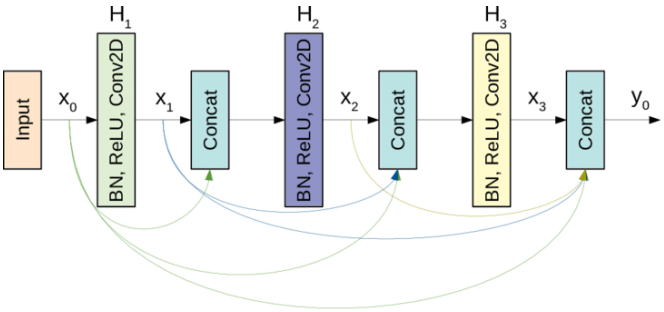


Figure 2.8: A schematic diagram of a 3-layer dense block used in the DenseNet architecture. Source: 10.7717/peerjcs.655/fig-3[37]

### 2.2.4 Machine learning for medical image diagnosis

In the last years there have been some notable examples of successful applications of computer vision and deep learning to medical imaging. For example, pneumonia can be detected from lung X-rays with a deep transfer learning-based classification ensemble, incorporating GoogLeNet, ResNet-18, and DenseNet-121 architectures[38]. Brain tumor can be diagnosed from brain MRI with 97.87% accuracy through the application of Convolutional Neural Networks (CNNs)[39]. Thyroid nodule classification[40] from ultrasound images has achieved an impressive accuracy of 97.63% by leveraging Convolutional Neural Networks (VGG16, EfficientNetB0[41], ResNet50) and Vision Transformers (ViT_B16 and Hybrid ViT[42]).

### 2.2.5 Clinical diagnosis through maternal-fetal ultrasound image

Deep learning (DL) and computer vision applications in the realm of ultrasound images have faced limited adoption. However, a few studies have shown promise in applying transfer learning techniques with Convolutional Neural Networks (CNNs) to categorize ultrasound images. One such study by Cheng and Malhi (2017) utilized transfer learning with CNNs to classify 2D abdominal ultrasound images into distinct categories[43]. A more recent paper by Burgos-Artizzu et al. (2020) focused on maternal-fetal ultrasound planes classification, utilizing different non-DL and the most used state-of-art CNN classifiers on a dataset of 12,400 images from 1,792 patients. The best-performing network, DenseNet-196, achieved comparable performance to a technician[44]. Additionally, Burgos-Artizzu and colleagues (2021) demonstrated that AI methods analyzing fetal brain morphology from standard ultrasound images can provide accurate estimates of gestational age, outperforming traditional biometric parameters[45]. Another study (Coronado-Gutierrez et al., 2023) showcased the potential of deep learning in delineating fetal head and brain structures, with automatic measurements achieving high accuracy in routine ultrasound examinations[46].

These studies demonstrate the potential of the application of ultrasound imaging to state-of-the-art deep learning techniques, and also show that current state-of-the-art computer vision models from standard pictures can work for maternal-fetal ultrasound images, and that the technology is mature enough to be applied to real maternal-fetal clinical settings.

## 2.3 Placental insufficiency

### 2.3.1 Overview of the clinical problem

Placental insufficiency is a condition in which the placenta does not function properly, resulting in reduced oxygen and nutrient transfer to the fetus. This can lead to a number of complications, including intrauterine growth restriction (IUGR), preterm labor, and perinatal morbidity and mortality[47]. Late-onset placental insufficiency typically occurs after 32 weeks of pregnancy.

It is generally understood that placental insufficiency is a process that involves progressive deterioration in placental function, reducing the transfer of oxygen and nutrients to the fetus, resulting in decompensated hypoxia and acidosis. Both MRI and ultrasound studies have revealed reductions in placental area and volume, increased placental thickness, and distinctive globular-shaped placentas on MRI when assessing placental insufficiency.

The underlying causes of placental insufficiency remain a subject of ongoing research and are not yet fully understood. However, several maternal risk factors are known to be associ-

ated with this condition. These factors include pre-eclampsia, maternal hypertensive disorders, maternal cigarette smoking, maternal use of drugs such as cocaine or heroin, maternal alcohol consumption, being a first-time mother (primiparity), advanced maternal age, and a prior history of delivering an infant with intrauterine fetal growth restriction (IUGR). Certain medications, such as anticonvulsants,or anticoagulants can also interfere with fetal growth.

Prematurity is the primary cause of perinatal death, closely followed by intrauterine fetal growth restriction (IUGR), which affects 4% to 6% of pregnancies. Placental insufficiency can lead to preterm labor, pre-eclampsia, IUGR, and stillbirth, impacting 10% to 15% of pregnancies. IUGR fetuses have a higher risk of preterm labor and perinatal death. Worryingly, about 50% of IUGR cases are only identified after birth, highlighting the need for improved prenatal monitoring and early detection in maternal and fetal healthcare.

### 2.3.2 Current diagnostic methods for placental insufficiency

Currently, there are no standardized methods for the diagnosis of placental insufficiency and the diagnosis can be challenging. When it comes to late-onset placental insufficiency, in many cases, the condition is diagnosed when it is already in advance stages due to the complications it causes.

Some common tools to diagnose the condition are blood tests to determine the fetus' alpha-fetoprotein levels, a fetal nonstress test that measures how fast the fetus' heart is beating, and regular ultrasounds to estimate the size of the fetus[48].

Another of the techniques used for the diagnosis of placental insufficiency is the Doppler ultrasound, which has proven to be useful in the evaluation of fetal and placental circulations in both healthy and diseased states. It is a noninvasive test that can be used to estimate the blood flow through blood vessels by bouncing high-frequency sound waves (ultrasound) off circulating red blood cells. Utilizing Doppler screening with the uterine artery to assess notching and resistance has demonstrated an approximate 85% sensitivity in identifying severe cases of intrauterine growth restriction (IUGR) and pre-eclampsia in high-risk situations. However, because of its limited predictive ability as a standalone test, uterine artery Doppler should be used in combination with other tests to guide clinical decisions[49].

These tests exhibit limitations in accurately predicting all adverse pregnancy outcomes, particularly in the context of late-onset placental insufficiency. For instance, the presence of a normal test result does not provide a definitive assurance that a woman will not experience preeclampsia or that her baby's growth will proceed as expected. Consequently, this underscores the potential benefit of exploring the development of a model utilizing ultrasound images to enhance early detection.

### 2.3.3 Placental insufficiency prediction with MRI imaging

There are some studies that show that MRI images together with machine learning algorithms can be utilized to predict late-onset placental insufficiency or some of the conditions derived from it, like fetal growth restriction (FGR).

In a notable study conducted by Dahdouh et al. in 2018[50], researchers assessed the ability of 3D MRI placental shape and textural features to predict FGR and birth weight (BW) for both healthy and FGR fetuses.

The study counted with MRI images of pregnant volunteers, comprising 46 healthy subjects and 34 individuals diagnosed with FGR, all between 18 and 39 weeks of gestation.

To predict FGR and BW, the researchers employed a combination of placental shape features (volume, thickness, and elongation) and textural features. The textural features included first-order characteristics such as mean, variance, kurtosis, and skewness of placental grey levels, along with textural features computed on grey-level co-occurrence and run-length matrices, providing insights into placental homogeneity, symmetry, and coarseness. They developed a prediction framework using RUSBoost, with 1,000 trees with a minimal depth of 5, adjusting prior probabilities to penalize false negatives more. FGR data were oversampled with a 2:1 ratio.

The method achieved an 86% accuracy rate for identifying FGR pregnancies, with a 77% precision and an 86% recall. Additionally, BW estimations were found to be 0.3% $\pm$ 13.4% (mean percentage error $\pm$ standard error) for healthy fetuses and -2.6% $\pm$ 15.9% for those with FGR. These results indicated the potential for early diagnosis and more accurate predictions regarding birth weight.

The high accuracy observed in both healthy and high-risk cohorts underscores the potential of combining MRI images and machine learning for FGR prediction.

### 2.3.4 Conclusion

Late-onset placental insufficiency can lead to a range of serious complications, including fetal growth restriction, preterm birth, and even stillbirth. Detecting late-onset placental insufficiency in a timely manner is of paramount importance for ensuring the well-being of both the mother and the developing fetus. Unfortunately, this condition is often diagnosed only when complications arise, which can limit the effectiveness of interventions. To address this critical issue, there is a growing need to develop more proactive and accessible methods for early detection.

Different types of medical imaging together with machine learning have emerged as a promising tool for clinical diagnosis. Ultrasound imaging is a non-invasive and widespread tool, fre-

quently used for routine prenatal screening.

By harnessing the power of advanced image analysis and machine learning, we believe it is possible to create a predictive model based on ultrasound images taken between weeks 27 and 29 of pregnancy that can identify subtle changes in the placenta that may be indicative of late-onset placental insufficiency. Such a model would not only improve the accuracy of diagnosis but also enable healthcare providers to implement timely interventions.

# Chapter 3

# Project development

## 3.1 Dataset

### 3.1.1 Study design and image acquisition

As mentioned before, this was a study in collaboration with the Preeclampsia and Intrauterine Growth Restriction Research Unit at the hospital Sant Joan de Déu in Barcelona. From September 2022 to November 2023, all pregnant women between weeks 27 and 29 of pregnancy, with a heightened baseline risk of placental insufficiency identified during routine second-trimester ultrasounds, attending Hospital Clínic in Barcelona, were included in the study. Inclusion criteria further specified that participants must be carrying a single, viable fetus without congenital malformations, be of legal age, and have no barriers compromising the ability to provide informed consent. To each woman, a minimum of three ultrasound images of the placenta were acquired. Subsequently, medical professionals of Hospital Sant Joan de Déu research team in Barcelona manually created masks for each one of these images, to outline the position of the placenta within the ultrasound image. After child delivery, perinatal data will be collected.

The study was approved by the ethical commitee of the Hospital (REF number HCB/2022/0135) and all patients signed an informed consent for the treatment of their data and images.

Access to these ultrasound images and associated clinical data was granted based on a signed agreement with the Sant Joan de Déu foundation.

### 3.1.2 Overview of the dataset

The final dataset comprised ultrasound images of 450 women. We also counted with a comprehensive patient database, cataloging a range of relevant medical variables for each patient. Out of these variables, clinicians have identified three diagnostic indicators that could pinpoint cases of late-onset placental insufficiency if any one of them is present:

1. The presence of preeclampsia (binary variable: 'Yes' or 'No').

2. Prenatal fetal growth restriction (CIR) diagnosis (binary variable: 'Yes' or 'No').

3. Birth weight percentile lower than 10 (LBW).

These markers are completed after childbirth. Therefore, we are trying to predict a condition 13 to 11 weeks before it is detected. While these markers may suggest potential placental insufficiency, it is crucial to acknowledge their nonspecific nature, as they may also arise from other medical conditions not affiliated with placental health. Therefore, our analysis treated these indicators both individually and in combination to account for their potential non-specificity. The main focus, however, is on treating these three indicators in combination.

### 3.1.3 Data characteristics

Even when our dataset initially comprised 450 women, not all patients could be included in the final analysis for each diagnostic criterion, as some had not yet given birth at the time of this study and some or all of the indicators were not available. The following statistics reflect the available information for each criterion:

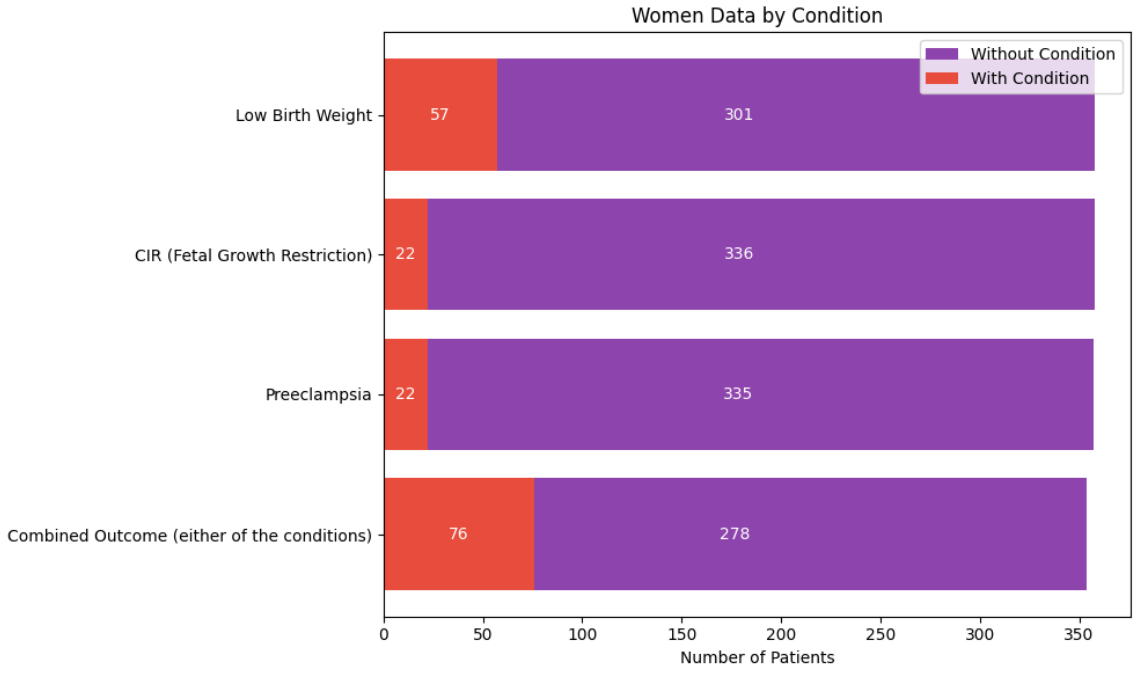


Figure 3.1: Summary of available patient data for diagnostic criteria.

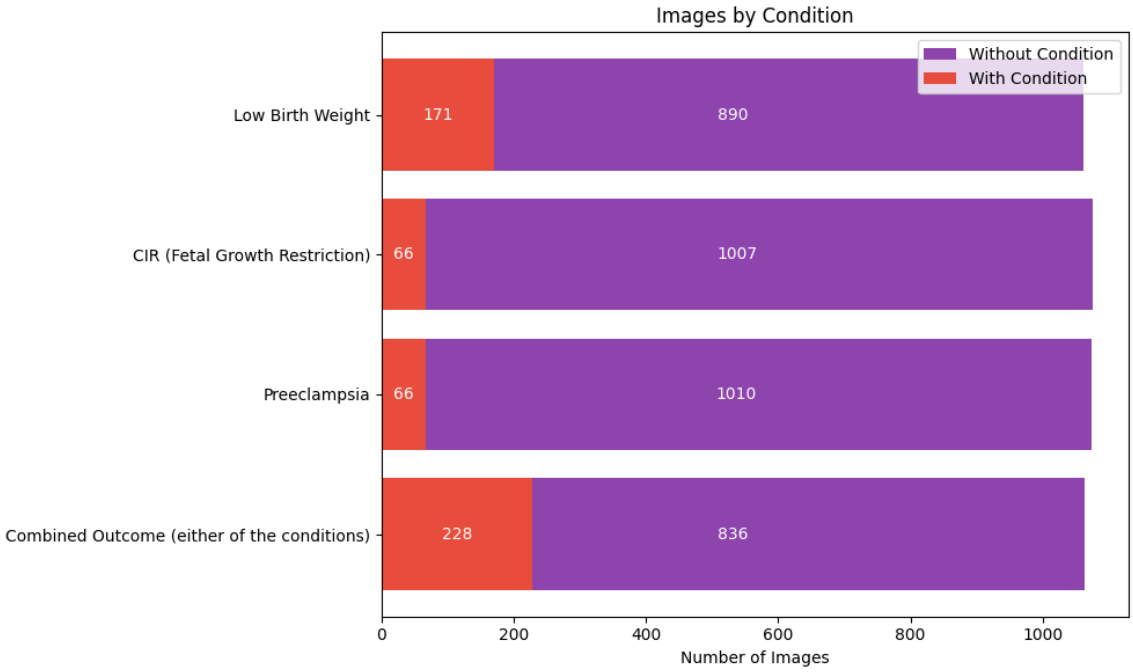


Figure 3.2: Summary of available images for diagnostic criteria.

Overall, we can observe significant imbalance in the data, particularly for fetal growth restriction and preeclampsia individually, where only 6% of instances involve women with these conditions. However, when considering the combined outcome, instances involving women with the condition increase to 21%. For the tenth percentile birth weight condition, the percentage of instances with the condition is 16%.

### Addressing class imbalance

The existing imbalance poses a risk of biased predictive performance, with models potentially favoring the majority class. We have addressed this by implementing normalized class weights in some of our experiments, as explained in section 3.3. We opted not to reduce the size of the majority class or artificially increase the minority class because these techniques may not translate well to a clinical setting, where the natural distribution of conditions is essential for model applicability.

### Variation in placental plane

Our dataset includes ultrasound images depicting the placenta in both front and posterior planes. These variations could have an impact on the results, but given the limited size of our dataset we conducted multiple trials using images from both planes. After these trials, we

selected the most effective model to be specifically trained on images from the anterior plane only, which provides clearer images, to evaluate if this change has an impact on performance. However, this reduced the number data available of women under study to 183, thus the sample size is not enough to draw statistically significant results for any of the different criteria. Data split for anterior planes followed a conventional 70-15-15 in all cases. Below we can observe the difference between anterior and posterior placental planes, on ultrasound images where the placenta is outlined:

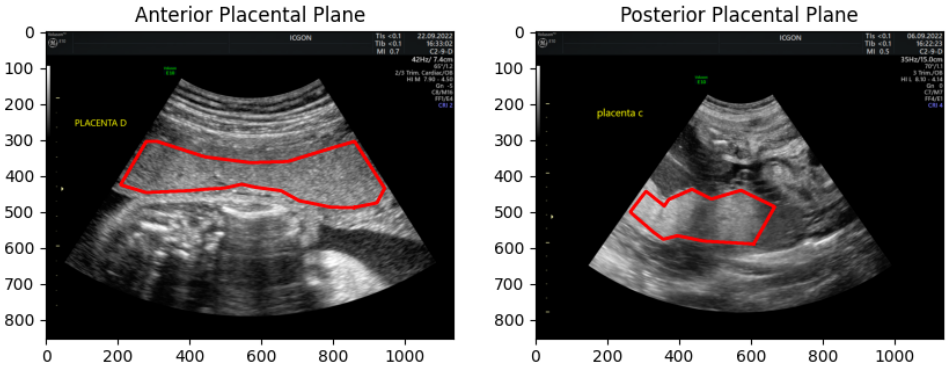


Figure 3.3: Examples of different placental planes.

## 3.2 Data preparation

**Sample size**

For any of the models to have clinical relevance, it was agreed that a minimum specificity of 1 - prevalence of the condition was required, as well as a minimum sensitivity of 0.5. In order to determine the minimum sample size for statistically significant results, we used Burderer's method[51] with pre-established alpha and beta coeficients of 0.05 and 0.20, respectively. The resulting minimum required test sizes for each criterion are presented in the following table:

| Condition | Prevalence | Min. test size | Sample available |
|---|---|---|---|
| Combined outcome | 20% | 121 | 354 |
| Preeclampsia | 6% | 401 | 357 |
| Fetal growth restriction (CIR) | 6% | 401 | 358 |
| Low birth weight (LBW) | 16% | 151 | 358 |

Table 3.1: *Minimum sample size of test from Burderer's method.*

With these considerations in mind, our main focus will be on the combined outcome, and 34% of the dataset will be allocated to test. For low birth weight, 42% of the dataset will be used for testing. Due to insufficient data for Preeclampsia and CIR, we will adopt a conventional 70-15-15 split in these instances, but the limited number of observations can mean that our analysis might not be as statistically strong as we would like, which is why we must be cautious when drawing conclusions from our data.

We split our dataset into training, validation, and test subsets, ensuring that each subset contained unique patients to prevent data leakage—no patient's images were included in more than one subset. The division was randomized but controlled to maintain a balanced ratio of positive (patients with the condition) to negative (patients without the condition) cases.

For image processing, we first cropped the images according to the placental masks provided by medical professionals, as shown below:

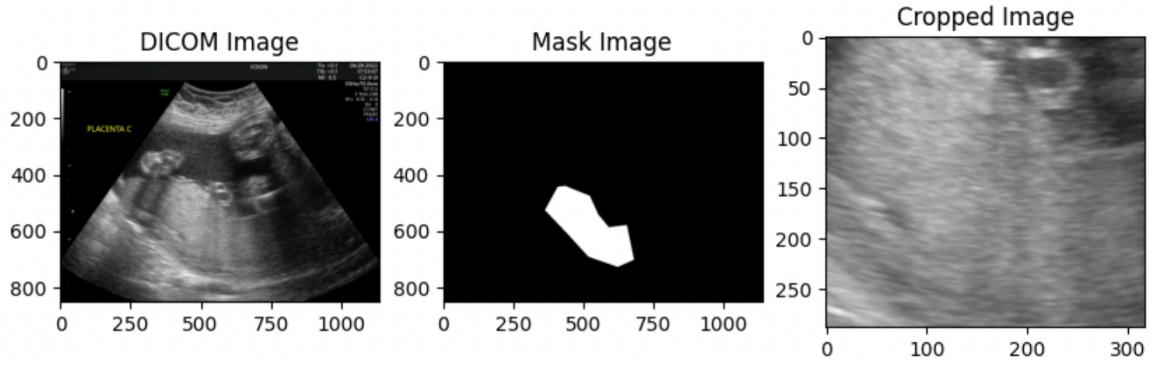


Figure 3.4: Example of placenta cropped image.

This step ensures that the analysis focuses exclusively on the placental region. We employed two cropping strategies: an exact match to the mask dimensions and an expanded crop at 1.3 times the mask size to include adjacent areas. We initially experimented with alternative methods, such as applying bitwise operations or multiplication across the image matrices, to isolate the placenta. However, this approach was quickly discarded due to unsatisfactory outcomes, predominantly resulting from excessive zero-padding.

The data was divided into 8 different groups: two main groups based on the two mask sizes used for image cropping, that were further subdivided into four subgroups, corresponding to the different labeling criteria.

| Condition | Mask Size |
|---|---|
| Combined outcome | Original |
| Combined outcome | Expanded |
| Fetal growth restriction | Original |
| Fetal growth restriction | Expanded |
| Preeclampsia | Original |
| Preeclampsia | Expanded |
| Tenth birth weight percentile | Original |
| Tenth birth weight percentile | Expanded |

Table 3.2: Groups of data

The distribution of data within these subsets remained consistent across both mask size groups. Below is the detailed structure of these datasets, indicating the prevalence of the condition for each subset:

### Overall criteria

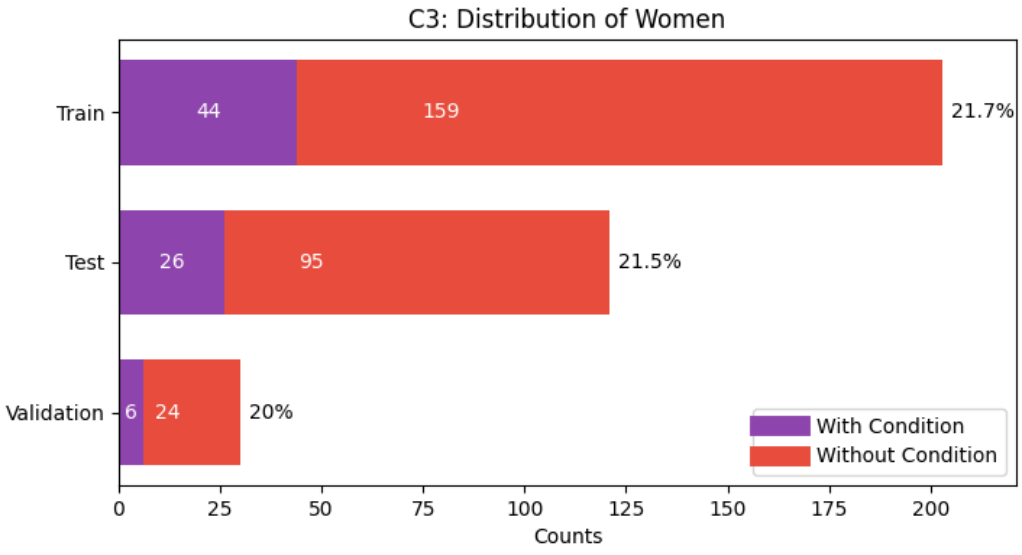


Figure 3.5: Distribution among train, validation and test for C3.

