Towards stable feature detection for 3D reconstruction of the placental surface during FLOVA surgery

Convolutional neural network based feature detection

Myrthe Buser

Master Thesis Technical Medicine

02-03-2020

University of Twente

Graduation thesis Technical Medicine

Towards stable feature detection for 3D reconstruction of the placental surface during the FLOVA procedure

Convolutional neural network based feature detection

Author:
M.A.D. Buser

Supervisor:
Dr. E. Sikkel
Dr.ir. F. van der Heijden
Drs. R.J. Haarman
A.M. van der Schot, MSc
C. Jeltes, MSc

Amalia kinderziekenhuis
Radboudumc

UNIVERSITY OF TWENTE.

02-03-2020

Voorwoord

Twee jaar geleden, op 4 maart, starte ik mijn eerste stage voor Technische Geneeskunde. De dag ná mijn verjaardag. Vandaag, op 2 maart, ligt mijn master scriptie voor jullie. De dag vóór mijn verjaardag.

Mijn afstudeerstage was gecentreerd rond de zorg voor ongeboren tweelingen. Mijn hoofdproject hierin ging over de software die de placenta in kaart brengt tijdens foetale therapie van tweelingen. Hiernaast heb ik een klinische studie opgezet die zich erop richt om het effect van het delen van een placenta op het geboortegewicht van tweelingen in kaart te brengen. De resultaten hiervan kunt u lezen in de Appendix.

Ik wil graag beginnen met het bedanken van het foetale therapieteam van het Raedboudumc. Het werk wat jullie verrichten is in mijn ogen heel speciaal. De steun die ik van jullie gekregen heb en de dingen die ik van jullie geleerd heb tijdens mijn afstuderen, zal ik nooit vergeten. In het bijzonder wil ik dokter Frank Vandenbussche bedanken, die mij in de eerste helft van mijn stage begeleid heeft. Helaas konden we het jaar niet samen afmaken, maar dokter Esther Sikkel bleek een meer dan waardige opvolger. Verder wil ik Claire Jeltes en Anouk van der Schot bedanken voor de dagelijkse begeleiding. Bedankt dat jullie me de ruimte, het vertrouwen en soms de benodigde steun gaven om aan mijn persoonlijke doelen te werken. Verder wil ik Marèll en Samantha bedanken voor de lol die we samen als stagiaires hebben. En voor Samantha een speciaal woord van dank: bedankt dat je de voorkant van mijn scriptie hebt gemaakt!

Natuurlijk kan ik de begeleiders van de UT niet vergeten. Rian Haarman, ik heb heel veel van je geleerd gedurende de twee jaar intervisie die we samen gehad hebben. Daarover gesproken wil ik Sarah, Marijn, Mechteld en Iris bedanken voor alle koekjes, tranen en goede gesprekken van het afgelopen jaar. Natuurlijk bedank ik ook Ferdi van der Heijden voor de goede gesprekken die we over de technische aspecten van mijn onderzoek gehad hebben.

Ik zou het niet hebben kunnen doen zonder de steun van mijn vrienden en familie. Mama, papa, Maarten: bedankt. Leonie, ik kan je niet genoeg bedanken voor me door de lastige momenten helpen, voor in me blijven geloven, voor het meerdere keren doorlezen van mijn scriptie en voor alle leuke momenten die we het afgelopen jaar hebben gehad.

Ik wil dit voorwoord afsluiten met het bedanken van iemand die speciaal voor mij is. Marc Wijnen, meer dan vijftien jaar geleden, kort na door jou

geopereerd te zijn, vertelde ik je dat ik dokter wilde worden. Dokter ben ik niet geworden, maar vijftien jaar later ontmoetten we elkaar opnieuw. Weer was ik met je op de operatiekamer, maar deze keer had ik het geluk om met je samen te mogen werken. Bedankt voor het altijd blijven geloven in mijn droom, die vijftien jaar geleden in jouw spreekkamer begon.

Abstract

Twin pregnancy can be complicated by Twin-to-Twin-transfusion Syndrome (TTTS). To cure this condition, the Fetoscopic Laser Occlusion of Vascular Anastomoses (FLOVA) procedure can be performed. During FLOVA, the placental anastomoses causing TTTS are coagulated under fetoscopic view. FLOVA is a challenging procedure, because the field of view is limited due the need to use instruments of a small size. To artificially aid the field of view, 3D FLOVA-SLAM was developed as a 3D placental surface reconstruction method. However, 3D FLOVA-SLAM uses the feature detection method ORB, which shows some problems when applied on placental data.

To solve the feature related problems in 3D FLOVA-SLAM, the new feature detection method F-net was developed, based on a Convolutional Neural Network (CNN). To create a training set for F-net, stable features were selected, using the information of the 3D reconstruction of 3D FLOVA-SLAM. CNN architecture and hyperparameter optimisation was performed and this information was used to create and train F-net. The performance of F-net was compared to the performance of the original feature detection method ORB. It was shown that F-net detects less features and has a higher prediction time than ORB. However, it was shown that F-net features are more distinct. Furthermore, feature selection before training F-net decreases feature quantity but increases feature distinctiveness even more. The biggest limitations at the moment are the long prediction times and the training of F-net specific for *ex-vivo* data. If these issues are resolved, implementing F-net in 3D FLOVA-SLAM is likely to increase the quality of the placental reconstruction.

Contents

List of Figures

List of Tables

List of Abbreviations

BRIEF	Binary Robust Independent Elementary Features
CNN	Convolutional Neural Network
FAST	Features from Accelerated Segment Test
FLOVA	Fetoscopic Laser Occlusion of Vascular Anastomoses
GAN	Generative Adversarial Network
MAD	Median Absolute Deviation
OF	Original Features
ORB	Oriented FAST and Rotated BRIEF
PPROM	Preterm Pre-labour Rupture of Membranes
RANSAC	Random Sample Consensus
ReLU	Rectified Linear Unit
SF	Selected Features
SIFT	Scale Invariant Feature Transform
SLAM	Simultaneous Localization And Mapping
SLCPV	Sequential Selective Laser Coagulation of Placental Vessels
TAPS	Twin Anaemia Polycythaemia Sequence
TOPS	Twin Oligohydramnios-Polyhydramnnios Sequence
TTTS	Twin-To-Twin Transfusion Syndrome
US	Ultrasound

1 General introduction

1.1 Twin-to-Twin Transfusion Syndrome

Twin pregnancy occurs in around 3% of all pregnancies [1]. A twin pregnancy can either be dichorionic, with both twins having their own placenta, or monochorionic, with both twins sharing the same placenta. Sharing a placenta can lead to specific complications unique for this type of pregnancy. One of these complications is Twin-to-Twin Transfusion Syndrome [2]. In most monochorionic pregnancies, arterial and venous connections called anastomoses are present on the placental surface, but in TTTS these anastomoses lead to an unbalanced blood supply. The result of the unbalanced blood supply is a hypovolaemia in the donor twin and hypervolaemia in the recipient twin, which leads to serious complications for both twins. A schematic overview of TTTS can be seen in figure 1.1. Untreated TTTS has a mortality rate of 90% for both fetuses [2, 3].

Maternal symptoms, such as a rapid growth of the womb or a feeling of abdominal tightness, can be a sign of TTTS, but TTTS is most often diagnosed after a routine ultrasound (US) examination [2]. On US, Twin Oligohydramnios-Polyhydramnios Sequence (TOPS) is the most important clinical sign, presenting as one fetus having an excessive amount of amniotic fluid (polyhydramnios) and the other fetus having little to no amniotic fluid (oligohydramnios) [2, 4]. TTTS is staged by the Quintero classification system, which consists of five stages [4]. An overview of the Quintero stages can be seen in table 1.1. The difference in amniotic fluid is quantified by the maximum vertical pocket and has to be $\leq$ 2 cm for the donor and $\geq$ 8 cm for the recipient to qualify as Quintero stage I at a gestational age of <20 weeks.

There are different methods for treating the symptoms of TTTS, such as draining the excess amniotic fluid of the recipient [5]. However, after the Eurofe-

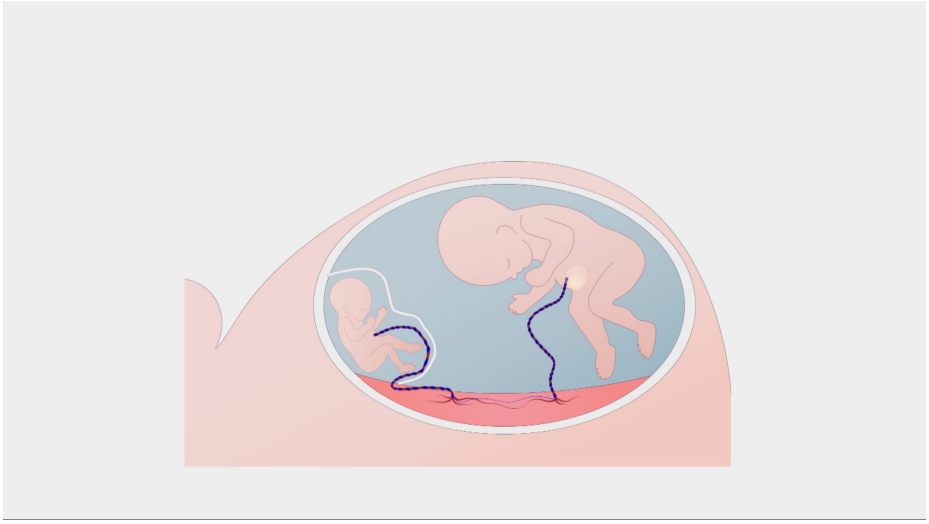


Figure 1.1: A schematic overview of the situation in TTTS. The volume shift caused by the anastomoses present in TTTS leads to several complications. An important clinical sign of TTTS is that there is less amniotic fluid in the donor, which is enveloped by the membrane. Note that the donor twin is wrongly depicted smaller that the recipient, which is often incorrectly associated with TTTS.

tus trial in 2004, Fetoscopic Laser Occlusion of Vascular Anastomoses (FLOVA) is considered the golden standard for treatment of TTTS [6], since it is the only treatment eliminating the cause of TTTS. In the FLOVA procedure, a fetoscope is used to visualise and coagulate the anastomoses on the surface of the placenta, eliminating the unbalanced blood supply between both twins.

1.2 FLOVA procedure

The FLOVA procedure starts with US examination to localize the position of the fetuses, to verify the position of the umbilical cord insertions and to determine the best localisation for the fetoscopic insertion [5]. Then, the fetoscope is inserted under US visualisation. This setup can be seen in figure 1.2. The insertion of the fetoscope marks the beginning of the orientation phase, in which the placental surface is visualised in order to establish the location, number and type of placental anastomoses. Usually, an observer manually draws an overview of the visualised topography. When the orientation phase is completed, the coagulation phase starts. In this stage, the anastomoses are laser coagulated one by one, according to the Sequential Selective Laser Coagulation of Placental Vessels (SLCPV) sequence. In this sequence, first the artery-to-vein anastomoses are coagulated, followed by the vein-to-artery anastomoses and completed with the vein-to-vein and artery-to-artery anastomoses [7]. The SLCPV sequence is used to create better neural outcomes for the recipient by preventing a blood

Table 1.1: Staging of TTTS.

Quintero stage	Symptoms
Stage I	· Difference in amniotic fluid · Normal Doppler measurements · Visible bladder in donor twin
Stage II	· Normal Doppler measurements · Non visible bladder in donor twin
Stage III	· Abnormal Doppler measurements
Stage IV	· Recipient twin with foetal hydrops
Stage V	· Death of one or both twins

pressure overload. After all anastomoses are coagulated, a Solomon procedure is performed, in which the vascular equator is coagulated along the placenta [8]. This ensures a complete separation of the circulation of the twins, including (deep) anastomoses that otherwise could have been missed. The Solomon procedure limits the chance on recurrent TTTS or Twin Anemia Polycythemia Sequence (TAPS). The Solomon procedure completes the FLOVA procedure. During the next days, extensive US follow-up ensure the health of both fetuses.

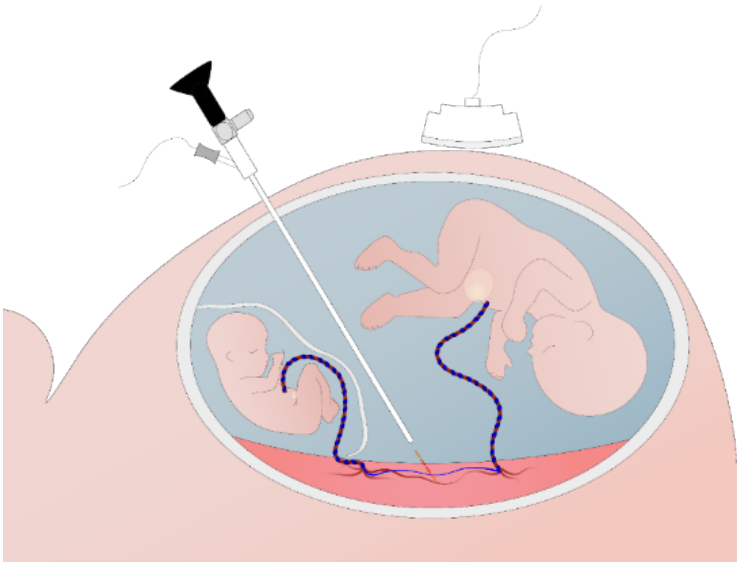


Figure 1.2: The set-up in the FLOVA-procedure. On the left, the donor twin is seen, in a small amniotic sac. On the right, the bigger recipient twin is located. Under ultrasound vision, the fetoscope is introduced into the amniotic sac of the recipient twin and the placental anastomoses are coagulated.

1.3 3D mapping of the placenta

FLOVA is a complex procedure with a limited view in a challenging intrauterine environment, subject to complicating factors like foetal movements and cloudy amniotic fluid. It is essential that the physician gets a clear overview of the placental surface, to prevent missing anastomoses. However, it is important as well to use a fetoscope that is as small as possible to minimise the chance of (Iatrogenic) Preterm Pre-labour Rupture Of Membranes ((i)PPROM) [9]. The balance between increasing the field of view and minimizing the change on PPROM is essential. A real-time placental overview could limit the effects of a small field of view without compromising on the size of the instruments. Using a placental overview during FLOVA can possible lead to shortened surgery times, less complications during and after the procedure and overall improved outcomes for the unborn twins.

Different researchers have attempted to create a placental overview. Most research has focused on 2D mosaicking or image stitching [10–12]. Image stitching combines multiple images to create an image with a wider field of view, creating what is commonly known as a panorama. These image stitching techniques use a planar surface to project the images on. However, Bergen et al. concluded that when the observed surface is a 3D structure or if the camera performs any translational motions, this technique becomes erroneous [13]. Both limiting factors are present in fetoscopic image sequences, suggesting that 2D image stitching is unsuitable for creating a stable placental overview.

Other attempts were made to use additional modalities, such as MRI, or (optic or electromagnetic) tracking to, create a 3D model of the placenta [14, 15]. These solutions require extra hardware, making practical implementation undesirable. This evokes the call for 3D mapping software techniques using only fetoscopic images with no additional hardware.

In different applications of computer vision, Simultaneous Localisation And Mapping (SLAM) is used to create reconstructions of 3D environments [16]. In SLAM, a 3D map is created and constantly updated (mapping) while the position of the camera is tracked (localisation). SLAM has been used in several medical applications, for example in minimal invasive surgery [17]. Researchers of the Radboudumc have used the work of Mur-Artal et al. to create a feature-based visual SLAM algorithm [18]. In the Mur-artal framework, recognizable pixel patterns, called features, are used for mapping of the 3D enviromnemnt and tracking of the camera position. The features are extracted by the Oriented FAST and Rotated BRIEF (ORB) algorithm, which gave the framework of Mur-artal the name ORB-SLAM. The adjusted ORB-SLAM framework for application in FLOVA was called 3D FLOVA-SLAM [19].

1.4 3D FLOVA-SLAM

The 3D FLOVA-SLAM framework was developed to create a textured 3D point cloud of the placental surface. The framework consists of the following five steps:

1) **Camera calibration**: Camera calibration is used to obtain the camera parameters needed for the rest of the steps. Because these parameters are dependent on the settings of the fetoscope, camera calibration has to be performed before each procedure.

2) **Image acquisition**: During the FLOVA-procedure, real-time video footage of the placenta is recorded.

3) **Feature detection**: On each frame, obtained in the previous step, ORB-features are detected.

4) **3D mapping**: The detected features are matched to create a 3D point cloud of the surface of the placenta. This 3D cloud is updated throughout the whole procedure.

5) **Texture reconstruction**: The recorded fetoscopic images are used to create a textured 3D map of the placenta.

A visual overview of the different steps in the 3D FLOVA-SLAM framework can be seen in figure 1.3.

