

Phase Recognition in Coronary Angiography Procedures

A Random Forest Approach Using
Video and C-arm System Logging Data

Biomedical Engineering Msc Thesis
Y Weijenberg

Phase Recognition in Coronary Angiography Procedures

A Random Forest Approach Using
Video and C-arm System Logging Data

by

Y Weijenberg

Student Name	Student Number
Yannick	4467930

Supervisor: Prof.dr. John J. van den Dobbelen TU Delft
Guidance: Teddy S. Vijfvinkel TU Delft
Project Duration: March, 2023 - November, 2023
Faculty: Faculty of Mechanical, Maritime
and Materials Engineering, Delft

Cover: Picture Generated By Photoshop
Style: TU Delft Report Style, with modifications by Daan Zwan-
eveld

summary

0.1. Introduction

The focus of this thesis is surgical phase recognition (SPR) for cardiac catheterization procedures. With the anticipated increase of the annual prevalence of coronary artery disease, this study gains relevance for its potential to enhance efficiency in coronary angiography procedures[23]. The study employs machine learning algorithms to analyze various data sources, including C-arm logs and video recordings, to assess its effectiveness for phase recognition for coronary angiography procedures.

0.2. Research Objective

The research question of the thesis is as follows:

”Can data from C-arm logs and video recordings collected in a catheterization laboratory be effectively used for phase recognition of coronary angiography procedures to aid in monitoring and predicting the remaining time of the procedure?”

0.3. Discussion and Results

- **Baseline Model:** A basic model using average phase durations achieved 45% accuracy. The model is limited by its inability to predict in real-time and its reliance on procedure duration.
- **C-arm Model** Achieved 80.73% accuracy, but this accuracy is influenced by class imbalance. High concentration of data in operative phases lead to a higher performance for these phases, however, performance was poor for other phases.
- **Object Detection Model** Recorded 63.8% accuracy, excelling in initial and final procedure phases but less reliable during operational stages.
- **Combined Data Model** Yielded 79.46% accuracy, demonstrating improved performance across most phases.
- **Reduced Granularity Model** Showed that decreasing detail in phase classification increased total accuracy to 88.23%, indicating a trade-off between granularity and performance.

0.4. Clinical Implementation

The combined model’s suitability for CAG phase recognition is promising, especially when granularity is reduced. The integration of C-arm and video data offers comprehensive procedure coverage and improved accuracy. However, challenges remain in predicting specific phases that only happen sometimes accurately.

0.5. Limitations and Future Research

The study acknowledges limitations in the non-temporal nature of the random forest model, and potential information loss from the object detection model. Future research directions include expanding dataset sizes, implementing models better suited for temporal data, and optimizing the balance between granularity and performance for clinical applications.

0.6. Conclusion

This thesis contributes to the understanding of surgical phase recognition in cardiac catheterization laboratories, particularly in coronary angiography procedures. The research demonstrates that the application of machine learning models to data from C-arm logs and video recordings can lead to accurate phase recognition, albeit with certain limitations in prediction of conditional phases and model granularity. The advancements in phase recognition as explored in this thesis could have implications for both the operational efficiency of cath labs and the overall quality of patient care in cardiovascular interventions. Future research should continue to explore these avenues, aiming to develop more robust and clinically applicable models for phase recognition in cath labs.

Contents

summary	iii
0.1 Introduction	iii
0.2 Research Objective	iii
0.3 Discussion and Results	iii
0.4 Clinical Implementation	iii
0.5 Limitations and Future Research	iii
0.6 Conclusion	iv
1 Introduction	1
1.0.1 Introduction	1
1.0.2 AI's Growing Role in Healthcare	1
1.0.3 Machine Learning and SPR	1
1.0.4 Annotations and Surgical Process Modeling	1
1.0.5 SPR for the Cathlab	2
1.1 Research Objective	2
1.1.1 Effectivity	3
2 Background	5
2.1 Overarching Project	5
2.2 Coronary Angiography	6
2.2.1 Catheterization Laboratory	6
2.3 Machine Learning	8
2.3.1 Decision Trees	9
2.4 Random Forest and Feature Importance Analysis	10
2.4.1 Random Forest	10
2.4.2 Feature Importance: Mean Decrease in Impurity	10
2.4.3 Model Selection	11
2.5 Literature Review: Machine learning in Coronary Angiography Procedures	11
2.5.1 Granularity in Phase Recognition	11
2.5.2 Potential Data Sources for CAG Procedures	11
2.5.3 Machine Learning Algorithms for CAG Procedures	12
3 Data	13
3.1 Video Data	14
3.1.1 Object Detection Data	14
3.2 C-Arm Data	16
3.2.1 Exam	16
3.2.2 Acquisitions	16
3.2.3 Movements	18
3.3 Clinical Data	19
3.4 Annotations Data	20
4 Methodology	21
4.1 Formalization of the Workflow	21