### Preeclampsia

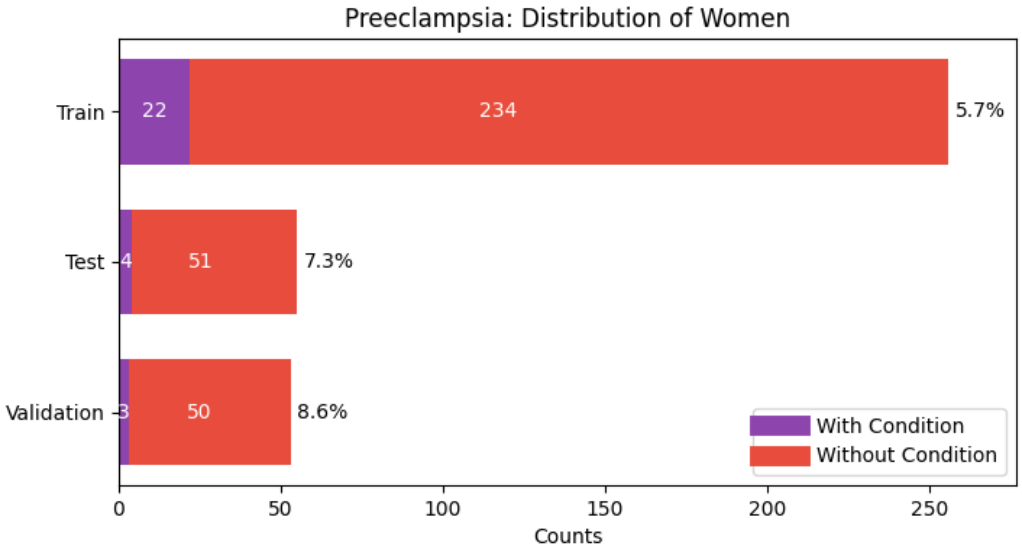


Figure 3.6: Distribution among train, validation and test for Preeclampsia.

### Fetal growth restriction (CIR)

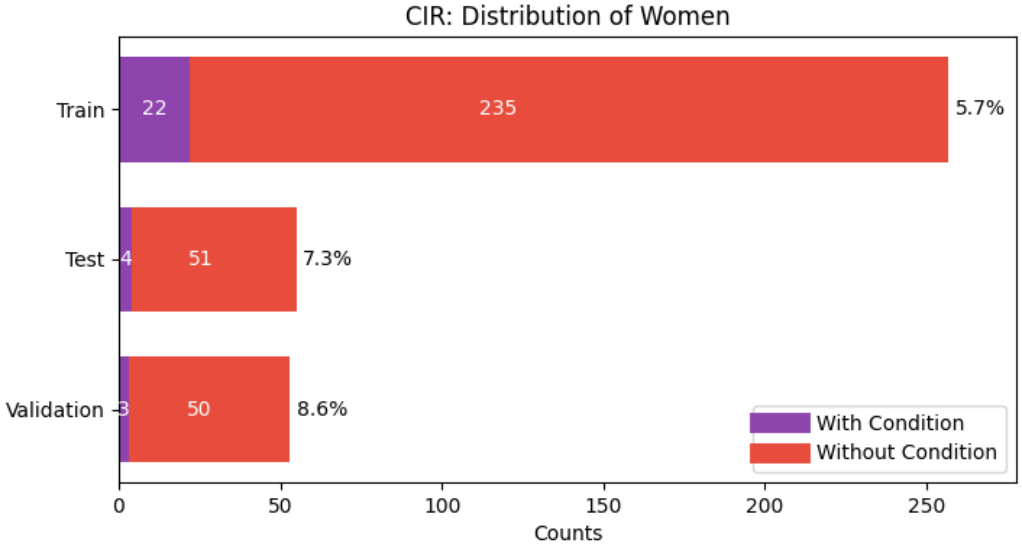


Figure 3.7: Distribution among train, validation and test for CIR.

### Low birth weight (LBW)

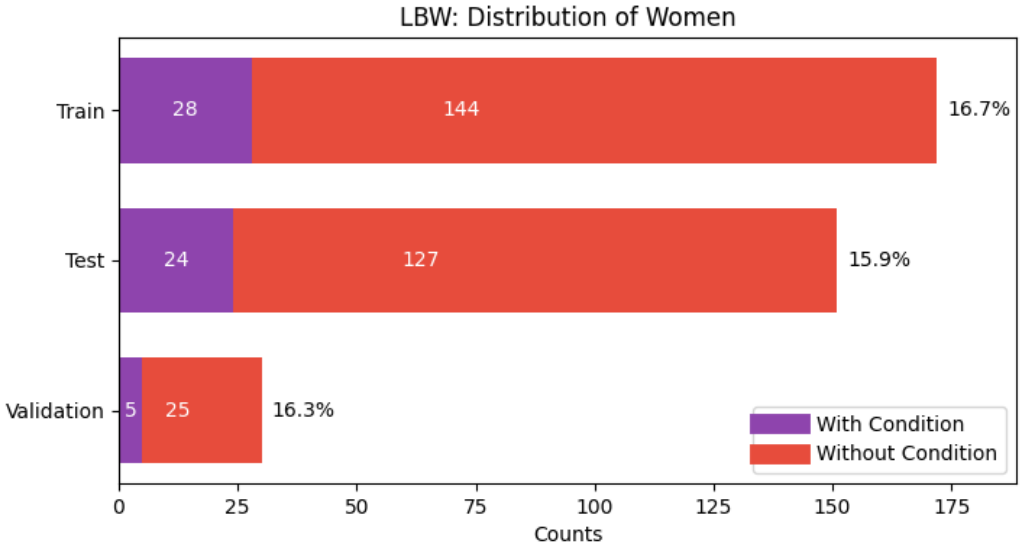


Figure 3.8: Distribution among train, validation and test for LBW.

## 3.3 Experimental design

We have carried out several experiments employing deep learning and computer vision techniques, following an agile methodology. This approach began with establishing baseline models, followed by incremental enhancements under the hypothesis that the performance would be improved.

In general, our methodology is structured as follows:

**1. Baseline Models**: We start with a series of baseline models, using the data as-is without any modifications or specialized hyperparameters.

**2. Class Weights Normalization**: The same models are then tested again, but applying normalized class weights to address class imbalance. Class weights are determined for each class using a simple formula that inversely relates the weight to the class frequency, with higher weights allocated to underrepresented classes. During model training, the normalized class weights are integrated into the loss function, and by assigning higher weights to underrepresented classes, the model becomes more sensitive to misclassifications of those classes. Therefore, the errors in the minority class have a greater impact on the overall loss.

**3. Image Augmentation**: We introduce image augmentation techniques to these models, both with and without class weights, to enhance their ability to generalize. These techniques increase the size of the training dataset artificially, by applying different changes to the training images, like rotations or increasing the brightness. We adopt two distinct approaches: an aggressive technique involving more extensive modifications to the images and a more refined

approach that focuses on adjusting parameters such as brightness or contrast while minimizing other aspects like rotation.

**4. Expanded Mask and Augmentation**: The images are cropped after increasing the mask size by 30%. Here, we also apply image augmentation, again testing both with and without class weights.

**5. Transfer Learning**: For Deep Learning approaches, we also tested whether doing transfer learning from a model pre-trained on ultrasound images improved performance compared to transfer learn from classical ImageNet weights. To achieve this, we used the 12,400 ultrasound images from the study by Burgos-Artizzu et al. (2020)[44], as introduced in Section 2. Specifically, we developed a classifier to discern whether an image contains a brain or not. Subsequently, we employed this trained classifier as a base for transfer learning, using it to train our placental ultrasound images. Our hypothesis was that a deep learning architecture pre-trained on relevant ultrasound imagery would exhibit enhanced proficiency in interpreting similar images, thereby improving the overall performance of our models. To explore this hypothesis, we employed two different architectures – one large and one small.

Finally, after conducting trials across both computer vision and deep learning paradigms, we selected the best-performing model for each technique. This model was then exclusively trained on anterior placental plane images to observe if we could get a boost in performance.

https://github.com/gwendysyd/Placenta-Insufficiency-Classification

### 3.3.1 Deep Learning

We have implemented a total of nine deep learning models to perform binary image classification. The models were developed using TensorFlow and Keras with use of GPU. The key objectives were to explore different architectures, including custom-built models and pre-trained models with transfer learning, and to compare their performance.

**Custom Artificial Neural Network (ANN) Model**

| **Component** | **Description** |
|---|---|
| Architecture | Sequential model using Keras' Sequential API |
| Preprocessing | Image resizing to 224x224 pixels, rescaling pixel values for normalization |
| Layers | • Flatten layer to convert 2D image data to 1D array<br>• Dense layer with 1024 neurons and ReLU activation for feature learning<br>• Dropout layer with a rate of 0.5 to reduce overfitting<br>• Output Dense layer with sigmoid activation for binary classification |

Table 3.3: Custom Artificial Neural Network (ANN) Model

**Pre-Trained Models: VGG16[7], ResNet50[8], MobileNet[52], and ResNet18[8]**

| Functionality | Description |
|---|---|
| Custom `build_model` function | • Modify trainability of base models<br>• Integrate custom layers for grayscale image processing and binary classification<br>• Utilize each model with ImageNet weights, testing both with and without transfer learning<br>• Implement functions for grayscale-to-RGB conversion, and vice versa, post-model preprocessing |

Table 3.4: Pre-Trained Models: VGG16, ResNet50, MobileNet, and ResNet18

For all models, we applied:

| Parameter | Value |
|---|---|
| Loss function | Binary Crossentropy |
| Optimizer | Adam |
| Batch size | 32 |
| Epochs | 150 |
| Callbacks | EarlyStopping (patience: 10) and ModelCheckpoint for optimal weight storage |
| Evaluation | Training time assessment for model efficiency |

Table 3.5: Common Settings for All Models

### 3.3.2 Computer vision

We have used two prominent computer vision techniques, Histogram of Oriented Gradients (HOG) and Gray Level Co-occurrence Matrix (GLCM), for feature extraction from image datasets. Subsequently, these extracted features were utilized to train three distinct machine learning models: Support Vector Classifier (SVC), XGBoost, and Logistic Regression (LR). In the initial phases of our experiment, we refrained from hyperparameter tuning of these

algorithms, with the intention to establish baseline models, allowing us to evaluate their fundamental performance prior to any optimization.

GLCM is particularly adept at texture analysis, crucial for identifying placental health indicators that manifest as textural changes in ultrasound imagery. Meanwhile, HOG may be good at discerning the internal structure of the placenta for detection of anomalies.

**HOG-Based Feature Extraction for Image Analysis**

The Histogram of Oriented Gradients (HOG)[24] is a popular method in computer vision to transform each image into a feature vector. First, images are resized to a standard dimension of 128x64 pixels, to ensure uniformity in feature extraction. Then, a HOG descriptor is created with specified parameters (window size, block size, block stride, cell size, number of bins) to define how the image is broken down into smaller regions for gradient computation. This all results in a feature vector that captures the shape and texture of the image based on gradient information. The feature vector is flattened to transform it into a 1D array suitable for machine learning algorithms. The specific parameters used for the implementation can be found in the appendix.

**GLCM-Based Feature Extraction for Image Analysis**

The Gray Level Co-occurrence Matrix (GLCM)[26] method turns images into vectors by statistical texture analysis. It involves the computation of a matrix based on the spatial relationships between pixel intensity values in an image. In our case, we read a grayscale image for GLCM to analyze textural features based on gray level intensities. Then images are resized to a standard dimension of 128x64 pixels, and finally the matrix is computed, quantifying how often different combinations of pixel brightness values (grey levels) occur in an image or a region of an image. We extract specific textural properties like Dissimilarity, Correlation, Homogeneity, Contrast, Angular Second Moment, and Energy. These properties provide a comprehensive characterization of the texture in terms of contrast, uniformity, and the arrangement of pixels in the image. The extracted properties are then aggregated into a single feature vector. This vector represents the textural attributes of the image and is suitable for subsequent analysis or classification in machine learning workflows. The specific parameters used for the implementation can be found in the appendix.

### 3.3.3 Evaluation metrics

We will monitor a variety of evaluation metrics to assess the performance of our models. When setting the threshold to assign labels, the metric to focus on is the maximum sensitivity achieved at the specific points on the ROC curve where specificity equals 1 - prevalence.

| Metric | Description | Formula |
|---|---|---|
| Accuracy | Overall correctness of the model | Accuracy = $\frac{TP+TN}{TP+TN+FP+FN}$ |
| Sensitivity (Recall) | Proportion of actual positives correctly predicted | Sensitivity = $\frac{TP}{TP+FN}$ |
| Specificity | Proportion of actual negatives correctly predicted | Specificity = $\frac{TN}{TN+FP}$ |
| Positive Predictive Value (Precision) | Proportion of predicted positives correctly predicted | Precision = $\frac{TP}{TP+FP}$ |
| Negative Predictive Value | Proportion of predicted negatives correctly predicted | NPV = $\frac{TN}{TN+FN}$ |
| Positive Likelihood Ratio | Ratio of the probability of a true positive to the probability of a false positive | PLR = $\frac{\text{Sensitivity}}{1-\text{Specificity}}$ |
| Negative Likelihood Ratio | Ratio of the probability of a false negative to the probability of a true negative | NLR = $\frac{1-\text{Sensitivity}}{\text{Specificity}}$ |

Table 3.6: Summary of Performance Metrics

In addition to those metrics, we will construct the Receiver Operating Characteristic (ROC) curve, a graphical representation illustrating the trade-off between the true positive rate (sensitivity) and false positive rate (1 - specificity) across different threshold settings. Subsequently, our objective is to optimize the threshold for our binary classification. This involves selecting the point that maximizes sensitivity while maintaining specificity at a minimum level, established as 1 minus the prevalence of the condition.

Once the optimized threshold is determined, we will apply it to assign labels and generate a confusion matrix. This matrix will be visualized as a heatmap, providing a detailed breakdown of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN), to provide insight into the model's accuracy in classification.

## 3.4 Deep Learning Experiments

### 3.4.1 Trial 1: baseline performance evaluation

In the first phase of our deep learning experimentation, we focused on establishing baseline performances for various neural network architectures. The goal was to assess the fundamental capabilities of each architecture without the influence of techniques like class weight adjustments or data modifications. The datasets were used in their original form, reflecting the real-world conditions under which the models must operate. This approach provided insight into the capability of each architecture to handle the data, serving as a benchmark for future experiments.

### 3.4.2 Trial 2: adding normalized class weights

In this second trial, we retained the core implementation framework from the previous experiment but introduced normalized class weights. The addition of class weights is a step to address the huge class imbalance in the datasets. Class weights are are calculated for each class using a simple formula that inversely relates the weight to the class frequency, with higher weights allocated to underrepresented classes, to ensure that the model does not become biased towards more prevalent categories.

### 3.4.3 Trial 3: adding image augmentation

In this third trial, we retained the core implementation framework from experiments 1 and 2, but introduced image augmentation techniques. Overall, this setup is a common approach in deep learning for preprocessing image data, particularly for training convolutional neural networks (CNNs) where augmented data helps the model learn from a more diverse set of examples, thereby improving its accuracy and robustness. We expected this introduction to enhance model generalization from exposing the model to a broader variety of image transformations, and to mitigate overfitting from increasing the data size and variability of our training dataset, making it more challenging for the model to simply memorize specific images. In addition, we have also adjusted the learning rate to 1e-4, in order to facilitate more gradual and precise model updates during training.

We experimented with a 'softer' and a more 'aggressive' augmentation techniques:

#### 3.4.3.1 Trial 3A: soft image augmentation

This adjustment was intended to keep a balance between adding variability to the dataset and maintaining the integrity of the original images. In this case, we decided to use only horizontal and vertical flips, along with adjustments to the images' brightness and contrast. We used TensorFlow's native image processing functions for the transformations.

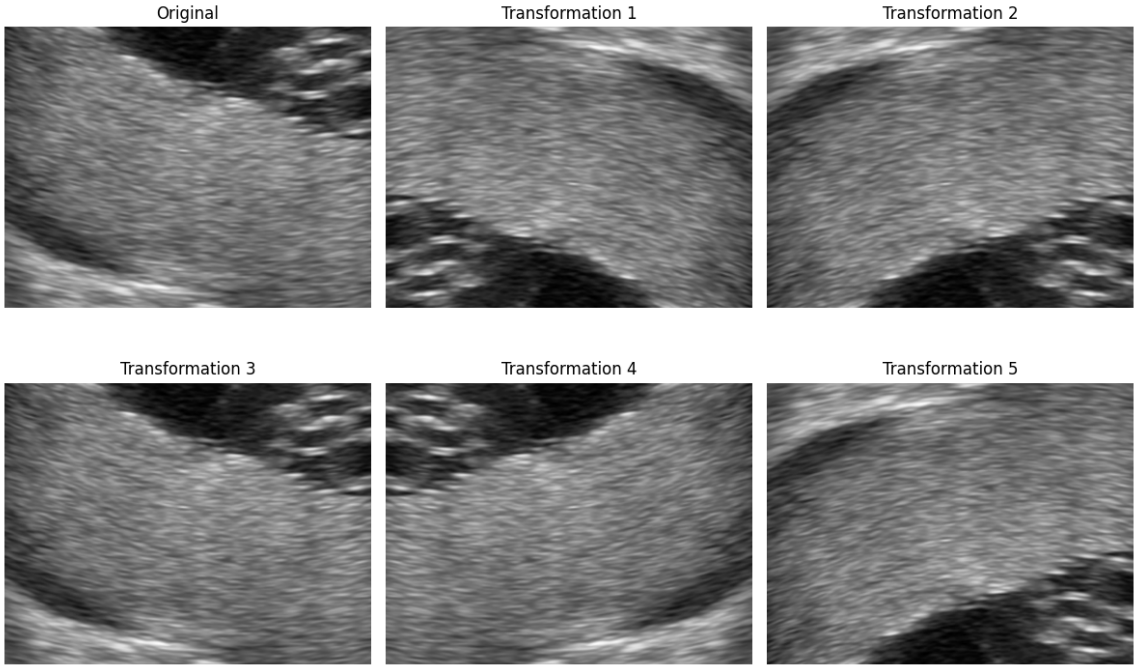


Figure 3.9: Results of Softer Image Augmentation.

#### 3.4.3.2 Trial 3B: aggressive image augmentation

In this case, the parameters for augmentation include random rotations, shifts, shear transformations, zooming and flipping. We used ImageDataGenerator from TensorFlow's Keras API for data augmentation in image processing. The generator works by taking batches of images from the original dataset and applying these transformations on-the-fly before passing them to the model. This approach is highly efficient as it reduces the need for storing a large number of transformed images on disk. Instead, ImageDataGenerator dynamically creates augmented images during the training process, providing a diverse range of examples for the model to learn from.

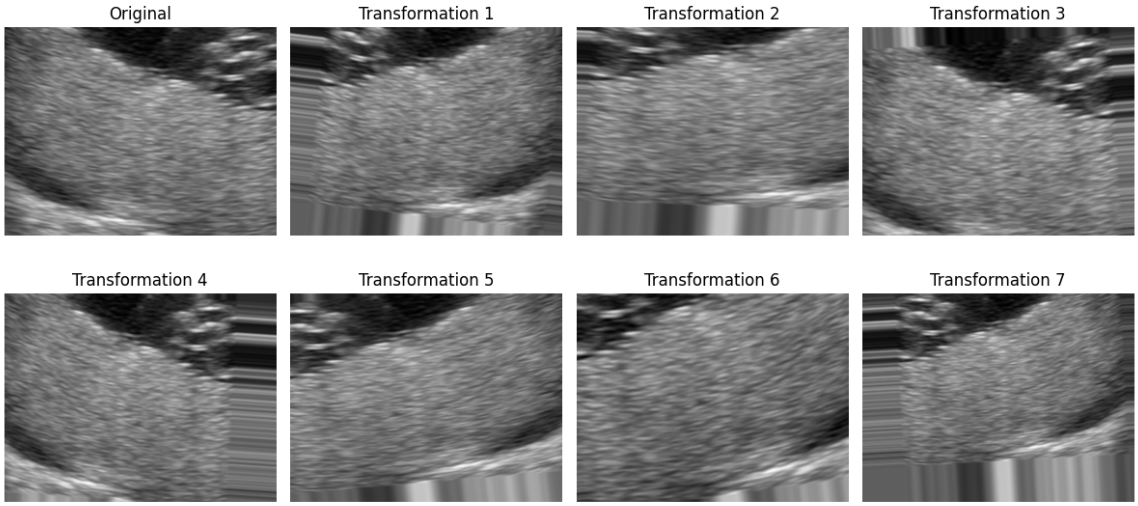


Figure 3.10: Results of Aggressive Image Augmentation.

### 3.4.4 Trial 4: expanding the mask

In this fourth trial, we closely followed the methodology of the preceding trial but introduced a key variation in our image preprocessing technique. Specifically, we expanded the placental mask used in image processing. Instead of utilizing the mask in its original dimensions, we enlarged it to 1.3 times its original size. This expansion aimed to capture a broader area around the placenta, potentially including additional contextual information that could be relevant for our analysis. We can see the difference in the resulting image below:

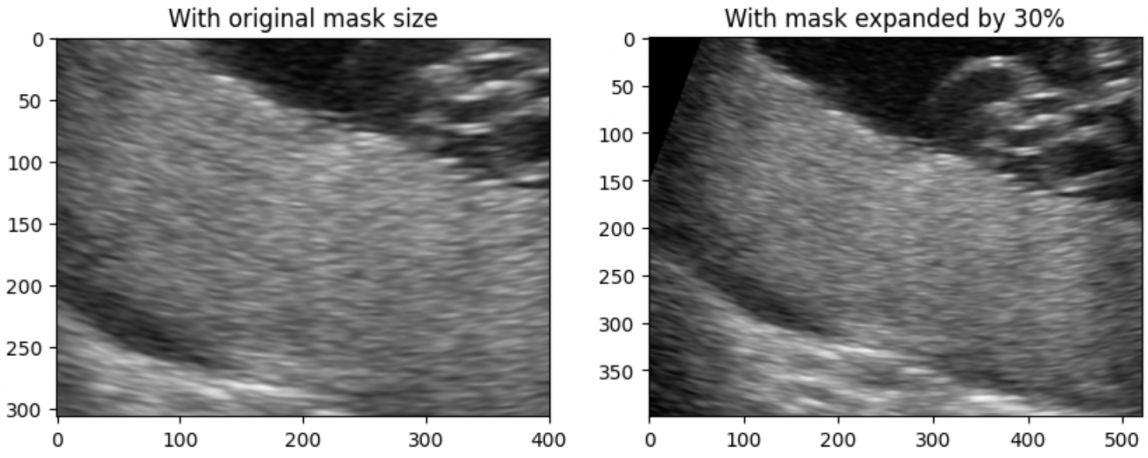


Figure 3.11: Difference in Images from Mask Size.

So far, we can summarize the key parameters for experiments 1 to 4 in the following table:

| Parameter | Experiment 1 | Experiment 2 | Experiment 3 | Experiment 4 |
|---|---|---|---|---|
| Optimizer | Adam | Adam | Adam | Adam |
| Learning Rate | 1e-3 | 1e-3 | 1e-4 | 1e-4 |
| Epochs | 150 | 150 | 150 | 150 |
| Early Stopping | Patience = 10 | Patience = 10 | Patience = 10 | Patience = 10 |
| Image Augmentation | No | No | Yes | Yes |
| Class Weights | Without | With | With and without | With and without |
| Placental Planes | Anterior + posterior | Anterior + posterior | Anterior + posterior | Anterior + posterior |
| Mask Size | 1:1 | 1:1 | 1:1 | 1:1.3 |

Table 3.7: Summary of Key Parameters for Experiments 1 - 4.

And the main architectures:

| Architecture | Configuration |
|---|---|
| Artificial Neural Network (ANN) | A basic ANN served as a straightforward baseline for comparison. |
| VGG16 | Evaluated with and without transfer learning. |
| ResNet50 | Evaluated with and without transfer learning. |
| MobileNet | Evaluated with and without transfer learning. |
| ResNet18 | Evaluated with and without transfer learning. |

Table 3.8: Summary of Architectures

### 3.4.5 Trial 5: transfer learning using a tailored classifier

Until now, all of the architectures used were trained on ImageNet. Those images are generic and not very similar to ultrasound imagery. So, we wanted to try to use an architecture that was trained on ultrasound images to then apply transfer learning, hypothesizing that a model effectively trained on general ultrasound images would demonstrate improved proficiency in classifying other types of ultrasound imagery, leveraging the learned features and patterns from the initial dataset. As there are not known architectures trained on ultrasound images, we trained our own by using a different dataset, comprising

The primary of this trial was to use a dataset comprising 12,400 ultrasound fetal plane images from a publicly available source[53] to develop a binary classifier to distinguish whether an image depicts a brain. Then, this classifier would serve as a foundational model for applying transfer learning to our placental ultrasound images, hypothesizing that a model effectively trained on general ultrasound images would demonstrate improved proficiency in classifying

other types of ultrasound imagery, leveraging the learned features and patterns from the initial dataset.

#### 3.4.5.1 Brain Classifier

We used two distinct neural network architectures: VGG16 and ResNet18, that we chose for their varying sizes—VGG16 being larger and ResNet18 smaller—to evaluate performance across different complexities. To tailor these models to our specific needs, we removed their original top layers and added four custom layers. Both architectures were trained from scratch, starting with weights initialized from ImageNet. In parallel, we also applied a transfer learning strategy, retaining the weights of all but the last four layers of each base model. These remaining layers were fine-tuned to adapt to our dataset in conjunction with our custom top layers.

In order to carry-out the implementation, we defined a function to adapt these pre-trained models for grayscale ultrasound image classification. This function begins by rendering the base model's layers non-trainable, thus retaining the original weights. Depending on the transfer learning strategy—'Y' for fine-tuning the last four layers only or 'N' for complete training from scratch—we dynamically adjust layer trainability. The process includes integrating a new input layer to accommodate grayscale images and a Conv2D layer to convert these to a three-channel format. Then, we append custom top layers, including Flatten and Dense layers, finalized by a sigmoid activation function for binary classification.