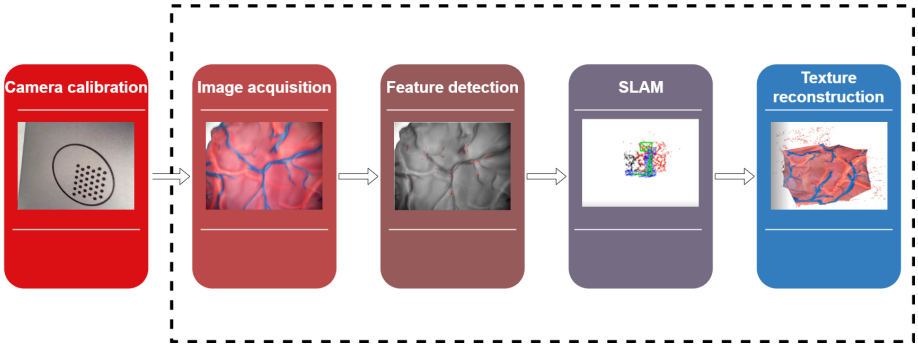


Figure 1.3: The 3D FLOVA-SLAM framework. The first step is the camera calibration. This is performed once at the start of the procedure. The last four steps, from image acquisition to feature detection are repeated throughout the process.

1.5 Problem description

The current 3D FLOVA-SLAM suffers from certain limitations, resulting from the application in a setting different from which it was developed for. ORB-SLAM was developed for non-medical, outdoor scenes. Because the nature of placental mapping is very different to this, errors in the placental mapping often occur, of which an example can be seen in figure 1.4. It can be observed that 3D points are located on a plane not part of the placental surface, which makes the final placental reconstruction interrupted and spike-like. This problem has to be resolved before 3D FLOVA-SLAM can reliable be applied *in-vivo*.

The incorrectly mapped 3D points can be the result of two different causes. First, unstable features can be detected, being not unique or not locatable. If this detected feature is matched with another feature on a different frame, the cosponsoring 3D point mapping is incorrect. An example of this are features detected on edges, which are not unique enough to be matched correctly on a different frame. Although ORB-SLAM removes a feature if it is not located on a number of subsequent frames, incorrect matched still can lead to incorrectly mapped 3D points. Second, features can be detected which are not located on the placenta, for example on flakes in the amniotic fluid, the fetus or the membranous equator. If these features are located in different frames, this will result in a 3D point which is originally not a part of the placental surface.

Finding a method that detects only stable features belonging to the placenta will likely increase the quality of the placental reconstruction. Implementation of such a method would bring 3D FLOVA-SLAM yet another step closer to clinical application.

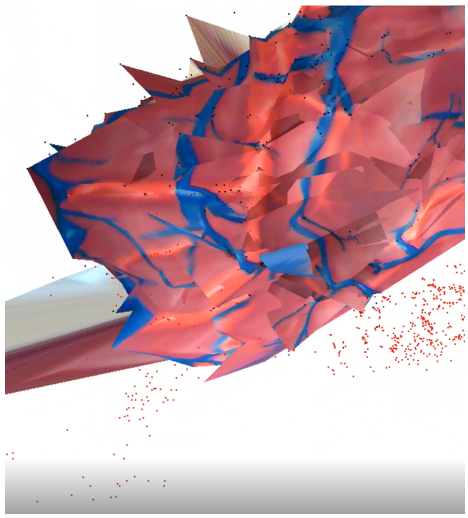


Figure 1.4: Incorrect placenta reconstruction of a 3D placental phantom, caused by incorrectly mapped features.

1.6 Feature detection and description

A feature is a point in the image containing useful information about the contents of the image. To go from an image to a feature which can be used in further application, two steps are required. First, a potential feature has to be located in the image, a process called feature detection. Second, feature description is to provide the feature with its own unique signature, which is used to identify the same feature in other frames. After this step, the features can be used for further processing, for example in feature matching.

As said before, feature detection determines which (set of) pixels in the image meet the requirements for a feature. Feature extraction is generally based on edge, corner or blob detection. Some examples of feature detection methods are SIFT, SURF and BRISK [20]. After a feature is detected, feature description is used to uniquely characterize the feature. Most common feature extraction methods such as SIFT contain both a feature detection as well as a feature description algorithm [21].

A described feature can be used for further processing. An important algorithm using described features is feature matching. In feature matching, the same feature is localised on multiple frames. Among other things, matched features can be used to calculate transformation between images, e.g. for stitching overlapping images.

3D FLOVA-SLAM uses ORB for both feature detection and feature description. In figure 1.5, an example of ORB feature detection can be seen. As feature detection part, the corner detection method Features from Accelerated Segment Test (FAST) is used. FAST extracts pixels in a circle around the pixel of interest and compares their intensities with the intensity of the pixel of interest. If 9 or more contiguous pixels have an intensity difference larger than a certain threshold, the pixel is considered a corner. This is visualised in figure 1.6. Because FAST has a low computational time, it is especially suitable for the real-time application in ORB-SLAM. In ORB, FAST is elaborated with an oriented and scaling component, hence the name Oriented FAST.

The feature description method used by ORB is Rotated Binary Robust Independent Elementary Features (BRIEF). BRIEF transforms a random patch around the feature pixel and converts this patch in a binary string by comparing random pairs of pixels. Originally, BRIEF is not invariant for rotation. Therefore, ORB uses rotated BRIEF, which adds invariance to rotation.

1.7 Possible solutions

A possible solution to limit the influence of the incorrect mapped features is to replace ORB with another feature detection method with the aim that this method does detect less incorrect features. Over the years, different handcrafted feature detection methods have been developed [23]. One example is Binary Robust Invariant Scalable Keypoints (BRISK), which has a higher accuracy and a higher invariance for rotation than ORB [24, 25]. However, these handcrafted

Figure 1.5: ORB feature detection. Every red dot is a detected ORB feature. It can be seen that ORB features are mostly located around corners with a large intensity difference.

feature detection methods are based on rigid algorithms, resulting in a lack of flexibility to learn what a 'good' feature for a specific application is. It would be much more useful to develop a feature detection method which has the flexibility to learn to detect application specific features.

Machine learning has the flexibility to 'learn different behaviours, when provided with labelled input in a structured way. This suggests that machine learning would be able to learn to detect only 'correct' features, provided that the application is trained with a clean dataset containing only these 'correct' features. A machine learning method especially suitable for such an application are Convolutional Neural Networks (CNNs), because CNNs are especially suitable for image related problems. CNNs are being widely used in classification and segmentation problems, but has been used for feature detection and matching problems as well. Developing a 3D FLOVA-SLAM specific CNN based feature detection method therefore is a possible solution for the feature related problems in 3D FLOVA-SLAM worth exploring.

1.8 Convolutional Neural Networks

CNNs are a (usually) supervised machine learning algorithm especially suitable for the use in image applications [26]. This type of neural network has been used in image classification, semantic segmentation and many more computer vision related challenges. CNNs consist, just like any artificial neural network, of different layers of compunctions, connected by activation functions. Each layer holds a set of parameters called, weights (multiplications to input) and biases (additions to input). These weights and biases can be trained, when the network is provided with labelled training examples, to map the input of the network to the required output.

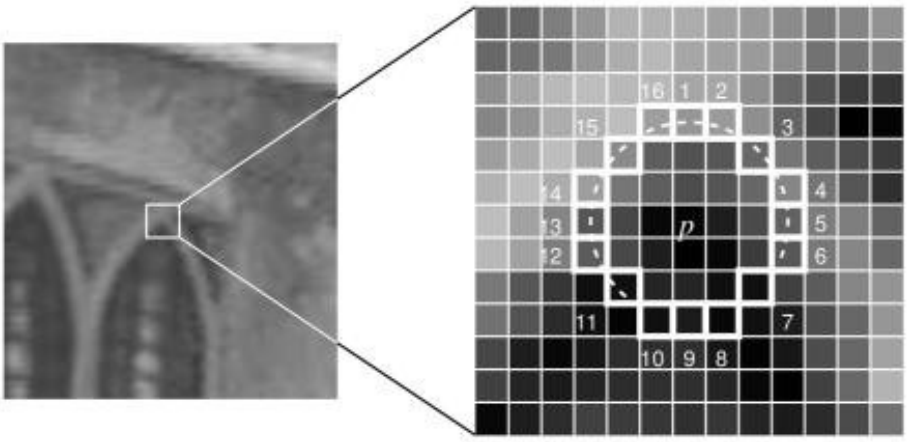


Figure 1.6: FAST feature detection. P is the pixel of interest. Around this pixel, a circle of interest pixels is used to determine if P is part of a corner [22].

The fundamental layer of a CNN is the convolutional layer. These convolutional layers are used to extract visual patterns from images by convolving parts of the image with kernels to extract certain features. This convolution operation can be seen in figure 1.7. By combining convolutional layers, different complex features can be learned. The lower layers extract local features such as edges and corners. The higher layers combine multiple low level features to be able to learn complex patterns in images.

Other important layers in a CNN are [26]:

- **Activation layer**: CNNs need non-linear activations to connect different layers. ReLU is one of the most used activation layers. Usually, the last layer of a network uses a different activation: the softmax activation.
- **Pooling layer**: this layer is used to decrease the number of features. This minimizes the chances of overfitting the network.
- **Drop-out layer**: A drop-out layer randomly deactivates weights during backpropagation. By adding this randomness, changes of overfitting decrease.

A CNN can learn requested behaviour providing the network with labelled example images. The term 'labelled image' means an image provided with the desired output of the CNN. This label can be assigned for the full image, for example the class 'dog' to indicate that a dog is present in the image. Class labels can also be applied to pixels in the image. For example, a pixel can be labelled 'grass', indicating that the pixel belongs to an area with grass. This is typically called semantic segmentation. Other types of labelling include numbers, either assigned to the whole image or individual pixels. A CNN learns by minimizing the difference between the output and the labels (the cost) in a

process called back-propagation. In back-propagation, the weights and biasses of the layers are adjusted by gradient descent to minimize this cost. Multiple passes of data are applied while updating the weights to minimize the cost. The learning behaviour of the CNN is controlled by the hyperparameters. The most important hyperparameters are discussed below.

- **Learning rate**: the learning rate controls the amount of which the weights of the CNN are changed during learning. This is one of the most important hyperparameters to optimize: a learning rate to too small will converge really slow into the global minimum when training, but a learning rate that is too high will likely not converge to the local minimum at all. This is visualised in figure 1.8.
- **Mini-batch size**: the mini-batch size determines how much data will be used in one pass of the network. A mini-batch size of one image is commonly called 'online learning'.
- **Number of epochs**: one epoch is the pass of all data through the network once. If the number of epochs is larger than one, the network will see all data more than once.

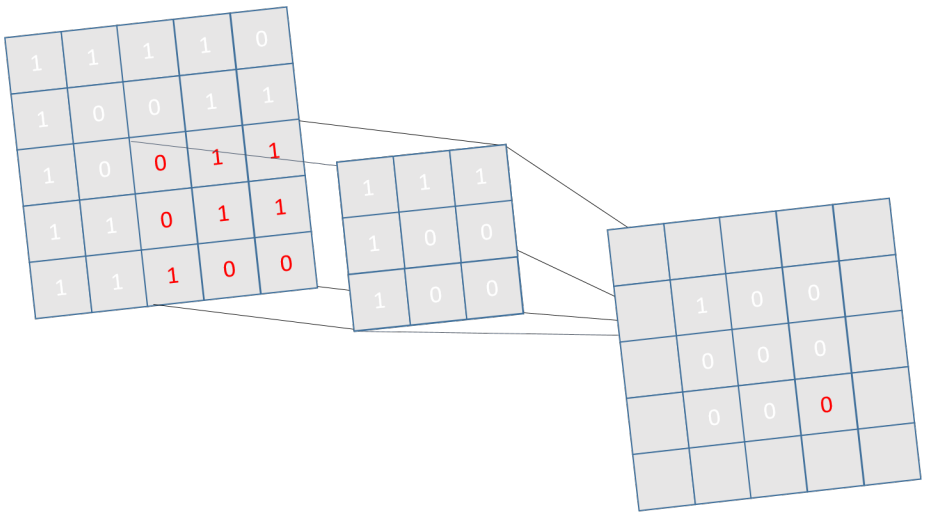


Figure 1.7: The convolutional operation. On the left,the pixel values of a binary input image are depicted. In the middle, a 3×3 kernel is located. When doing a convolution on a certain pixel, the value of that pixel is replaced by the result of that convolution operation in the output image.

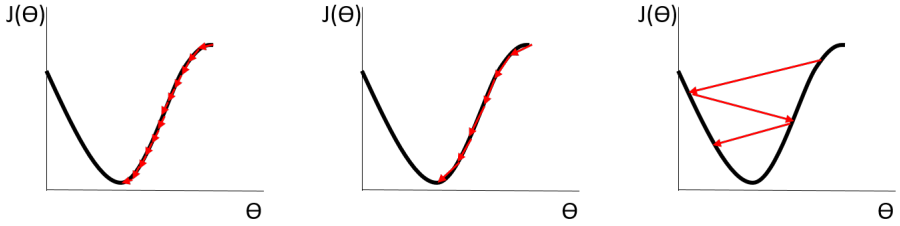


Figure 1.8: The learning rate in a 2D example of gradient descent. The learning rate is depicted by the length of the arrows. The aim of gradient descent is to find the value of $\theta$for which $J(\theta)$ is minimal. If the learning rate is too low, many steps of gradient descent are required to find the minimum. If the learning rate is to high, it is likely that gradient descent overshoots the minimum. Only at a optimal learning rate the minimum is found at a decent pace.

1.9 CNNs in feature applications

CNNs have been applied in feature related problems before, for all steps in the pipeline of feature detection, feature description and feature matching. For example, Altwaijry et al. trained a CNN to do both feature detection as feature matching [27]. To do so, a large annotated dataset from outdoor scenes was used. What most CNN based feature applications have in common is the use of extensive, annotated datasets of outdoor scenery [28–30]. This makes application in 3D FLOVA-SLAM unfeasible, but this research shows that CNNs can be successfully applied in feature related application.

The work of Gaisser et al. did focus on placental application [31]. In this research, a CNN was used to improve the feature detection and matching for 2D image stitching. As training examples manually annotated bounding boxes on the centre of placental vessels were used. A Siamese CNN was trained on these annotated images, which was able to do both feature extraction as feature matching. Although the results of Gaisser et al. are promising in 2D image stitching, the developed method cannot be used in 3D FLOVA-SLAM because 3D FLOVA-SLAM uses point features, which makes the box approach inapplicable. Furthermore, only vessels were used to extract features from, which would make creating an overview on part of the placenta without vessels impossible. Moreover, Gaisser used extensive manual annotation of features, which makes labelling labour intensive and user dependent. Altough the work of Gaisser is not a solution for the problems in 3D FLOVA-SLAM, it is an inspiring example of using a CNN to improve placental mapping in FLOVA.

1.10 Research objectives

The aim of this research is to create a CNN based feature detection method which detects stable features specific for 3D FLOVA-SLAM. This leads to the

following research question:
What is the added value of using a CNN based feature detection method to substitute ORB in 3D FLOVA-SLAM?

To answer this question, three sub-questions were formulated:

1. What are stable features specific for the use in 3D FLOVA-SLAM?
2. How can an existing CNN architecture be adjusted and trained to be able to function as a feature detection method?
3. What is the *ex-vivo* performance of the new feature detection method in comparison with ORB?

To ensure that the newly developed CNN feature detection method can be applied in 3D FLOVA-SLAM, the following requirements were considered throughout the development:

- Comparable or increased stability of detected features.
- No manual labelling of features required.
- Process times short enough to enable real-time application.
- Detects point features.
- Is translatable to C++ to work in the existing framework.

1.11 Overview of thesis

In Chapter 2, sub-question 1 will be answered and the creation of a dataset containing stable 3D FLOVA-SLAM features will be discussed. Chapter 3 answers sub-question 2, in the search for a CNN architecture capable of detecting features. In Chapter 4, the newly developed feature detection method called F-net will be introduced and the performance compared with the current feature detection method ORB. Chapter 5 concludes this thesis by answering the research question and discussing the implications of this answer for application in 3D FLOVA-SLAM.

2 Identifying stable features for data labelling

2.1 Introduction

In order to train a CNN feature detection method with the aim of improving the detected feature stability, it is essential to first analyse what stable features are. Salahat et al. described the following general properties for a stable feature detection [32]:

- **Distinctiveness**: The pixel pattern of a detected feature should be distinguishing enough be able to perform matching.
- **Locality**: A feature should be local, to prevent occlusion of part of the feature by another structure. This would decrease the chances of a feature being detected.
- **Quantity**: The number of detected features needs to be high enough for the specific application.
- **Accuracy**: A feature should be detected accurate with respect to its position independent of the location, scale and shape of the feature.
- **Efficiency**: A feature detection method should be fast enough to be used in the (real-time) application.

- **Repeatability**: A feature should be able to be detected in different frames under different circumstances. The most important properties for repeatability are scale and rotation invariance and robustness to noise.

As suggested by the definitions of Salahat et al., different applications can be sensitive for different feature qualities [32]. In the previous chapter it was explained that incorrect mapped 3D points in 3D FLOVA-SLAM are most likely the result of ORB features being not distinct and repeatable enough. A differentiation between stable and non stable 3D FLOVA-SLAM features can be made by making a selection between correct and incorrect 3D FLOVA-features. The goal of this chapter is to identify the stable 3D FLOVA-SLAM features by identifying inlier points in the 3D point cloud.