4.2	Annotation Process	23
4.3	Video Data Object Detection	23
4.4	Criteria for Assessing Effectiveness	24
4.4.1	Performance Metric	24
4.4.2	Defining Performance Benchmarks	24
4.5	Data Preprocessing	26
4.5.1	Logging Data	26
4.5.2	Clinical	27
4.5.3	Object Detection Data	27
4.5.4	Annotations	29
4.6	Data Integration	30
4.6.1	Merge 1: Logging: Acquisitions & Logging: Exam	31
4.6.2	Merge 2: Add Movements to Merge 1	31
4.6.3	Merge 3: Add Annotations to Merge 2	32
4.6.4	Merge 4: Add Video Data to Merge 3	37
4.6.5	Merge 5: Add Clinical Data to Merge 4	38
4.7	Feature Engineering	38
4.7.1	Temporal Features	38
4.7.2	Object Detection Features	40
4.8	Feature Matrix	41
4.8.1	Splitting Data Sources For Recognition	43
4.9	One-Hot Encoding	43
4.10	Random Forest Classification	44
4.10.1	hyper-parameter Tuning: Random Search and Grid Search	44
4.10.2	Validation	45
4.11	Reducing Granularity	46
4.11.1	Random Forest Model Training	46
4.12	Phase Recognition for Predicting Procedure Delay	48
4.12.1	Data	48
4.12.2	Annotations	48
4.12.3	Random Forest Classification	48
4.12.4	Model Performance	48
5	Results	49
5.1	Performance	49
5.2	Feature Importance	51
5.3	Example Procedure Visualization	51
5.4	Performance of Model with Reduced Granularity	51
5.5	Using Phase Recognition for Predicting Procedure Delay	51
6	Discussion	57
6.1	Interpretation of the Classification Results	57
6.1.1	Baseline Model	57
6.1.2	C-arm Model	57
6.1.3	Object Detection Model	58
6.1.4	Combined Data Model	58
6.1.5	Reduced Granularity Model	59
6.1.6	Predictions of Delay Model	59
6.1.7	Clinical Implementation and Research Objectives	59
6.2	Limitations and Future Research Directions	60

References	63
A Full Dataset Tables	65
A.1 C-Arm Logging Data	65
A.1.1 Clinical Data	66

1

Introduction

1.0.1. Introduction

The application of artificial intelligence (AI) in healthcare has significantly grown in recent years [1]. One of the applications of AI in the healthcare sector that has seen rapid growth is surgical phase recognition (SPR). As cardiovascular disease remains a leading cause of mortality worldwide, the potential of SPR in cardiac catheterization procedures warrants exploration [5].

1.0.2. AI's Growing Role in Healthcare

The integration of artificial intelligence (AI) into the healthcare sector has been a topic of increasing research and discussion in recent years [1]. One growing area of research over the last years is SPR. SPR focuses on recognizing the procedural phases of clinical procedures. This innovation can lead to a variety of applications. Examples of such applications include contributions to an intelligent operating room (OR) where phase recognition can help realize context-aware assistance systems. These systems will be able to work in partnership with clinical workers to actively tackle increasingly complex procedures [10]. Other potential applications include the implementation of dynamic scheduling systems and automatic procedure analysis. Overall, employing AI for phase recognition could enhance patient care and streamline clinical processes, either directly or through subsequent technological advancements.

1.0.3. Machine Learning and SPR

Machine learning (ML) is a subfield of AI that focuses on the development of algorithms that can be trained to recognize patterns in data. These algorithms can be used to make predictions based on these data and are used in SPR to predict the procedural phase of procedures. Although laparoscopic video data has been the predominant data source for SPR, especially in laparoscopic cholecystectomy procedures [24] [11], other data sources like audio [15], instrument usage [6], and robot kinematic data [8] have also been explored.