Both VGG16 and ResNet18 architectures with and without transfer learning underwent training for 15 epochs with a patience of 10. We utilized the Adam optimizer with a learning rate of 1e-4. Class weights were applied. The average Area Under the Curve (AUC) across all four methodologies was 0.9975. Moreover, the average F1-score was 0.98 for the positive class and 0.99 for the negative class. These results indicate that all models achieved near-perfect classification accuracy.

#### 3.4.5.2 Transfer Learning Approach

For classifying placental ultrasound images, we replicated the architectures employed in our brain classifier, freezing the weights of all but the last four layers. This strategy allowed the models to adapt specifically to placental images. We extended the training to 150 epochs for these four architectures, and used an early stopping mechanism with a patience of 10. The learning rate was reduced to 1e-5 to facilitate more precise and gradual adjustments in the weights, which is particularly beneficial in the latter stages of training. This slower learning rate helps in fine-tuning the model more effectively, as it prevents overfitting and ensures that the model does not miss subtle features specific to placental images. We performed several trials,

that involved training each architecture with and without class weights, with and without image augmentation.

### 3.4.6 Final trial: optimization

The primary objective of this trial was to identify and optimize the best-performing model based for each predefined criteria. The optimization process was conducted in two main stages:

**Model Retraining on Anterior Planes**

In this stage, we aimed to retrain the best-performing model on set of images that correspond to an anterior plane of the placenta only. The purpose of this step was to evaluate and compare the model's performance when trained exclusively on this plane type, which offers a clearer visibility, to verify whether the model exhibited superior performance.

**Model Tuning with Optuna**

After identifying the most effective model—trained either on anterior planes, or a combination of both anterior and posterior—we proceeded to hyperparameter tuning using Optuna, an advanced framework specifically designed for optimizing machine learning models. Optuna automates the process of finding the most effective hyperparameters for our model. The parameters we focused on can be found in the appendix. These parameters were consistently incorporated into our optimization process whenever the selected best-performing architecture could accommodate them. In instances where the architecture was not conducive to all these adjustments, we selectively included only those parameters that were compatible. These parameters were iteratively refined across 10 trials with the objective of maximizing Sensitivity within a minimum Specificity of 1 - prevalence of the condition.

## 3.5 Computer Vision Experiments

### 3.5.1 Trial 1: baseline performance evaluation

In the first phase of our computer vision experimentation, we focused on establishing baseline performances for various neural network architectures. The goal was to assess the fundamental capabilities of each algorithm without the influence of techniques like class weight adjustments or data modifications. The datasets were used in their original form, reflecting the real-world conditions under which the models must operate. This approach provided insight into the capability of each design to handle the data, serving as a benchmark for future experiments.

### 3.5.2 Trial 2: adding normalized class weights

In this second trial, we retained the core implementation framework from the previous experiment but introduced normalized class weights. The addition of class weights is a step to address the huge class imbalance in the datasets. This method assigns a higher significance to less represented classes, ensuring that the model does not become biased towards more prevalent categories.

Class weights were calculated based on the frequency of each class in the training dataset. This method ensures that classes with fewer samples have a greater impact on the loss function during training, effectively balancing the influence of each class on the model's learning process. With this, we anticipated a more balanced classification performance across all classes, reducing the likelihood of overlooking critical but less frequent patterns in the data.

### 3.5.3 Trial 3: adding image augmentation

In this third trial, we retained the core implementation framework from experiments 1 and 2, but introduced image augmentation techniques. Even though this approach is more common in deep learning, we wanted to see if it would have some effect on improving the computer vision models by enhancing the model's robustness to orientation and positional variances, enabling the model to recognize and classify objects across different scales and angles, and increasing the variability of the dataset, preventing overfitting and improving generalization.

### 3.5.4 Trial 4: expanding the mask

In this fourth trial, we closely followed the methodology of the preceding trial but introduced a key variation in our image preprocessing technique. Specifically, we expanded the placental mask used in image processing. Instead of utilizing the mask in its original dimensions, we enlarged it to 1.3 times its original size. This expansion aimed to capture a broader area around the placenta, potentially including additional contextual information that could be relevant for our analysis.

We can summarize the key parameters for experiments 1 to 4 in the following table:

| Parameter | Experiment 1 | Experiment 2 | Experiment 3 | Experiment 4 |
|---|---|---|---|---|
| Class Weights | Not applied | Applied | Tested with and without | Tested with and without |
| Image Augmentation | Not applied | Not applied | Applied | Applied |
| Placental Planes | Anterior + posterior | Anterior + posterior | Anterior + posterior | Anterior + posterior |
| Mask Size | 1:1 | 1:1 | 1:1 | 1:1.3 |

Table 3.9: Summary of Key Parameters for Experiments 1 - 4, Computer Vision.

### 3.5.5 Trial 5: optimizing feature extraction

Given the unsatisfactory performance in prior trials, our approach in Trial 5 consisted on refining the feature extraction process.

For that, we focused solely on GLCM for feature extraction, enhancing its parameters to capture more nuanced textural information. This focus was driven by the inherent suitability of GLCM for capturing the subtle textural patterns present in placental ultrasounds, which are critical for differentiating between various conditions. In this trial, we expected to potentially extract a richer set of features due to the use of multiple distances in the GLCM, through `distance` parameter. Previously confined to a single distance value of `[1]`, we have now broadened this parameter to encompass multiple distances `[1, 2, 3]`.

We also performed a milder form of image augmentation in order to maintain the integrity of textural features, the rationale being that aggressive augmentation could distort the intrinsic patterns that GLCM seeks to analyze, potentially reducing the classifier's ability to accurately discern relevant textural differences. We used the *imgaug* library again to apply horizontal (iaa.Fliplr) and vertical (iaa.Flipud) flips with a 50% probability, as well as to adjust the brightness (iaa.Multiply) and contrast (iaa.LinearContrast) of images with random factors chosen between 0.8 and 1.2. In this case, we created three augmentations per image on the training set.

Regarding the classifiers, Support Vector Classifier (SVC) and Logistic Regression (LR) were chosen for their efficacy in high-dimensional spaces.

We trained the models based on the enhanced parameters of GLCM in two distinct scenarios: one with image augmentation and the other without it. Additionally, we also explored the effects of incorporating class weights versus not including them. This multifaceted approach was aimed at understanding the influence of these variables on model performance and accuracy.

### 3.5.6 Final trial: optimization

We selected the Support Vector Classifier (SVC) for hyperparameter tuning and implemented two different strategies: one incorporating image augmentation and the other without it. The strategy that demonstrated superior performance, whether with or without image augmentation, was subsequently chosen for fine-tuning the models on the anterior placental planes.

We used GridSearchCV for hyperparameter tuning. This method entails an exhaustive search over a specified range of hyperparameters to identify the most effective combination for our model. The hyperparameters under consideration can be found in the appendix.

# Chapter 4

# Results

## 4.1 Deep Learning

### 4.1.1 Comparison Across Trials

As established before, our main focus is to predict placental insufficiency by the presence of one of the three specific conditions (preeclampsia (PRE), tenth percentile birth weight (LBW) or fetal growth restriction(CIR)), and the results shown in this chapter correspond to this approach. However, at the end of this chapter we will also provide the best classifiers for each condition separately. We have chosen the maximum sensitivity within the minimum required specificity (1 - prevalence) as metrics to compare performance across different trials. In the table below, for each best architecture, we have indicated whether class weights (CW), transfer learning (TL), or image augmentation (AUG) were applied. In general, we observe that the best result was obtained through three different experiments: introducing image augmentation in both aggressive and softer forms, and utilizing transfer learning from the custom brains classifier. We can also observe that a slightly larger mask size (Trial 4) decreased model performance.

| TRIAL | Best Architecture | Max Sensitivity |
|---|---|---|
| 1 | ANN | 0.26 |
| 2 | ResNet18 [TL, CW] | 0.32 |
| **3A** | **ResNet18 [TL, AUG]** | **0.36** |
| **3B** | **ResNet18 [TL, AUG]** | **0.36** |
| 4 | ResNet18 [TL, CW, AUG] | 0.32 |
| **5** | **ResNet18 [TL, CW, AUG]** | **0.36** |

Table 4.1: Best Architecture and Max Sensitivity by Trial. 1 - Baseline; 2 - Class Weights; 3A - Soft image augmentation; 3B - Aggressive image augmentation; 4 - Expanded mask; 5 - Transfer learning.

We can observe that the effect that class weights had on model performance was not consistent on average, leading to slight increases or decreases in sensitivity for different trials. In the table below, we compare the average maximum sensitivity achieved with class weights (AMS-CW) and without (AMS) by trial.

| Comparison | AMS | AMS-CW |
|---|---|---|
| TRIAL 1-2 | 0.21 | 0.18 |
| TRIAL 3 | 0.23 | 0.24 |
| TRIAL 4 | 0.21 | 0.24 |
| TRIAL 5 | 0.27 | 0.26 |

Table 4.2: Impact of Class Weights

### 4.1.2 Comparison Across Architectures

Extending our analysis to compare the performance of different architectures, we observed that ResNet18, with transfer learning and image augmentation, outperformed the other architectures.

| Architecture | Configuration | Best Sensitivity |
|---|---|---|
| ANN | [CW, AUG] | 0.29 |
| VGG16 | [TL, AUG] | 0.32 |
| ResNet50 | [TL, CW] | 0.32 |
| MobileNet | [CW, AUG] | 0.33 |
| **ResNet18** | **[TL, AUG]** | **0.36** |

Table 4.3: Performance of Different Neural Network Architectures Trials 1 - 4

### 4.1.3 Results after optimization

The architecture we chose to optimize was ResNet18, with softer image augmentation, transfer learning, and no class weights. Out of the three architectures that yielded the same sensitivity, this was the one that returned the highest PLR.

Initially, we re-trained the same architecture and configuration on anterior planes of the placenta only. However, this did not result in any significant improvement in sensitivity. Subsequently, we attempted to optimize parameters with Optuna for the original best architecture. Unfortunately, this optimization strategy also failed to enhance sensitivity.

## 4.2 Computer Vision

### 4.2.1 Comparison Across Trials

As before, we have chosen the maximum sensitivity within the minimum required specificity (1 - prevalence) as metrics to compare performance across different trials. In the table below, for each best architecture, we have indicated whether class weights (CW) or image augmentation (AUG) were applied. In general, we observe that the best result was obtained in trial 5, that included a more refined version of GLCM and softer image augmentation techniques. Overall, the rest of trials performed similarly with no great differences.

| TRIAL | Best Architecture | Max Sensitivity |
|---|---|---|
| 1 | HOG + LR | 0.26 |
| 2 | HOG + SVC [CW] | 0.25 |
| 3 | HOG + XGBOOST [CW, AUG] | 0.27 |
| 4 | GLCM + SVC [CW, AUG] | 0.29 |
| **5** | **GLCM + SVC** | **0.32** |

Table 4.4: Best Architecture and Max Sensitivity by Trial

### 4.2.2 Comparison Across Architectures

GLCM with SVC and HOG with XGBOOST yielded the best results. At the time of optimization, we will focus on GLCM due to its ability to analyze textures, which is more relevant to the complexity of placental tissue.

| Architecture | Configuration | Best Sensitivity |
|---|---|---|
| **GLCM + SVC** | **[CW, AUG]** | **0.29** |
| GLCM + XGBOOST | [CW, AUG] | 0.23 |
| GLCM + LR | [CW] | 0.23 |
| HOG + SVC | [CW, AUG] | 0.27 |
| **HOG + XGBOOST** | **[CW, AUG]** | **0.29** |
| HOG + LR | [CW, AUG] | 0.27 |

Table 4.5: Performance of Different Architectures Trials 1 - 4

### 4.2.3 Results after optimization

In order to optimize the best performing architecture, we undertook hyperparameter tuning on the Support Vector Classifier (SVC), systematically exploring various configurations, with and without class weights and image augmentation. Then, we also attempted to refine the architecture by training it exclusively on the anterior planes of the placenta, which offers better visibility of the region. In all cases, the original prevalence of the condition when considering anterior planes only remained the same. However, none of these techniques resulted in a discernible improvement in performance.

## 4.3 Overall best performers

While building a classifier for diagnosing placental insufficiency based on the triple criteria—tenth birth weight percentile (LBW), fetal growth restriction (CIR), and preeclampsia (PRE)—we also pursued individual implementations for each criterion. Employing the same systematic approach, we conducted trials and applied optimization methodologies tailored to the unique characteristics of each condition. The detailed results of these approaches are documented in the appendix. In summary, these are the definitive best-performing models identified for each specific criterion:

| **Criteria** | C3 | LBW | PRE | CIR |
|---|---|---|---|---|
| **Best Architecture** | ResNet18 [TL, AUG] | ANN | GLCM + SVC [CW] | ANN [CW] |
| **Placental Plane** | Both | Both | Anterior | Both |
| **Sensitivity** | 0.36 | 0.29 | **0.56** | 0.33 |
| **AUC** | 0.60 | 0.53 | 0.76 | 0.55 |
| **F1-Score Pos Class** | 0.33 | 0.27 | 0.53 | 0.29 |
| **Accuracy** | 0.69 | 0.77 | 0.96 | 0.91 |
| **Specificity** | 0.79 | 0.86 | 0.99 | 0.96 |
| **PPV** | 0.31 | 0.27 | 0.67 | 0.33 |
| **NPV** | 0.82 | 0.86 | 0.97 | 0.94 |
| **PLR** | 1.64 | 1.96 | 34.67 | 6.42 |
| **NLR** | 0.83 | 0.84 | 0.56 | 0.78 |

Table 4.6: Best Classifiers by Criteria

For the classifiers to have clinical relevance, a specificity of 1 - prevalence of the condition and a sensitivity of at least 0.50 was necessary. Unfortunately, this threshold was only met for the prediction of Preeclampsia, using a SVC classifier with features extracted by GLCM and trained on anterior placental planes. The classifier underwent hyperparameter tuning, exploring various combinations of $C$ and $\gamma$ values, different kernels, and the inclusion/exclusion of class weights, getting {'C': 100, 'class_weight': 'balanced', 'gamma': 0.001, 'kernel': 'rbf'} as best parameters. GLCM was calculated for pixel pairs at distances 1, 2, and 3, and angles 0, $\pi/4$, $\pi/2$, and $3\pi/4$. The symmetric 256-level GLCM was normalized, treating values as probabilities. Extracted features included dissimilarity, correlation, homogeneity, contrast, Angular Second Moment (ASM), and energy.

### Combined outcome: ROC and training curves

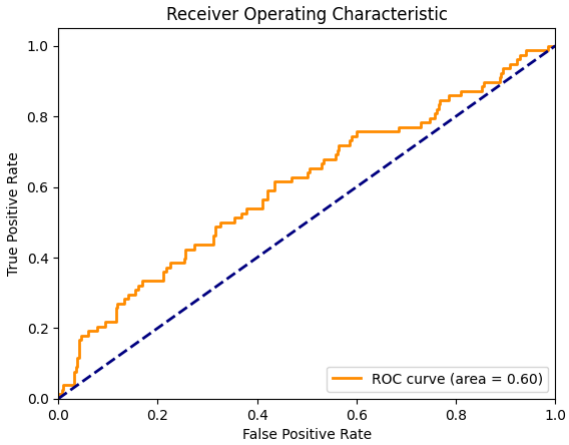


Figure 4.1: C3: ROC curve

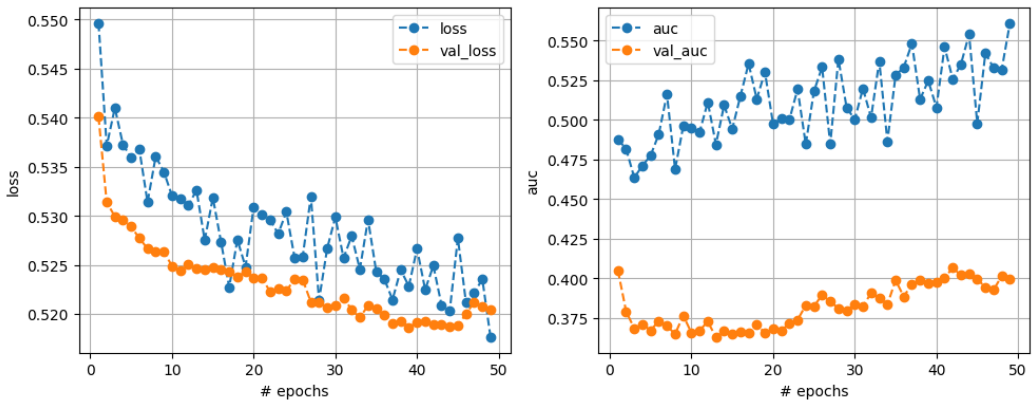


Figure 4.2: C3: training

For the combined outcome, the performance of the chosen model (ResNet18 with image augmentation and transfer learning from ImageNet) is modest. The maximum sensitivity of 0.36 suggests that the model's ability to correctly identify true positive cases is limited. The F1-Score for the positive class is 0.33, indicating a moderate balance between making accurate positive predictions and capturing a reasonable proportion of actual positive instances. The positive predictive value (PPV) is only 0.31, revealing a considerable proportion of false positives. While the negative predictive value (NPV) is relatively high at 0.82, suggesting reliability in negative predictions, the positive likelihood ratio (PLR) and negative likelihood ratio (NLR) of 1.64 and 0.83, respectively, underscore the model's limitations. In conclusion, there is room for improvement, and further investigation may be needed to enhance performance.

### Low Birth Weight: ROC and training curves

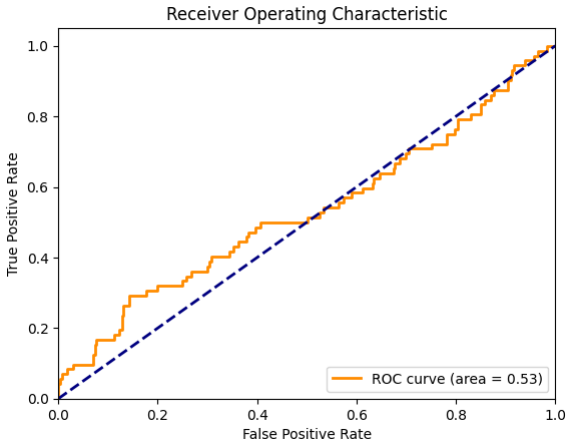


Figure 4.3: LBW: ROC curve

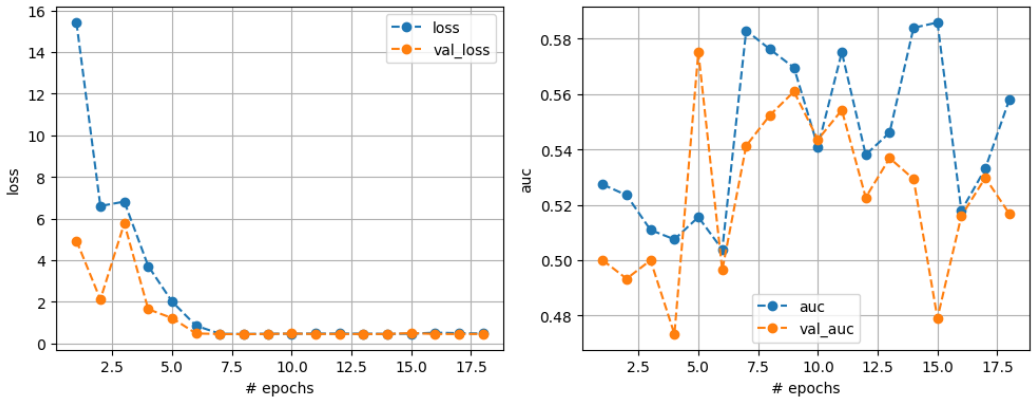


Figure 4.4: LBW: training

For the assessment of LBW prediction, the performance of the chosen model (an Artificial Neural Network), is characterized by modest outcomes. The model achieves a maximum sensitivity of 0.29, indicating a limitation in its ability to accurately identify true positive cases. The F1-Score for the positive class at 0.27 suggests a moderate balance between making accurate positive predictions and capturing a reasonable proportion of actual positive instances. The positive predictive value (PPV) is low at 0.31, pointing towards a considerable proportion of false positives. The positive likelihood ratio (PLR) and negative likelihood ratio (NLR) are calculated at 1.96 and 0.84, respectively, revealing certain limitations in the model's diagnostic performance. In conclusion, the presented results suggest that the ANN model, while demonstrating some effectiveness, leaves room for improvement. Further investigation and refinement may be essential to enhance its performance.

### Fetal Growth Restriction: ROC and training curves

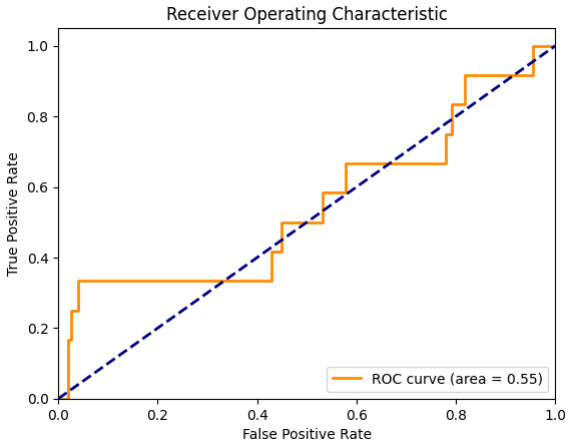


Figure 4.5: CIR: ROC curve

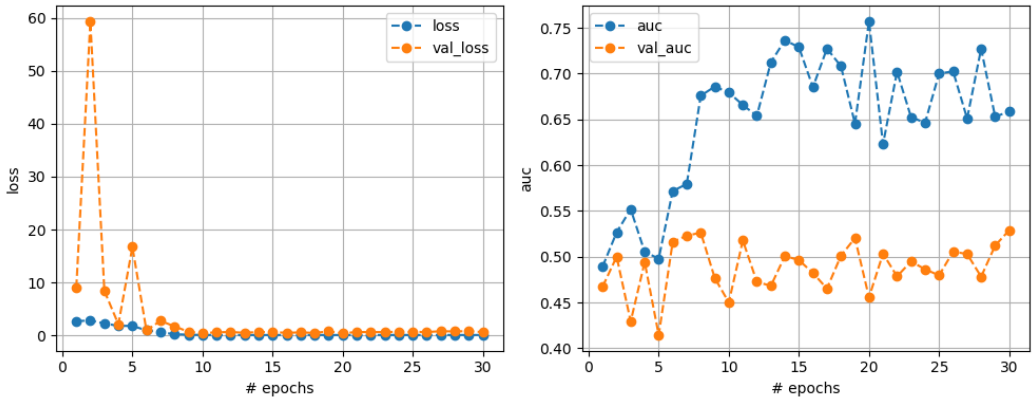


Figure 4.6: CIR: training

In evaluating the performance of CIR prediction for the selected model (an Artificial Neural Network with class weighting), the outcomes indicate a modest level of effectiveness. The model achieves a maximum sensitivity of 0.33, showcasing poor ability to correctly identify true positive cases. The F1-Score for the positive class at 0.29 suggests a moderate balance between making accurate positive predictions and capturing a reasonable proportion of actual positive instances. In summary, the ANN model with class weighting to predict CIR lacks strength.

**Preeclampsia: ROC curve**

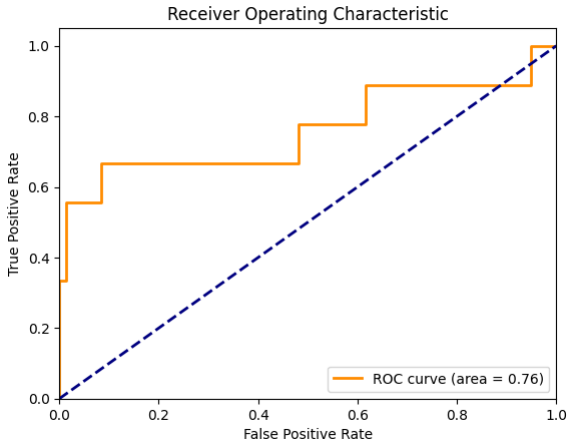


Figure 4.7: PRE: ROC curve

For this model, a PLR of 34.67 indicates a strong likelihood of accurately identifying preeclampsia when the test result is positive. The PPV of 0.67 is fairly high, which is encouraging for clinical use, although there's room to improve to reduce false positive rates since it suggests that two thirds of positive results are true cases of preeclampsia, a common challenge in conditions with low prevalence.

However, this result needs to be taken with caution. Given the low prevalence of preeclampsia (6%), Burderer's method dictates a minimum sample size of 401 patients in the test set for the results to attain statistical significance. As of the current project development stage, the dataset falls short of this requirement. Therefore, a reevaluation of this test should be considered upon the expansion of the dataset to ensure robust statistical conclusions.

# Chapter 5

# Conclusion and Outlook

## 5.1 Conclusion

In the context of identifying placental insufficiency, the clinical relevance of accurate and reliable models is important. When these models demonstrate high efficacy and yield satisfactory results, their implementation translates into an improvement in maternal-fetal healthcare. Women identified as at risk of placental insufficiency through these models can benefit from enhanced monitoring protocols. The heightened vigilance allows healthcare professionals to conduct more periodic and closer assessments, enabling early detection of potential complications. This proactive approach empowers medical practitioners to implement timely interventions and personalized care plans, ultimately aiming to mitigate the risks associated with placental insufficiency and to contribute to improved outcomes for both the mother and the developing fetus.