2.2 Methods

The goal of this chapter is to create multiple datasets to train a CNN based feature detection method on. Two types of datasets were created: with the original detected 3D FLOVA-SLAM features and with selected stable 3D FLOVA-SLAM features. To select stable features, the assumption is made that all 3D point cloud outliers are the result of unstable detected ORB features. By selecting the 3D outliers, the unstable features can be detected in the images. The feature outlier detection was based on the assumption that the 3D points are located on a placenta like surface. By using a mean outlier filter the inlier 3D points were identified and coupled with the corresponding features, after which a dataset was created containing these stable features.

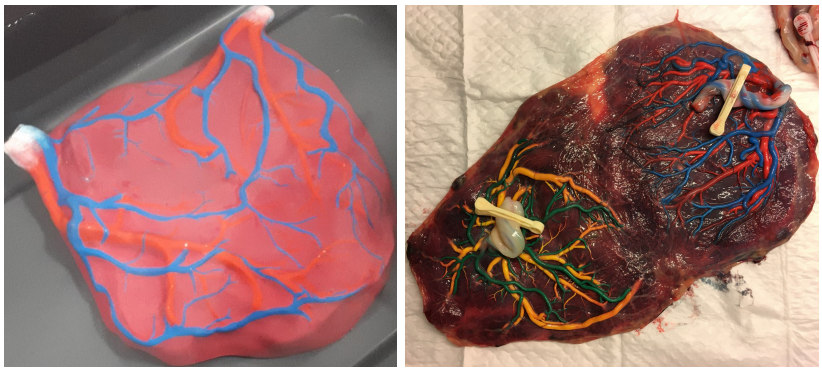
2.2.1 Data recording

To obtain data, the 3D FLOVA-SLAM application was used to make 3D point clouds. The OTG&Micro USB Inspection Borescope Camera (resolution 640×480, focal distance 3-8 cm) was used the data recording. Data recording was performed for both a phantom placenta as well as a *ex-vivo* placenta. The calibration parameters needed for 3D FLOVA-SLAM were obtained by calibrating with the Endocal method in air [33].

Phantom placenta: First, data was obtained with a 3D phantom of a placenta, which can be seen in figure 2.1a. This phantom was created by creating a 3D scan of a dye-injected placenta, after which the model was 3D printed and painted in three different colours. During data recording, the phantom was placed on a flat surface. A 3D point cloud was created by slowly scanning the placental surface, while observing the created 3D point cloud. Data recording was stopped when the whole surface of the placenta was covered. The phantom dataset was created with the aim of obtaining a simple dataset with big contrast differences, little detail and little specular reflections.

***Ex-vivo* placenta:**. Second, a dye injected twin placenta was used to create an *ex-vivo* dataset. The placenta which was used can be seen in figure 2.1b. Dye

injection of the placenta was performed with water soluble paint in realistic colours to enlighten the veins and arteries to compensate for the loss of blood in the placental vessels [34]. Upon data recording, the placenta was placed on a flat surface. The light of the endoscope and light of the room was dimmed and an OK light was placed pointing slightly behind the placenta, to limit the reflections on the placental surface. The aim of creating the *ex-vivo* dataset was to create a dataset approximating the *in-vivo* images.



(a) The 3D placenta phantom. *(b) Dye injected placenta.*

Figure 2.1: Objects used for data recording. Note that this is a recording of the dye injected placenta before dye injection with realistic colours.

The results of each scan were saved for further processing. The saved output of 3D FLOVA-SLAM consisted of all keyframe images and an output table with the properties of all points of the 3D point cloud. From each 3D point, the name of the image it was detected on, the x- and y-coordinates of the feature in this image and the X- Y-, and Z-coordinates of the point in the 3D cloud was saved. An example of the output of 3D FLOVA-SLAM can be seen in table 2.1.

Table 2.1: Output from 3D FLOVA-SLAM.

	2D Image			3D-Point Cloud		
Feature	Image Name	x-coordinate	y-coordinate	X-coordinate	Y-coordinate	Z-coordinate
1_1	Image-1	245	277	0.16	0.08	0.99
2_1	Image-1	254	136	0.16	0.12	1.02
...	...	...	...	...	...	...
n_1	Image-1	210	160	0.13	0.08	0.04
1_2	Image-2	113	217	0.34	0.12	0.07
...	...	...	...	...	...	...
m_2	Image-2	270	276	0.77	0.30	0.13
...	...	...	...	...	...	...
k_i	Image-i	222	106	0.07	0.12	0.01

2.2.2 Outlier selection

In MATLAB R2019a the 3D point cloud was recreated by plotting the coordinates of the 3D points as saved in the output table. To differentiate between

in- and outlier 3D points, a mean outlier filter was used on the point cloud. The optimal mean absolute deviation (MAD) threshold was tested by using a threshold of one to six and visualisation of the result. The choice of threshold was based on the trade-off between removing all the outlier points and preserving the structural placenta shape. A threshold of four was used in the next steps.

2.2.3 Data labelling

For the phantom model, one labelled dataset containing original features was made. For the *ex-vivo* placenta, two different datasets were created.

Original features: As said before, the properties of the corresponding feature were known for each 3D point. The feature properties consisted of the name of the image it was located on and the 2D coordinates of the detected feature on that image. Multiple features shared the same origin image, so the first step for creating a dataset was extracting all unique image names. Second, all features corresponding to this image were identified. Next, the 2D coordinates of these features were binary encoded in an image with the same. This resulted in an image with 0 for a non feature pixel and 1 for a feature pixel. Lastly, both the original image as the label image saved together to create a feature labelled dataset.

Selected features: Next, a dataset was created containing only stable 3D FLOVA-SLAM features. To do so, the features belonging to outlier 3D points were removed from the feature set before performing data labelling the way described above. This resulted in a dataset containing labelled with only stable features.

In total, three datasets were created:

1) **Phantom: original features**: Dataset created on a placental phantom, with labelling containing all 3D FLOVA-SLAM features.

2) ***Ex-vivo*: original features**: Dataset created on a dye injected *ex-vivo* placenta, with labelling containing all 3D FLOVA-SLAM features.

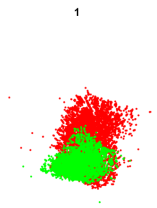
3) ***Ex-vivo*: selected features**: Dataset created on a dye injected *ex-vivo* placenta, with labelling containing only the features selected as inlier 3D points.

2.3 Results

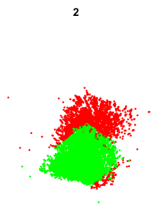
In figure 2.2, the resulting point clouds of the 3D FLOVA-SLAM scans can be seen. It can be observed that both the placental phantom as the *ex-vivo* placenta have the structural form of the scanned object, but do contain outlier points. In figure 2.3, the visualisations for different filtering MAD threshold factors can be seen. An increasing threshold factor indicates less 3D point as outlier points. The statistics of the three datasets can be seen in table 2.2. The phantom dataset contains 1253 images, with a total of 341643 features. This is a mean number of features of 272 per image. For the *ex-vivo* dataset, the mean number of features was 349 on 1341 images, which is a total of 469040 features. The percentage of outlier 3D points was 0.062%, which totals for a mean of 321 features on each image of the dataset containing selected features. An example of the labelling images for both the phantom as the *ex-vivo* placenta can be seen in figure 2.4.

(a) Placental phantom. *(b) Ex-vivo placenta.*

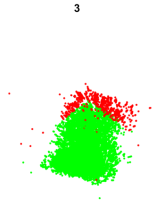
Figure 2.2: Point clouds of the placenta phantom and the ex-vivo placenta.



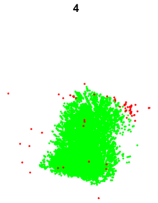
(a) Threshold 1.



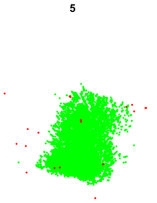
(b) Threshold 2.



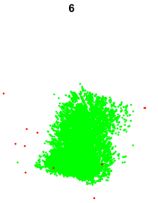
(c) Threshold 3.



(d) Threshold 4.



(e) Threshold 5.

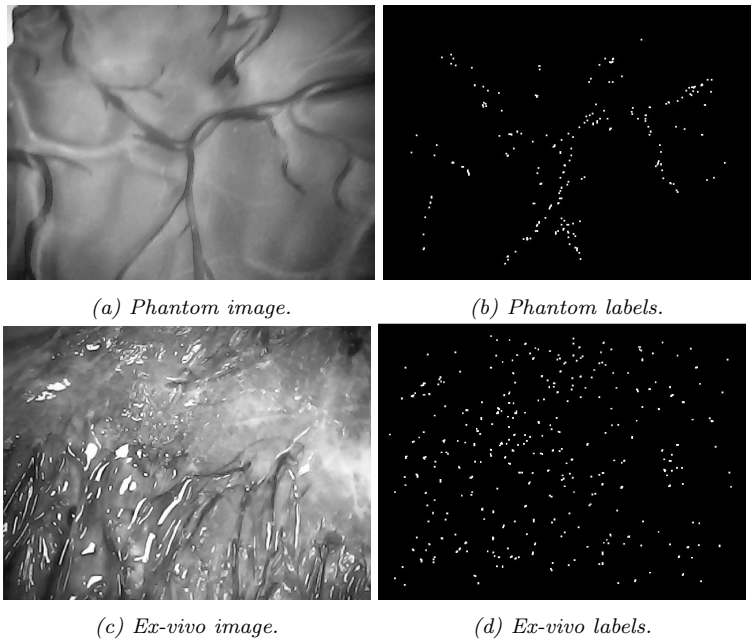


(f) Threshold 6.

Figure 2.3: Filtering with different MAD threshold values. Inlier points are depicted in green, outier points are dipicted in red.

Table 2.2: Description of dataset. The features corresponding with outlier 3D points were removed from the original dataset to create the selected dataset.

	Images	Features	Mean per image	Outlier features	% outlier features
Phantom-original	1253	341643	272	20586	6.0%
***Ex-vivo* original**	1341	469040	349	294	0.06%
***Ex-vivo* selected**	1341	468746	321	-	-



(a) Phantom image. *(b) Phantom labels.*

(c) Ex-vivo image. *(d) Ex-vivo labels.*

Figure 2.4: Example of images and corresponding labels. Note that in this visualisation the features have a width of three pixels instead of one pixel in the original labels.

2.4 Discussion

The goal of this chapter was to create datasets for training a new feature detection method specific for 3D FLOVA-SLAM on. First,two datasets with the original ORB features as extracted by 3D FLOVA-SLAM was created, one with phantom placenta images and one with *ex-vivo* placenta images. Next, a dataset containing only stable features was created from the *ex-vivo* scan. To do so, the assumption was made that all outlier 3D point in the placental reconstruction were the result of a detected unstable feature. To determine outlier 3D points,

a mean outlier filter was used and the features cosponsoring to the outlier 3D points were removed from the set of features.The resulting dataset of the placental phantom will be used in Chapter 3 to develop a CNN based feature detection. Feature detection trained on both the original as the selected features of the *ex-vivo* dataset will be compared in Chapter 4 to determine the added value of training a CNN feature detection method on selected features.

In figure 2.2, it can be seen that the 3D reconstructions indeed do have a lot of outlier points, stressing the importance of a new feature detection method. It is noticeable that the placental phantom point cloud shows more outlier 3D points. This is likely the result of the lack of detail in on the placental image, which probably leads to less distinctive features being detected. The difference in detail for the phantom and *ex-vivo* placenta is really clear in figure 2.4, in which a labelled example for both the datasets can be seen. However, it can be seen that in the image from the phantom dataset the features are almost exclusively detected along the edges of the vessels, which makes the uniformity of the features in the phantom dataset bigger. The uniformity of the features along with the simplicity of the images lead to the choice of developing the CNN feature detector initially on the phantom dataset.

In figure 2.3, the different MAD thresholds can be seen. It can be seen that increasing this threshold decreases the number of detected outlier 3D points. In subfigure 2.3a, a big part of the points on the placental surface are considered outlier points. In subfigure 2.3b, a bigger part of the placental 3D points are depicted as inlier points, but still the structural integerity of the placenta is not intact any more. Using a threshold of three only depicts a border at one side of the placenta as outlier points. For threshold four, almost all points of the placenta are considered inlier points and most outlier points are removed as well, in contrast to a threshold of five, in which the whole placenta is preserved but 3D points not located on the placental plane are considered inlier 3D points. In subfigure 2.3f, even more points which are not located on the placental surface are considered inlier points. A threshold of four was chosen for further application because of the balance between removing most outlier 3D points while preserving the structural integrity of the placental plane.

The current way of 3D outlier selection showed some limitations. The biggest limitation was that not all outliers were removed, even at a MAD threshold of one. This is a result of the thin but stretched shape of the placenta, which influences the behaviour of the mean outlier filter. If the current selection of the 3D points and features do not lead to the desired effect after training the CNN feature detection with, another 3D point selection method has to be implemented. Manual selection of all outlier 3D points might be an option, but this is time consuming and likely user dependent. Another option would be to use a ground truth for the 3D reconstruction of the placenta and registering this on the 3D point cloud as outputted by 3D FLOVA-SLAM. Outliers would be all 3D points located deviant to the 3D ground truth. Because the phantom placenta was created using 3D printing, such 3D ground truth was available, as can be seen in figure 2.5.

At the moment, the quality of the outlier selection was not quantified. Man-

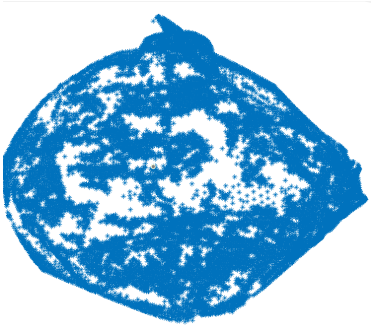


Figure 2.5: 3D point cloud the used for making the placenta phantom.

ual selection or ground truth 3D information can be used to validate the quality of the 3D point selection. The selection of features following the outlier selection of 3D point was made on the assumption that 3D outlier points are the result of unstable features. This assumption will be validated in Chapter 4, in which a comparison between a CNN based feature detection trained on original and selected features will be made to determine the stability of the features as detected by both methods.

A last important note to made is that a CNN requires enough training examples to learn a certain behaviour. The current feature selection only discard a features on a small amount of the images. If the hypothesised effect of feature selection for the training of the CNN is not present, it is recommended to select training examples where at least one unstable feature was removed. It might be needed to merge datasets of different 3D point clouds to obtain enough training images when using this method.

For application in a CNN, it is important that enough training information is present. However, only small numbers of features were removed, which might not be enough information for each type of unstable feature. For a future application, it is recommended to select training examples where at least one unstable feature was removed. If this will lead to a not sufficient number of training images, multiple point clouds have to be merged.

3 CNN as feature detection method

3.1 Introduction

This chapter will discuss the implementation of a CNN based feature detection. To profit from already existing knowledge, it was chosen to base this method on an already established CNN architecture. Because feature detection is close to semantic segmentation, U-net was chosen to base the feature detection method on [35]. Another argument to chose for U-net is that U-net was developed for application in biomedical images. U-net is named after its distinct shape with a downsampling arm, followed by a bottleneck structure and a final upsampling arm. The shape of U-net can be seen in figure 3.1. The downsampling steps consist of pairs of convolutional layers, followed by max-pooling steps. The downsampling is followed by a bottleneck consisting of multiple convolutional layers. The resulting feature map is upsampled until it results in the output segmentation map. The unique feature of U-net are the horizontal skips between the down- and upsampling arms, depicted in grey in figure 3.1. These connections combine the deep, low detailed feature maps with the shallow, high detailed feature maps, to create a more detailed result.

The development of the U-net based feature detection was based on the phantom dataset with the original 3D FLOVA-SLAM features for two reasons. First, using this simple dataset will create the most simplified situation so it could be proved that U-net is indeed capable of functioning as a feature detection. Second, because the dataset is cleaner, the output is easier to correlate with the input images, which makes the effect of optimisation changes more

insightful.

Three steps of developing the U-net based feature detection method will be discussed. First, the adjustments needed to make U-net capable of detecting point features will be described. Second, optimisation to find the best adjusted architecture of U-net is discussed. Lastly, hyperparameter optimisation of the feature detection method is reviewed.