1.0.4. Annotations and Surgical Process Modeling

The annotation of the data used for SPR is typically done manually. The effectiveness of this annotation process relies on a clear definition of phases, which is where the field of surgical process modeling (SPM) becomes relevant [20]. SPM is a methodological approach aimed at systematically describing, analyzing, and modeling surgical procedures. It seeks to structure and make sense of the complex series of tasks that make up a surgical procedure. A key aspect of SPM is the concept of 'granularity,' which refers to the level of detail at which a

procedure is described [20]. Lalys and Jannin describe six levels of granularity to describe a procedure, namely, low-level information, motions, activities, steps, phases, and procedures [20].

1.0.5. SPR for the Cathlab

A catheterization laboratory (cathlab) is a clinical examination room that is used for cardiac catheterization purposes. These procedures involve the introduction of a catheter into the radial or femoral artery for diagnostic or therapeutic purposes. The primary objective is to diagnose and treat cardiovascular diseases, which are the leading cause of death worldwide [5]. In the Netherlands, the annual prevalence of coronary artery disease, the most common type of cardiovascular disease, is expected to increase by 43% between 2018 and 2040, according to the Dutch National Institute of Health and Environment [23]. This projected increase in the annual prevalence of cardiovascular disease could lead to a higher demand for cardiac catheterization procedures.

While phase recognition has seen significant advancements in fields like laparoscopic surgery, with some studies showcasing accuracy rates exceeding 90% [22][6], its potential applications in cathlabs remain largely unexplored [24]. The relevance of phase recognition in the cathlab lies in its potential to improve the efficiency of cardiac catheterization procedures. By tracking the progress of these procedures, phase recognition can help identify and address potential delays or complications, which can reduce the overall duration of the surgery. This can result in cost savings for the hospital and less pressure on the healthcare system, as shorter procedures require fewer resources and can be performed on more patients in a given time period. Furthermore, the implementation of context-aware assistance systems, which utilize phase recognition technology to adapt to the real-time circumstances and surgical needs, can improve the accuracy and precision of the procedures. This could reduce the likelihood of complications that can reduce the need for additional treatment or follow-up care. This can also result in cost savings for hospitals and can contribute to the achievement of key medical ethical principles, as it can help ensure that patients receive the best possible care.

1.1. Research Objective

Given the context presented in the introduction above, the research objective of this paper is to address the following research question:

”Can data from C-arm logs and video recordings collected in a catheterization laboratory be effectively used for phase recognition of coronary angiography procedures to aid in monitoring and predicting the remaining time of the procedure?”

To address this question, the following subquestions have been formulated:

- Is the model’s accuracy suited for basic applications of CAG phase recognition, such as remaining time predictions?
- How does the model’s performance compare to a baseline prediction based on average phase length?
- In terms of predictive accuracy, what advantages are offered by combining data from C-arm logs and video recordings for the purpose of phase prediction?

After initial phase recognition, this study also aims to utilize the results to perform a simple estimated remaining-time classification to assess the usability of the results in a clinical context.

Additionally, the model's effectiveness will be evaluated using a broader, less granular workflow definition that retains clinical relevance, to determine any performance improvements and the resulting trade-offs in usability.

1.1.1. Effectivity

Effectivity in this research is the ability of phase recognition to achieve a level of accuracy, consistency, and reliability that makes it practical for monitoring of coronary angiography procedures and for predicting the remaining duration of the procedure with a margin of error deemed acceptable within clinical settings. Specific requirements are formalized in Section 4.4

2

Background

This chapter provides the necessary background for this thesis. Section 2.1 details the project within which this thesis was conducted. The fundamental concepts of coronary angiography procedures are outlined in section 2.2, while section 2.2.1 gives a comprehensive overview of what a catheterization laboratory is. In section 2.3, the basic principles of machine learning are discussed, with a specific focus on the random forest model and its functionality. A review of the existing literature on phase recognition for medical procedures can be found in section 2.5.