The outcomes of our study, which aimed to predict malfunctioning of the placenta (inferred from conditions such as tenth percentile weight, fetal growth restriction, and preeclampsia) using traditional Deep Learning and Computer Vision architectures, reveal that, in general, these methods fall short of the robustness required for clinical diagnostic tools.

The model for predicting preeclampsia shows moderate strength. In theory, with a sensitivity higher than 0.50 within the minimum specifity required of 1 - prevalence, clinicians could endorse its clinical application, which means that medical professionals would conduct more frequent and in-depth assessments for individuals identified by the model. However, the results lack statistical significance due to the insufficient sample size. The required sample size for statistical significance exceeds the current available data.

In general, only the combination of all three criteria and low birth weight alone met the size requirements for statistically significant models. However, this combination includes both anterior and posterior placental planes. Anterior planes, which provide clearer images, could

have the potential to enhance results. Unfortunately, at the time of developing this project, the dataset size for anterior planes did not meet the minimum test size requirement according to Burderer's method.

In summary, while these models show promise in specific areas, their utility as standalone diagnostic tools is limited at the moment.

## 5.2 Future work

Moving forward, enhancements in sensitivity are essential for the models to have clinical use, to enhance patient outcomes in conditions such as tenth percentile weight, fetal growth restriction, and preeclampsia.

Despite extensive exploration of fine-tuning, its limited success arises a need to contemplate alternative strategies. The model's enhanced performance in predicting preeclampsia, following training on anterior placental planes, may suggest that acquiring more images of anterior planes could result in improving predictive power.

Moreover, it is crucial to acknowledge the possibility that ultrasound images may have inherent limitations in predicting conditions such as tenth percentile weight, fetal growth restriction, and preeclampsia, given the current capabilities of deep learning and computer vision techniques. Improving the quality of images, or using a different type of medical imaging, could improve performance of predictive models.

Taking a nuanced view of the capabilities of ultrasound images and considering alternative approaches beyond those explored in this project could prove advantageous. This approach enables a comprehensive grasp of both the potential and limitations of predictive models within the specific context of the studied conditions.

Adjusting expectations and strategies may entail incorporating these tools into a more extensive diagnostic process rather than depending solely on them as independent solutions. This holistic approach guarantees a more in-depth exploration of predictive models' capabilities and their relevance in various clinical scenarios.

# Bibliography


[1] Natalie M. Gude, Claire T. Roberts, Bill Kalionis, and Rosalind G. King. Growth and function of the normal human placenta. *Thrombosis Research*, 114(5-6):397–407, 2004. doi: 10.1016/j.thromres.2004.06.038.

[2] Sarosh Rana, Eliza Lemoine, Jeffrey P. Granger, and S. Ananth Karumanchi. Preeclampsia: Pathophysiology, challenges, and perspectives. *Circulation Research*, 124(7):1094–1112, Mar 2019. doi: 10.1161/CIRCRESAHA.118.313276. Erratum in: Circ Res. 2020 Jan 3;126(1):e8.

[3] Y. Ohgiya, H. Nobusawa, N. Seino, O. Miyagami, N. Yagi, S. Hiroto, J. Munechika, M. Hirose, N. Takeyama, N. Ohike, R. Matsuoka, A. Sekizawa, and T. Gokan. Mr imaging of fetuses to evaluate placental insufficiency. *Magnetic Resonance in Medical Sciences*, 15(2): 212–219, 2016. doi: 10.2463/mrms.mp.2015-0051.

[4] Sonia Dahdouh, Nickie Andescavage, Shantanu Yewale, Alexa Yarish, Danielle Lanham, Dorothy Bulas, Adré J. du Plessis, and Catherine Limperopoulos. In vivo placental mri shape and textural features predict fetal growth restriction and postnatal outcome. *Journal of Magnetic Resonance Imaging*, 47(2):449–458, Feb 2018. doi: 10.1002/jmri.25806.

[5] Yann LeCun, Leon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 586–591, 1998. doi: 10.1109/CVPR.1998.694492.

[6] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, 2012.

[7] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for

image recognition. *arXiv preprint arXiv:1512.03385*, 2015. doi: 10.48550/arXiv.1512.03385.

[9] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015. doi: 10.1109/CVPR.2015.7298594.

[10] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *arXiv:1406.2661 [stat.ML]*, 2014. Focus: To learn more, DOI: https://doi.org/10.48550/arXiv.1406.2661.

[11] Mengping Yang, Ceyuan Yang, Yichi Zhang, Qingyan Bai, Yujun Shen, and Bo Dai. Revisiting the evaluation of image synthesis with gans. *arXiv:2304.01999 [cs.CV]*, 2023. NeurIPS 2023 datasets and benchmarks track, DOI: https://doi.org/10.48550/arXiv.2304.01999.

[12] Huan Ling, Karsten Kreis, Daiqing Li, Seung Wook Kim, Antonio Torralba, and Sanja Fidler. Editgan: High-precision semantic image editing. *arXiv:2111.03186 [cs.CV]*, 2021. DOI: https://doi.org/10.48550/arXiv.2111.03186.

[13] Sai Ashrith Aduwala, Manish Arigala, Shivan Desai, Heng Jerry Quan, and Magdalini Eirinaki. Deepfake detection using gan discriminators. In *2021 IEEE Seventh International Conference on Big Data Computing Service and Applications (BigDataService)*, pages 69–77, 2021. doi: 10.1109/BigDataService52369.2021.00014.

[14] Ronan Collobert, Jason Weston, Leon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *arXiv:1103.0398 [cs.LG]*, 2011. DOI: https://doi.org/10.48550/arXiv.1103.0398.

[15] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv:1312.5602 [cs.LG]*, 2013. DOI: https://doi.org/10.48550/arXiv.1312.5602.

[16] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. *arXiv:1411.4555 [cs.CV]*, 2014. DOI: https://doi.org/10.48550/arXiv.1411.4555.

[17] Ana Cláudia Akemi Matsuki de Faria, Felype de Castro Bastos, José Victor Nogueira Alves da Silva, Vitor Lopes Fabris, Valeska de Sousa Uchoa, Décio Gonçalves

de Aguiar Neto, and Claudio Filipi Goncalves dos Santos. Visual question answering: A survey on techniques and common trends in recent literature. *arXiv:2305.11033*, 2023. doi: 10.48550/arXiv.2305.11033. URL https://doi.org/10.48550/arXiv.2305.11033.

[18] MedlinePlus - X-Rays. URL https://medlineplus.gov/xrays.html. https://arthdiagnostics.com/difference-between-x-ray-mri-ultrasound-and-ct-scan. Accessed on 2023-10-17.

[19] National Institute of Biomedical Imaging and Bioengineering - X-Rays. URL https://www.nibib.nih.gov/science-education/science-topics/x-rays. https://arthdiagnostics.com/difference-between-x-ray-mri-ultrasound-and-ct-scan. Accessed on 2023-10-17.

[20] iStockphoto. URL https://www.istockphoto.com/. https://www.istockphoto.com/. Accessed on 2023-10-17.

[21] Arth Diagnostics. Difference Between X-Ray, MRI, Ultrasound, and CT Scan. URL https://arthdiagnostics.com/difference-between-x-ray-mri-ultrasound-and-ct-scan. https://arthdiagnostics.com/difference-between-x-ray-mri-ultrasound-and-ct-scan. Accessed on 2023-10-17.

[22] Case Western Reserve University School of Medicine. Magnetic resonance imaging (mri) of the brain and spine: Basics, 10 2023. URL https://case.edu/med/neurology/NR/MRI%20Basics.htm. https://case.edu/med/neurology/NR/MRI%20Basics.htm/. Accessed on 2023-10-17.

[23] Mayo Clinic. Ultrasound, 10 2023. URL https://www.mayoclinic.org/tests-procedures/ultrasound/about/pac-20395177. https://www.mayoclinic.org/tests-procedures/ultrasound/about/pac-20395177/. Accessed on October 17, 2023.

[24] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893 vol. 1, 2005. doi: 10.1109/CVPR.2005.177.

[25] ResearchGate. The description of the hog feature extraction process, 2023. URL https://www.researchgate.net/figure/The-description-of-the-HOG-feature-extraction-process_fig2_362428076. Accessed on December 24, 2023.

[26] Robert M Haralick, Kannappan Shanmugam, and Its'hak Dinstein. Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, 3(6):610–621, 1973.

[27] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014.

[28] Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, page 807–814, Madison, WI, USA, 2010. Omnipress. ISBN 9781605589077.

[29] Herbert Robbins and Sutton Monro. A stochastic approximation method. *Annals of Mathematical Statistics*, 22(3):400–407, September 1951. doi: 10.1214/aoms/1177729586.

[30] DataHacker. Deep learning - alexnet architecture, Year 2018. URL https://datahacker.rs/deep-learning-alexnet-architecture/. https://datahacker.rs/deep-learning-alexnet-architecture/. Accessed on Date October 17, 2023.

[31] ImageNet Large Scale Visual Recognition Challenge 2014, 2014. URL https://www.image-net.org/challenges/LSVRC/2014/. https://www.image-net.org/challenges/LSVRC/2014/.

[32] Max Ferguson, Ronay Ak, Yung-Tsun Tina Lee, and Kincho H. Law. Automatic localization of casting defects with convolutional neural networks. In *Proceedings of the IEEE International Conference on Big Data (Big Data)*. IEEE, December 2017. doi: 10.1109/BigData.2017.8258115.

[33] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by error propagation. Technical Report ICS 8504, Institute for Cognitive Science, University of California, San Diego, California, September 1985.

[34] Michael I Jordan. Serial order: A parallel distributed processing approach. Technical Report ICS 8604, Institute for Cognitive Science, University of California, San Diego, California, May 1986.

[35] Wikipedia contributors. Residual neural network, 2023. URL https://en.wikipedia.org/wiki/Residual_neural_network. Accessed on Date October 18, 2023.

[36] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. doi: 10.48550/arXiv.1608.06993. URL https://arxiv.org/abs/1608.06993.

[37] Hammam Alshazly, Christoph Linse, Mohamed Abdalla, Erhardt Barth, and Thomas Martinetz. Covid-nets: Deep cnn architectures for detecting covid-19 using chest ct scans. *PeerJ Comput Sci*, 7:e655, July 2021. doi: 10.7717/peerj-cs.655.

[38] R. Kundu, R. Das, Z.W. Geem, G.T. Han, and R. Sarkar. Pneumonia detection in chest x-ray images using an ensemble of deep learning models. *PLoS One*, 16(9):e0256630, Sep 2021. doi: 10.1371/journal.pone.0256630.

[39] Tonmoy Hossain, Fairuz Shadmani Shishir, Mohsena Ashraf, MD Abdullah Al Nasim, and Faisal Muhammad Shah. Brain tumor detection using convolutional neural network. In *2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT)*, pages 1–6, 2019. doi: 10.1109/ICASERT.2019.8934561.

[40] Feres Jerbi, Noura Aboudi, and Nawres Khhlifa. Automatic classification of ultrasound thyroids images using vision transformers and generative adversarial networks. *Scientific African*, 20:e01679, 2023. ISSN 2468-2276. doi: 10.1016/j.sciaf.2023.e01679. URL https://www.sciencedirect.com/science/article/pii/S2468227623001357.

[41] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2019. doi: 10.48550/arXiv.1905.11946. URL https://arxiv.org/abs/1905.11946.

[42] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021. doi: 10.48550/arXiv.2010.11929. URL https://arxiv.org/abs/2010.11929.

[43] Philip M. Cheng and Harmeet S. Malhi. Transfer learning with convolutional neural networks for classification of abdominal ultrasound images. *Journal of Digital Imaging*, 30:234–243, 2017. doi: 10.1007/s10278-016-9929-2.

[44] X. P. Burgos-Artizzu, D. Coronado-Gutiérrez, B. Valenzuela-Alcaraz, E. Bonet-Carne, E. Eixarch, F. Crispi, and E. Gratacós. Evaluation of deep convolutional neural networks for automatic classification of common maternal fetal ultrasound planes. *Sci Rep*, 10: 10200, 2020. doi: 10.1038/s41598-020-67076-5.

[45] X. P. Burgos-Artizzu, D. Coronado-Gutiérrez, B. Valenzuela-Alcaraz, K. Vellvé, E. Eixarch, F. Crispi, E. Bonet-Carne, M. Bennasar, and E. Gratacos. Analysis of maturation features in fetal brain ultrasound via artificial intelligence for the estimation of gestational age. *Am J Obstet Gynecol MFM*, 3:100462, 2021. doi: 10.1016/j.ajogmf.2021.100462.

[46] D. Coronado-Gutiérrez, E. Eixarch, E. Monterde, I. Matas, P. Traversi, E. Gratacos, E. Bonet-Carne, and X. P. Burgos-Artizzu. Automatic deep learning-based pipeline for automatic delineation and measurement of fetal brain structures in routine mid-trimester ultrasound images. *Fetal Diagn Ther*, 2023. doi: 10.1159/000533203. Epub ahead of print.

[47] J.E. Wardinger and S. Ambati. Placental insufficiency. *Treasure Island*, 2023. URL https://www.ncbi.nlm.nih.gov/books/NBK563171/. Updated 2022 Oct 3.

[48] WebMD. What is placental insufficiency?, 2023. URL https://www.webmd.com/baby/what-is-placental-insufficiency. https://www.webmd.com/baby/what-is-placental-insufficiency/. Accessed on 2023-10-18.

[49] R. J. Martinez-Portilla, J. Caradeux, E. Meler, D. L. Lip-Sosa, A. Sotiriadis, and F. Figueras. Third-trimester uterine artery doppler for prediction of adverse outcome in late small-for-gestational-age fetuses: systematic review and meta-analysis. *Ultrasound in Obstetrics & Gynecology*, 54:723–730, 2019. doi: 10.1002/uog.21940. URL https://doi.org/10.1002/uog.21940.

[50] Sahar Dahdouh, Nickie Andescavage, Sanika Yewale, and et al. In vivo placental mri shape and textural features predict fetal growth restriction and postnatal outcome. *J Magn Reson Imaging*, 47(2):449–458, 2018. doi: 10.1002/jmri.25806.

[51] Nancy M Buderer. Statistical methodology: I. incorporating the prevalence of disease into the sample size calculation for sensitivity and specificity. *Academic Emergency Medicine*, 3(9):895–900, 1996. doi: 10.1111/j.1553-2712.1996.tb03538.x.

[52] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. doi: 10.48550/arXiv.1704.04861. URL https://arxiv.org/abs/1704.04861.

[53] X. P. Burgos-Artizzu, D. Coronado-Gutierrez, B. Valenzuela-Alcaraz, E. Bonet-Carne, E. Eixarch, F. Crispi, and E. Gratacós. Fetal_planes_db: Common maternal-fetal ultrasound images. *Nature Scientific Reports*, 10:10200, 2020. doi: 10.5281/zenodo.3904280. Data set.

# Appendix

## 5.3 Trials Overview

### 5.3.1 Summary of Deep Learning trials

| TRIAL | DESCRIPTION |
|---|---|
| TRIAL 1 | Data as-is. |
| TRIAL 2 | Data as-is with class weights. |
| TRIAL 3 | Data with image augmentation, with and without class weights. |
| TRIAL 4 | Data with expanded mask, image augmentation, with and without class weights. |
| TRIAL 3B | Repeat TRIAL 3 with softer image augmentation. |
| TRIAL 5 | Train brain classifier to use for transfer-learning, with softer image augmentation. |
| LAST TRIAL | Fine-tune with anterior planes and optuna based on best model for each criterion. |

Table 5.1: Overview of Deep Learning Trials

### 5.3.2 Summary of Computer Vision trials

| TRIAL | DESCRIPTION |
|---|---|
| TRIAL 1 | Data as-is. |
| TRIAL 2 | Data as-is with class weights. |
| TRIAL 3 | Data with image augmentation, with and without class weights. |
| TRIAL 4 | Data with expanded mask, image augmentation, with and without class weights. |
| TRIAL 5 | Refined GLCM, softer image augmentation, GLCM with SVC and LR. |
| LAST TRIAL | 1. Hyperparameter tuning for SVC, with and without image augmentation. |
| LAST TRIAL | 2. Fine-tune on anterior planes based on best results. |

Table 5.2: Overview of Computer Vision Trials

## 5.4 Parameters

### 5.4.1 Aggressive Image Augmentation

- `rotation_range=20`: Random rotations of the image within a range of 20 degrees.
- `width_shift_range=0.2` and `height_shift_range=0.2`: Random shifts of the image horizontally and vertically by up to 20% of the image's width and height.
- `shear_range=0.2`: Applying shear transformations to the image.
- `zoom_range=0.2`: Random zooming of the image by up to 20%.
- `horizontal_flip=True`: Randomly flipping the image horizontally.
- `fill_mode='nearest'`: The strategy used to fill in newly created pixels after a rotation or shift, in this case, by replicating the nearest pixel value.

### 5.4.2 Soft Image Augmentation

- Horizontal Flip: `tf.image.random_flip_left_right(image)` - This line randomly flips the image horizontally (left to right).
- Vertical Flip: `tf.image.random_flip_up_down(image)` - This line randomly flips the image vertically (upside down).
- Brightness Adjustment: `tf.image.random_brightness(image, max_delta=0.2)` - This function alters the brightness of the image. The `max_delta` parameter controls the maximum amount of brightness change. A random value within this range is selected for each image, making some images brighter and others darker. This step ensures the model can recognize objects under varying lighting conditions.
- Contrast Adjustment: `tf.image.random_contrast(image, lower=0.8, upper=1.2)` - This line adjusts the contrast of the image within the specified range. The lower and upper parameters define the range for the contrast scaling. Altering the contrast helps the model in identifying features under different levels of image clarity and sharpness.

### 5.4.3 HOG Feature Extraction

The `extract_features` function is central to processing the image dataset. Its role is to transform each image into a feature vector using the Histogram of Oriented Gradients (HOG) method, a popular feature descriptor in computer vision. The steps in this process are:

- Image Preprocessing

  - The image is loaded in grayscale using OpenCV (cv2.imread), which simplifies the HOG computation as it relies on gradient information rather than color.
  - Images are resized to a standard dimension (128x64 pixels) to ensure uniformity in feature extraction. This resizing is crucial because HOG requires a fixed-size input.

- HOG Descriptor Configuration

  - **Window Size (`win_size`)**: The size of the window over which the HOG descriptor is calculated, set to $(128, 64)$, which is the size of the resized image.
  - **Block Size (`block_size`)**: The size of the blocks within the window, set to $(16, 16)$. Blocks are square regions from which HOG features are normalized.
  - **Block Stride (`block_stride`)**: The stride of the blocks across the window, set to $(8, 8)$. This parameter determines the overlap between consecutive blocks.
  - **Cell Size (`cell_size`)**: The size of the cells within a block, set to $(8, 8)$. Cells are sub-regions within blocks for which histograms are computed.
  - **Number of Bins (`n_bins`)**: The number of histogram bins used to count gradient orientations, set to 9. This defines the granularity of the orientation binning.

### 5.4.4 GLCM Feature Extraction

Here we count with another `extract_features` function to transform each image into a feature vector using the Gray Level Co-occurrence Matrix (GLCM) method. The steps in this process are:

- Image Preprocessing

  - The image is initially read in grayscale using OpenCV's cv2.imread function. Grayscale conversion is essential for GLCM, which analyzes textural features based on gray level intensities.
  - The image is resized to a standard dimension (128x64 pixels). If resizing fails (e.g., due to an unusual image format), the original image is used. Consistent image sizing ensures uniformity in the feature extraction process.

- GLCM Key Parameters

  - **Distances**: This parameter in the GLCM function is set to `[1]`, indicating the pixel pair distance for computing the matrix.
  - **Angles**: Defined as `[0, π/4, π/2, 3π/4]`, it represents the four directions (0, 45, 90, and 135 degrees) used in the GLCM calculations.

- **Levels**: Set to `256`, it specifies the number of gray levels in the GLCM, thereby defining the granularity of the texture analysis.
- **Symmetric**: A boolean parameter, set to `True`, indicating that the GLCM is symmetric, which is an important aspect of the texture feature representation.
- **Normed**: Also a boolean parameter, set to `True`. This indicates that the GLCM is normalized, ensuring that the texture features are standardized for comparison.
- **GLCM Properties**: The properties extracted from the GLCM include 'dissimilarity', 'correlation', 'homogeneity', 'contrast', 'ASM', and 'energy'. These properties are crucial for texture analysis and are the primary features extracted from the GLCM.

### 5.4.5 Deep Learning optimization: Optuna

After identifying the most effective model—trained either on anterior planes, or a combination of both anterior and posterior—we proceeded to hyperparameter tuning using Optuna, an advanced framework specifically designed for optimizing machine learning models. Optuna automates the process of finding the most effective hyperparameters for our model. The parameters we focused on refining included:

- **Learning Rate:** Optimized within a logarithmic range from $1 \times 10^{-6}$ to $1 \times 10^{-4}$, using `trial.suggest_loguniform('learning_rate', 1e-6, 1e-4)`. This range allows exploration of a wide spectrum of learning rates, facilitating fine-tuning of the model's training process.

- **Number of Layers:** Determined by `trial.suggest_int('n_layers', 1, 3)`, allowing the model to have 1, 2, or 3 custom Dense layers.

- **Number of Neurons in Each Layer:** Configured for each layer $i$ through `trial.suggest_int('`
`50, 200)`, with a range from 50 to 200 units per layer.

- **Dropout Rate:** Set following each Dense layer, with values ranging from 0.2 to 0.5 as per `trial.suggest_uniform('dropout_l', 0.2, 0.5)`. This assists in reducing overfitting by randomly nullifying a portion of input units during training.

These parameters were consistently incorporated into our optimization process whenever the selected best-performing architecture could accommodate them. In instances where the architecture was not conducive to all these adjustments, we selectively included only those parameters that were compatible. These parameters were iteratively refined across 10 trials with the objective of maximizing the Area Under the Curve (AUC) metric.

### 5.4.6 Computer Vision optimization: Grid Search

We selected the Support Vector Classifier (SVC) for hyperparameter tuning and implemented two different strategies: one incorporating image augmentation and the other without it. The strategy that demonstrated superior performance, whether with or without image augmentation, was subsequently chosen for fine-tuning the models on the anterior placental planes.

We used GridSearchCV for hyperparameter tuning. This method entails an exhaustive search over a specified range of hyperparameters to identify the most effective combination for our model. The hyperparameters under consideration can be found in the appendix.

**For rbf and poly kernels:**

- `C`: Regularization parameter, explored across a range of values: [1, 10, 100, 1000, 10000].

- `kernel`: Types of kernels evaluated include 'rbf' and 'poly'.

- `gamma`: Kernel coefficient, with values [0.001, 0.01, 0.1, 0.5].

- `class_weight`: Options for class balancing include [None, 'balanced'].

**For linear kernel:**

- `C`: Regularization parameter, with the same range as in Test 1: [1, 10, 100, 1000, 10000].

- `class_weight`: Class balancing options, similar to Test 1: [None, 'balanced'].

**General Key parameters**

- Image augmentation: Tested with and without image augmentation.

- Class weights: tuned as hyperparameter.

- Placental planes: Anterior, posterior, combined.

- Mask size: 1:1.