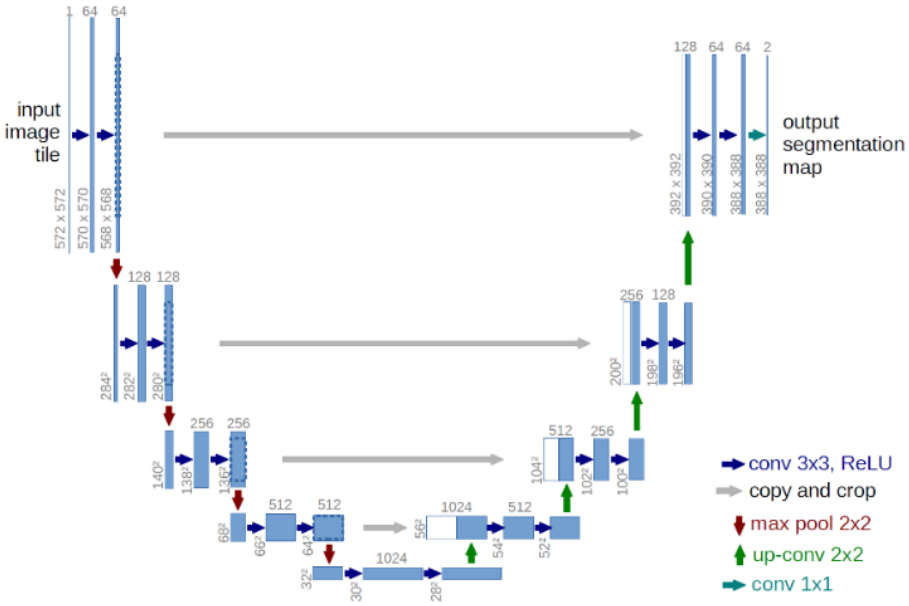
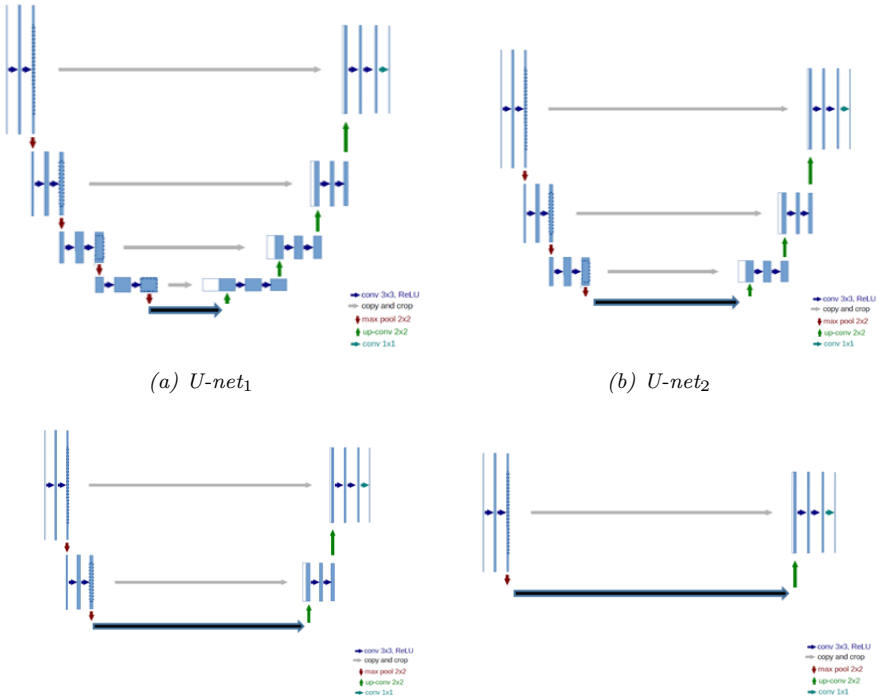


Figure 3.1: The architecture of U-net. In darker blue, the 3×3 main convolutional operations are depicted. In lighter blue, the 1×1 convolutional step is displayed. Horizontal skips can be seen in grey. The max-pooling layers are visualised as red arrows. The up convolutional steps are depicted in green. Furthermore, the number of channels is displayed on top of each operation. [35].



(a) U-net$_1$ *(b) U-net$_2$*

(c) U-net$_3$ *(d) U-net$_4$*

Figure 3.2: The different adjusted versions of U-net. From the top left, one convolutional block at the time was removed. Input and output sizes remained the same between versions.

3.2 Methods

The U-net based feature detection method was implemented in Python 3.7 using the Keras framework [36]. A non pre-trained version of U-net was used. The feature detection method was based on a non pre-trained version of U-net. The original phantom dataset was used to train and test in the optimisation process. From this dataset, a smaller dataset of 500 images was extracted. A random split of 0.02 with a fixed seed was used to obtain a reproducible test set of 10 images.

3.2.1 U-net as feature detection method

Before implementation of U-net as a feature detection method, three adjustments were made to U-net. First, the input size was changed from 512×512 to 640×480 to match the size of the images of the dataset. Second, the original implementation of U-net used no padding in the upsampling steps, leading to an output segmentation map of 128x128. However, to be implemented as a feature detection method, the output of U-net is required to be the same size as the input images. Padding was added in the convolutional steps to solve this. Lastly, the Rectified Linear Unit (ReLU) activation layer was replaced by the the state-of-the-art leaky ReLU activation [37]. This was done in order to prevent the situation of 'dying ReLU', in which ReLU outputs zero when provided with a negative input, a problem which can stop back propagation entirely. Using leaky ReLU, which outputs a slightly negative value instead of zero at a negative input value, prevents this problem.

3.2.2 Architecture optimisation

As said before, U-net is developed for the complex task of semantic segmentation. Spatial relationships are of uttermost importance in segmentation tasks. However, for feature detection, only local information is of importance. Therefore, the number of layers of U-net could be decreased. To find the optimum depth, layers of U-net were removed creating four different versions of U-net. The original U-net architecture was called U-net$_0$. Then, for U-net$_1$ to U-net$_4$ one convolutional block per step was removed. The resulting architectures can be seen in figure 3.2.

To find the optimum architecture U-net architecture, each version of U-net was trained on 490 images of the phantom dataset, with Adam as optimizer with a learning rate of 0.0001, batch size of 1 and the default value for the rest of the parameters ($\beta 1 = 0.900$, $\beta 2 = 0.999$ and learning rate decay $= 1 \times 10^{-8}$ [38]. Training was performed for 1 epoch, with a validation split of 0.2 and a mini-batch size of 1. The following list of parameters was obtained for each trained version of U-net:

- Binary accuracy.
- Binary cross entropy loss.

- Binary validation accuracy.
- Binary cross entropy loss.
- Mean prediction time of one image.
- Ten thresholded output images.

The optimal U-net architecture was based on the outcome parameters as described above. The binary accuracy and binary validation accuracy were used as a general parameter regarding the quality of training of the CNN. The binary cross entropy loss was used to quantify the classification error between the predicted images and the labelled images. Both binary accuracy as binary cross entropy loss were used obtain a general view on the influence of the adjusted architectures on the training process. However, earlier test showed that these parameters were somewhat of limited value for assessing the quality of the resulting feature detection method, as will be explained in the discussion. Therefore, the output on ten test images were saved and analysed as well. On the visual inspection of the output images, the number, location and separation of the point features was considered.

3.2.3 Hyperparameter optimisation

With U-net$_3$, which was chosen as the optimal architecture in the previous section, two steps of hyperparameter optimisation were performed. First, U-net was trained with a grid of combinations of the learning rate and the mini-batch size. These hyperparameters were chosen because the expected large effect on the training outcomes. For the learning rate, a range of 1×10^{-1} to 1×10^{-5} was tested, decreasing a factor 10 with each step. The mini-batch size was tested in a range of 1-5 with a step size of 1. For each training of the U-net, the same outcome parameters as in the architecture optimisation step were analysed.

After the optimal combination between the learning rate and mini-batch size was determined, an optimalisation test was performed to find the best number of training epochs. This experiment was conducted to find the optimal balance between obtaining optimal training results while preventing overfitting. This test was performed by training the network with a learning rate of 1×10^{-4} and a mini-batch size of 1 for an epoch range of 1-10. Again, the same outcome parameters were used to quantify the U-net training.

3.3 Results

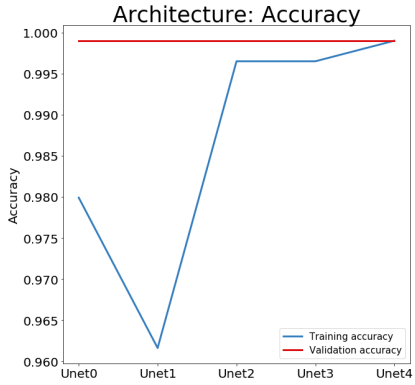
3.3.1 Architecture optimisation

The results of the architecture optimisation can be seen in figure 3.3. In figure 3.3a, the training and validation accuracy are visualised for the different U-net versions. The training accuracy is lower than the validation accuracy. Furthermore, the training accuracy is the lowest after training U-net$_1$ and highest in

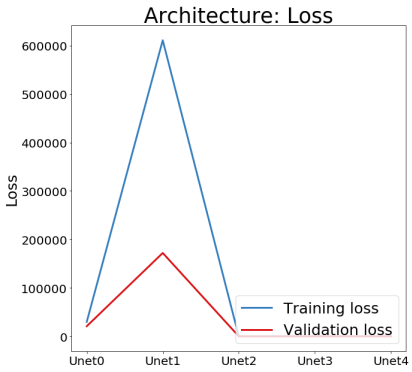
U-net$_4$. In figure 3.3b, the training and validation loss can be observed. Because of the large difference in values, the exact values for both the losses is depicted in table 3.1. The training loss is the lowest for U-net$_4$, the validation loss for U-net$_3$. In figure 3.3c, it can be seen that prediction times decrease with decreasing the depth of U-net. Examples of the output for the different U-net architectures can be seen in figure 3.4. It can be seen that for U-net$_0$ to U-net$_3$ almost no pixels were classified as feature. For U-net$_3$ and U-net$_4$ detected features mostly can be seen in the middle lower part of the image.

Table 3.1: U-net architecture optimisation.

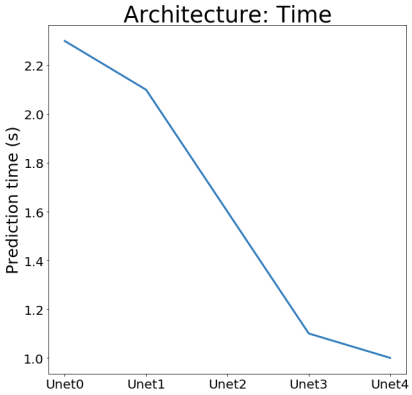
Name	Accuracy	Loss	Validation accuracy	Validation loss
U-net$_0$	0.98	2.97×10^{4}	0.99	2.08×10^{4}
U-net$_1$	0.96	6.11×10^{5}	0.99	1.72×10^{5}
U-net$_2$	0.99	1.42×10^{-2}	0.99	6.70×10^{-3}
U-net$_3$	0.99	1.28×10^{-2}	0.99	6.30×10^{-3}
U-net$_4$	0.99	1.19×10^{-2}	0.99	6.40×10^{-3}



(a) Training and validation accuracy.

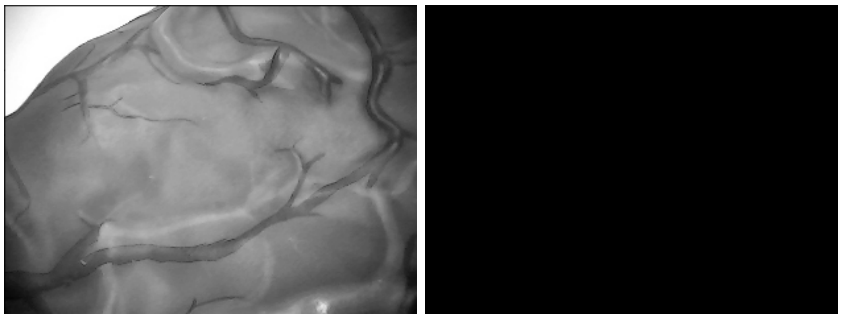


(b) Training and validation loss.

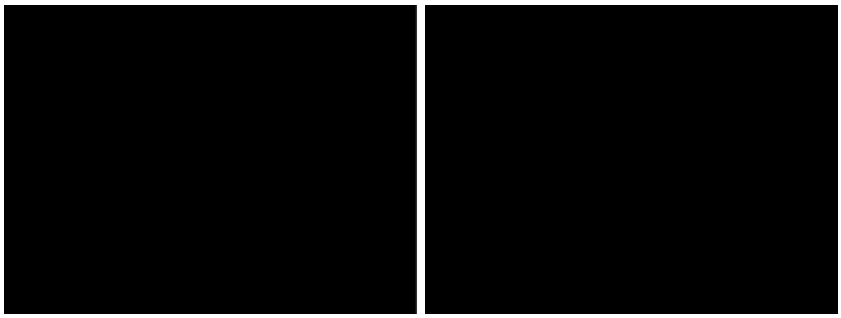


(c) Prediction time

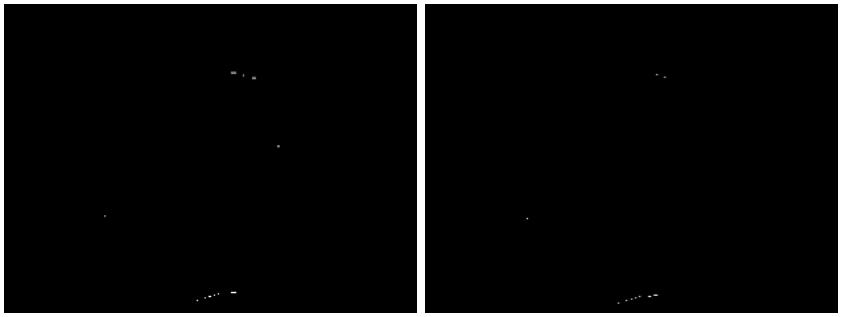
Figure 3.3: U-net architecture optimisation. For each parameter, U-net$_0$ - U-net$_4$ was trained with a learning rate of 0.0001 and a batch size of 1 on 500 images



(a) The test image. *(b) The output of U-net$_0$*



(c) The output of U-net$_1$ *(d) The output of U-net$_3$*



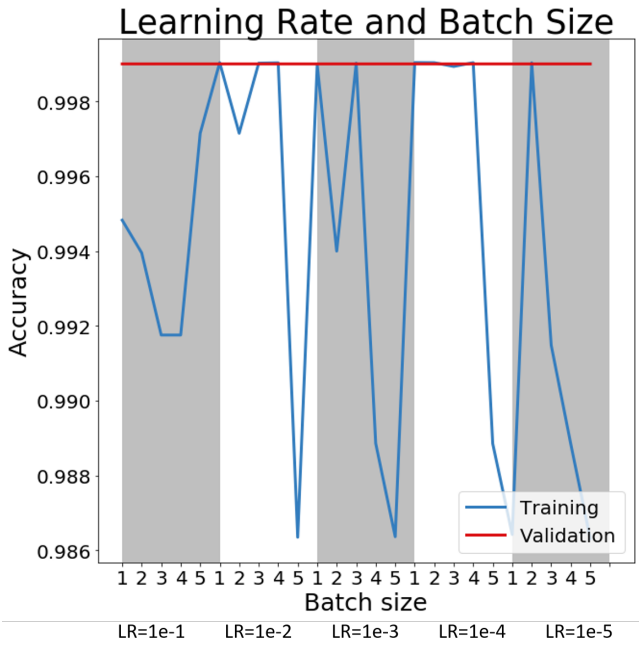
(e) U-net$_3$ *(f) The output of U-net$_4$*

Figure 3.4: Examples of the same output image of training different versions of U-net.

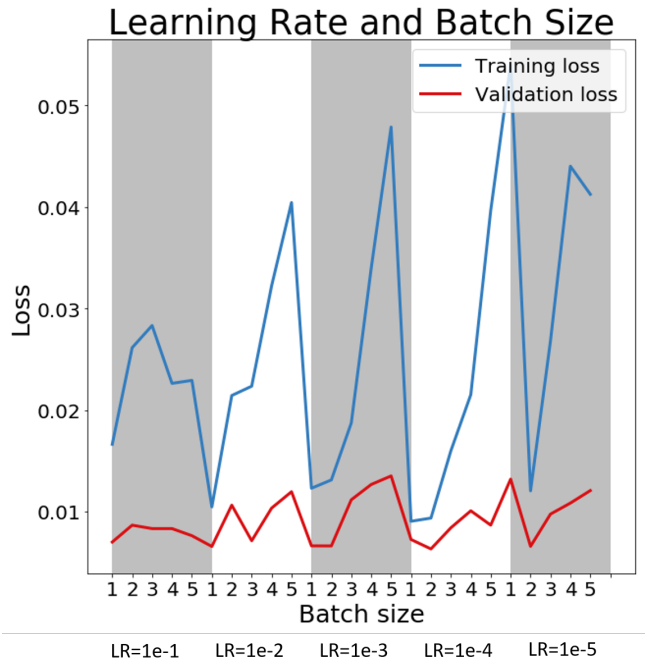
3.3.2 Hyperparameter optimisation

In figure 3.5a, the training and validation accuracy of training with different values for learning rate and mini-batch size can be seen. The training accuracy

lies between 0.98 and 0.99 for all hyperparameter combinations. The validation accuracy seems to be stable around 0.999. Peaks in the training accuracy can be observed, with minima for the batch sizes of 5. There is no clear pattern in the training accuracy for different learning rates, expecpt for the learning rate of 1×10^{-4}, for which the minimal values are not as high as the minimal values for learning rate 1×10^{-2} to 1×10^{-5}. For the training loss, which can be seen in figure 3.6b, minima can be seen at the batch sizes of 1. The validation loss is lower than the training loss, with minima for batch size of 1 as well. The minimal training and validation loss are located at a batch size of 1 and a learning rate of 1×10^{-4}.