2.1. Overarching Project

This thesis is part of an overarching project that aims to study the catheterization laboratory workflow at the Reinier de Graaf Gasthuis (RDGG) hospital in Delft. The goal of the project is to optimize cathlab efficiency in the context of a high demand for cardiac catheterization procedures. At the basis lies a collaboration between Phillips Healthcare, the RDGG hospital, and the Delft University of Technology (TU Delft).

Work until now focussed on the formalization and quantification of the procedure specific workflow [12], identification of useful data sources for workflow monitoring [18], and the usability of interventional X-ray data for procedure duration prediction [16]. Furthermore, a variety of data has been collected that can be used for an analysis of the workflow of coronary angiography procedures. This data consists of videos gathered by five cameras that have been placed at different angles in the cathlab and a log of the C-arm device. In figure 2.1 the camera positions can be seen, and look at Chapter 2.2.1 for more on the C-arm machine. An object detection algorithm developed by R. Dai as part of his thesis in 2022 is able to extract useful information regarding the size and position of objects in the recorded procedures using the video data collected [7].

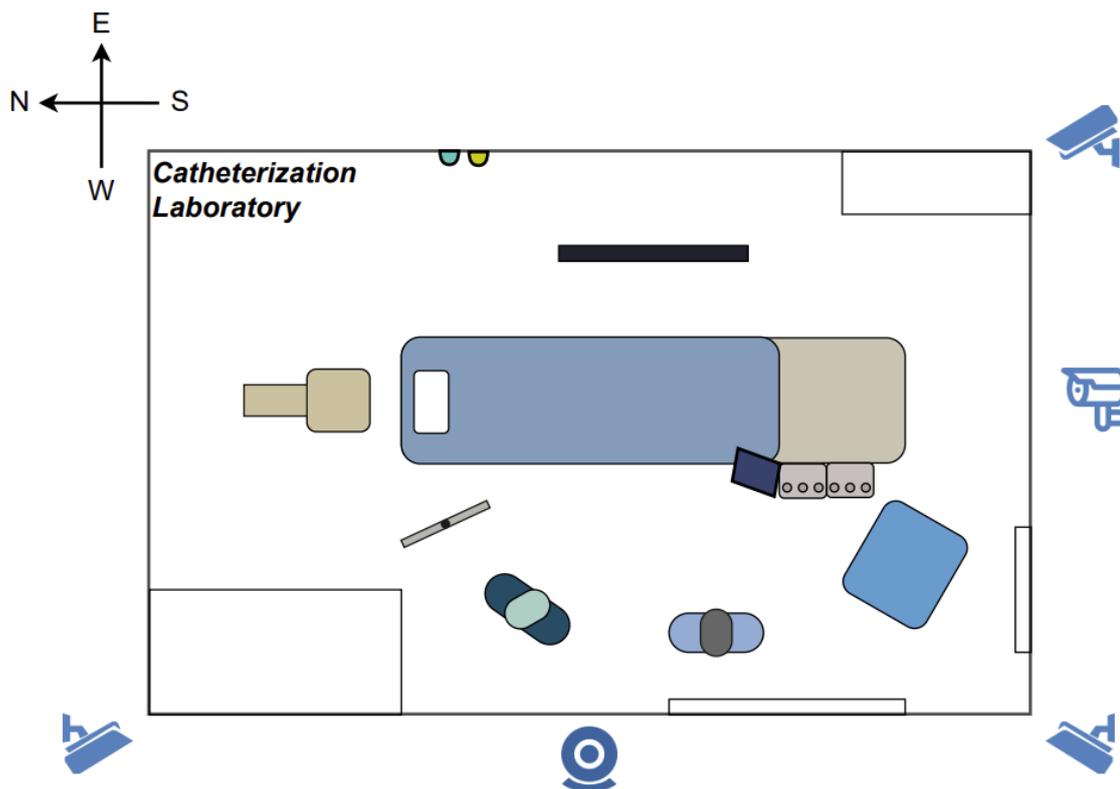


Figure 2.1: The positions of the cameras placed in the cathlab of RDGG. Figure reprinted from R. Dai [7]

2.2. Coronary Angiography

A coronary angiography is a common coronary catheterization procedure that aims to visualize blood flow through the coronary arteries of a patient. Coronary arteries are the main blood vessels that supply the heart with blood, oxygen, and nutrients. They wrap around the surface of the heart and branch out to reach the different muscle layers of the heart walls, see Figure 2.2. Coronary angiography uses X-rays in combination with a contrast fluid to visualize blood flow through the arteries. The contrast fluid is delivered via a catheter that is introduced into the coronary arteries through the radial or femoral arteries. The goal of a coronary angiography procedure is to diagnose stenosis in the coronary arteries. An example of a coronary angiogram in which stenosis can be observed, taken during a coronary angiography, can be seen in figure 2.3.