## 5.5 Deep Learning full results

#### 5.5.0.1 C3 (Preeclampsia, fetal growth restriction or birth weight percentile below 10)

| Best Architecture | Max Sensitivity | AUC | F1-Score Pos Class | Accuracy | Specificity | PPV | NPV | PLR | NLR |
|---|---|---|---|---|---|---|---|---|---|
| ANN | 0.26 | 0.52 | 0.26 | 0.70 | 0.82 | 0.27 | 0.80 | 1.36 | 0.92 |
| VGG16-no-TL | 0.18 | 0.48 | 0.18 | 0.68 | 0.82 | 0.20 | 0.78 | 0.91 | 1.02 |
| VGG16-TL | 0.18 | 0.45 | 0.18 | 0.67 | 0.81 | 0.19 | 0.78 | 0.88 | 1.03 |
| ResNet50-no-TL | 0.18 | 0.57 | 0.19 | 0.70 | 0.85 | 0.23 | 0.79 | 1.08 | 0.99 |
| ResNet50-TL | 0.23 | 0.57 | 0.23 | 0.68 | 0.80 | 0.23 | 0.79 | 1.11 | 0.97 |
| MobileNet-no-TL | 0.23 | 0.50 | 0.23 | 0.68 | 0.81 | 0.24 | 0.79 | 1.13 | 0.97 |
| MobileNet-TL | 0.15 | 0.48 | 0.15 | 0.67 | 0.81 | 0.17 | 0.78 | 0.76 | 1.06 |
| ResNet18-no-TL | 0.23 | 0.54 | 0.22 | 0.67 | 0.80 | 0.23 | 0.79 | 1.09 | 0.98 |
| ResNet18-TL | 0.22 | 0.46 | 0.21 | 0.67 | 0.80 | 0.22 | 0.79 | 1.03 | 0.99 |
| ResNet18-TL | 0.23 | 0.46 | 0.22 | 0.67 | 0.80 | 0.23 | 0.79 | 1.07 | 0.98 |

Table 5.3: C3 - Trial 1 results

| Best Architecture | Max Sensitivity | AUC | F1-Score Pos Class | Accuracy | Specificity | PPV | NPV | PLR | NLR |
|---|---|---|---|---|---|---|---|---|---|
| ANN | 0.26 | 0.52 | 0.26 | 0.70 | 0.82 | 0.27 | 0.80 | 1.36 | 0.92 |
| VGG16-no-TL | 0.18 | 0.48 | 0.18 | 0.68 | 0.82 | 0.20 | 0.78 | 0.91 | 1.02 |
| VGG16-TL | 0.19 | 0.44 | 0.19 | 0.68 | 0.81 | 0.21 | 0.78 | 0.97 | 1.01 |
| ResNet50-no-TL | 0.13 | 0.44 | 0.14 | 0.71 | 0.87 | 0.19 | 0.78 | 0.87 | 1.02 |
| ResNet50-TL | 0.32 | 0.56 | 0.29 | 0.68 | 0.79 | 0.28 | 0.81 | 1.44 | 0.88 |
| MobileNet-no-TL | 0.27 | 0.51 | 0.25 | 0.67 | 0.79 | 0.25 | 0.79 | 1.20 | 0.95 |
| MobileNet-TL | 0.17 | 0.50 | 0.17 | 0.68 | 0.82 | 0.19 | 0.78 | 0.86 | 1.03 |
| ResNet18-no-TL | 0.18 | 0.47 | 0.19 | 0.69 | 0.84 | 0.22 | 0.79 | 1.01 | 1.00 |
| ResNet18-TL | 0.32 | 0.57 | 0.30 | 0.70 | 0.80 | 0.30 | 0.81 | 1.57 | 0.86 |
| ResNet18-TL | 0.23 | 0.46 | 0.22 | 0.67 | 0.79 | 0.23 | 0.79 | 1.07 | 0.98 |

Table 5.4: C3 - Trial 2 results

| Model Name | Max Sensitivity | AUC | F1-Score Pos Class | Accuracy | Specificity | PPV | NPV | PLR | NLR |
|---|---|---|---|---|---|---|---|---|---|
| ANN | 0.24 | 0.52 | 0.24 | 0.69 | 0.81 | 0.25 | 0.79 | 1.24 | 0.94 |
| ANN-no-CW | 0.19 | 0.56 | 0.18 | 0.66 | 0.79 | 0.19 | 0.78 | 0.84 | 1.04 |
| VGG16-no-TL | 0.22 | 0.51 | 0.21 | 0.66 | 0.79 | 0.21 | 0.78 | 0.97 | 1.01 |
| VGG16-no-TL-no-CW | 0.17 | 0.50 | 0.17 | 0.67 | 0.81 | 0.18 | 0.78 | 0.83 | 1.04 |
| VGG16-TL | 0.26 | 0.49 | 0.25 | 0.68 | 0.80 | 0.25 | 0.80 | 1.24 | 0.94 |
| VGG16-TL-no-CW | 0.26 | 0.54 | 0.24 | 0.67 | 0.79 | 0.24 | 0.79 | 1.16 | 0.96 |
| ResNet50-no-TL | 0.18 | 0.50 | 0.18 | 0.66 | 0.80 | 0.19 | 0.78 | 0.83 | 1.04 |
| ResNet50-no-TL-no-CW | 0.24 | 0.54 | 0.23 | 0.67 | 0.79 | 0.23 | 0.79 | 1.08 | 0.98 |
| ResNet50-TL | 0.19 | 0.48 | 0.20 | 0.69 | 0.84 | 0.23 | 0.79 | 1.09 | 0.98 |
| ResNet50-TL-no-CW | 0.10 | 0.44 | 0.10 | 0.67 | 0.82 | 0.12 | 0.77 | 0.51 | 1.10 |
| MobileNet-no-TL | 0.33 | 0.57 | 0.30 | 0.69 | 0.79 | 0.29 | 0.81 | 1.50 | 0.86 |
| MobileNet-no-TL-no-CW | 0.26 | 0.55 | 0.25 | 0.68 | 0.80 | 0.25 | 0.80 | 1.24 | 0.94 |
| MobileNet-TL | 0.27 | 0.53 | 0.25 | 0.68 | 0.79 | 0.25 | 0.80 | 1.24 | 0.94 |
| MobileNet-TL-no-CW | 0.26 | 0.53 | 0.24 | 0.67 | 0.79 | 0.24 | 0.79 | 1.16 | 0.96 |
| ResNet18-no-TL | 0.26 | 0.56 | 0.24 | 0.67 | 0.79 | 0.24 | 0.79 | 1.16 | 0.96 |
| ResNet18-no-TL-no-CW | 0.27 | 0.54 | 0.26 | 0.68 | 0.80 | 0.26 | 0.80 | 1.28 | 0.93 |
| ResNet18-TL | 0.17 | 0.54 | 0.17 | 0.67 | 0.81 | 0.18 | 0.78 | 0.80 | 1.05 |
| ResNet18-TL-no-CW | 0.36 | 0.58 | 0.33 | 0.69 | 0.79 | 0.31 | 0.82 | 1.64 | 0.83 |

Table 5.5: C3 - Trial 3 results

| Model Name | Max Sensitivity | AUC | F1-Score Pos Class | Accuracy | Specificity | PPV | NPV | PLR | NLR |
|---|---|---|---|---|---|---|---|---|---|
| ANN | 0.18 | 0.48 | 0.18 | 0.67 | 0.81 | 0.19 | 0.78 | 0.88 | 1.03 |
| ANN-no-CW | 0.31 | 0.54 | 0.29 | 0.69 | 0.79 | 0.28 | 0.80 | 1.42 | 0.89 |
| VGG16-no-TL | 0.31 | 0.54 | 0.29 | 0.68 | 0.79 | 0.28 | 0.80 | 1.40 | 0.89 |
| VGG16-no-TL-no-CW | 0.17 | 0.48 | 0.16 | 0.65 | 0.79 | 0.17 | 0.77 | 0.73 | 1.07 |
| VGG16-TL | 0.22 | 0.53 | 0.22 | 0.69 | 0.82 | 0.24 | 0.79 | 1.12 | 0.97 |
| VGG16-TL-no-CW | 0.21 | 0.53 | 0.20 | 0.66 | 0.79 | 0.20 | 0.78 | 0.91 | 1.02 |
| ResNet50-no-TL | 0.26 | 0.51 | 0.24 | 0.67 | 0.79 | 0.24 | 0.79 | 1.18 | 0.95 |
| ResNet50-no-TL-no-CW | 0.27 | 0.55 | 0.25 | 0.67 | 0.79 | 0.25 | 0.80 | 1.22 | 0.94 |
| ResNet50-TL | 0.21 | 0.48 | 0.20 | 0.68 | 0.81 | 0.22 | 0.79 | 1.01 | 1.00 |
| ResNet50-TL-no-CW | 0.15 | 0.45 | 0.15 | 0.66 | 0.80 | 0.16 | 0.77 | 0.69 | 1.08 |
| MobileNet-no-TL | 0.22 | 0.53 | 0.21 | 0.67 | 0.80 | 0.22 | 0.79 | 1.04 | 0.99 |
| MobileNet-no-TL-no-CW | 0.29 | 0.54 | 0.28 | 0.68 | 0.79 | 0.27 | 0.80 | 1.34 | 0.91 |
| MobileNet-TL | 0.19 | 0.50 | 0.18 | 0.66 | 0.79 | 0.19 | 0.78 | 0.85 | 1.04 |
| MobileNet-TL-no-CW | 0.21 | 0.47 | 0.20 | 0.67 | 0.81 | 0.21 | 0.78 | 1.00 | 1.00 |
| ResNet18-no-TL | 0.29 | 0.51 | 0.28 | 0.68 | 0.79 | 0.27 | 0.80 | 1.34 | 0.91 |
| ResNet18-no-TL-no-CW | 0.24 | 0.51 | 0.24 | 0.69 | 0.82 | 0.26 | 0.80 | 1.26 | 0.94 |
| ResNet18-TL | 0.21 | 0.52 | 0.19 | 0.66 | 0.79 | 0.20 | 0.78 | 0.90 | 1.03 |

Table 5.6: C3 - Trial 3B results

| Model Name | Max Sensitivity | AUC | F1-Score Pos Class | Accuracy | Specificity | PPV | NPV | PLR | NLR |
|---|---|---|---|---|---|---|---|---|---|
| ANN | 0.29 | 0.51 | 0.28 | 0.68 | 0.79 | 0.27 | 0.8 | 1.36 | 0.91 |
| ANN-no-CW | 0.23 | 0.5 | 0.22 | 0.67 | 0.79 | 0.22 | 0.79 | 1.04 | 0.99 |
| VGG16-no-TL | 0.17 | 0.47 | 0.17 | 0.67 | 0.81 | 0.18 | 0.78 | 0.8 | 1.05 |
| VGG16-no-TL-no-CW | 0.13 | 0.44 | 0.12 | 0.65 | 0.79 | 0.13 | 0.77 | 0.56 | 1.12 |
| VGG16-TL | 0.22 | 0.48 | 0.21 | 0.67 | 0.8 | 0.22 | 0.79 | 1.01 | 1 |
| VGG16-TL-no-CW | 0.21 | 0.45 | 0.2 | 0.66 | 0.79 | 0.2 | 0.78 | 0.91 | 1.02 |
| ResNet50-no-TL | 0.31 | 0.52 | 0.28 | 0.68 | 0.79 | 0.27 | 0.8 | 1.38 | 0.9 |
| ResNet50-no-TL-no-CW | 0.27 | 0.52 | 0.26 | 0.69 | 0.81 | 0.27 | 0.8 | 1.33 | 0.92 |
| ResNet50-TL | 0.21 | 0.5 | 0.19 | 0.66 | 0.79 | 0.2 | 0.78 | 0.9 | 1.03 |
| ResNet50-TL-no-CW | 0.24 | 0.54 | 0.23 | 0.67 | 0.79 | 0.23 | 0.79 | 1.08 | 0.98 |
| MobileNet-no-TL | 0.18 | 0.48 | 0.17 | 0.66 | 0.79 | 0.18 | 0.78 | 0.79 | 1.06 |
| MobileNet-no-TL-no-CW | 0.17 | 0.46 | 0.16 | 0.66 | 0.79 | 0.17 | 0.77 | 0.74 | 1.07 |
| MobileNet-TL | 0.22 | 0.45 | 0.21 | 0.66 | 0.79 | 0.21 | 0.78 | 0.97 | 1.01 |
| MobileNet-TL-no-CW | 0.19 | 0.45 | 0.18 | 0.66 | 0.79 | 0.19 | 0.78 | 0.84 | 1.04 |
| ResNet18-no-TL | 0.22 | 0.52 | 0.21 | 0.66 | 0.79 | 0.21 | 0.78 | 0.96 | 1.01 |
| ResNet18-no-TL-no-CW | 0.26 | 0.54 | 0.24 | 0.67 | 0.79 | 0.24 | 0.79 | 1.16 | 0.96 |
| ResNet18-TL | 0.32 | 0.54 | 0.29 | 0.68 | 0.79 | 0.28 | 0.81 | 1.44 | 0.88 |
| ResNet18-TL-no-CW | 0.17 | 0.5 | 0.17 | 0.67 | 0.81 | 0.18 | 0.78 | 0.83 | 1.04 |

Table 5.7: C3 - Trial 4 results

| Model Name | Max Sensitivity | AUC | F1-Score Pos Class | Accuracy | Specificity | PPV | NPV | PLR | NLR |
|---|---|---|---|---|---|---|---|---|---|
| VGG16-TL | 0.26 | 0.52 | 0.25 | 0.69 | 0.24 | 0.81 | 0.8 | 1.29 | 0.93 |
| VGG16-TL-AUG | 0.27 | 0.55 | 0.25 | 0.67 | 0.26 | 0.79 | 0.79 | 1.2 | 0.95 |
| VGG16-TL-AUG-no-CW | 0.32 | 0.52 | 0.31 | 0.7 | 0.31 | 0.81 | 0.81 | 1.59 | 0.86 |
| VGG16-no-TL | 0.15 | 0.52 | 0.16 | 0.67 | 0.14 | 0.82 | 0.78 | 0.77 | 1.05 |
| VGG16-no-TL-AUG | 0.23 | 0.56 | 0.23 | 0.68 | 0.22 | 0.81 | 0.79 | 1.15 | 0.96 |
| VGG16-no-TL-AUG-no-CW | 0.18 | 0.47 | 0.17 | 0.65 | 0.17 | 0.79 | 0.78 | 0.78 | 1.06 |
| ResNet18-TL | 0.26 | 0.52 | 0.25 | 0.68 | 0.24 | 0.8 | 0.79 | 1.22 | 0.95 |
| ResNet18-TL-AUG | 0.23 | 0.51 | 0.22 | 0.67 | 0.22 | 0.8 | 0.79 | 1.07 | 0.98 |
| ResNet18-TL-AUG-no-CW | 0.23 | 0.53 | 0.22 | 0.67 | 0.22 | 0.79 | 0.79 | 1.04 | 0.99 |
| ResNet18-no-TL | 0.22 | 0.47 | 0.21 | 0.67 | 0.21 | 0.79 | 0.78 | 0.99 | 1 |
| ResNet18-no-TL-AUG | 0.36 | 0.58 | 0.33 | 0.69 | 0.35 | 0.79 | 0.82 | 1.64 | 0.83 |
| ResNet18-no-TL-no-CW | 0.31 | 0.54 | 0.29 | 0.69 | 0.29 | 0.79 | 0.8 | 1.42 | 0.89 |

Table 5.8: C3 - Trial 5 results

| Model Name | Max Sensitivity | AUC | F1-Score Pos Class | Accuracy | Specificity | PPV | NPV | PLR | NLR |
|---|---|---|---|---|---|---|---|---|---|
| ResNet18-TL-no-CW-anterior | 0.11 | 0.35 | 0.07 | 0.67 | 0.83 | 0.08 | 0.76 | 0.33 | 1.13 |
| ResNet18-TL-no-CW-tuned | 0.24 | 0.47 | 0.23 | 0.67 | 0.79 | 0.23 | 0.79 | 1.08 | 0.98 |

Table 5.9: C3 - Trial Optimization results

#### 5.5.0.2 LBW (Birth Weight Percentile below 10)

| Best Architecture | Max Sensitivity | AUC | F1-Score Pos Class | Accuracy | Specificity | PPV | NPV | PLR | NLR |
|---|---|---|---|---|---|---|---|---|---|
| ANN | 0.29 | 0.53 | 0.27 | 0.77 | 0.86 | 0.27 | 0.86 | 1.96 | 0.84 |
| VGG16-no-TL | 0.00 | 0.5 | 0.00 | 0.84 | 1.00 | 0.00 | 0.84 | | 1.00 |
| VGG16-TL | 0.15 | 0.5 | 0.14 | 0.73 | 0.85 | 0.14 | 0.84 | 0.90 | 1.02 |
| ResNet50-no-TL | 0.19 | 0.51 | 0.18 | 0.74 | 0.85 | 0.19 | 0.85 | 1.21 | 0.96 |
| ResNet50-TL | 0.21 | 0.56 | 0.19 | 0.74 | 0.85 | 0.19 | 0.85 | 1.28 | 0.95 |
| MobileNet-no-TL | 0.15 | 0.53 | 0.14 | 0.74 | 0.85 | 0.15 | 0.84 | 0.93 | 1.01 |
| MobileNet-TL | 0.21 | 0.59 | 0.19 | 0.74 | 0.85 | 0.19 | 0.85 | 1.28 | 0.95 |
| ResNet18-no-TL | 0.19 | 0.51 | 0.19 | 0.75 | 0.86 | 0.20 | 0.85 | 1.32 | 0.95 |
| ResNet18-TL | 0.14 | 0.47 | 0.13 | 0.73 | 0.85 | 0.13 | 0.84 | 0.81 | 1.04 |

Table 5.10: LBW - Trial 1 results

| Best Architecture | Max Sensitivity | AUC | F1-Score Pos Class | Accuracy | Specificity | PPV | NPV | PLR | NLR |
|---|---|---|---|---|---|---|---|---|---|
| ANN | 0.25 | 0.55 | 0.25 | 0.78 | 0.88 | 0.27 | 0.86 | 1.96 | 0.87 |
| VGG16-no-TL | 0.24 | 0.53 | 0.22 | 0.74 | 0.84 | 0.21 | 0.85 | 1.41 | 0.92 |
| VGG16-TL | 0.17 | 0.54 | 0.15 | 0.74 | 0.85 | 0.16 | 0.84 | 0.99 | 1.00 |
| ResNet50-no-TL | 0.18 | 0.51 | 0.18 | 0.76 | 0.88 | 0.20 | 0.85 | 1.35 | 0.95 |
| ResNet50-TL | 0.22 | 0.56 | 0.20 | 0.74 | 0.84 | 0.20 | 0.85 | 1.32 | 0.94 |
| MobileNet-no-TL | 0.14 | 0.56 | 0.13 | 0.74 | 0.86 | 0.15 | 0.84 | 0.90 | 1.02 |
| MobileNet-TL | 0.25 | 0.59 | 0.24 | 0.76 | 0.86 | 0.24 | 0.86 | 1.67 | 0.89 |
| ResNet18-no-TL | 0.21 | 0.50 | 0.19 | 0.74 | 0.84 | 0.19 | 0.85 | 1.23 | 0.96 |
| ResNet18-TL | 0.18 | 0.51 | 0.17 | 0.74 | 0.84 | 0.17 | 0.84 | 1.06 | 0.99 |

Table 5.11: LBW - Trial 2 results

| Model Name | Max Sensitivity | AUC | F1-Score Pos Class | Accuracy | Specificity | PPV | NPV | PLR | NLR |
|---|---|---|---|---|---|---|---|---|---|
| ANN | 0.28 | 0.55 | 0.25 | 0.75 | 0.85 | 0.24 | 0.86 | 1.7 | 0.87 |
| ANN-no-CW | 0.28 | 0.57 | 0.27 | 0.77 | 0.87 | 0.28 | 0.86 | 2.05 | 0.84 |
| VGG16-no-TL | 0.14 | 0.5 | 0.14 | 0.75 | 0.87 | 0.15 | 0.84 | 0.95 | 1.01 |
| VGG16-no-TL-no-CW | 0.15 | 0.5 | 0.15 | 0.75 | 0.86 | 0.16 | 0.84 | 1 | 1 |
| VGG16-TL | 0.12 | 0.45 | 0.12 | 0.73 | 0.85 | 0.12 | 0.83 | 0.72 | 1.05 |
| VGG16-TL-no-CW | 0.17 | 0.5 | 0.15 | 0.73 | 0.84 | 0.15 | 0.84 | 0.97 | 1.01 |
| ResNet50-no-TL | 0.14 | 0.51 | 0.13 | 0.74 | 0.86 | 0.15 | 0.84 | 0.9 | 1.02 |
| ResNet50-no-TL-no-CW | 0.21 | 0.53 | 0.2 | 0.75 | 0.85 | 0.2 | 0.85 | 1.3 | 0.95 |
| ResNet50-TL | 0.17 | 0.51 | 0.16 | 0.75 | 0.87 | 0.18 | 0.84 | 1.14 | 0.98 |
| ResNet50-TL-no-CW | 0.14 | 0.48 | 0.13 | 0.74 | 0.86 | 0.14 | 0.84 | 0.87 | 1.02 |
| MobileNet-no-TL | 0.17 | 0.52 | 0.16 | 0.74 | 0.85 | 0.16 | 0.84 | 1 | 1 |
| MobileNet-no-TL-no-CW | 0.14 | 0.51 | 0.14 | 0.75 | 0.87 | 0.16 | 0.84 | 0.97 | 1 |
| MobileNet-TL | 0.15 | 0.49 | 0.14 | 0.74 | 0.85 | 0.15 | 0.84 | 0.91 | 1.02 |
| MobileNet-TL-no-CW | 0.14 | 0.46 | 0.13 | 0.73 | 0.85 | 0.13 | 0.84 | 0.82 | 1.03 |
| ResNet18-no-TL | 0.17 | 0.52 | 0.16 | 0.74 | 0.86 | 0.17 | 0.84 | 1.06 | 0.99 |
| ResNet18-no-TL-no-CW | 0.15 | 0.56 | 0.14 | 0.73 | 0.84 | 0.14 | 0.84 | 0.88 | 1.02 |
| ResNet18-TL | 0.18 | 0.5 | 0.17 | 0.75 | 0.86 | 0.18 | 0.84 | 1.15 | 0.97 |
| ResNet18-TL-no-CW | 0.17 | 0.48 | 0.17 | 0.76 | 0.87 | 0.18 | 0.84 | 1.19 | 0.97 |

Table 5.12: LBW - Trial 3 results

| Model Name | Max Sensitivity | AUC | F1-Score Pos Class | Accuracy | Specificity | PPV | NPV | PLR | NLR |
|---|---|---|---|---|---|---|---|---|---|
| ANN | 0.19 | 0.51 | 0.18 | 0.74 | 0.84 | 0.18 | 0.84 | 1.15 | 0.97 |
| ANN-no-CW | 0.22 | 0.52 | 0.20 | 0.74 | 0.84 | 0.20 | 0.85 | 1.32 | 0.94 |
| VGG16-no-TL | 0 | 0.5 | 0 | 0.84 | 1 | 0 | 0.84 | - | 1 |
| VGG16-no-TL-no-CW | 0.25 | 0.55 | 0.23 | 0.75 | 0.85 | 0.23 | 0.85 | 1.55 | 0.9 |
| VGG16-TL | 0.17 | 0.5 | 0.17 | 0.77 | 0.89 | 0.20 | 0.85 | 1.35 | 0.96 |
| VGG16-TL-no-CW | 0.19 | 0.57 | 0.18 | 0.74 | 0.84 | 0.18 | 0.84 | 1.15 | 0.97 |
| ResNet50-no-TL | 0.1 | 0.48 | 0.1 | 0.75 | 0.88 | 0.12 | 0.84 | 0.69 | 1.04 |
| ResNet50-no-TL-no-CW | 0.19 | 0.52 | 0.19 | 0.75 | 0.86 | 0.19 | 0.85 | 1.27 | 0.95 |
| ResNet50-TL | 0.24 | 0.52 | 0.22 | 0.75 | 0.85 | 0.22 | 0.85 | 1.51 | 0.91 |
| ResNet50-TL-no-CW | 0.14 | 0.51 | 0.13 | 0.74 | 0.86 | 0.14 | 0.84 | 0.87 | 1.02 |
| MobileNet-no-TL | 0.17 | 0.51 | 0.15 | 0.73 | 0.84 | 0.15 | 0.84 | 0.97 | 1.01 |
| MobileNet-no-TL-no-CW | 0.28 | 0.62 | 0.25 | 0.75 | 0.85 | 0.24 | 0.86 | 1.7 | 0.87 |
| MobileNet-TL | 0.08 | 0.45 | 0.08 | 0.73 | 0.86 | 0.08 | 0.83 | 0.48 | 1.09 |
| MobileNet-TL-no-CW | 0.08 | 0.52 | 0.08 | 0.74 | 0.87 | 0.09 | 0.83 | 0.53 | 1.07 |
| ResNet18-no-TL | 0.11 | 0.47 | 0.1 | 0.73 | 0.85 | 0.11 | 0.83 | 0.66 | 1.06 |
| ResNet18-no-TL-no-CW | 0.22 | 0.54 | 0.21 | 0.75 | 0.85 | 0.21 | 0.85 | 1.37 | 0.93 |
| ResNet18-TL | 0.15 | 0.5 | 0.15 | 0.75 | 0.87 | 0.17 | 0.84 | 1.06 | 0.99 |
| ResNet18-TL-no-CW | 0.21 | 0.51 | 0.19 | 0.74 | 0.85 | 0.19 | 0.85 | 1.28 | 0.95 |

Table 5.13: LBW - Trial 3B results

| Model Name | Max Sensitivity | AUC | F1-Score Pos Class | Accuracy | Specificity | PPV | NPV | PLR | NLR |
|---|---|---|---|---|---|---|---|---|---|
| ANN | 0.14 | 0.48 | 0.13 | 0.74 | 0.86 | 0.14 | 0.84 | 0.87 | 1.02 |
| ANN-no-CW | 0.17 | 0.51 | 0.16 | 0.74 | 0.85 | 0.16 | 0.84 | 1.04 | 0.99 |
| VGG16-no-TL | 0.11 | 0.49 | 0.1 | 0.74 | 0.86 | 0.11 | 0.83 | 0.67 | 1.06 |
| VGG16-no-TL-no-CW | 0.12 | 0.5 | 0.12 | 0.74 | 0.86 | 0.13 | 0.84 | 0.8 | 1.03 |
| VGG16-TL | 0.17 | 0.52 | 0.16 | 0.75 | 0.86 | 0.17 | 0.84 | 1.1 | 0.98 |
| VGG16-TL-no-CW | 0.17 | 0.49 | 0.16 | 0.74 | 0.85 | 0.16 | 0.84 | 1 | 1 |
| ResNet50-no-TL | 0.11 | 0.44 | 0.1 | 0.73 | 0.85 | 0.11 | 0.83 | 0.63 | 1.07 |
| ResNet50-no-TL-no-CW | 0.18 | 0.51 | 0.18 | 0.76 | 0.87 | 0.2 | 0.85 | 1.32 | 0.95 |
| ResNet50-TL | 0.11 | 0.5 | 0.11 | 0.75 | 0.87 | 0.13 | 0.84 | 0.77 | 1.03 |
| ResNet50-TL-no-CW | 0.1 | 0.45 | 0.09 | 0.73 | 0.85 | 0.1 | 0.83 | 0.56 | 1.08 |
| MobileNet-no-TL | 0.15 | 0.49 | 0.14 | 0.74 | 0.85 | 0.15 | 0.84 | 0.91 | 1.02 |
| MobileNet-no-TL-no-CW | 0.17 | 0.5 | 0.15 | 0.74 | 0.85 | 0.16 | 0.84 | 0.99 | 1 |
| MobileNet-TL | 0.18 | 0.52 | 0.17 | 0.75 | 0.86 | 0.18 | 0.84 | 1.15 | 0.97 |
| MobileNet-TL-no-CW | 0.15 | 0.48 | 0.16 | 0.76 | 0.88 | 0.18 | 0.84 | 1.15 | 0.98 |
| ResNet18-no-TL | 0.21 | 0.52 | 0.19 | 0.74 | 0.84 | 0.19 | 0.85 | 1.23 | 0.96 |
| ResNet18-no-TL-no-CW | 0.19 | 0.52 | 0.18 | 0.74 | 0.85 | 0.18 | 0.85 | 1.17 | 0.97 |
| ResNet18-TL | 0.22 | 0.5 | 0.21 | 0.76 | 0.86 | 0.22 | 0.85 | 1.5 | 0.92 |
| ResNet18-TL-no-CW | 0.17 | 0.48 | 0.16 | 0.75 | 0.86 | 0.17 | 0.84 | 1.1 | 0.98 |