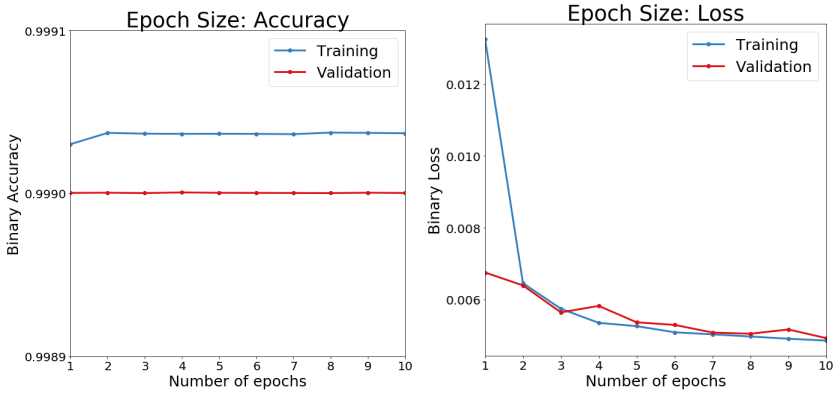
(a) Accuracy



(b) Loss

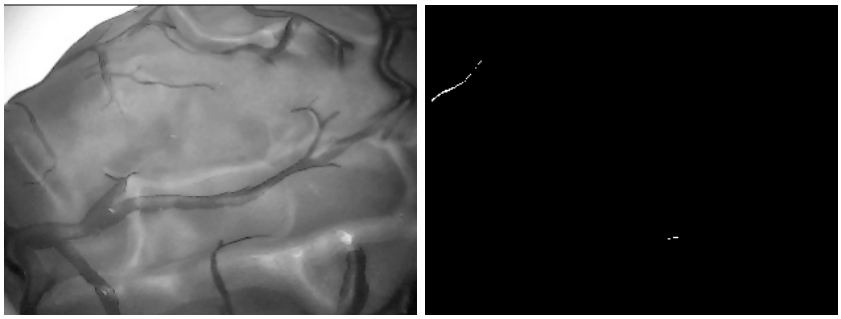
Figure 3.5: Hyperparameter optimization. Each learning rate was tested with all 5 batch sizes.

In figure 3.6, the training and validation accuracy can be seen for training a decreasing number of epochs. It is clear that both the training and the validation accuracy decrease with the number of epochs, whereas the training loss increases. The validation loss shows a decreasing trend as well with an increasing epoch size. In figure 3.7, examples of the output of U-net$_3$ trained with different learning rates with a mini-batch size of 1 can be seen. Throughout the images, differences in the location, isolation and shape of the positive areas can be seen.



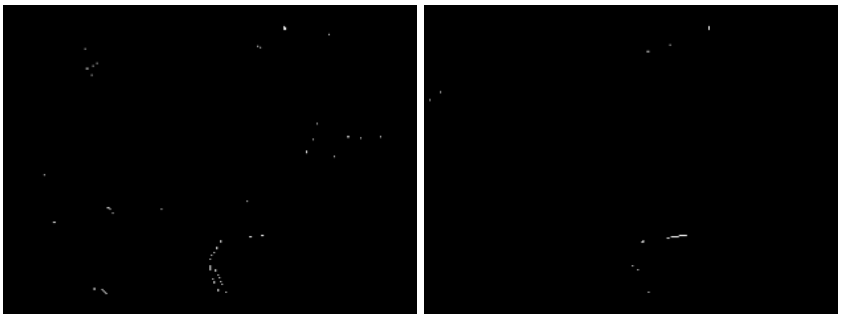
(a) Training and validation accuracy *(b) Training and validation loss.*

Figure 3.6: Epoch optimisation. Training was performed for 10 epochs.



(a) The test image. *(b) Learning rate* 1×10^{-1}

(c) Learning rate 1×10^{-2} *(d) Learning rate* 1×10^{-3}



(e) Learning rate 1×10^{-4} *(f) Learning rate* 1×10^{-5}

Figure 3.7: Examples of the same output image of U-net after training with different learning rates with mini-batch size 1.

3.4 Discussion

In this chapter, the optimal architecture and hyperparameters for using a U-net based architecture as a feature detection method were analysed. First,

the optimal architecture of U-net had to be determined. This might seem a straightforward problem, but due to the nature of point feature detection classic optimisation strategies were of limited use. In figure 3.3a, it can be observed that the training and validation accuracies are close to 0.99 and differ only slightly for the different training instances. This was caused by the skewed balance between pixels labelled as features and pixels labelled as non feature. For example, predicting a totally black image would lead to an accuracy of 0.99 (which is, of course, dependent on feature density). This rules out looking at training and validation accuracy as means to optimise the outcomes of this feature detection method.

Where the accuracy showed minimal differences, the binary cross entropy loss did show large differences, with training losses ranging from 6.11×10^5 for U-net$_2$ to 1.19×10^{-2} for U-net$_4$. The validation losses did show large differences as well, which can be seen in table 3.1. That binary cross entropy loss is of better predictive value than accuracy can be explained because loss, in contrast to accuracy, is an accumulative value instead of an percentage. This makes loss less influenced by the skewed distribution between classes. Based on the validation loss, the choice for U-net$_3$ was made. The visual inspection of the threshold predictions on the test images did support this decision. The low loss and good visual outcomes when training a shallower U-net supports the theory that the deep architecture of U-net is unsuitable to learn to predict local (point) features.

For the hyperparameter optimisation, accuracy was again of limited value.In figure 3.5a it can be seen that again that training accuracy only differs in a small range. The binary cross entropy loss did show larger differences. The loss is clearly the lowest when mini-batch size is 1. However, the exact best value for the learning rate in combination with a mini-batch size of 1 is not clear. To differentiate this, test images for training with a mini-batch size of 1 and the four different learning rates were visualised together. An example of this visualisation can be seen in figure 3.7. The visual inspection of these results showed that a 1×10^{-4} in combination with a mini-batch size of 1 is the best set of hyperparameters for this application. A last note to make is that the validation loss is lower than the training loss for both figure 3.3b as figure 3.5b. It might seem counter intuitive that the validation loss is smaller than the training loss, but in the Keras implementation the validation loss is calculated after one epoch while the training loss is calculated during training. Because these hyperparameter tests were performed during one epoch, the different time of calculating loss gave this unexpected result.

The last parameter to optimize was the number of training epochs. In figure 3.6b it can be seen that both the training and the validation loss decreased after training for more epochs, which was as expected. The validation loss decreases until 3 epochs, after which some increasing can be seen. To prevent the network from overfitting, an epoch size of 3 was selected as ideal number for training in this application. The training loss did start higher than the validation loss for one epoch, but after three epochs the validation loss is higher, as expected. This support the theory as described above.

The loss and accuracy are general performance measures for the quality of training of U-net, but it did not say anything about the sensitivity or specificity of the resulting feature detection method. Sensitivity is an important parameter for the feature detection method, because ORB-SLAM needs enough features to create the 3D point cloud. However, detecting a few more or less features will likely not be a very big influence on the quality of the resulting point cloud. The specificity of the feature detection method is more important for the use in 3D FLOVA-SLAM. Although 3D FLOVA-SLAM removes incorrect detected features, incorrect detected features can have a big effect on feature mapping. When design choices have to be made, for example in choosing the threshold for the output images, the specificity should be valued more.

Something which was not taken into account during the optimisation process was the different mistakes which the feature detection method can make, all with different influences on the accuracy, loss and final application. First, a point feature can be missed altogether. This will increase the binary cross entropy loss and decrease the accuracy. The effect on 3D FLOVA-SLAM would be clear: fewer detected features to create the point cloud with. Next, a point feature can be located on a slightly different location. This increases both the loss as decreases the accuracy. The effect of this is difficult to predict for 3D FLOVA-SLAM: it depends on the stability of the feature on the slightly different location. Lastly, a point feature can be detected multiple times. This will increase loss and decrease accuracy. This mistake is expected to have the biggest influence in 3D FLOVA-SLAM, because the repeated features are likely to be less unique and thus more unstable. This increases the changes of creating an incorrectly mapped 3D point in the final reconstruction. Further research on the different feature detection mistakes is recommended after implementation of the feature detection in 3D FLOVA-SLAM.

In summary, this chapter introduced the use of U-net as a feature detection method. To do so, the best architecture was found to be U-net with two instead of the originally four down- and upsampling steps. The optimal hyperparameters were shown to be a learning rate of 1×10^{-4}, a mini-batch size of 1 and an epoch size of 3. In the next chapter, this information is used to train a feature detection method on *ex-vivo* placenta data and evaluating its performance.

4 Performance evaluation of F-net

4.1 Introduction

In this chapter, F-net will be introduced and its performance will be analysed. F-net was developed by training the adjusted architecture of U-net as described in Chapter 3 on the *ex-vivo* placental dataset. Two versions of F-net are discussed: one trained on the original and one trained on the selected 3D FLOVA-SLAM features. First, a comparison between F-net and ORB will be made to quantify the ability of F-net of learning to detect ORB-features. Second, a comparison between F-net trained on original 3D FLOVA-SLAM features and selected, stable 3D FLOVA-SLAM features will be made to assess the added value of training on selected features.

Comparison of feature stability will be guided by the definition of stable features as provided by Salahat et al. [32]. The feature qualities distinctiveness, quantity, efficiency, sensitivity to rotation and sensitivity for noise will be evaluated. Distinctiveness is the most important feature stability, because F-net was developed with the aim of increasing distinctiveness specific for 3D FLOVA-SLAM. The repeatability of the features will be compared by testing how the feature detection methods react to noise and rotation, situations which occur during FLOVA. Quantity is of importance because 3D FLOVA-SLAM needs a sufficient amount of features to create the 3D placental reconstruction. Lastly, it will be discussed if F-net is efficient enough to be used in the real-time 3D FLOVA-SLAM application.

4.2 Methods

4.2.1 Training of F-net

F-net was based on the U-net architecture as depicted in subfigure 3.2c, with two down- and upsampling blocks with corresponding horizontal skips. Training of F-net was performed on the *ex-vivo* placental datasets. Two different versions of F-net were trained, one on the dataset with the original 3D FLOVA-SLAM features, called F-net$_{OF}$, and one on the dataset with the selected features, F-net$_{SF}$. From both datasets, the same 50 pairs of images were randomly selected to create a test set on which feature matching could be performed. After selection of the test pairs, both datasets were randomized the same way, leading to datasets with identical training examples but different training labels. The resulting datasets contained 1240 images. A training- and validation split of 0.2 was used, resulting in a training set of 992 images and a validation set of 248 images. Training of F-net was performed with Adam as optimizer with a learning rate of 1x10^{-4} , $\beta 1 = 0.900$, $\beta 2 = 0.999$ and learning rate decay = 1×10^{-8}, a batch size of 1. Training was performed during 3 epochs.

4.2.2 Comparison between ORB features and F-net features

First, F-net$_{OF}$ and ORB features as outputted by 3D FLOVA-SLAM were compared to analyse the ability of F-net learning to detect ORB features. If possible, this comparison was based on the original features from 3D FLOVA-SLAM, encoded in the labels of the dataset. This was the case for the quantity of features and matches and the distinctiveness. For the comparison based on sensitivity for rotation and noise and prediction times, the Python OpenCV implementation of ORB was used to detect features, because the required information could not be extracted from the labelled images [39].

F-net$_{OF}$ was used to predict feature coordinates. First, an input image was provided to F-net, after which the resulting prediction was a greyscale image with values between zero and one, being the probability of a pixel being a feature. Next, this probability image was thresholded with a value of 0.09 to obtain a binary image with feature locations. After this, the coordinates of the non-zero elements were extracted to obtain the feature coordinates. A visual overview of this pipeline can be seen in figure 4.1. The quantity of F-net features and the efficiency could be calculated from this step. The prediction times of F-net were calculated as the time needed to do the prediction, the thresholding and the feature coordinate extraction step.

Next, feature matching with the OpenCV implementation of the Brute-Force Matcher was performed with both the F-net features as the ORB. Each pair of test images was matched together, so the number of matches could be determined for 50 instances. After feature matching, Random Sample Consensus (RANSAC) was used with a projection threshold value of 5.0 to calculate the inlier and outlier matches for each image [40]. The feature distinctiveness was

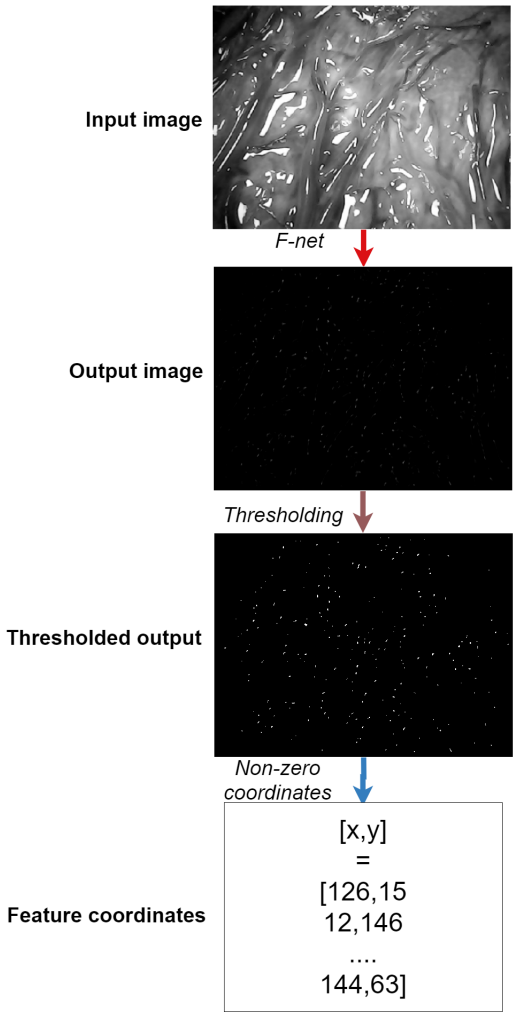


Figure 4.1: The F-net pipeline for going from input image to feature coordinates.

calculated by dividing the number of inlier matches by the total number of matches for each pair of images.

To compare sensitivity to rotation, F-net was compared with the Python implementation of ORB. To do so, rectangles from the test images were selected and rotated, after which the resulting outcome images were substracted to find the overlap in features. First, a rectangle of 400×400 pixels was selected from the middle of the image. Second, this rectangle was rotated 0°, 90°, 180°and 270°, after which zeros were added to match the input size of F-net. The first two steps were necessary to be able to rotate 90 °and 270 °, because otherwise the size of the images did not match the required input size of F-net. The big rotation angles were chosen to see the effect of rotation in the most extreme

case, so differences in sensitivity to rotation would be the most distinct. The third step was feature detection with both F-net as well as ORB on the four created images. A threshold of 0.09 was applied to the output images of F-net to obtain binary feature images. Then, the feature coordinates of ORB were binary encoded into an image of 400×400, to be able to do an comparison with F-net. After this, the feature images were rotated back to the original orientation. Next, for both ORB as F-net, all four image orientations were subtracted. From the resulting images, the non-zero elements were counted to obtain the number of non overlapping features. This number was divided by the total number of features in the image with 0°of rotation to obtain the sensitivity of noise of F-net and ORB. To do a comparison based on sensitivity to noise, salt and pepper noise with a probability of 0.10% was added to test images. ORB and F-net were used to predict images both on the noisy as the original image. The resulting images were subtracted and the sensitivity for noise was calculated the same way as the sensitivity for rotation.

4.2.3 Comparison between F-net trained with original and selected features

For the comparison between F-net$_{OF}$ and F-net$_{SF}$, the feature quantity, the quantity of matches, the quantity of inlier matches and the feature distinctiveness were calculated. These were calculated with the same methods as for the comparison between F-net and ORB.

4.3 Results

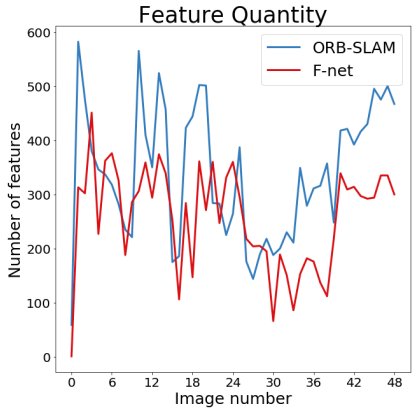
4.3.1 Training F-net versions

The final training of F-net on the original feature dataset resulted in a training accuracy of 0.99 and a training loss of 5.20×10^{-3}. The validation accuracy was 0.99 and the validation loss 5.10×10^{-3}. The final training of F-net trained on the original dataset resulted in a training accuracy of 0.99 and a training loss of 5.20×10^{-3}. The validation accuracy was 0.99 and the validation loss 5.30×10^{-3}.