2.2.1. Catheterization Laboratory

A coronary angiography is usually performed in a catheterization laboratory (cathlab). A cathlab is a hospital room which is specialized for medical procedures that utilize catheters. The procedures are guided by real-time imaging technology, such as fluoroscopy. Fluoroscopy utilizes x-ray radiation to make real-time video of the patient's internal anatomy, which allows physicians to navigate catheters within the body and helps visualize the blood flow in the coronary arteries during a coronary angiography.

A cathlab typically includes two primary areas: a control room and a procedure room [19]. The control room serves as the monitoring hub, equipped with control panels and display screens for observation and supervision of procedures [16]. The procedure room is the active

site where the medical procedures take place, equipped with all necessary instruments and equipment. The centerpiece of the procedure room's equipment is the C-arm, as shown in figure 3.3. This device plays a critical role in generating and visualizing X-ray images. The C-arm has the ability to rotate around the patient, capturing images from multiple angles.

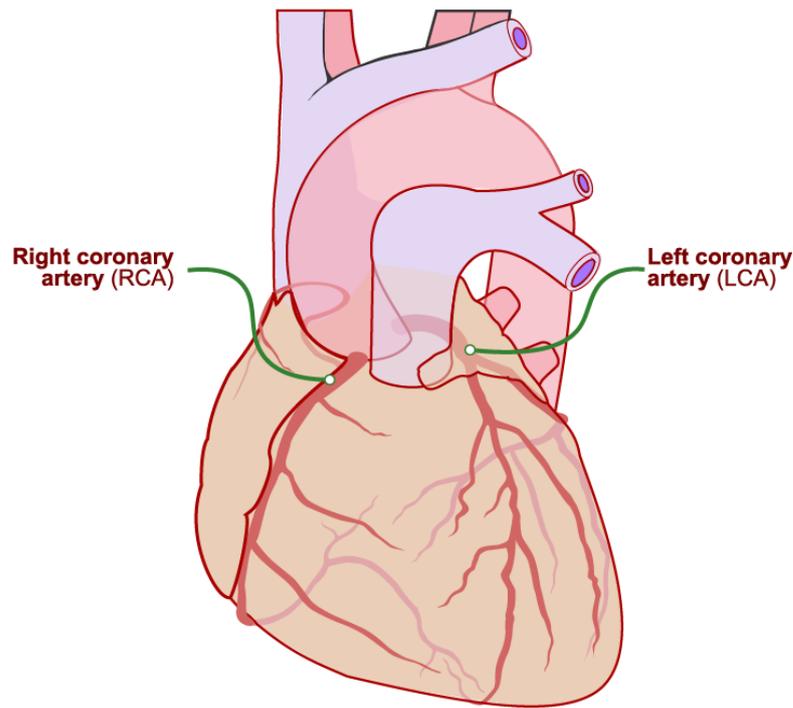


Figure 2.2: The main coronary arteries

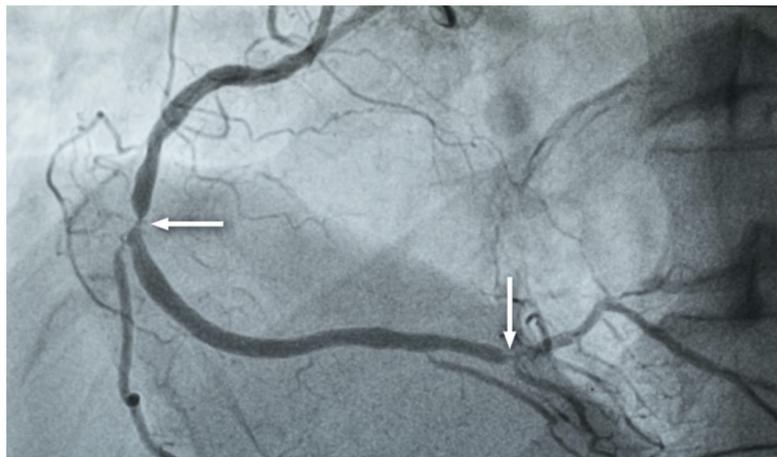


Figure 2.3: Coronary angiogram with stenosis on 2 locations

2.3. Machine Learning

Machine learning (ML) involves the development of algorithms and statistical models that enable computers to identify patterns in data, leading to informed decision making with minimal human intervention. A subset of ML is supervised learning. In supervised learning, algorithms are trained on labeled data, where each instance is paired with a corresponding output or label. Once trained, the algorithm can predict outputs for new, unseen data.