Table 5.14: LBW - Trial 4 results

| Model Name | Max Sensitivity | AUC | F1-Score Pos Class | Accuracy | Specificity | PPV | NPV | PLR | NLR |
|---|---|---|---|---|---|---|---|---|---|
| VGG16-TL | 0.17 | 0.51 | 0.16 | 0.74 | 0.15 | 0.85 | 0.84 | 1.02 | 1 |
| VGG16-TL-AUG | 0.21 | 0.51 | 0.19 | 0.74 | 0.19 | 0.85 | 0.85 | 1.28 | 0.95 |
| VGG16-TL-AUG-no-CW | 0.26 | 0.52 | 0.27 | 0.78 | 0.25 | 0.88 | 0.86 | 2.16 | 0.85 |
| VGG16-no-TL | 0.14 | 0.55 | 0.13 | 0.73 | 0.12 | 0.84 | 0.84 | 0.79 | 1.04 |
| VGG16-no-TL-AUG | 0.11 | 0.54 | 0.1 | 0.72 | 0.1 | 0.84 | 0.83 | 0.62 | 1.07 |
| VGG16-no-TL-AUG-no-CW | 0.18 | 0.51 | 0.17 | 0.74 | 0.17 | 0.84 | 0.84 | 1.06 | 0.99 |
| ResNet18-TL | 0.08 | 0.48 | 0.07 | 0.73 | 0.07 | 0.85 | 0.83 | 0.46 | 1.09 |
| ResNet18-TL-AUG | 0.24 | 0.55 | 0.22 | 0.75 | 0.22 | 0.86 | 0.85 | 1.54 | 0.91 |
| ResNet18-TL-AUG-no-CW | 0.15 | 0.5 | 0.15 | 0.75 | 0.14 | 0.87 | 0.84 | 1.06 | 0.99 |
| ResNet18-no-TL | 0.15 | 0.51 | 0.14 | 0.74 | 0.14 | 0.85 | 0.84 | 0.93 | 1.01 |
| ResNet18-no-TL-AUG | 0.17 | 0.52 | 0.15 | 0.74 | 0.15 | 0.85 | 0.84 | 0.99 | 1 |
| ResNet18-no-TL-no-CW | 0.14 | 0.51 | 0.16 | 0.79 | 0.12 | 0.91 | 0.85 | 1.4 | 0.96 |

Table 5.15: LBW - Trial 5 results

| Model Name | Max Sensitivity | AUC | F1-Score Pos Class | Accuracy | Specificity | PPV | NPV | PLR | NLR |
|---|---|---|---|---|---|---|---|---|---|
| ANN-anterior | 0.28 | 0.58 | 0.27 | 0.75 | 0.88 | 0.33 | 0.81 | 1.92 | 0.88 |
| ANN-tuned | 0.28 | 0.54 | 0.25 | 0.75 | 0.85 | 0.24 | 0.86 | 1.7 | 0.87 |

Table 5.16: LBW - Trial Optimization results

#### 5.5.0.3 PRE (Preeclampsia)

| Best Architecture | Max Sensitivity | AUC | F1-Score Pos Class | Accuracy | Specificity | PPV | NPV | PLR | NLR |
|---|---|---|---|---|---|---|---|---|---|
| ANN | 0.17 | 0.59 | 0.09 | 0.88 | 0.94 | 0.10 | 0.93 | 1.43 | 0.97 |
| VGG16-no-TL | 0.00 | 0.27 | 0.00 | 0.93 | 1.00 | 0.00 | 0.93 | | 1.00 |
| VGG16-TL | 0.08 | 0.47 | 0.00 | 0.91 | 0.98 | 0.00 | 0.93 | 0.00 | 1.02 |
| ResNet50-no-TL | 0.08 | 0.42 | 0.00 | 0.91 | 0.98 | 0.00 | 0.93 | 0.00 | 1.02 |
| ResNet50-TL | 0.00 | 0.40 | 0.00 | 0.93 | 1.00 | 0.00 | 0.93 | | 1.00 |
| MobileNet-no-TL | 0.33 | 0.71 | 0.25 | 0.89 | 0.94 | 0.25 | 0.94 | 4.28 | 0.80 |
| MobileNet-TL | 0.00 | 0.29 | 0.00 | 0.93 | 1.00 | 0.00 | 0.93 | | 1.00 |
| ResNet18-no-TL | 0.08 | 0.49 | 0.00 | 0.88 | 0.95 | 0.00 | 0.92 | 0.00 | 1.05 |
| ResNet18-TL | 0.00 | 0.51 | 0.00 | 0.93 | 1.00 | 0.00 | 0.93 | | 1.00 |

Table 5.17: PRE - Trial 1 results

| Best Architecture | Max Sensitivity | AUC | F1-Score Pos Class | Accuracy | Specificity | PPV | NPV | PLR | NLR |
|---|---|---|---|---|---|---|---|---|---|
| ANN | 0.00 | 0.46 | 0.00 | 0.93 | 1.00 | 0.00 | 0.93 | | 1.00 |
| VGG16-no-TL | 0.08 | 0.33 | 0.00 | 0.90 | 0.97 | 0.00 | 0.93 | 0.00 | 1.03 |
| VGG16-TL | 0.08 | 0.42 | 0.00 | 0.86 | 0.93 | 0.00 | 0.92 | 0.00 | 1.08 |
| ResNet50-no-TL | 0.00 | 0.47 | 0.00 | 0.93 | 1.00 | 0.00 | 0.93 | | 1.00 |
| ResNet50-TL | 0.17 | 0.42 | 0.10 | 0.89 | 0.95 | 0.12 | 0.93 | 1.83 | 0.96 |
| MobileNet-no-TL | 0.08 | 0.56 | 0.00 | 0.87 | 0.94 | 0.00 | 0.92 | 0.00 | 1.07 |
| MobileNet-TL | 0.00 | 0.23 | 0.00 | 0.93 | 1.00 | 0.00 | 0.93 | | 1.00 |
| ResNet18-no-TL | 0.08 | 0.46 | 0.00 | 0.86 | 0.93 | 0.00 | 0.92 | 0.00 | 1.08 |
| ResNet18-TL | 0.08 | 0.58 | 0.00 | 0.87 | 0.94 | 0.00 | 0.92 | 0.00 | 1.06 |

Table 5.18: PRE - Trial 2 results

| Model Name | Max Sensitivity | AUC | F1-Score Pos Class | Accuracy | Specificity | PPV | NPV | PLR | NLR |
|---|---|---|---|---|---|---|---|---|---|
| ANN | 0.00 | 0.15 | 0.00 | 0.93 | 1.00 | 0.00 | 0.93 | - | 1 |
| ANN-no-CW | 0.08 | 0.61 | 0.00 | 0.87 | 0.94 | 0.00 | 0.92 | 0.00 | 1.06 |
| VGG16-no-TL | 0.25 | 0.55 | 0.16 | 0.87 | 0.93 | 0.15 | 0.93 | 2.33 | 0.90 |
| VGG16-no-TL-no-CW | 0.17 | 0.58 | 0.10 | 0.89 | 0.95 | 0.11 | 0.93 | 1.60 | 0.97 |
| VGG16-TL | 0.08 | 0.49 | 0.00 | 0.88 | 0.95 | 0.00 | 0.92 | 0.00 | 1.05 |
| VGG16-TL-no-CW | 0.08 | 0.36 | 0.00 | 0.93 | 1.00 | 0.00 | 0.93 | - | 1 |
| ResNet50-no-TL | 0.00 | 0.31 | 0.00 | 0.93 | 1.00 | 0.00 | 0.93 | - | 1 |
| ResNet50-no-TL-no-CW | 0.00 | 0.34 | 0.00 | 0.93 | 1.00 | 0.00 | 0.93 | - | 1 |
| ResNet50-TL | 0.00 | 0.31 | 0.00 | 0.93 | 1.00 | 0.00 | 0.93 | - | 1 |
| ResNet50-TL-no-CW | 0.00 | 0.30 | 0.00 | 0.93 | 1.00 | 0.00 | 0.93 | - | 1 |
| MobileNet-no-TL | 0.00 | 0.27 | 0.00 | 0.93 | 1.00 | 0.00 | 0.93 | - | 1 |
| MobileNet-no-TL-no-CW | 0.00 | 0.31 | 0.00 | 0.93 | 1.00 | 0.00 | 0.93 | - | 1 |
| MobileNet-TL | 0.00 | 0.37 | 0.00 | 0.93 | 1.00 | 0.00 | 0.93 | - | 1 |
| MobileNet-TL-no-CW | 0.00 | 0.44 | 0.00 | 0.93 | 1.00 | 0.00 | 0.93 | - | 1 |
| ResNet18-no-TL | 0.00 | 0.26 | 0.00 | 0.93 | 1.00 | 0.00 | 0.93 | - | 1 |
| ResNet18-no-TL-no-CW | 0.00 | 0.30 | 0.00 | 0.93 | 1.00 | 0.00 | 0.93 | - | 1 |
| ResNet18-TL | 0.00 | 0.46 | 0.00 | 0.93 | 1.00 | 0.00 | 0.93 | - | 1 |
| ResNet18-TL-no-CW | 0.00 | 0.26 | 0.00 | 0.93 | 1.00 | 0.00 | 0.93 | - | 1 |

Table 5.19: PRE - Trial 3 results

| Model Name | Max Sensitivity | AUC | F1-Score Pos Class | Accuracy | Specificity | PPV | NPV | PLR | NLR |
|---|---|---|---|---|---|---|---|---|---|
| ANN | 0.00 | 0.38 | 0.00 | 0.93 | 1.00 | 0.00 | 0.93 | | 1.00 |
| ANN-no-CW | 0.17 | 0.60 | 0.10 | 0.89 | 0.95 | 0.12 | 0.93 | 1.83 | 0.96 |
| VGG16-no-TL | 0.00 | 0.26 | 0.00 | 0.93 | 1.00 | 0.00 | 0.93 | | 1.00 |
| VGG16-no-TL-no-CW | 0.00 | 0.23 | 0.00 | 0.93 | 1.00 | 0.00 | 0.93 | | 1.00 |
| VGG16-TL | 0.17 | 0.54 | 0.09 | 0.87 | 0.94 | 0.09 | 0.93 | 1.28 | 0.98 |
| VGG16-TL-no-CW | 0.08 | 0.39 | 0.00 | 0.93 | 1.00 | 0.00 | 0.93 | | 1.00 |
| ResNet50-no-TL | 0.08 | 0.44 | 0.00 | 0.90 | 0.97 | 0.00 | 0.93 | 0.00 | 1.03 |
| ResNet50-no-TL-no-CW | 0.08 | 0.39 | 0.00 | 0.91 | 0.98 | 0.00 | 0.93 | 0.00 | 1.02 |
| ResNet50-TL | 0.00 | 0.39 | 0.00 | 0.93 | 1.00 | 0.00 | 0.93 | | 1.00 |
| ResNet50-TL-no-CW | 0.00 | 0.32 | 0.00 | 0.93 | 1.00 | 0.00 | 0.93 | | 1.00 |
| MobileNet-no-TL | 0.08 | 0.57 | 0.00 | 0.90 | 0.97 | 0.00 | 0.93 | 0.00 | 1.03 |
| MobileNet-no-TL-no-CW | 0.25 | 0.52 | 0.16 | 0.87 | 0.93 | 0.15 | 0.93 | 2.33 | 0.90 |
| MobileNet-TL | 0.00 | 0.29 | 0.00 | 0.93 | 1.00 | 0.00 | 0.93 | | 1.00 |
| MobileNet-TL-no-CW | 0.00 | 0.45 | 0.00 | 0.93 | 1.00 | 0.00 | 0.93 | | 1.00 |
| ResNet18-no-TL | 0.00 | 0.32 | 0.00 | 0.93 | 1.00 | 0.00 | 0.93 | | 1.00 |
| ResNet18-no-TL-no-CW | 0.08 | 0.44 | 0.00 | 0.90 | 0.97 | 0.00 | 0.93 | 0.00 | 1.03 |
| ResNet18-TL | 0.17 | 0.56 | 0.10 | 0.89 | 0.95 | 0.12 | 0.93 | 1.83 | 0.96 |
| ResNet18-TL-no-CW | 0.42 | 0.73 | 0.31 | 0.89 | 0.94 | 0.29 | 0.95 | 5.13 | 0.71 |

Table 5.20: PRE - Trial 3B results

| Model Name | Max Sensitivity | AUC | F1-Score Pos Class | Accuracy | Specificity | PPV | NPV | PLR | NLR |
|---|---|---|---|---|---|---|---|---|---|
| ANN | 0.08 | 0.42 | 0 | 0.92 | 0.99 | 0 | 0.93 | 0 | 1.01 |
| ANN-no-CW | 0.08 | 0.51 | 0 | 0.90 | 0.97 | 0 | 0.93 | 0 | 1.03 |
| VGG16-no-TL | 0.17 | 0.51 | 0.08 | 0.87 | 0.93 | 0.08 | 0.93 | 1.17 | 0.99 |
| VGG16-no-TL-no-CW | 0 | 0.28 | 0 | 0.93 | 1 | 0 | 0.93 | | 1 |
| VGG16-TL | 0.08 | 0.27 | 0 | 0.87 | 0.94 | 0 | 0.92 | 0 | 1.07 |
| VGG16-TL-no-CW | 0.08 | 0.38 | 0 | 0.93 | 1 | 0 | 0.93 | | 1 |
| ResNet50-no-TL | 0 | 0.36 | 0 | 0.93 | 1 | 0 | 0.93 | | 1 |
| ResNet50-no-TL-no-CW | 0 | 0.4 | 0 | 0.93 | 1 | 0 | 0.93 | | 1 |
| ResNet50-TL | 0.17 | 0.35 | 0.11 | 0.90 | 0.96 | 0.14 | 0.93 | 2.14 | 0.95 |
| ResNet50-TL-no-CW | 0 | 0.37 | 0 | 0.93 | 1 | 0 | 0.93 | | 1 |
| MobileNet-no-TL | 0.25 | 0.43 | 0.17 | 0.88 | 0.94 | 0.17 | 0.94 | 2.57 | 0.89 |
| MobileNet-no-TL-no-CW | 0.17 | 0.37 | 0.12 | 0.91 | 0.97 | 0.2 | 0.93 | 3.21 | 0.94 |
| MobileNet-TL | 0 | 0.43 | 0 | 0.93 | 1 | 0 | 0.93 | | 1 |
| MobileNet-TL-no-CW | 0.08 | 0.57 | 0 | 0.90 | 0.97 | 0 | 0.93 | 0 | 1.03 |
| ResNet18-no-TL | 0 | 0.42 | 0 | 0.93 | 1 | 0 | 0.93 | | 1 |
| ResNet18-no-TL-no-CW | 0 | 0.28 | 0 | 0.93 | 1 | 0 | 0.93 | | 1 |
| ResNet18-TL | 0 | 0.3 | 0 | 0.93 | 1 | 0 | 0.93 | | 1 |
| ResNet18-TL-no-CW | 0 | 0.39 | 0 | 0.93 | 1 | 0 | 0.93 | | 1 |

Table 5.21: PRE - Trial 4 results

| Model Name | Max Sensitivity | AUC | F1-Score Pos Class | Accuracy | Specificity | PPV | NPV | PLR | NLR |
|---|---|---|---|---|---|---|---|---|---|
| VGG16-TL | 0.08 | 0.43 | 0 | 0.93 | 1 | 0 | 0.93 | | 1 |
| VGG16-TL-AUG | 0.17 | 0.52 | 0.09 | 0.88 | 0.94 | 0.1 | 0.93 | 1.43 | 0.97 |
| VGG16-TL-AUG-no-CW | 0.08 | 0.4 | 0 | 0.87 | 0.94 | 0 | 0.92 | 0 | 1.06 |
| VGG16-no-TL | 0.25 | 0.78 | 0.16 | 0.87 | 0.93 | 0.15 | 0.93 | 2.33 | 0.9 |
| VGG16-no-TL-AUG | 0 | 0.67 | 0 | 0.93 | 1 | 0 | 0.93 | | 1 |
| VGG16-no-TL-AUG-no-CW | 0.08 | 0.41 | 0 | 0.93 | 1 | 0 | 0.93 | | 1 |
| ResNet18-TL | 0.08 | 0.57 | 0 | 0.89 | 0.95 | 0 | 0.92 | 0 | 1.05 |
| ResNet18-TL-AUG | 0 | 0.47 | 0 | 0.93 | 1 | 0 | 0.93 | | 1 |
| ResNet18-TL-AUG-no-CW | 0 | 0.38 | 0 | 0.93 | 1 | 0 | 0.93 | | 1 |
| ResNet18-no-TL | 0 | 0.53 | 0 | 0.93 | 1 | 0 | 0.93 | | 1 |
| ResNet18-no-TL-AUG | 0.25 | 0.63 | 0.17 | 0.89 | 0.94 | 0.18 | 0.94 | 2.85 | 0.89 |
| ResNet18-no-TL-no-CW | 0 | 0.4 | 0 | 0.93 | 1 | 0 | 0.93 | | 1 |

Table 5.22: PRE - Trial 5 results

| Model Name | Max Sensitivity | AUC | F1-Score Pos Class | Accuracy | Specificity | PPV | NPV | PLR | NLR |
|---|---|---|---|---|---|---|---|---|---|
| ResNet18-TL-no-CW-anterior | 0 | 0.53 | 0 | 0.93 | 1 | 0 | 0.93 | | 1 |
| ResNet18-TL-no-CW-tuned | 0 | 0.4 | 0 | 0.93 | 1 | 0 | 0.93 | | 1 |

Table 5.23: PRE - Trial Optimization results

#### 5.5.0.4 CIR (Fetal Growth Restriction)

| Best Architecture | Max Sensitivity | AUC | F1-Score Pos Class | Accuracy | Specificity | PPV | NPV | PLR | NLR |
|---|---|---|---|---|---|---|---|---|---|
| ANN | 0.17 | 0.58 | 0.13 | 0.92 | 0.99 | 0.33 | 0.93 | 6.42 | 0.93 |
| VGG16-no-TL | 0.00 | 0.52 | 0.00 | 0.93 | 1.00 | 0.00 | 0.93 | | 1.00 |
| VGG16-TL | 0.33 | 0.56 | 0.25 | 0.89 | 0.94 | 0.25 | 0.94 | 4.28 | 0.80 |
| ResNet50-no-TL | 0.08 | 0.57 | 0.00 | 0.87 | 0.94 | 0.00 | 0.92 | 0.00 | 1.06 |
| ResNet50-TL | 0.08 | 0.38 | 0.00 | 0.88 | 0.95 | 0.00 | 0.92 | 0.00 | 1.05 |
| MobileNet-no-TL | 0.08 | 0.63 | 0.00 | 0.87 | 0.94 | 0.00 | 0.92 | 0.00 | 1.07 |
| MobileNet-TL | 0.00 | 0.49 | 0.00 | 0.93 | 1.00 | 0.00 | 0.93 | | 1.00 |
| ResNet18-no-TL | 0.08 | 0.56 | 0.00 | 0.89 | 0.96 | 0.00 | 0.92 | 0.00 | 1.04 |
| ResNet18-TL | 0.00 | 0.41 | 0.00 | 0.93 | 1.00 | 0.00 | 0.93 | | 1.00 |

Table 5.24: CIR - Trial 1 results

| Best Architecture | Max Sensitivity | AUC | F1-Score Pos Class | Accuracy | Specificity | PPV | NPV | PLR | NLR |
|---|---|---|---|---|---|---|---|---|---|
| ANN | 0.33 | 0.55 | 0.29 | 0.91 | 0.96 | 0.33 | 0.94 | 6.42 | 0.78 |
| VGG16-no-TL | 0.00 | 0.47 | 0.00 | 0.93 | 1.00 | 0.00 | 0.93 | | 1.00 |
| VGG16-TL | 0.17 | 0.55 | 0.10 | 0.89 | 0.95 | 0.12 | 0.93 | 1.83 | 0.96 |
| ResNet50-no-TL | 0.00 | 0.38 | 0.00 | 0.93 | 1.00 | 0.00 | 0.93 | | 1.00 |
| ResNet50-TL | 0.08 | 0.36 | 0.00 | 0.89 | 0.95 | 0.00 | 0.92 | 0.00 | 1.05 |
| MobileNet-no-TL | 0.25 | 0.45 | 0.16 | 0.87 | 0.93 | 0.15 | 0.93 | 2.33 | 0.90 |
| MobileNet-TL | 0.00 | 0.46 | 0.00 | 0.93 | 1.00 | 0.00 | 0.93 | | 1.00 |
| ResNet18-no-TL | 0.00 | 0.57 | 0.00 | 0.93 | 1.00 | 0.00 | 0.93 | | 1.00 |
| ResNet18-TL | 0.00 | 0.46 | 0.00 | 0.93 | 1.00 | 0.00 | 0.93 | | 1.00 |

Table 5.25: CIR - Trial 2 results

| Model Name | Max Sensitivity | AUC | F1-Score Pos Class | Accuracy | Specificity | PPV | NPV | PLR | NLR |
|---|---|---|---|---|---|---|---|---|---|
| ANN | 0.08 | 0.36 | 0 | 0.9 | 0.97 | 0 | 0.93 | 0 | 1.03 |
| ANN-no-CW | 0 | 0.4 | 0 | 0.93 | 1 | 0 | 0.93 | - | 1 |
| VGG16-no-TL | 0.08 | 0.44 | 0 | 0.9 | 0.97 | 0 | 0.93 | 0 | 1.03 |
| VGG16-no-TL-no-CW | 0.08 | 0.42 | 0 | 0.9 | 0.97 | 0 | 0.93 | 0 | 1.03 |
| VGG16-TL | 0 | 0.43 | 0 | 0.93 | 1 | 0 | 0.93 | - | 1 |
| VGG16-TL-no-CW | 0.08 | 0.55 | 0 | 0.93 | 1 | 0 | 0.93 | - | 1 |
| ResNet50-no-TL | 0.17 | 0.52 | 0.11 | 0.9 | 0.96 | 0.14 | 0.93 | 2.14 | 0.95 |
| ResNet50-no-TL-no-CW | 0 | 0.5 | 0 | 0.93 | 1 | 0 | 0.93 | - | 1 |
| ResNet50-TL | 0.08 | 0.56 | 0 | 0.91 | 0.98 | 0 | 0.93 | 0 | 1.02 |
| ResNet50-TL-no-CW | 0.17 | 0.57 | 0.12 | 0.92 | 0.98 | 0.25 | 0.93 | 4.28 | 0.93 |
| MobileNet-no-TL | 0.08 | 0.53 | 0 | 0.86 | 0.93 | 0 | 0.92 | 0 | 1.08 |
| MobileNet-no-TL-no-CW | 0 | 0.61 | 0 | 0.93 | 1 | 0 | 0.93 | - | 1 |
| MobileNet-TL | 0 | 0.51 | 0 | 0.93 | 1 | 0 | 0.93 | - | 1 |
| MobileNet-TL-no-CW | 0 | 0.39 | 0 | 0.93 | 1 | 0 | 0.93 | - | 1 |
| ResNet18-no-TL | 0.17 | 0.42 | 0.12 | 0.91 | 0.97 | 0.2 | 0.93 | 3.21 | 0.94 |
| ResNet18-no-TL-no-CW | 0.25 | 0.71 | 0.27 | 0.93 | 0.99 | 0.67 | 0.94 | 25.67 | 0.84 |
| ResNet18-TL | 0.25 | 0.57 | 0.16 | 0.87 | 0.93 | 0.15 | 0.93 | 2.33 | 0.9 |
| ResNet18-TL-no-CW | 0.25 | 0.58 | 0.17 | 0.88 | 0.94 | 0.17 | 0.94 | 2.57 | 0.89 |