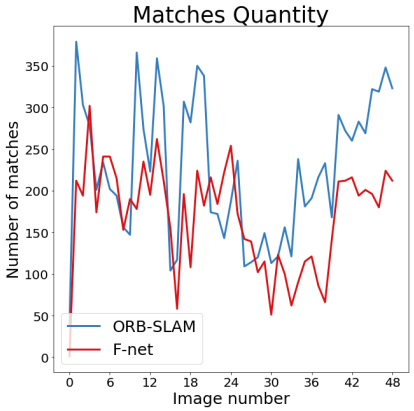
4.3.2 Comparison between ORB and F-net features

In figure 4.2, the comparison between ORB and F-net features can be seen. In subfigure 4.2a, it can be observed that the quantity of ORB features is larger for almost all images, with a mean number of ORB features of 340 and F-net features of 257. The mean number of matches is higher for ORB features as well (224-168), as can be seen in subfigure 4.2b. For both feature detection methods it can be seen that the number of matches is lower than the number of detected features. Subfigure 4.2c shows the distinctiveness of the detected features, which shows that F-net features are overall more distinct, with an inlier ratio of 86 for

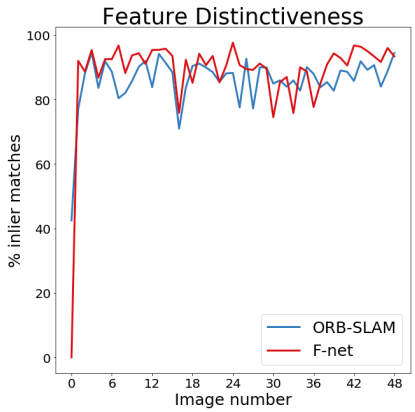
ORB features and 88 for F-net features. Both feature detection times can be observed in subfigure 4.2d, for which it can be seen that the prediction times of F-net are almost one second longer on all images. Subfigure 4.2e shows that F-net detects more deviant features when predicting rotated images, with a percentage of 2.7 for ORB and 3.1 for F-net. In figure 4.2f, it can be seen that the percentage of deviant features for ORB is higher than for F-net when predicting images with different levels of noise. An example of feature matching before and after RANSAC can be seen in figure 4.2c. The method of calculating rotation and noise sensitivity can be seen in figure 4.5 and 4.6.



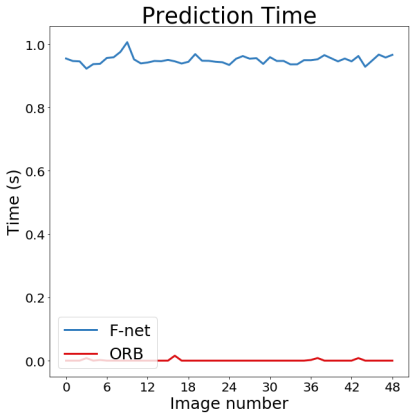
(a) Number of features. The mean number of features was 340 for ORB and 257 for F-net.



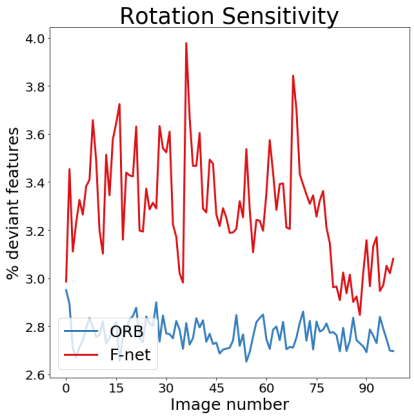
(b) Number of matches. The mean number of features was 224 for ORB and 168 for F-net.



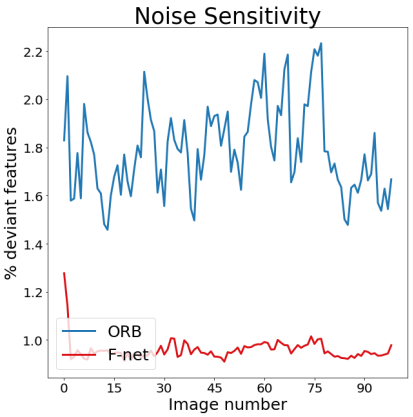
(c) Ratio of matches after RANSAC. The mean ratio was 86 for ORB and 88 for F-net.



(d) Prediction time. The mean prediction time was 0.002 seconds for ORB and 0.950 seconds for F-net.



(e) Sensitivity to rotation, the mean sensitivity was 2.7 for ORB and 3.1 for F-net.

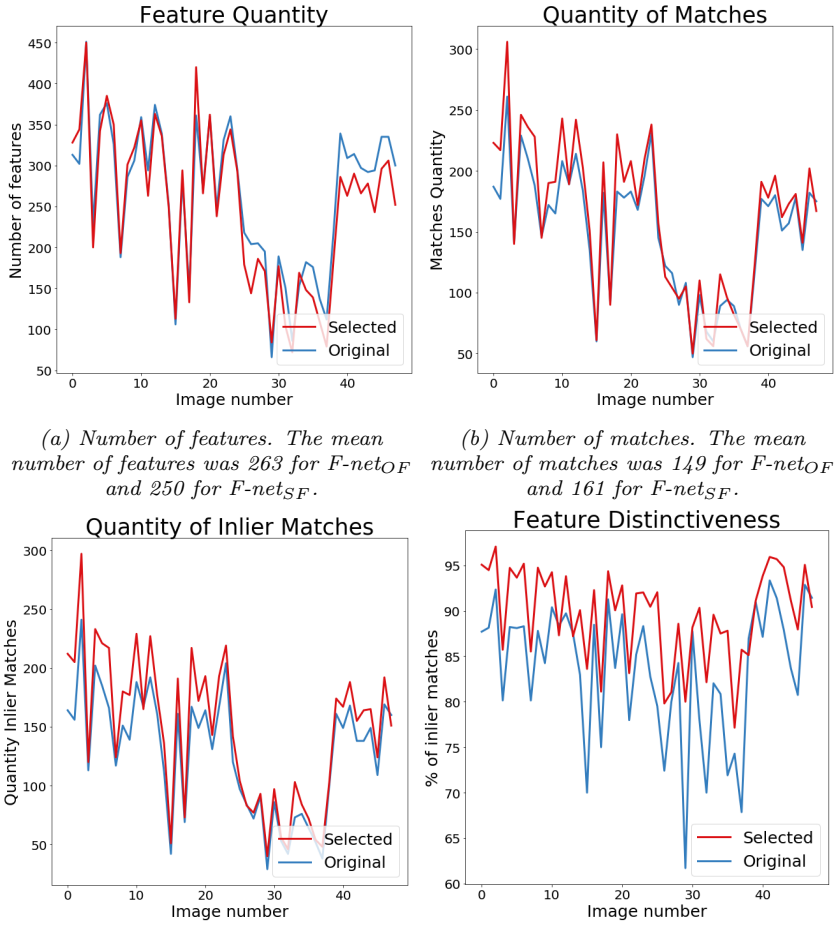


(f) Sensitivity to noise, the mean sensitivity was 1.8 for ORB and 0.95 for F-net.

Figure 4.2: Comparison between ORB and F-net features. Note that for the sensitivity for noise and rotation not the ORB features from 3D FLOVA-SLAM, but the python implementation of ORB has been used.

4.3.3 Comparison between F-net trained on original and selected features

The comparison between the F-net trained with original and selected features can be seen in figure 4.3. In subfigure 4.3a, it can be seen that the number of features is higher for F-net$_{OF}$ (263-250). Subfigure 4.3b shows the difference in the quantity of matches: with a mean of 149 for F-net$_{OF}$ and a mean of 161 for F-net$_{SF}$. In subfigure 4.3c, it can be seen that the total number of inlier matches is 127 for F-net$_{OF}$ and 146 for F-net$_{OF}$.In figure 4.3d, it can be seen that the ratio of inlier matches is bigger for F-net$_{SF}$, with a mean number of 83.6% for F-net$_{OF}$ versus 89.6 % for F-net$_{SF}$.



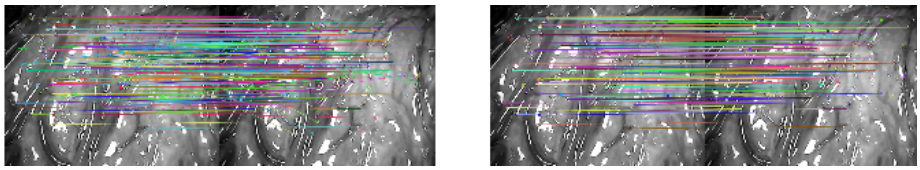
(a) Number of features. The mean number of features was 263 for F-net$_{OF}$ and 250 for F-net$_{SF}$.

(b) Number of matches. The mean number of matches was 149 for F-net$_{OF}$ and 161 for F-net$_{SF}$.

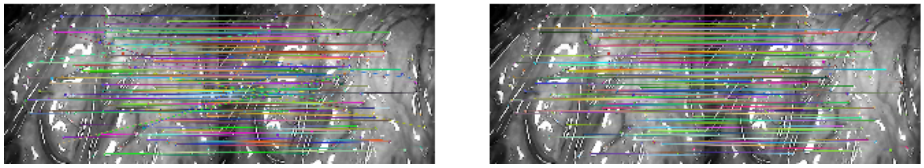
(c) Quantity matches after RANSAC. The mean number of inlier features was 127 for F-net$_{OF}$ and 146 for F-net$_{SF}$

(d) Ratio of inlier matches. The mean ratio was 83.6 for F-net$_{OF}$ and 89.6 for F-net$_{SF}$.

Figure 4.3: Comparison between F-net trained on original and selected features.

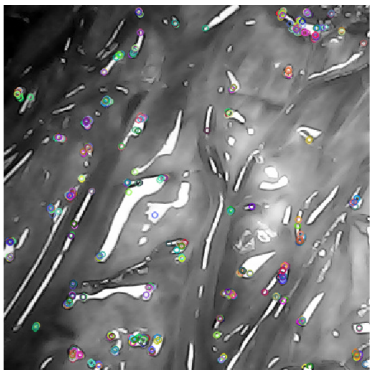


(a) ORB matches before and after RANSAC.

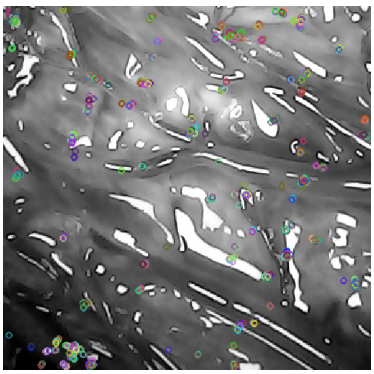


(b) F-net matches before and after RANSAC.

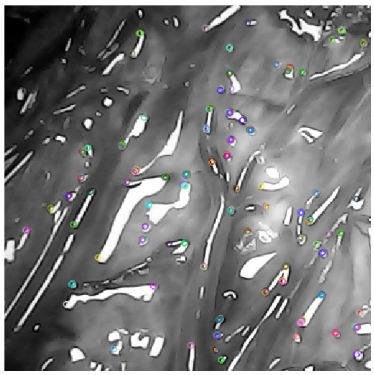
Figure 4.4: Example of feature matching before and after RANSAC.



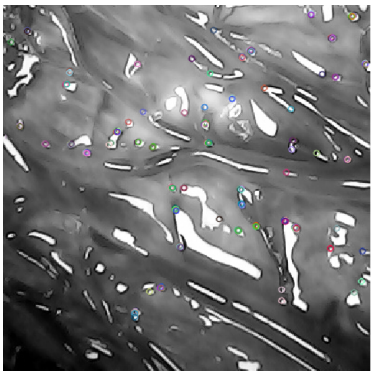
(a) The output of ORB with 0° rotation.



(b) The output of ORB with 90° rotation.



(c) The output of F-net with 0° rotation.

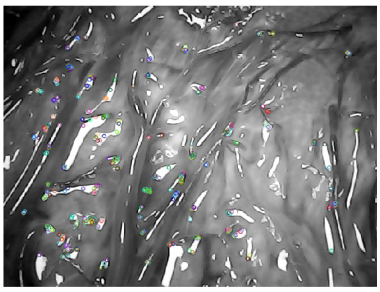


(d) The output of F-net with 90° rotation.

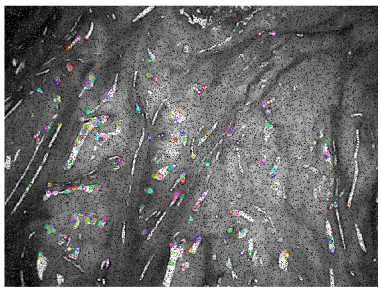
(e) The substracted output for ORB.

(f) The substracted output for F-net.

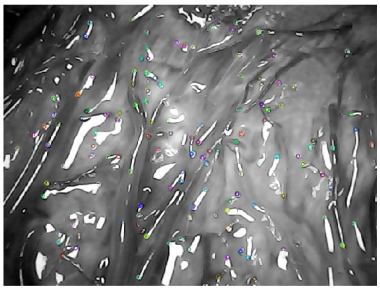
Figure 4.5: Sensitivity to rotation.



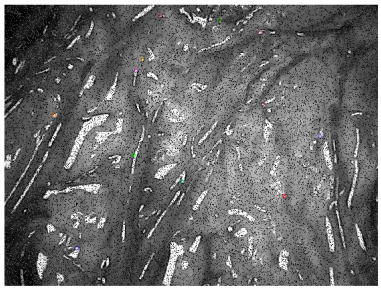
(a) The output of ORB without noise.



(b) The output of ORB with noise.



(c) The output of F-net without noise.



(d) The output of F-net with noise.

(e) The substracted output for ORB.

(f) The substracted output for F-net.

Figure 4.6: Sensitivity to noise.

4.4 Discussion

In this chapter, F-net was developed by training an adjusted U-net architecture on *ex-vivo* placenta data. Two different versions of F-net were trained, F-net$_{OF}$ on the original 3D FLOVA-SLAM labels and F-net$_{SF}$ on the 3D FLOVA-SLAM

features selected on feature stability. First, F-net F-net$_{OF}$ was compared to the original 3D FLOVA-SLAM features to quantify the ability of F-net to detect ORB features. Next, F-net$_{OF}$ was compared with F-net$_{SF}$ to test the hypothesis that feature selection before training F-net will increase stability of the detected features.

4.4.1 Comparison between ORB and F-net features

The comparison between F-net and ORB as feature detection started with the quantity of detected features. In figure 4.2a it can be seen that the difference in the number of features is really dependent on the specific image. However, the mean number of detected features was 258 for F-net, in contrast to 340 for ORB. The effect of the feature quantity on 3D FLOVA-SLAM is likely not large, because a higher number of features does not neseccary mean a higher number of feature matches. In subfigure 4.2b, it can be seen that the total number of matches is larger for ORB-SLAM features as well. However, the difference in the mean number of matches is smaller in comparison to the difference in number of detected features. This is an indication that F-net detects less in-matchable features. The feature distinctiveness for F-net is bigger than for ORB, as can be seen in figure 4.2c. Because F-net predicts less, but more stable features, the effect of introducing F-net as a feature detection method in 3D FLOVA-SLAM is unclear and should be validated after implementation of F-net in 3D FLOVA-SLAM. It should be noted that increasing the threshold applied for the calculation of feature coordinates decreases the number of features while likely increasing the feature stability. It is recommended that the effect of the output image threshold is tested after implementation in 3D FLOVA-SLAM. The efficiency of F-net should be high enough to be used in the real-time 3D FLOVA-SLAM application, which means a feature prediction time of 0.1 seconds or less (for a frame rate of 10 frames per second). At the moment, the mean prediction time of F-net is close to 1 second, so F-net cannot be used real-time yet. Decreasing of the computational time of F-net is likely after implementation in C++.

Noise robustness, a low sensitivity to noise, and rotation invariance, a low sensitivity to rotation, are of importance in 3D FLOVA-SLAM to maximize the changes of a feature being detected on two different frames in different circumstances. The sensitivity to rotation seemed to be bigger for F-net than for ORB features with a mean 3.1 % of deviant features. However, it should be noted that rotations of 90°, 180°and 270°were extreme test cases and are not likely to happen during the fetoscopy. Identifying the most extreme rotation angels during fetoscopy and using these angles to augment the training dataset with would be an improvement in the next version of F-net. The sensitivity of ORB to noise seems to be higher than the sensitivity of F-net to noise, as can be seen in figure 4.2f. However, it also can be seen in 4.6d, F-net detects a lot less features after adding noise. The conclusion has to be made that F-net is more sensitive to noise as well. It is recommended to add different levels of noise to the training dataset and using feature matching instead of feature

detection as a measure for rotation invariance. The comparison between F-net and ORB features was made to prove that F-net can predict features with the same stability. However, in this chapter it was shown that predicted F-net features are more distinct than ORB features. This might seem an unexpected result, but can be explained by the stable ORB features being more uniform which makes training for F-net of this features easier.

4.4.2 Comparison between F-net trained with original and selected features

The comparison between F-net$_{SF}$ and F-net$_{OF}$ started with the different training outcomes. Because the datasets were completely comparable, besides the labels, differences in training outcomes can only be explained by the different labels. The training of F-net$_{OF}$ resulted in a training loss of 5.20×10^{-3} and a validation loss of 5.10×10^{-3}. Training of F-net$_{SF}$ lead to a training loss of 5.20×10^{-3} and a validation loss of 5.30×10^{-3}. The training losses are comparable, but the validation loss is lower for F-net$_{OF}$. The difference in validation loss can be explained by the larger number of features in the original test set, which leads to a more stable training.