Table 5.26: CIR - Trial 3 results

| Model Name | Max Sensitivity | AUC | F1-Score Pos Class | Accuracy | Specificity | PPV | NPV | PLR | NLR |
|---|---|---|---|---|---|---|---|---|---|
| ANN | 0.25 | 0.6 | 0.17 | 0.88 | 0.94 | 0.17 | 0.94 | 2.57 | 0.89 |
| ANN-no-CW | 0.17 | 0.55 | 0.13 | 0.92 | 0.99 | 0.33 | 0.93 | 6.42 | 0.93 |
| VGG16-no-TL | 0 | 0.42 | 0 | 0.93 | 1 | 0 | 0.93 |  | 1 |
| VGG16-no-TL-no-CW | 0 | 0.35 | 0 | 0.93 | 1 | 0 | 0.93 |  | 1 |
| VGG16-TL | 0 | 0.32 | 0 | 0.93 | 1 | 0 | 0.93 |  | 1 |
| VGG16-TL-no-CW | 0.25 | 0.55 | 0.16 | 0.87 | 0.93 | 0.15 | 0.93 | 2.33 | 0.90 |
| ResNet50-no-TL | 0 | 0.39 | 0 | 0.93 | 1 | 0 | 0.93 |  | 1 |
| ResNet50-no-TL-no-CW | 0.08 | 0.47 | 0 | 0.87 | 0.94 | 0 | 0.92 | 0 | 1.06 |
| ResNet50-TL | 0.17 | 0.56 | 0.12 | 0.91 | 0.97 | 0.2 | 0.93 | 3.21 | 0.94 |
| ResNet50-TL-no-CW | 0.08 | 0.52 | 0 | 0.92 | 0.99 | 0 | 0.93 | 0 | 1.01 |
| MobileNet-no-TL | 0 | 0.45 | 0 | 0.93 | 1 | 0 | 0.93 |  | 1 |
| MobileNet-no-TL-no-CW | 0 | 0.48 | 0 | 0.93 | 1 | 0 | 0.93 |  | 1 |
| MobileNet-TL | 0.08 | 0.52 | 0 | 0.90 | 0.97 | 0 | 0.93 | 0 | 1.03 |
| MobileNet-TL-no-CW | 0 | 0.39 | 0 | 0.93 | 1 | 0 | 0.93 |  | 1 |
| ResNet18-no-TL | 0.25 | 0.53 | 0.18 | 0.89 | 0.95 | 0.2 | 0.94 | 3.21 | 0.88 |
| ResNet18-no-TL-no-CW | 0 | 0.4 | 0 | 0.93 | 1 | 0 | 0.93 |  | 1 |
| ResNet18-TL | 0 | 0.58 | 0 | 0.93 | 1 | 0 | 0.93 |  | 1 |
| ResNet18-TL-no-CW | 0 | 0.56 | 0 | 0.93 | 1 | 0 | 0.93 |  | 1 |

Table 5.27: CIR - Trial 3B results

| Model Name | Max Sensitivity | AUC | F1-Score Pos Class | Accuracy | Specificity | PPV | NPV | PLR | NLR |
|---|---|---|---|---|---|---|---|---|---|
| ANN | 0.08 | 0.36 | 0 | 0.87 | 0.94 | 0 | 0.92 | 0 | 1.06 |
| ANN-no-CW | 0 | 0.35 | 0 | 0.93 | 1 | 0 | 0.93 | | 1 |
| VGG16-no-TL | 0 | 0.55 | 0 | 0.93 | 1 | 0 | 0.93 | | 1 |
| VGG16-no-TL-no-CW | 0.08 | 0.41 | 0 | 0.87 | 0.94 | 0 | 0.92 | 0 | 1.06 |
| VGG16-TL | 0 | 0.51 | 0 | 0.93 | 1 | 0 | 0.93 | | 1 |
| VGG16-TL-no-CW | 0.17 | 0.65 | 0.09 | 0.87 | 0.94 | 0.09 | 0.93 | 1.28 | 0.98 |
| ResNet50-no-TL | 0.17 | 0.44 | 0.1 | 0.89 | 0.95 | 0.12 | 0.93 | 1.83 | 0.96 |
| ResNet50-no-TL-no-CW | 0.08 | 0.46 | 0 | 0.93 | 1 | 0 | 0.93 | | 1 |
| ResNet50-TL | 0.08 | 0.52 | 0 | 0.89 | 0.96 | 0 | 0.92 | 0 | 1.04 |
| ResNet50-TL-no-CW | 0.17 | 0.56 | 0.1 | 0.89 | 0.95 | 0.11 | 0.93 | 1.6 | 0.97 |
| MobileNet-no-TL | 0 | 0.5 | 0 | 0.93 | 1 | 0 | 0.93 | | 1 |
| MobileNet-no-TL-no-CW | 0 | 0.52 | 0 | 0.93 | 1 | 0 | 0.93 | | 1 |
| MobileNet-TL | 0.08 | 0.58 | 0 | 0.88 | 0.95 | 0 | 0.92 | 0 | 1.05 |
| MobileNet-TL-no-CW | 0 | 0.39 | 0 | 0.93 | 1 | 0 | 0.93 | | 1 |
| ResNet18-no-TL | 0.08 | 0.53 | 0 | 0.88 | 0.95 | 0 | 0.92 | 0 | 1.05 |
| ResNet18-no-TL-no-CW | 0.08 | 0.45 | 0 | 0.87 | 0.94 | 0 | 0.92 | 0 | 1.07 |
| ResNet18-TL | 0.25 | 0.56 | 0.16 | 0.87 | 0.93 | 0.15 | 0.93 | 2.33 | 0.9 |
| ResNet18-TL-no-CW | 0 | 0.52 | 0 | 0.93 | 1 | 0 | 0.93 | | 1 |

Table 5.28: CIR - Trial 4 results

| Model Name | Max Sensitivity | AUC | F1-Score Pos Class | Accuracy | Specificity | PPV | NPV | PLR | NLR |
|---|---|---|---|---|---|---|---|---|---|
| VGG16-TL | 0.08 | 0.44 | 0 | 0.93 | 1 | 0 | 0.93 | | 1 |
| VGG16-TL-AUG | 0.25 | 0.55 | 0.17 | 0.88 | 0.94 | 0.17 | 0.94 | 2.57 | 0.89 |
| VGG16-TL-AUG-no-CW | 0.08 | 0.4 | 0 | 0.87 | 0.94 | 0 | 0.92 | 0 | 1.06 |
| VGG16-no-TL | 0.08 | 0.42 | 0 | 0.93 | 1 | 0 | 0.93 | | 1 |
| VGG16-no-TL-AUG | 0 | 0.37 | 0 | 0.93 | 1 | 0 | 0.93 | | 1 |
| VGG16-no-TL-AUG-no-CW | 0.08 | 0.43 | 0 | 0.93 | 1 | 0 | 0.93 | | 1 |
| ResNet18-TL | 0.08 | 0.53 | 0 | 0.89 | 0.95 | 0 | 0.92 | 0 | 1.05 |
| ResNet18-TL-AUG | 0.17 | 0.57 | 0.11 | 0.9 | 0.96 | 0.14 | 0.93 | 2.14 | 0.95 |
| ResNet18-TL-AUG-no-CW | 0.08 | 0.51 | 0 | 0.88 | 0.95 | 0 | 0.92 | 0 | 1.05 |
| ResNet18-no-TL | 0.17 | 0.48 | 0.09 | 0.87 | 0.94 | 0.09 | 0.93 | 1.28 | 0.98 |
| ResNet18-no-TL-AUG | 0.17 | 0.48 | 0.12 | 0.91 | 0.97 | 0.2 | 0.93 | 3.21 | 0.94 |
| ResNet18-no-TL-no-CW | 0 | 0.4 | 0 | 0.93 | 1 | 0 | 0.93 | | 1 |

Table 5.29: CIR - Trial 5 results

| Model Name | Max Sensitivity | AUC | F1-Score Pos Class | Accuracy | Specificity | PPV | NPV | PLR | NLR |
|---|---|---|---|---|---|---|---|---|---|
| ANN-anterior | 0.17 | 0.63 | 0 | 0.91 | 0.98 | 0 | 0.93 | 0 | 1.03 |
| ANN-tuned | 0.25 | 0.5 | 0.16 | 0.87 | 0.93 | 0.15 | 0.93 | 2.33 | 0.9 |

Table 5.30: CIR - Trial Optimization results

## 5.6 Computer Vision full results

#### 5.6.0.1 C3 (Preeclampsia, fetal growth restriction or birth weight percentile below 10)

| Model Name | Max Sensitivity | AUC | F1-Score Pos Class | Accuracy | Specificity | PPV | NPV | PLR | NLR |
|---|---|---|---|---|---|---|---|---|---|
| ResNet18-TL-no-CW-anterior | 0 | 0.53 | 0 | 0.93 | 1 | 0 | 0.93 | | 1 |
| ResNet18-TL-no-CW-tuned | 0 | 0.4 | 0 | 0.93 | 1 | 0 | 0.93 | | 1 |
| ANN-anterior | 0.17 | 0.63 | 0 | 0.91 | 0.98 | 0 | 0.93 | 0 | 1.03 |
| ANN-tuned | 0.25 | 0.5 | 0.16 | 0.87 | 0.93 | 0.15 | 0.93 | 2.33 | 0.9 |
| C3_GLCM_SVC | 0.21 | 0.49 | 0.2 | 0.67 | 0.79 | 0.2 | 0.79 | 0.95 | 1.01 |
| C3_GLCM_XGBOOST | 0.19 | 0.46 | 0.18 | 0.67 | 0.8 | 0.19 | 0.78 | 0.88 | 1.03 |
| C3_GLCM_LR | 0.2 | 0.49 | 0.19 | 0.67 | 0.8 | 0.2 | 0.78 | 0.92 | 1.02 |
| C3_HOG_SVC | 0.24 | 0.52 | 0.24 | 0.69 | 0.81 | 0.24 | 0.8 | 1.2 | 0.95 |
| C3_HOG_XGBOOST | 0.25 | 0.54 | 0.24 | 0.68 | 0.8 | 0.24 | 0.8 | 1.19 | 0.95 |
| C3_HOG_LR | 0.26 | 0.52 | 0.25 | 0.68 | 0.79 | 0.24 | 0.8 | 1.21 | 0.95 |

Table 5.31: C3 - Trial 1 results

| Model Name | Max Sensitivity | AUC | F1-Score Pos Class | Accuracy | Specificity | PPV | NPV | PLR | NLR |
|---|---|---|---|---|---|---|---|---|---|
| ResNet18-TL-no-CW-anterior | 0 | 0.53 | 0 | 0.93 | 1 | 0 | 0.93 | | 1 |
| ResNet18-TL-no-CW-tuned | 0 | 0.4 | 0 | 0.93 | 1 | 0 | 0.93 | | 1 |
| ANN-anterior | 0.17 | 0.63 | 0 | 0.91 | 0.98 | 0 | 0.93 | 0 | 1.03 |
| ANN-tuned | 0.25 | 0.5 | 0.16 | 0.87 | 0.93 | 0.15 | 0.93 | 2.33 | 0.9 |
| C3_HOG_SVC | 0.25 | 0.52 | 0.24 | 0.68 | 0.8 | 0.24 | 0.8 | 1.2 | 0.95 |
| C3_HOG_XGBOOST | 0.2 | 0.52 | 0.21 | 0.69 | 0.83 | 0.23 | 0.79 | 1.1 | 0.98 |
| C3_HOG_LR | 0.24 | 0.51 | 0.23 | 0.67 | 0.79 | 0.23 | 0.79 | 1.09 | 0.98 |
| C3_GLCM_SVC | 0.24 | 0.5 | 0.23 | 0.68 | 0.8 | 0.24 | 0.79 | 1.15 | 0.96 |
| C3_GLCM_XGBOOST | 0.19 | 0.45 | 0.18 | 0.66 | 0.79 | 0.19 | 0.78 | 0.85 | 1.04 |
| C3_GLCM_LR | 0.23 | 0.5 | 0.22 | 0.68 | 0.8 | 0.23 | 0.79 | 1.12 | 0.97 |

Table 5.32: C3 - Trial 2 results

| Model Name | Max Sensitivity | AUC | F1-Score Pos Class | Accuracy | Specificity | PPV | NPV | PLR | NLR |
|---|---|---|---|---|---|---|---|---|---|
| C3_HOG_SVC | 0.23 | 0.5 | 0.22 | 0.67 | 0.79 | 0.22 | 0.79 | 1.04 | 0.99 |
| C3_HOG_SVC-no-CW | 0.2 | 0.49 | 0.21 | 0.71 | 0.85 | 0.25 | 0.79 | 1.22 | 0.96 |
| C3_HOG_XGBOOST | 0.27 | 0.54 | 0.26 | 0.68 | 0.79 | 0.25 | 0.8 | 1.24 | 0.94 |
| C3_HOG_XGBOOST-no-CW | 0.21 | 0.51 | 0.21 | 0.68 | 0.8 | 0.21 | 0.79 | 1.01 | 1 |
| C3_HOG_LR | 0.18 | 0.51 | 0.17 | 0.66 | 0.79 | 0.18 | 0.78 | 0.79 | 1.05 |
| C3_HOG_LR-no-CW | 0.19 | 0.5 | 0.18 | 0.66 | 0.79 | 0.18 | 0.78 | 0.84 | 1.04 |
| C3_GLCM_SVC | 0.24 | 0.51 | 0.23 | 0.67 | 0.79 | 0.23 | 0.79 | 1.09 | 0.98 |
| C3_GLCM_SVC-no-CW | 0.15 | 0.49 | 0.15 | 0.67 | 0.82 | 0.17 | 0.78 | 0.74 | 1.06 |
| C3_GLCM_XGBOOST | 0.19 | 0.46 | 0.18 | 0.67 | 0.8 | 0.19 | 0.78 | 0.88 | 1.03 |
| C3_GLCM_XGBOOST-no-CW | 0.18 | 0.43 | 0.17 | 0.66 | 0.8 | 0.18 | 0.78 | 0.82 | 1.05 |
| C3_GLCM_LR | 0.2 | 0.49 | 0.19 | 0.66 | 0.79 | 0.19 | 0.78 | 0.89 | 1.03 |
| C3_GLCM_LR-no-CW | 0.16 | 0.47 | 0.15 | 0.66 | 0.79 | 0.16 | 0.78 | 0.7 | 1.08 |

Table 5.33: C3 - Trial 3 results

| Model Name | Max Sensitivity | AUC | F1-Score Pos Class | Accuracy | Specificity | PPV | NPV | PLR | NLR |
|---|---|---|---|---|---|---|---|---|---|
| C3_HOG_SVC | 0.27 | 0.55 | 0.26 | 0.68 | 0.8 | 0.26 | 0.8 | 1.29 | 0.93 |
| C3_HOG_SVC-no-CW | 0.25 | 0.55 | 0.24 | 0.68 | 0.8 | 0.24 | 0.8 | 1.17 | 0.96 |
| C3_HOG_XGBOOST | 0.29 | 0.52 | 0.28 | 0.69 | 0.8 | 0.28 | 0.81 | 1.43 | 0.89 |
| C3_HOG_XGBOOST-no-CW | 0.22 | 0.54 | 0.21 | 0.67 | 0.79 | 0.21 | 0.79 | 1.01 | 1 |
| C3_HOG_LR | 0.27 | 0.55 | 0.26 | 0.69 | 0.8 | 0.26 | 0.8 | 1.33 | 0.92 |
| C3_HOG_LR-no-CW | 0.3 | 0.55 | 0.29 | 0.69 | 0.8 | 0.28 | 0.81 | 1.45 | 0.89 |
| C3_GLCM_SVC | 0.29 | 0.52 | 0.27 | 0.68 | 0.79 | 0.27 | 0.8 | 1.36 | 0.91 |
| C3_GLCM_SVC-no-CW | 0.15 | 0.47 | 0.14 | 0.65 | 0.79 | 0.15 | 0.77 | 0.64 | 1.09 |
| C3_GLCM_XGBOOST | 0.23 | 0.5 | 0.22 | 0.68 | 0.8 | 0.23 | 0.79 | 1.1 | 0.98 |
| C3_GLCM_XGBOOST-no-CW | 0.17 | 0.45 | 0.16 | 0.66 | 0.79 | 0.17 | 0.78 | 0.75 | 1.06 |
| C3_GLCM_LR | 0.17 | 0.5 | 0.16 | 0.66 | 0.8 | 0.17 | 0.78 | 0.76 | 1.06 |
| C3_GLCM_LR-no-CW | 0.15 | 0.47 | 0.14 | 0.66 | 0.8 | 0.15 | 0.77 | 0.66 | 1.09 |

Table 5.34: C3 - Trial 4 results

| Model Name | Max Sensitivity | AUC | F1-Score Pos Class | Accuracy | Specificity | PPV | NPV | PLR | NLR |
|---|---|---|---|---|---|---|---|---|---|
| C3-SVC-AUG | 0.18 | 0.48 | 0.17 | 0.66 | 0.79 | 0.18 | 0.78 | 0.79 | 1.05 |
| C3-SVC-AUG-no-CW | 0.14 | 0.48 | 0.14 | 0.67 | 0.81 | 0.15 | 0.78 | 0.67 | 1.08 |
| C3-LR-AUG | 0.18 | 0.51 | 0.18 | 0.68 | 0.82 | 0.2 | 0.78 | 0.9 | 1.02 |
| C3-LR-AUG-no-CW | 0.18 | 0.5 | 0.17 | 0.66 | 0.8 | 0.18 | 0.78 | 0.82 | 1.05 |
| C3-SVC-no-AUG | 0.23 | 0.49 | 0.23 | 0.7 | 0.82 | 0.25 | 0.8 | 1.24 | 0.95 |
| C3-SVC-no-AUG-no-CW | 0.32 | 0.52 | 0.3 | 0.7 | 0.8 | 0.3 | 0.81 | 1.57 | 0.86 |
| C3-LR-no-AUG | 0.29 | 0.53 | 0.28 | 0.69 | 0.8 | 0.27 | 0.8 | 1.38 | 0.9 |
| C3-LR-no-AUG-no-CW | 0.26 | 0.53 | 0.26 | 0.7 | 0.82 | 0.28 | 0.8 | 1.42 | 0.91 |

Table 5.35: C3 - Trial 5 results

| Model Name | Max Sensitivity | AUC | F1-Score Pos Class | Accuracy | Specificity | PPV | NPV | PLR | NLR |
|---|---|---|---|---|---|---|---|---|---|
| C3-SVC-noaug-poly | 0.29 | 0.51 | 0.28 | 0.69 | 0.8 | 0.28 | 0.81 | 1.43 | 0.89 |
| C3-SVC-noaug-linear | 0.27 | 0.54 | 0.26 | 0.69 | 0.81 | 0.27 | 0.8 | 1.35 | 0.92 |
| C3-SVC-aug-poly | 0 | 0.5 | 0 | 0.79 | 1 | 0 | 0.79 | 0 | 1 |
| C3-SVC-aug-linear | 0.19 | 0.51 | 0.18 | 0.66 | 0.8 | 0.19 | 0.78 | 0.87 | 1.03 |
| C3-SVC-noaug-poly-anterior | 0.28 | 0.55 | 0.25 | 0.67 | 0.78 | 0.24 | 0.79 | 1.15 | 0.96 |
| C3-SVC-noaug-linear-anterior | 0.17 | 0.54 | 0.18 | 0.72 | 0.88 | 0.24 | 0.78 | 1.12 | 0.98 |

Table 5.36: C3 - Trial Optimization results

#### 5.6.0.2 LBW (Birth weight percentile below 10)

| Model Name | Max Sensitivity | AUC | F1-Score Pos Class | Accuracy | Specificity | PPV | NPV | PLR | NLR |
|---|---|---|---|---|---|---|---|---|---|
| LBW_HOG_SVC | 0.21 | 0.48 | 0.19 | 0.74 | 0.84 | 0.19 | 0.85 | 1.22 | 0.96 |
| LBW_HOG_XGBOOST | 0.18 | 0.57 | 0.18 | 0.74 | 0.85 | 0.18 | 0.84 | 1.17 | 0.97 |
| LBW_HOG_LR | 0.18 | 0.51 | 0.17 | 0.73 | 0.84 | 0.17 | 0.84 | 1.08 | 0.99 |
| LBW_GLCM_SVC | 0.17 | 0.5 | 0.17 | 0.74 | 0.85 | 0.17 | 0.84 | 1.1 | 0.98 |
| LBW_GLCM_XGBOOST | 0.08 | 0.47 | 0.07 | 0.72 | 0.84 | 0.08 | 0.83 | 0.44 | 1.11 |
| LBW_GLCM_LR | 0.15 | 0.49 | 0.14 | 0.73 | 0.84 | 0.14 | 0.84 | 0.89 | 1.02 |

Table 5.37: LBW - Trial 1 results

| Model Name | Max Sensitivity | AUC | F1-Score Pos Class | Accuracy | Specificity | PPV | NPV | PLR | NLR |
|---|---|---|---|---|---|---|---|---|---|
| LBW_HOG_SVC | 0.18 | 0.48 | 0.18 | 0.76 | 0.87 | 0.2 | 0.85 | 1.29 | 0.96 |
| LBW_HOG_XGBOOST | 0.21 | 0.54 | 0.2 | 0.74 | 0.85 | 0.2 | 0.85 | 1.27 | 0.95 |
| LBW_HOG_LR | 0.2 | 0.51 | 0.19 | 0.74 | 0.85 | 0.19 | 0.85 | 1.23 | 0.96 |
| LBW_GLCM_SVC | 0.16 | 0.51 | 0.16 | 0.74 | 0.86 | 0.16 | 0.84 | 1.03 | 0.99 |
| LBW_GLCM_XGBOOST | 0.1 | 0.47 | 0.1 | 0.72 | 0.84 | 0.1 | 0.83 | 0.58 | 1.08 |
| LBW_GLCM_LR | 0.15 | 0.5 | 0.15 | 0.75 | 0.86 | 0.16 | 0.84 | 1 | 1 |

Table 5.38: LBW - Trial 2 results

| Model Name | Max Sensitivity | AUC | F1-Score Pos Class | Accuracy | Specificity | PPV | NPV | PLR | NLR |
|---|---|---|---|---|---|---|---|---|---|
| LBW_HOG_SVC | 0.17 | 0.51 | 0.16 | 0.73 | 0.84 | 0.16 | 0.84 | 1.01 | 1 |
| LBW_HOG_SVC-no-CW | 0.13 | 0.51 | 0.12 | 0.73 | 0.85 | 0.13 | 0.83 | 0.77 | 1.04 |
| LBW_HOG_XGBOOST | 0.2 | 0.5 | 0.19 | 0.75 | 0.85 | 0.19 | 0.85 | 1.25 | 0.96 |
| LBW_HOG_XGBOOST-no-CW | 0.23 | 0.53 | 0.21 | 0.74 | 0.84 | 0.21 | 0.85 | 1.38 | 0.93 |
| LBW_HOG_LR | 0.14 | 0.5 | 0.14 | 0.75 | 0.86 | 0.15 | 0.84 | 0.93 | 1.01 |
| LBW_HOG_LR-no-CW | 0.11 | 0.5 | 0.11 | 0.74 | 0.86 | 0.13 | 0.83 | 0.76 | 1.04 |
| LBW_GLCM_SVC | 0.13 | 0.5 | 0.13 | 0.76 | 0.88 | 0.15 | 0.84 | 0.94 | 1.01 |
| LBW_GLCM_SVC-no-CW | 0.14 | 0.48 | 0.13 | 0.73 | 0.84 | 0.13 | 0.83 | 0.8 | 1.04 |
| LBW_GLCM_XGBOOST | 0.1 | 0.42 | 0.11 | 0.76 | 0.89 | 0.14 | 0.84 | 0.86 | 1.02 |
| LBW_GLCM_XGBOOST-no-CW | 0.14 | 0.45 | 0.13 | 0.73 | 0.84 | 0.13 | 0.83 | 0.79 | 1.04 |
| LBW_GLCM_LR | 0.11 | 0.48 | 0.11 | 0.73 | 0.85 | 0.12 | 0.83 | 0.69 | 1.05 |
| LBW_GLCM_LR-no-CW | 0.14 | 0.45 | 0.13 | 0.73 | 0.84 | 0.13 | 0.83 | 0.79 | 1.04 |

Table 5.39: LBW - Trial 3 results

| Model Name | Max Sensitivity | AUC | F1-Score Pos Class | Accuracy | Specificity | PPV | NPV | PLR | NLR |
|---|---|---|---|---|---|---|---|---|---|
| LBW_HOG_SVC | 0.1 | 0.44 | 0.1 | 0.72 | 0.84 | 0.1 | 0.83 | 0.59 | 1.08 |
| LBW_HOG_SVC-no-CW | 0.13 | 0.44 | 0.13 | 0.76 | 0.88 | 0.15 | 0.84 | 0.94 | 1.01 |
| LBW_HOG_XGBOOST | 0.13 | 0.53 | 0.12 | 0.73 | 0.85 | 0.13 | 0.83 | 0.77 | 1.04 |
| LBW_HOG_XGBOOST-no-CW | 0.16 | 0.53 | 0.15 | 0.73 | 0.85 | 0.16 | 0.84 | 0.97 | 1 |
| LBW_HOG_LR | 0.15 | 0.55 | 0.15 | 0.75 | 0.87 | 0.17 | 0.84 | 1.07 | 0.99 |
| LBW_HOG_LR-no-CW | 0.17 | 0.55 | 0.17 | 0.75 | 0.87 | 0.19 | 0.84 | 1.2 | 0.97 |
| LBW_GLCM_SVC | 0.11 | 0.51 | 0.12 | 0.75 | 0.87 | 0.13 | 0.84 | 0.81 | 1.03 |
| LBW_GLCM_SVC-no-CW | 0.15 | 0.52 | 0.14 | 0.73 | 0.84 | 0.14 | 0.84 | 0.87 | 1.02 |
| LBW_GLCM_XGBOOST | 0.2 | 0.52 | 0.18 | 0.74 | 0.85 | 0.19 | 0.84 | 1.2 | 0.96 |
| LBW_GLCM_XGBOOST-no-CW | 0.21 | 0.52 | 0.2 | 0.74 | 0.85 | 0.2 | 0.85 | 1.29 | 0.95 |
| LBW_GLCM_LR | 0.13 | 0.51 | 0.12 | 0.74 | 0.86 | 0.14 | 0.84 | 0.82 | 1.03 |
| LBW_GLCM_LR-no-CW | 0.15 | 0.48 | 0.15 | 0.75 | 0.86 | 0.16 | 0.84 | 1 | 1 |

Table 5.40: LBW - Trial 4 results

| Model Name | Max Sensitivity | AUC | F1-Score Pos Class | Accuracy | Specificity | PPV | NPV | PLR | NLR |
|---|---|---|---|---|---|---|---|---|---|
| LBW_HOG_SVC | 0.1 | 0.44 | 0.1 | 0.72 | 0.84 | 0.1 | 0.83 | 0.59 | 1.08 |
| LBW_HOG_SVC-no-CW | 0.13 | 0.44 | 0.13 | 0.76 | 0.88 | 0.15 | 0.84 | 0.94 | 1.01 |
| LBW_HOG_XGBOOST | 0.13 | 0.53 | 0.12 | 0.73 | 0.85 | 0.13 | 0.83 | 0.77 | 1.04 |
| LBW_HOG_XGBOOST-no-CW | 0.16 | 0.53 | 0.15 | 0.73 | 0.85 | 0.16 | 0.84 | 0.97 | 1 |
| LBW_HOG_LR | 0.15 | 0.55 | 0.15 | 0.75 | 0.87 | 0.17 | 0.84 | 1.07 | 0.99 |
| LBW_HOG_LR-no-CW | 0.17 | 0.55 | 0.17 | 0.75 | 0.87 | 0.19 | 0.84 | 1.2 | 0.97 |
| LBW_GLCM_SVC | 0.11 | 0.51 | 0.12 | 0.75 | 0.87 | 0.13 | 0.84 | 0.81 | 1.03 |
| LBW_GLCM_SVC-no-CW | 0.15 | 0.52 | 0.14 | 0.73 | 0.84 | 0.14 | 0.84 | 0.87 | 1.02 |
| LBW_GLCM_XGBOOST | 0.2 | 0.52 | 0.18 | 0.74 | 0.85 | 0.19 | 0.84 | 1.2 | 0.96 |
| LBW_GLCM_XGBOOST-no-CW | 0.21 | 0.52 | 0.2 | 0.74 | 0.85 | 0.2 | 0.85 | 1.29 | 0.95 |
| LBW_GLCM_LR | 0.13 | 0.51 | 0.12 | 0.74 | 0.86 | 0.14 | 0.84 | 0.82 | 1.03 |
| LBW_GLCM_LR-no-CW | 0.15 | 0.48 | 0.15 | 0.75 | 0.86 | 0.16 | 0.84 | 1 | 1 |
| LBW-SVC-AUG | 0.09 | 0.44 | 0.09 | 0.74 | 0.86 | 0.1 | 0.83 | 0.58 | 1.07 |
| LBW-SVC-AUG-no-CW | 0.15 | 0.46 | 0.14 | 0.73 | 0.85 | 0.15 | 0.84 | 0.9 | 1.02 |
| LBW-LR-AUG | 0.08 | 0.43 | 0.08 | 0.75 | 0.88 | 0.1 | 0.83 | 0.55 | 1.06 |
| LBW-LR-AUG-no-CW | 0.11 | 0.48 | 0.11 | 0.73 | 0.85 | 0.12 | 0.83 | 0.69 | 1.05 |
| LBW-SVC-no-AUG | 0.17 | 0.54 | 0.16 | 0.73 | 0.84 | 0.16 | 0.84 | 1.01 | 1 |
| LBW-SVC-no-AUG-no-CW | 0.17 | 0.54 | 0.18 | 0.76 | 0.88 | 0.2 | 0.85 | 1.31 | 0.96 |
| LBW-LR-no-AUG | 0.1 | 0.52 | 0.1 | 0.73 | 0.85 | 0.1 | 0.83 | 0.61 | 1.07 |
| LBW-LR-no-AUG-no-CW | 0.16 | 0.53 | 0.15 | 0.73 | 0.84 | 0.15 | 0.84 | 0.96 | 1.01 |

Table 5.41: LBW - Trial 5 results

| Model Name | Max Sensitivity | AUC | F1-Score Pos Class | Accuracy | Specificity | PPV | NPV | PLR | NLR |
|---|---|---|---|---|---|---|---|---|---|
| LBW-SVC-noaug-poly | 0.16 | 0.46 | 0.15 | 0.73 | 0.85 | 0.16 | 0.84 | 0.97 | 1 |
| LBW-SVC-noaug-linear | 0.16 | 0.5 | 0.15 | 0.74 | 0.85 | 0.16 | 0.84 | 0.99 | 1 |
| LBW-SVC-aug-poly | 0.2 | 0.54 | 0.19 | 0.74 | 0.85 | 0.19 | 0.85 | 1.23 | 0.96 |
| LBW-SVC-aug-linear | 0.15 | 0.49 | 0.14 | 0.73 | 0.84 | 0.14 | 0.84 | 0.86 | 1.03 |
| LBW-SVC-noaug-poly-anterior | 0.23 | 0.53 | 0.2 | 0.71 | 0.82 | 0.2 | 0.82 | 1.12 | 0.97 |
| LBW-SVC-noaug-linear-anterior | 0.13 | 0.45 | 0.11 | 0.7 | 0.84 | 0.12 | 0.81 | 0.61 | 1.08 |

Table 5.42: LBW - Trial Optimization results

#### 5.6.0.3 PRE (Preeclampsia)

| Model Name | Max Sensitivity | AUC | F1-Score Pos Class | Accuracy | Specificity | PPV | NPV | PLR | NLR |
|---|---|---|---|---|---|---|---|---|---|
| PRE_HOG_SVC | 0.14 | 0.57 | 0.1 | 0.88 | 0.94 | 0.1 | 0.94 | 1.52 | 0.97 |
| PRE_HOG_XGBOOST | 0.05 | 0.54 | 0 | 0.89 | 0.95 | 0 | 0.93 | 0 | 1.06 |
| PRE_HOG_LR | 0.1 | 0.47 | 0.05 | 0.89 | 0.95 | 0.06 | 0.94 | 0.9 | 1.01 |
| PRE_GLCM_SVC | 0.1 | 0.61 | 0.08 | 0.93 | 0.99 | 0.33 | 0.94 | 7.24 | 0.96 |
| PRE_GLCM_XGBOOST | 0 | 0.42 | 0 | 0.94 | 1 | 0 | 0.94 | | 1 |
| PRE_GLCM_LR | 0.19 | 0.55 | 0.15 | 0.9 | 0.95 | 0.17 | 0.94 | 2.9 | 0.9 |

Table 5.43: PRE - Trial 1 results

| Model Name | Max Sensitivity | AUC | F1-Score Pos Class | Accuracy | Specificity | PPV | NPV | PLR | NLR |
|---|---|---|---|---|---|---|---|---|---|
| PRE_HOG_SVC | 0 | 0.54 | 0 | 0.94 | 1 | 0 | 0.94 | 0 | 1 |
| PRE_HOG_XGBOOST | 0.14 | 0.48 | 0.13 | 0.92 | 0.97 | 0.2 | 0.94 | 3.62 | 0.93 |
| PRE_HOG_LR | 0.14 | 0.48 | 0.1 | 0.88 | 0.94 | 0.1 | 0.94 | 1.52 | 0.97 |
| PRE_GLCM_SVC | 0.05 | 0.51 | 0 | 0.93 | 0.99 | 0 | 0.93 | 0 | 1.01 |
| PRE_GLCM_XGBOOST | 0 | 0.48 | 0 | 0.94 | 1 | 0 | 0.94 | 0 | 1 |
| PRE_GLCM_LR | 0.19 | 0.52 | 0.14 | 0.89 | 0.94 | 0.14 | 0.94 | 2.29 | 0.91 |

Table 5.44: PRE - Trial 2 results

| Model Name | Max Sensitivity | AUC | F1-Score Pos Class | Accuracy | Specificity | PPV | NPV | PLR | NLR |
|---|---|---|---|---|---|---|---|---|---|
| PRE_HOG_SVC | 0.05 | 0.51 | 0 | 0.89 | 0.95 | 0 | 0.93 | 0 | 1.05 |
| PRE_HOG_SVC-no-CW | 0 | 0.54 | 0 | 0.94 | 1 | 0 | 0.94 | 0 | 1 |
| PRE_HOG_XGBOOST | 0.1 | 0.45 | 0.05 | 0.88 | 0.94 | 0.05 | 0.93 | 0.8 | 1.01 |
| PRE_HOG_XGBOOST-no-CW | 0.05 | 0.46 | 0 | 0.88 | 0.94 | 0 | 0.93 | 0 | 1.06 |
| PRE_HOG_LR | 0 | 0.47 | 0 | 0.94 | 1 | 0 | 0.94 | 0 | 1 |
| PRE_HOG_LR-no-CW | 0 | 0.46 | 0 | 0.94 | 1 | 0 | 0.94 | 0 | 1 |
| PRE_GLCM_SVC | 0 | 0.42 | 0 | 0.94 | 1 | 0 | 0.94 | 0 | 1 |
| PRE_GLCM_SVC-no-CW | 0.05 | 0.47 | 0 | 0.88 | 0.94 | 0 | 0.93 | 0 | 1.07 |
| PRE_GLCM_XGBOOST | 0.05 | 0.4 | 0 | 0.9 | 0.97 | 0 | 0.93 | 0 | 1.03 |
| PRE_GLCM_XGBOOST-no-CW | 0.05 | 0.43 | 0 | 0.92 | 0.98 | 0 | 0.93 | 0 | 1.02 |
| PRE_GLCM_LR | 0.05 | 0.41 | 0 | 0.92 | 0.98 | 0 | 0.93 | 0 | 1.02 |
| PRE_GLCM_LR-no-CW | 0.05 | 0.48 | 0 | 0.9 | 0.96 | 0 | 0.93 | 0 | 1.04 |

Table 5.45: PRE - Trial 3 results

| Model Name | Max Sensitivity | AUC | F1-Score Pos Class | Accuracy | Specificity | PPV | NPV | PLR | NLR |
|---|---|---|---|---|---|---|---|---|---|
| PRE_HOG_SVC | 0.05 | 0.52 | 0 | 0.91 | 0.98 | 0 | 0.93 | 0 | 1.02 |
| PRE_HOG_SVC-no-CW | 0.1 | 0.54 | 0.06 | 0.9 | 0.96 | 0.07 | 0.94 | 1.11 | 0.99 |
| PRE_HOG_XGBOOST | 0.1 | 0.49 | 0.07 | 0.92 | 0.98 | 0.14 | 0.94 | 2.41 | 0.97 |
| PRE_HOG_XGBOOST-no-CW | 0.05 | 0.54 | 0 | 0.9 | 0.96 | 0 | 0.93 | 0 | 1.04 |
| PRE_HOG_LR | 0.1 | 0.47 | 0.05 | 0.88 | 0.94 | 0.05 | 0.93 | 0.8 | 1.01 |
| PRE_HOG_LR-no-CW | 0.1 | 0.48 | 0.06 | 0.9 | 0.96 | 0.08 | 0.94 | 1.21 | 0.99 |
| PRE_GLCM_SVC | 0.19 | 0.59 | 0.14 | 0.89 | 0.94 | 0.14 | 0.94 | 2.29 | 0.91 |
| PRE_GLCM_SVC-no-CW | 0.05 | 0.36 | 0 | 0.89 | 0.95 | 0 | 0.93 | 0 | 1.05 |
| PRE_GLCM_XGBOOST | 0 | 0.46 | 0 | 0.94 | 1 | 0 | 0.94 | 0 | 1 |
| PRE_GLCM_XGBOOST-no-CW | 0.05 | 0.46 | 0 | 0.89 | 0.95 | 0 | 0.93 | 0 | 1.05 |
| PRE_GLCM_LR | 0.05 | 0.43 | 0 | 0.9 | 0.97 | 0 | 0.93 | 0 | 1.03 |
| PRE_GLCM_LR-no-CW | 0.05 | 0.48 | 0 | 0.92 | 0.99 | 0 | 0.93 | 0 | 1.01 |

Table 5.46: PRE - Trial 4 results

| Model Name | Max Sensitivity | AUC | F1-Score Pos Class | Accuracy | Specificity | PPV | NPV | PLR | NLR |
|---|---|---|---|---|---|---|---|---|---|
| PRE_SVC_AUG | 0 | 0.33 | 0 | 0.94 | 1 | 0 | 0.94 | 0 | 1 |
| PRE_SVC_AUG_no_CW | 0.1 | 0.59 | 0.05 | 0.88 | 0.94 | 0.05 | 0.93 | 0.76 | 1.02 |
| PRE_LR_AUG | 0.05 | 0.5 | 0 | 0.9 | 0.97 | 0 | 0.93 | 0 | 1.03 |
| PRE_LR_AUG_no_CW | 0.05 | 0.46 | 0 | 0.92 | 0.99 | 0 | 0.93 | 0 | 1.01 |
| PRE_SVC_no_AUG | 0.05 | 0.51 | 0 | 0.91 | 0.97 | 0 | 0.93 | 0 | 1.03 |
| PRE_SVC_no_AUG_no_CW | 0.1 | 0.55 | 0.05 | 0.89 | 0.94 | 0.06 | 0.93 | 0.85 | 1.01 |
| PRE_LR_no_AUG | 0.1 | 0.52 | 0.06 | 0.9 | 0.95 | 0.07 | 0.94 | 1.03 | 1 |
| PRE_LR_no_AUG_no_CW | 0.1 | 0.52 | 0.06 | 0.9 | 0.96 | 0.08 | 0.94 | 1.32 | 0.99 |

Table 5.47: PRE - Trial 5 results

| Model Name | Max Sensitivity | AUC | F1-Score Pos Class | Accuracy | Specificity | PPV | NPV | PLR | NLR |
|---|---|---|---|---|---|---|---|---|---|
| PRE_SVC_noaug_poly | 0.05 | 0.45 | 0 | 0.88 | 0.94 | 0 | 0.93 | 0 | 1.07 |
| PRE_SVC_noaug_linear | 0.05 | 0.57 | 0 | 0.88 | 0.94 | 0 | 0.93 | 0 | 1.06 |
| PRE_SVC_aug_poly | 0 | 0.5 | 0 | 0.94 | 1 | 0 | 0.94 | 0 | 1 |
| PRE_SVC_aug_linear | 0.05 | 0.5 | 0 | 0.92 | 0.98 | 0 | 0.93 | 0 | 1.02 |
| PRE_SVC_noaug_poly_anterior | 0.56 | 0.76 | 0.53 | 0.96 | 0.99 | 0.67 | 0.97 | 34.67 | 0.56 |
| PRE_SVC_noaug_linear_anterior | 0.33 | 0.75 | 0.22 | 0.92 | 0.96 | 0.22 | 0.96 | 4.95 | 0.81 |

Table 5.48: PRE - Trial Optimization results

#### 5.6.0.4 CIR (Fetal growth restriction)

| Model Name | Max Sensitivity | AUC | F1-Score Pos Class | Accuracy | Specificity | PPV | NPV | PLR | NLR |
|---|---|---|---|---|---|---|---|---|---|
| CIR_HOG_SVC | 0 | 0.55 | 0 | 0.94 | 1 | 0 | 0.94 | 1 | |
| CIR_HOG_XGBOOST | 0.19 | 0.6 | 0.15 | 0.9 | 0.95 | 0.16 | 0.94 | 2.71 | 0.9 |
| CIR_HOG_LR | 0.14 | 0.47 | 0.1 | 0.89 | 0.94 | 0.1 | 0.94 | 1.61 | 0.96 |
| CIR_GLCM_SVC | 0 | 0.46 | 0 | 0.94 | 1 | 0 | 0.94 | 1 | |
| CIR_GLCM_XGBOOST | 0.1 | 0.35 | 0.05 | 0.88 | 0.94 | 0.05 | 0.93 | 0.8 | 1.01 |
| CIR_GLCM_LR | 0.19 | 0.58 | 0.14 | 0.89 | 0.94 | 0.14 | 0.94 | 2.41 | 0.91 |

Table 5.49: CIR - Trial 1 results

| Model Name | Max Sensitivity | AUC | F1-Score Pos Class | Accuracy | Specificity | PPV | NPV | PLR | NLR |
|---|---|---|---|---|---|---|---|---|---|
| CIR_HOG_SVC | 0 | 0.53 | 0 | 0.94 | 1 | 0 | 0.94 | 0 | 1 |
| CIR_HOG_XGBOOST | 0.14 | 0.61 | 0.12 | 0.91 | 0.97 | 0.18 | 0.94 | 3.22 | 0.93 |
| CIR_HOG_LR | 0.14 | 0.48 | 0.1 | 0.89 | 0.94 | 0.11 | 0.94 | 1.7 | 0.96 |
| CIR_GLCM_SVC | 0.14 | 0.57 | 0.12 | 0.91 | 0.97 | 0.17 | 0.94 | 2.9 | 0.94 |
| CIR_GLCM_XGBOOST | 0.05 | 0.33 | 0 | 0.9 | 0.96 | 0 | 0.93 | 0 | 1.04 |
| CIR_GLCM_LR | 0.1 | 0.59 | 0.07 | 0.92 | 0.98 | 0.12 | 0.94 | 2.07 | 0.97 |

Table 5.50: CIR - Trial 2 results

| Model Name | Max Sensitivity | AUC | F1-Score Pos Class | Accuracy | Specificity | PPV | NPV | PLR | NLR |
|---|---|---|---|---|---|---|---|---|---|
| CIR_HOG_SVC | 0.14 | 0.53 | 0.1 | 0.89 | 0.94 | 0.11 | 0.94 | 1.7 | 0.96 |
| CIR_HOG_SVC-no-CW | 0.1 | 0.53 | 0.07 | 0.92 | 0.98 | 0.17 | 0.94 | 2.9 | 0.97 |
| CIR_HOG_XGBOOST | 0.19 | 0.54 | 0.15 | 0.9 | 0.95 | 0.17 | 0.94 | 2.9 | 0.9 |
| CIR_HOG_XGBOOST-no-CW | 0.19 | 0.61 | 0.14 | 0.89 | 0.94 | 0.14 | 0.94 | 2.41 | 0.91 |
| CIR_HOG_LR | 0.14 | 0.5 | 0.1 | 0.88 | 0.94 | 0.1 | 0.94 | 1.52 | 0.97 |
| CIR_HOG_LR-no-CW | 0.14 | 0.52 | 0.13 | 0.92 | 0.97 | 0.2 | 0.94 | 3.62 | 0.93 |
| CIR_GLCM_SVC | 0.19 | 0.56 | 0.14 | 0.89 | 0.94 | 0.14 | 0.94 | 2.29 | 0.91 |
| CIR_GLCM_SVC-no-CW | 0 | 0.46 | 0 | 0.94 | 1 | 0 | 0.94 | 0 | 1 |
| CIR_GLCM_XGBOOST | 0.05 | 0.47 | 0 | 0.9 | 0.97 | 0 | 0.93 | 0 | 1.03 |
| CIR_GLCM_XGBOOST-no-CW | 0.05 | 0.49 | 0 | 0.92 | 0.98 | 0 | 0.93 | 0 | 1.02 |
| CIR_GLCM_LR | 0.14 | 0.55 | 0.15 | 0.93 | 0.99 | 0.4 | 0.94 | 9.65 | 0.91 |
| CIR_GLCM_LR-no-CW | 0.05 | 0.54 | 0 | 0.88 | 0.94 | 0 | 0.93 | 0 | 1.07 |

Table 5.51: CIR - Trial 3 results

| Model Name | Max Sensitivity | AUC | F1-Score Pos Class | Accuracy | Specificity | PPV | NPV | PLR | NLR |
|---|---|---|---|---|---|---|---|---|---|
| CIR_HOG_SVC | 0.19 | 0.58 | 0.15 | 0.89 | 0.94 | 0.15 | 0.94 | 2.55 | 0.91 |
| CIR_HOG_SVC-no-CW | 0.1 | 0.55 | 0.06 | 0.9 | 0.96 | 0.08 | 0.94 | 1.32 | 0.99 |
| CIR_HOG_XGBOOST | 0 | 0.5 | 0 | 0.94 | 1 | 0 | 0.94 | 0 | 1 |
| CIR_HOG_XGBOOST-no-CW | 0 | 0.47 | 0 | 0.94 | 1 | 0 | 0.94 | 0 | 1 |
| CIR_HOG_LR | 0.24 | 0.61 | 0.18 | 0.89 | 0.94 | 0.17 | 0.94 | 3.05 | 0.86 |
| CIR_HOG_LR-no-CW | 0.24 | 0.59 | 0.18 | 0.89 | 0.94 | 0.17 | 0.94 | 3.05 | 0.86 |
| CIR_GLCM_SVC | 0 | 0.46 | 0 | 0.94 | 1 | 0 | 0.94 | 0 | 1 |
| CIR_GLCM_SVC-no-CW | 0 | 0.52 | 0 | 0.94 | 1 | 0 | 0.94 | 0 | 1 |
| CIR_GLCM_XGBOOST | 0.05 | 0.5 | 0 | 0.94 | 1 | 0 | 0.94 | 0 | 1 |
| CIR_GLCM_XGBOOST-no-CW | 0.05 | 0.49 | 0 | 0.94 | 1 | 0 | 0.94 | 0 | 1 |
| CIR_GLCM_LR | 0.05 | 0.45 | 0 | 0.89 | 0.95 | 0 | 0.93 | 0 | 1.05 |
| CIR_GLCM_LR-no-CW | 0 | 0.39 | 0 | 0.94 | 1 | 0 | 0.94 | 0 | 1 |

Table 5.52: CIR - Trial 4 results

| Model Name | Max Sensitivity | AUC | F1-Score Pos Class | Accuracy | Specificity | PPV | NPV | PLR | NLR |
|---|---|---|---|---|---|---|---|---|---|
| CIR-SVC-AUG | 0.14 | 0.52 | 0.1 | 0.88 | 0.94 | 0.1 | 0.94 | 1.52 | 0.97 |
| CIR-SVC-AUG-no-CW | 0.14 | 0.55 | 0.11 | 0.9 | 0.95 | 0.12 | 0.94 | 1.93 | 0.95 |
| CIR-LR-AUG | 0 | 0.48 | 0 | 0.94 | 1 | 0 | 0.94 | 0 | 1 |
| CIR-LR-AUG-no-CW | 0 | 0.4 | 0 | 0.94 | 1 | 0 | 0.94 | 0 | 1 |
| CIR-SVC-no-AUG | 0 | 0.38 | 0 | 0.94 | 1 | 0 | 0.94 | 0 | 1 |
| CIR-SVC-no-AUG-no-CW | 0.05 | 0.47 | 0 | 0.91 | 0.97 | 0 | 0.93 | 0 | 1.03 |
| CIR-LR-no-AUG | 0.14 | 0.61 | 0.1 | 0.89 | 0.94 | 0.1 | 0.94 | 1.61 | 0.96 |
| CIR-LR-no-AUG-no-CW | 0.14 | 0.61 | 0.11 | 0.9 | 0.95 | 0.12 | 0.94 | 2.07 | 0.95 |

Table 5.53: CIR - Trial 5 results

| Model Name | Max Sensitivity | AUC | F1-Score Pos Class | Accuracy | Specificity | PPV | NPV | PLR | NLR |
|---|---|---|---|---|---|---|---|---|---|
| CIR-SVC-noaug-poly | 0.1 | 0.55 | 0.05 | 0.89 | 0.95 | 0.06 | 0.94 | 0.9 | 1.01 |
| CIR-SVC-noaug-linear | 0.05 | 0.52 | 0 | 0.91 | 0.97 | 0 | 0.93 | 0 | 1.03 |
| CIR-SVC-aug-poly | 0.1 | 0.63 | 0.05 | 0.89 | 0.94 | 0.06 | 0.93 | 0.85 | 1.01 |
| CIR-SVC-aug-linear | 0.14 | 0.56 | 0.11 | 0.9 | 0.95 | 0.12 | 0.94 | 2.07 | 0.95 |
| CIR-SVC-noaug-poly-anterior | 0.22 | 0.47 | 0.12 | 0.92 | 0.96 | 0.14 | 0.95 | 2.94 | 0.92 |
| CIR-SVC-noaug-linear-anterior | 0 | 0.55 | 0 | 0.95 | 1 | 0 | 0.95 | 0 | 1 |

Table 5.54: CIR - Trial Optimization results