The goal of training F-net on selected features was to increase the stability of the detected features. In figure 4.3a, it can be seen that the mean number of detected features is slightly lower for F-net$_{SF}$, 250 instead of 253 features. This can be expected, because F-net$_{SF}$ was trained to not detect unstable features. However, it can be seen that this for some images the number of predicted images is higher for F-net$_{SF}$. This can be explained by the used threshold: if the predicted features for F-net$_{SF}$ have a higher certainty, using the same threshold for both versions will lead to more predicted features for F-net$_{SF}$. The mean number of matches for F-net$_{SF}$ features is bigger than for F-net$_{OF}$ (149-161), which is not to be expected from the lower number of detected features. This means that F-net$_{SF}$ features are better matchable than F-net$_{OF}$, which suggests that training F-net on selected features will lead to a more stable feature detection. This is supported by fact that the quantity of inlier matches and the feature distinctiveness is bigger for F-net$_{SF}$. The distinctiveness of F-net$_{SF}$ was 89.6% inlier features, which was 83.6% for F-net$_{OF}$. This is an important finding because increasing the feature distinctiveness strongly suggest that there will be less outlier features in the 3D FLOVA-SLAM reconstruction.

Gaisser et al. showed an increased feature distinctiveness for their CNN feature detection method in comparison to BRIEF, 26.2% versus 50.9% [31]. These numbers are lower than expected based on the results discussed in this chapter and are likely the result of different matching results. The match pairs which Gaisser used were artificially adjusted with rotation, translation, scale and noise, in contrast to the matching of subsequent images. Although the methods of feature detection and feature analysis were different, the results of Gaisser support the findings in this chapter that F-net is capable of predicting more stable features.

The main drawback of the method of comparing feature stability in this

chapter is that feature stability specific for 3D FLOVA-SLAM could only be analysed indirectly. The effect of the differences in the presented feature detection methods can be predicted, but the true effect will remain unclear until implementation in 3D FLOVA-SLAM is performed. Another point of discussion is the method of comparing rotation- and noise sensitivity. The current method has to shortcomings. First, the penalty of detecting a feature on a slightly different location is the same as predicting a feature on a completely different location. Second, predicting less features will lead to a decreased sensitivity, which is an incorrect conclusion. For further research, it is recommended to analyse feature repeatability indirectly by feature matching to limit the effects of both limitations.

In conclusion, F-net features are of lower quantity, but slightly more distinct than ORB features. The sensitivity to noise and rotation of detecting features with F-net seems to be higher than of ORB, but augmentation of the training set with noise and rotation might solve this issue. The main drawback at the moment is the efficiency of F-net. If the prediction times of F-net can be shortened, F-net should be able to replace ORB features in the 3D FLOVA-SLAM application.

5 General discussion

In this thesis, the development and performance evaluation of F-net, a CNN based feature detection method for application in 3D FLOVA-SLAM, was discussed. The aim of this newly developed feature detection method was to increase the stability of features used in 3D FLOVA-SLAM, with the aim of improving the quality of the placental reconstruction. To do so, three sub questions were answered. Chapter 3 discussed feature stability, in general and specific for 3D FLOVA-SLAM. Salahat et al. stated that in general, a feature is stable if it is distinct, local, of sufficient quantity, accurate, efficient, invariant and robust [32]. Stable features specific for 3D FLOVA-SLAM were defined as features not leading to outlier points in the 3D placental point cloud. This definition was used to create a training dataset containing only stable features, with the aim of training F-net on these. Chapter 3 described the adjustments and optimisation process for the existing CNN framework U-net needed to to function as a feature detection method. Using these conclusions, F-net was developed and the performance was validated in Chapter 4. Two comparisons were made, first between F-net trained on the dataset with the not selected features and ORB, the original feature detection method of 3D FLOVA-SLAM. Next, a comparison between F-net trained on original and selected features was performed. This chapter will shortly discuss the results of these chapters and provide a general discussion about these conclusions.

5.1 Interpretations of results

In Chapter 2, a 3D point cloud filtering method was used to distinguish between stable and non stable 3D FLOVA-SLAM features. To do so, the assumption was

made that all outlier 3D points were the result of unstable features. The ratio of outlier features was 0.06%, which resulted a low number of rejected features on each image. For the next development stage of F-net, it is recommended to select images with multiple removed features and merge the result of different 3D FLOVA-SLAM scans to get a dataset of a sufficient size.

In Chapter 3, it was showed that the optimal adjusted architecture of U-net for feature detection was a architecture with two instead of the original five convolutional down- and upsampling steps. With this adjusted architecture, hyperparameter optimisation was performed, concluding to an optimal learning rate of 1×10^{-4} in combination with a batch size of 1. Training for three epochs was shown to maximize training results while avoiding overfitting of the network.

With the information provided by Chapter 3 and the dataset as described in Chapter 2, F-net was created. Two different versions of F-net were trained, on on the dataset containing the original 3D FLOVA-SLAM features and one with the selected features. First, a comparison between F-net and ORB features was performed to be able to say something about the possibility of F-net learning to detect original ORB features. In this comparison, it was shown that the quantity of F-net features is slightly lower, but the distinctiveness is higher in comparison with the 3D FLOVA-SLAM features. F-net is less invariant to noise and rotation, in comparison to ORB. It is recommended to augment the training dataset with different angles of rotation and levels of noise, to enhance the repeatability of F-net. The biggest limitation of F-net was shown to be the efficiency, with prediction times of around 1 second for each image. This is to slow for real-time application in 3D FLOVA-SLAM. Before application in 3D FLOVA-SLAM it is essential that the prediction times of F-net are improved, for example by implementation in C++. Next, the comparison of F-net trained on the original and selected features showed that the training of F-net on selected features is effective in improving feature stability. It was shown that the feature and matches quantity is slightly for F-net trained on the selected features, but that this is compensated after feature matching. Based on these results it is expected that implementation of F-net trained on selected features in 3D FLOVA-SLAM will lead to an improved quality of the 3D reconstruction of the placenta.

The comparison between F-net trained on the original features and F-net trained on the selected features showed that the latter detects more distinct features. This is in line with the hypothesis and the initial goal of the feature selection. Based on these results it is expected that implementation of F-net trained on selected features in 3D FLOVA-SLAM will lead to an improved quality of the 3D reconstruction of the placenta.

5.2 Strengths and weaknesses

F-net uses the established architecture of U-net, but with the decreased complexity created by decreasing the number of convolutional blocks. This is beneficial

for detecting (point) features because these are highly local, whereas deeper convolutional layers focus on more complex spatial relations. Training F-net with a less complex architecture therefore increases the chance of F-net successfully learning to detect features, with the added advantage of a lower training and prediction time.

The biggest strength of F-net is training on stable features specific for 3D FLOVA-SLAM. These labels were labelled without human supervision, which limits observer variability and decreases the time needed to label big quantities of data. Together with the short training times, this adds the flexibility to multiple times when needed. This is likely necessary, because the training of F-net is hard to generalize to data from different fetoscopes or under different conditions, because of the limited variation in the used dataset. It would be easier to retrain F-net when needed to be used in another application than trying to make F-net generalisable for all instances. Adding to this, if a fetoscope with different images sizes than 640×480 will be used, F-net has to be retrained with the adjusted input size.

The biggest limitation from F-net was the performance evaluation. The definition of a stable feature from Salahat et al. was used to compare the detected features with the original 3D FLOVA-SLAM features [32]. However, the effects of changes in these feature qualities can only be assessed indirectly, because F-net is not implemented in 3D FLOVA-SLAM yet.The exact effect of F-net on the quality of the 3D point cloud therefore remains unknown. When F-net is implemented in the existing 3D FLOVA-SLAM network, the effect of different design choices can be better quantified as well. For example, both the threshold used in the filtering of the point clouds as the threshold for extracting feature coordinates from the output of F-net can be reasoned better when the effect of the placental reconstruction is more clear. The recommended way to do so is to make a 3D scan of the placenta and using this as a ground truth for the placental reconstruction. The error between the registered point clouds can be used as a performance and evaluation measure.

5.3 Further perspectives

As said before, the next important step is to implement the developed F-net in the 3D FLOVA-SLAM framework. To do so, the trained version of F-net can be implemented in C++, without needing to retrain in C++. F-net, with the required post processing steps, will be able to function in place of ORB. However, because F-net is only a feature detection method, while ORB is both a feature detection as feature description method, a separate feature description needs to be implemented as well. It is recommended to initially use BRIEF for this, because this is the description method implemented in ORB. Different research has used CNNs to perform feature description with, so an addition to F-net could be developed as well [27, 29]. However, the effect of adding this possibly complicated step on the quality of the 3D FLOVA-SLAM placental reconstruction is unknown and might be minimal.

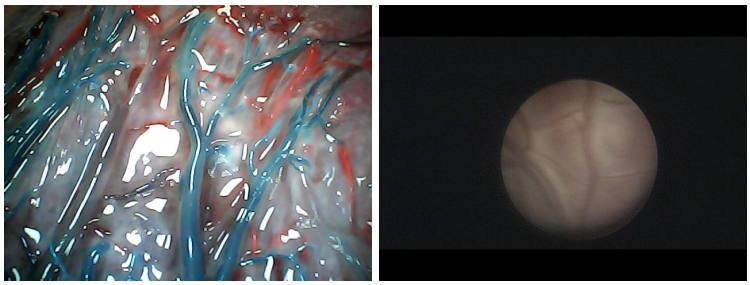
The biggest limitation for using F-net aided 3D FLOVA-SLAM *in-vivo* is that it was trained on *ex-vivo* placenta data. Because the current 3D FLOVA-SLAM method has not successfully been used on *in-vivo* yet, and thus *in-vivo* labels, F-net could not be trained on *in-vivo* data. A way to break this vicious circle could be to use *ex-vivo* data with labels, but artificial change the *ex-vivo* data to match the images obtained *in-vivo*. It is recommended to add one property of *in-vivo* data to the training set at the time, to observe the effect on detecting features on *in-vivo* test images. The following properties of *in-vivo* data, in order of the expected effect on F-net, need to be taken into account:

- **Circular field of view**: The *ex-vivo* data was recorded with a endoscope with a square field of view. The fetoscope used in FLOVA has a circular field of view, with a size dependent on the used zoom. F-net could be trained to detect only features within the field of view and not on the edges of the circle or in the black, noisy surrounding remainder of the image. However, if the calibration method requires an image crop, this step can be skipped.

- **Lower contrast**: The contrast in *in-vivo* images is lower than in the *ex-vivo* images, resulting of the amniotic fluid and low lightning conditions. If F-net is not able to detect features in these low contrast images, the contrast of the training images can be decreased.

- **Occlusion in image**: Occlusion of the placental view can occur with parts of the fetus, membranous equator or surgical instruments. Features might be detected on these obstructions. Adding occlusions in the training set, with removed labels at these parts in the image, will create a version of F-net that only detects features on the placental views.

- **Light distribution**: The light distribution is uneven in fetoscopic images. This influences feature detection and feature matching. Artificial illumination changes on the edges of the training images while remaining the original labels might make the feature detection invariant to these lighting differences.



(a) Ex-vivo image *(b) In-vivo image*

Figure 5.1: Example of in- and ex-vivo recorded data. The in-vivo data is recorded during a FLOVA procedure. It can be seen that the in-vivo image contains a round field of view and has less contrast than the ex-vivo image.

If training data can be made containing these properties, Generative Adversarial Networks (GANs) can be used to create large amounts of this labelled artificial data [41]. A dataset with more variation can be created this way. Developing F-net trained specificity on data mimicking the images *in-vivo* from *ex-vivo* data to the challenging *in-vivo* 3D FLOVA-SLAM application.

5.4 Clinical relevance

The presented results implicate that implementation of F-net in the 3D FLOVA-SLAM framework is likely to increase the quality of the 3D placental reconstruction. This might be sufficient to start validating and using 3D FLOVA-SLAM *in-vivo*. 3D mapping of the placenta during the FLOVA procedure is likely to decrease surgery times and increase both maternal as well as fetal outcomes. Implementing F-net in 3D FLOVA-SLAM might bring the physicians of the Radboudumc a step closer to saving even more unborn lives.

6 Conclusion

The 3D FLOVA-SLAM framework, which was developed to create a real-time, 3D reconstruction of the placenta during the FLOVA procedure, suffers from incorrectly detected features. This leads to erroneous placental reconstructions. F-net, convolutional neural network based feature detection method, was developed to detect only stable 3D FLOVA-SLAM features. F-net was created by optimising the architecture of the medical segmentation framework U-net. F-net is able to find more distinct features, which is likely to improve the quality of the 3D FLOVA-SLAM placental reconstruction. The biggest limitations at the moment are the long prediction times of F-net and the development of F-net on *ex-vivo* data. The prediction times will likely to be shorter when F-net is implemented in the original 3D FLOVA-SLAM framework. Artificial augmentation of *ex-vivo* data will F-net suitable for using *in-vivo*. Implementation of F-net in 3D FLOVA-SLAM brings this placental mapping software yet another step closer to support physicians during the complex FLOVA procedure.

Bibliography

[1] J. Collins, "Global epidemiology of multiple birth," *Reproductive biomedicine online*, vol. 15, pp. 45–52, 2007.

[2] V. Berghella and M. Kaufmann, "Natural history of twin-twin transfusion syndrome.," *The Journal of reproductive medicine*, vol. 46, no. 5, pp. 480–484, 2001.

[3] E. Ortibus, E. Lopriore, J. Deprest, F. P. Vandenbussche, F. J. Walther, A. Diemert, K. Hecher, L. Lagae, P. De Cock, and P. J. Lewi, "The pregnancy and long-term neurodevelopmental outcome of monochorionic diamniotic twin gestations: a multicenter prospective cohort study from the first trimester onward," vol. 200, no. 5, pp. 494.e1–494.e8, 2009.

[4] R. A. Quintero, W. J. Morales, M. H. Allen, P. W. Bornick, P. K. Johnson, and M. Kruger, "Staging of twin-twin transfusion syndrome.," *Journal of Perinatology*, vol. 19, 1999.

[5] M. Van Gemert, J. Van den Wijngaard, E. Lopriore, L. Lewi, J. Deprest, and F. Vandenbussche, "Simulated Sequential Laser Therapy of Twin-Twin Transfusion Syndrome," vol. 29, no. 7, pp. 609–613, 2008.

[6] L. L. Simpson, "SMFM clinical guideline: Twin-twin transfusion syndrome," *American Journal of Obstetrics and Gynecology*, vol. 208, no. 1, pp. 3–18, 2013.

[7] R. A. Quintero, K. Ishii, R. H. Chmait, P. W. Bornick, M. H. Allen, and E. V. Kontopoulos, "Sequential selective laser photocoagulation of communicating vessels in twin–twin transfusion syndrome," *The Journal of Maternal-Fetal & Neonatal Medicine*, vol. 20, no. 10, pp. 763–768, 2007.

[8] F. Slaghekke and D. Oepkes, "Solomon Technique Versus Selective Coagulation for Twin–Twin Transfusion Syndrome," *Twin Research and Human Genetics*, vol. 19, no. 3, pp. 217–221, 2016.

[9] S. G. Petersen, K. S. Gibbons, F. I. Luks, L. Lewi, A. Diemert, K. Hecher, J. E. Dickinson, J. J. Stirnemann, Y. Ville, R. Devlieger, G. Gardener,

and J. A. Deprest, "The Impact of Entry Technique and Access Diameter on Prelabour Rupture of Membranes Following Primary Fetoscopic Laser Treatment for Twin-Twin Transfusion Syndrome," *Fetal Diagnosis and Therapy*, vol. 40, no. 2, pp. 100–109, 2016.

[10] M. Reeff, *Mosaicing of endoscopic placenta images*. PhD thesis, ETH ZURICH, 2011.

[11] S. Fransson, *Mosaicing of Fetoscopic Acquired Images using SIFT and FAST*. PhD thesis, KTH Royal Institute of Technology School of Computer Science and Communication, 2017.

[12] F. Gaisser, S. H. Peeters, B. Lenseigne, P. P. Jonker, and D. Oepkes, "Fetoscopic panorama reconstruction: moving from ex-vivo to in-vivo," pp. 581–593, 2017.

[13] T. Bergen and T. Wittenberg, "Stitching and surface reconstruction from endoscopic image sequences: A review of applications and methods," *IEEE Journal of Biomedical and Health Informatics*, vol. 20, pp. 304–321, jan 2016.

[14] M. Tella, P. Daga, F. Chadebecq, S. Thompson, D. I. Shakir, G. Dwyer, R. Wimalasundera, J. Deprest, D. Stoyanov, T. Vercauteren, and S. Ourselin, "A Combined em and Visual Tracking Probabilistic Model for Robust Mosaicking: Application to Fetoscopy," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 524–532, 2016.

[15] H. Werner, J. L. Dos Santos, R. A. Sá, P. Daltro, E. Gasparetto, E. Marchiori, S. Campbell, and E. Araujo Júnior, "Visualisation of the vascular equator in twin-to-twin transfusion syndrome by virtual fetoscopy," *Archives of Gynecology and Obstetrics*, vol. 292, pp. 1183–1184, dec 2015.

[16] S. Thrun and J. J. Leonard, "Simultaneous localization and mapping," *Springer handbook of robotics*, pp. 871–889, 2008.

[17] D. Scaradozzi, S. Zingaretti, and A. Ferrari, "Simultaneous localization and mapping (slam) robotics techniques: a possible application in surgery," *Shanghai Chest*, vol. 2, no. 1, 2018.

[18] R. Mur-Artal and J. D. Tardos, "ORB-SLAM2: an Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras," oct 2016.

[19] C. Jeltes, *The introduction of foetal therapy for TTTS patients in the Radboudumc*. PhD thesis, University of Twente, 2018.

[20] S. Li, "A review of feature detection and match algorithms for localization and mapping," vol. 231, no. 1, p. 012003, 2017.

[21] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[22] T. Deepanshu, "Introduction to FAST (Features from Accelerated Segment Test)." `https://medium.com/analytics-vidhya/introduction-to-fast-features-from-accelerated-segment-test-4ed33dde6d65`.

[23] S. Li, "A review of feature detection and match algorithms for localization and mapping," in *IOP Conference Series: Materials Science and Engineering*, vol. 231, p. 012003, IOP Publishing, 2017.

[24] S. Leutenegger, M. Chli, and R. Siegwart, "Brisk: Binary robust invariant scalable keypoints," in *2011 IEEE international conference on computer vision (ICCV)*, pp. 2548–2555, Ieee, 2011.

[25] S. A. K. Tareen and Z. Saleem, "A comparative analysis of sift, surf, kaze, akaze, orb, and brisk," in *2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, pp. 1–10, IEEE, 2018.

[26] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, "A Survey of the Recent Architectures of Deep Convolutional Neural Networks," *arXiv*, jan 2019.

[27] H. Altwaijry, A. Veit, and S. Belongie, "Learning to Detect and Match Keypoints with Deep Architectures," 2016.

[28] Y. Ono, E. Trulls, P. Fua, and K. Moo Yi, "LF-Net: Learning Local Features from Images,"

[29] K. M. Yi, E. Trulls, V. Lepetit, and P. Fua, "LIFT: Learned Invariant Feature Transform," mar 2016.

[30] Y. Verdie, K. Moo Yi, P. Fua, and V. Lepetit, "TILDE: A Temporally Invariant Learned DEtector,"

[31] F. Gaisser, S. Peeters, B. Lenseigne, P. Jonker, and D. Oepkes, "Stable Image Registration for In-Vivo Fetoscopic Panorama Reconstruction," *Journal of Imaging*, vol. 4, no. 1, p. 24, 2018.

[32] E. Salahat and M. Qasaimeh, "Recent advances in features extraction and description algorithms: A comprehensive survey," *Proceedings of the IEEE International Conference on Industrial Technology*, pp. 1059–1063, 2017.

[33] "GitHub - gift-surg/endocal: A cross-platform, compact GUI application for the optical distortion calibration of fluid-immersed endoscopes." `https://github.com/gift-surg/endocal`.

[34] E. B. Jelin, S. C. Schecter, K. D. Gonzales, S. Hirose, H. Lee, G. A. Machin, L. Rand, and V. A. Feldstein, "Guide Wire Assisted Catheterization and Colored Dye Injection for Vascular Mapping of Monochorionic Twin Placentas," *Journal of Visualized Experiments*, no. 55, p. 2837, 2011.

[35] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," *: Medical Image Computing and Computer-Assisted Intervention*, 2015.

[36] Chollet and François, "Keras."

[37] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," *arXiv*, 2015.

[38] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, International Conference on Learning Representations, ICLR, 2015.

[39] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[40] M. Zuliani, "Ransac for dummies with examples using the ransac toolbox for matlab^{TM} & octave and more," tech. rep.

[41] I. Goodfellow, "NIPS 2016 Tutorial: Generative Adversarial Networks," dec 2016.

Appendices

Placental surface in relation to birth weight in MCDA twin pregnancy

Myrthe Buser

February 19, 2020

Abstract

Birth weight discordance can be present between both members of a monochorionic amniotic twin. Placental share discordance is likely an important factor in the development of birth weight discordance, but the exact influence of this placental share in combination with other placental factors remains unclear. This article presents the first six inclusions in a study aiming to determine the effects of these placental factors. In the first six inclusions, a relationship between the placental share and the birth weight was suggested, but this could not be statistical confirmed yet. It is recommended to remain the inclusion in this study, while performing an inter- and intra observer analyses with the first included results.

1 Introduction

Monochorionic pregnancies are associated with higher complication rates than dichorionic pregnancy. Monochorionic pregnancy can be subdivided in twins sharing one amniotic sac, called monochorionic monoamniotic (MCMA) pregnancy, or each twin having their own amniotic sac, called monochorionic diamniotic (MCDA) pregnancy. Both MCMA and MCDA pregnancy can be complicated by Twin-to-Twin-Transfusion Syndrome (TTTS) or Twin Anemia Polycythemia Sequence (TAPS), both caused by shared placental vessels. Even when a MCDA or MCMA pregnancy is not complicated with any of these conditions, (birth) weight discordance between both members of the twins can be present. Feng et al. showed that that there is a significant larger incidence of a weight difference >25% in MCDA twins compared to dichorionic twins (6% incidence versus incidence 11%) [1]. Although in the review of D'Antonio et al. it is discussed that a weight discordance of >15% is a risk factor for intrauterine death, the effects on perinatal outcome remain unclear [2]. However, the systematic review of Di Mascio et al. did show birth weight discordance to be an indicator for neonatal morbidity [3].

Because the implications twin weight discordance might have, several research have been conducted to identify placental factors involved in weight discordance in monochorionic twins. The first important factor is the placental surface present for each twin. Victoria et al. showed a relation between small placental weight and severe weight discordance in DCDA twins (>25%) [4]. However, placental ratio's for MCDA twins were not calculated. Souza et al. showed that IUGR is related to a lower placental weight, how-

ever they did not take into account the vascularity in MCDA pregnancies [5]. Chang et al. performed a small study in which placentas were divided along the vascular equator and weighted [6]. They showed that all twins with IUGR had the smaller placental share. Fick et al. used dye injection of the placenta to determine the placental share and found that unequal placental sharing is a risk factor for birth weight discordance [7]. Lewi et al. used placental angiography to calculate net transfusion and found that this is a bigger number for twins with a placental territory discordance, suggesting a self-limiting effect for unequal placental sharing [8]. Other factors in relation to birth weight discordance has been researched as well. De Paepe et al. showed that the type of umbilical cord insertion is related to birth weight discordance [9]. Konno et al. compared the type of anastomoses (arterial-arterial (AA), venous-venous (VV), venous-arterial (VA)) showed that VV anastomoses are an risk factor for birth weight discordance [10]. Sun et al. propose that the presence of large AA anastomoses and a short umbilical cord distance protect the twin with the smaller placental surface for a smaller birth weight [11].

As shown by the research discussed above, a lot of different (placental) factors play a role in the birth weight of MCDA twins. This study aims to combine the placental share ratios with the vascularity in order the combined effect on birth weight. This article describes the first six results and provides guidance for the continuation of the study.

2 Method

2.1 Study population

A single centre prospective cohort study was performed. All MCDA twins born in Radboudumc were included during the period from August 2019 to January 2020. Informed consent was signed by the mother of the twins. Exclusion criteria were:

- No informed consent
- No placenta available
- Technical impossibility to dye inject the placenta or determine placental surface area, including:
  - Prenatal death of one or both of the foetuses leading to hardening of the placenta
  - MCMA pregnancy
- Incomplete labelling of birth order

2.2 Analysis of placental surface

Dye injection of the placenta was performed to visualise the placental vascularity [12]. Catheters were introduced in the umbilical artery, after which for child 1 the veins were injected with red dye and the arteries with blue dye. For child 2, green and yellow dye was used for respectively the veins and arteries [12]. The distance between the umbilical cord insertions was measured. Pictures of the placental surface were taken for further analyses.

2.3 Data analyses

From the medical records of the mother the following parameters were obtained:

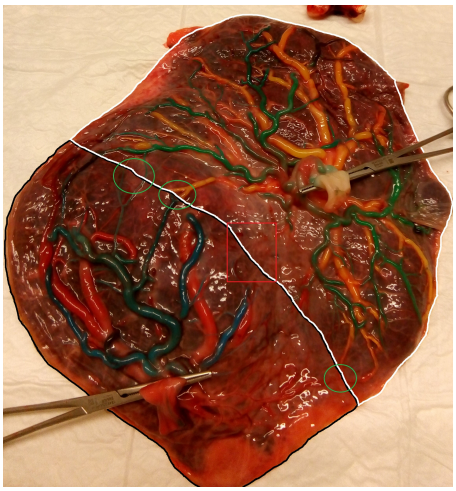


Figure 1: An example of the analyses of the placental surface. Child 1 is annotated in black. Child 2 is annotated in white. The red circle shows an anastomoses were a line could be drawn in-between. The green area is an area without a clear separation.

- Gestational age
- Maternal co morbidity
- Antenatal complications, including:
  - TTTS
  - TAPS
  - IUG

From the records of both the twins the following parameters were obtained:

- Birth weight
- Inter-twin reticulocyte count ratio
- Apgar scores at one and five minutes
- Hemoglobine (HB)

The pictures of the placental surface were used to obtain the following parameters:

- Relative placental surfaces
- Absolute placental surfaces
- Position of umbilical cord insertion Absolute placental surfaces for both foetuses

Before analyses, all pairs of children were grouped together based on birth weight, with the larger children in one group and the smaller children in the second. In *IBM SPSS Statistics 23,* the descriptives of the dataset were calculated. A independent samples T-test was performed on the placental ratios to calculate the difference between the birth weight groups.

3 Results

In table 1, the characteristics of the study population can be seen. Note that the total count

of maternal complications is bigger than six, because one mother had multiple antenatal complications. The mean birth weight of group 1 (M = 2388, SD =808) was lower than of group 2 (M = 1870, SD = 518). However, this difference was not significant ($t(10) = 1.321$, p = .308). The mean of placental ratio of the group with a higher birth weight was bigger ($M = 0.55$, $SD = 0.8$) than for the group with the higher birth weight ($M = 0.44$, $SD = 0.08$). However, this difference was not significant either ($t(10) = 0.933$, $p = 0.933$).

4 Discussion

This article presented the first six inclusions of the MCDA twin birth weight study. In table 1, the characteristics of the study population can be seen. The two members of a twin were divided based on birth weight. The mean birth weight for the smaller child group was 2388 grams and 1870 grams for the smaller child group. However, this difference was not significant, due to the low number of inclusions. Furthermore, it could not be proven that the placental ratio for both groups was significantly different. The main research question therefore cannot be answered until more twins are included. However, based on the results there does seem to be a trend, with a mean placental ratio of 0.55 for the group of bigger children versus 0.44 for the group of smaller children.

When enough children are included, different parameters can be combined with the placental ratio. Sun et al. found a limiting effect of AA anastomoses and short umbilical cord distance on uneven placental share [11]. In our population, a mean number of 0.7 AA anastomoses were present. This number was comparable to the number of VV anastomoses, but a lot lower than the VA and AV anastomoses. It is important to make the distinction between the VA and AVs, because the difference in orientation is likely to have a different effect on placental weight. When more twins are included, the anastomoses, umbilical cord distance and placental ratios can be combined together in a logistic regression model to analyse the effect of these parameters in different combinations.

In figure 1, the method of analyses of the placental surfaces is depicted. The dividing of the placenta was guided by the placental anastomoses. In the green circles, three examples of a clear division between placental vessels can be seen. In the part without anastomoses or in anastomoses in which the separation is not clear, the midline between the anastomoses before and after this area was chosen. Although these rules were followed, the analyses of the placentas are likely to are observer depended. Therefore, it is recommended to do a inter- and intra-observer analyses. This analyses can be done parallel with the continuation of the inclusions, with the first inclusions. Depending on the outcome of this analyses, it might be necessary to take the mean of several observers for the final analyses.

Table 1: The characteristics of the study population.

Maternal complication	**Count**
Medicated colitis ulcerosa	1
Pre-enclampsia	1
Anemic	1
Gestational diabetes	1
None	3
Antenatal complications	**Count**
TAPS	1
IUGR	2
TTTS	1
None	2
Gestational age (days)	**Mean, Min, Max (std)**
(n=6)	240, 210, 256, (17.9)
HB	**Mean, Min, Max (std)**
Bigger child (n=4)	10.4, 9.0, 13.5 (2.0)
Smaller child (n=5)	10.8, 8.6, 14.6 (2.3)
Birth weight (g)	**Mean, Min, Max (std)**
Bigger child (n=6)	2388, 1410, 3688 (808)
Smaller child (n=6)	1870, 1210, 2368 (518)
Relative placental surface	**Mean, Min, Max (std)**
Bigger child (n=6)	0.55, 0.36, 0.84 (0.19)
Smaller child (n=6)	0.44, 0.15, 0.63 (0.20)
Absolute placental surface (cm^2)	**Mean, Min, Max (std)**
Bigger child (n=3)	238, 59, 321 (75)
Smaller child (n=3)	238, 59, 386 (165)
Number of anastomoses	**Mean, Min, Max (std)**
Total (n=6)	6.7, 2, 12 (3.8)
AA	0.7, 0, 2 (0.8)
VA	2.0, 0, 5 (2.0)
AV	3.0, 1, 6 (2,1)
VV	0.7, 0, 1 (0.7)
Umbilical cord distance	**Mean, Min, Max (std)**
(n=4)	17.0, 12, 30 (8.7)

5 Conclusion

The trend of birth weight and placental ratio in the included twins suggest that there might be a correlation between placental ratio and birth weight in MCDA twins. It is recommended to continue including MCDA twins, after which a more thorough analysis of the effect of placental ratio, placental anastomoses and umbilical cord distance can be performed. This would provide valuable information, not only about healthy twins, but about the placental factors related to TTTS and TAPS as well.

References

[1] B. Feng, J. Zhai, and Y. Cai, "Effect of twin pregnancy chorionic properties on maternal and fetal outcomes," *Taiwanese Journal of Obstetrics and Gynecology*, vol. 57, pp. 351–354, jun 2018.

[2] F. D'Antonio, A. O. Odibo, F. Prefumo, A. Khalil, D. Buca, M. E. Flacco, M. Liberati, L. Manzoli, and G. Acharya, "Weight discordance and perinatal mortality in twin pregnancy: systematic review and meta-analysis.," *Ultrasound in obstetrics & gynecology : the official journal of the International Society of Ultrasound in Obstetrics and Gynecology*, vol. 52, no. 1, pp. 11–23, 2018.

[3] D. Di Mascio, G. Acharya, A. Khalil, A. Odibo, F. Prefumo, M. Liberati, D. Buca, L. Manzoli, M. E. Flacco, R. Brunelli, P. Benedetti Panici, and F. D'Antonio, "Birthweight discordance and neonatal morbidity in twin pregnancies: A systematic review and meta-analysis," 2019.

[4] A. Victoria, G. Mora, and F. Arias, "Perinatal outcome, placental pathology, and severity of discordance in monochorionic and dichorionic twins," *Obstetrics and Gynecology*, vol. 97, no. 2, pp. 310–315, 2001.

[5] M. A. Souza, M. D. L. Brizot, S. E. Biancolin, R. Schultz, M. H. Burlacchini de Carvalho, R. P. Vieira Francisco, and M. Zugaib, "Placental weight and birth weight to placental weight ratio in monochorionic and dichorionic growth-restricted and non-growth-restricted twins," *Clinics*, vol. 72, no. 5, pp. 265–271, 2017.

[6] Y. L. Chang, S. D. Chang, A. S. Chao, P. C. Hsieh, C. N. Wang, and L. H. Tseng, "The individual fetal weight/estimated placental weight ratios in monochorionic twins with selective intrauterine growth restriction," *Prenatal Diagnosis*, vol. 28, no. 3, pp. 217–221, 2008.

[7] A. L. Fick, V. A. Feldstein, M. E. Norton, C. Wassel Fyr, A. B. Caughey, and G. A. Machin, "Unequal placental sharing and birth weight discordance in monochorionic diamniotic twins.," *American journal of obstetrics and gynecology*, vol. 195, pp. 178–83, jul 2006.

[8] L. Lewi, M. Cannie, I. Blickstein, J. Jani, A. Huber, K. Hecher, S. Dymarkowski, E. Gratacós, P. Lewi, and J. Deprest, "Placental sharing, birthweight discordance, and vascular anastomoses in monochorionic diamniotic twin placentas," *American Journal of Obstetrics and Gynecology*, vol. 197, no. 6, pp. 587.e1–587.e8, 2007.

[9] M. E. De Paepe, S. Shapiro, L. E. Young, and F. I. Luks, "Placental weight, birth weight and fetal:placental weight ratio in dichorionic and monochorionic twin gestations in function of gestational age, cord insertion type and placental partition," *Placenta*, vol. 36, pp. 213–220, feb 2015.

[10] H. Konno, T. Murakoshi, A. Yamashita, and M. Matsushita, "Roles of venovenous anastomosis and umbilical cord insertion abnormalities in birthweight discordance in monochorionic–diamniotic twin pregnancies without twin–twin transfusion syndrome," *Journal of Obstetrics and Gynaecology Research*, vol. 44, pp. 623–629, apr 2018.

[11] L. M. Sun, Y. Li, G. Zou, F. Zhou, X. Lei, D. S. Cram, D. Oepkes, and J. Wu, "Monochorionic twins with unequal placental sharing: Why can the outcome still be favorable?," *Journal of Maternal-Fetal and Neonatal Medicine*, vol. 29, pp. 1261–1264, apr 2016.