The use of machine learning has become increasingly prevalent in all areas of research, including healthcare. This is evident from the rapidly growing number of articles on the subject in databases such as PubMed in recent years [1]. With this trend comes an increased need for the availability of well-annotated data in healthcare settings [3].

It is important to note that, while increasing the availability of data offers numerous opportuni-

ties for advancements in healthcare, it also brings about its own set of challenges, including, but not limited to, maintaining data privacy, which can be especially important in the clinical setting [21].

The next sub-chapters will be on a specific ML model which was used in this thesis.

2.3.1. Decision Trees

A decision tree (DT) is a type of ML model that has a tree-like structure. When a DT is used for classification, it is called a classification tree, and when it is used to predict numerical values, it is called a regression tree. For the purpose of this thesis, we will focus on classification trees.

Decision Trees: Splits, Purity, and Gini Impurity

Decision trees work by recursively splitting data into subsets based on feature values, with the aim of maximizing the purity of data in each subset.

Splits: A split divides the data into two sets based on a feature value. These sets are the 'child nodes' of the data before the split. The goal is to segregate the data so that each child node is more homogeneous (or pure) than before the split.

Purity: Purity in the context of decision trees refers to how homogeneous the data is in a node. A node is considered pure if it contains data points from only one class. The objective of the decision tree is to create splits that increase the purity of child nodes.

Gini Impurity: Gini impurity is a metric that quantifies the impurity of a set. For classification:

$$Gini(p) = 1 - \sum_{i=1}^J (p_i)^2$$

Where $Gini(p)$ is the Gini impurity, J is the number of classes in the classification problem, and p_i is the proportion of instances that belong to class i . The lower the Gini impurity, the more pure the node is. When making a split, the decision tree algorithm aims to minimize the Gini impurity in the child nodes.

Example: Predicting Fruit Based on Color and Size

Imagine we have a dataset of fruits: apples and strawberries. We want to predict the type of fruit based on two features: color (green or red) and size (small or large).

Our dataset:

Color	Size	Fruit
Green	Small	Apple
Green	Large	Apple
Red	Small	Strawberry
Red	Large	Apple
Red	Small	Strawberry

To create a decision tree, we start with the full dataset, called the 'root' and evaluate potential splits. Let's consider splitting by color first. If we split by color:

- Green node: 2 Apples
- Red node: 1 Apple, 2 Strawberries

Purity in our split by color:

- The Green node is pure (100% Apples).

- The Red node is impure (33% Apples, 67% Strawberries).

For the red node (with 1 Apple and 2 Strawberries):

$$GiniImpurity = 1 - \left[\left(\frac{1}{3} \right)^2 + \left(\frac{2}{3} \right)^2 \right] = \frac{4}{9}$$

The Gini impurity is $\frac{4}{9}$, indicating some level of impurity.

The example shows the basic principle of a decision tree. Once the tree has been generated from the labeled data it can be used for predictions on new data. The main benefits of using a decision tree are that it is easy to understand, has a transparent prediction process, and can work with both numerical and categorical data [13]. A limitation of decision trees is that they can easily overfit, meaning that they perform well on training data, but may not generalize well to new data [13].

2.4. Random Forest and Feature Importance Analysis

2.4.1. Random Forest

A random forest (RF) is an ensemble machine learning (ML) method that combines multiple decision trees to produce a more robust and accurate model. Ensemble learning techniques, such as RF, combine the predictions of several base estimators to improve the generalizability and robustness of a single estimator [13]. RFs use two main methods to achieve ensemble learning: bootstrap aggregation, which is often termed as 'bagging', and feature randomness.

Bootstrap Aggregation (Bagging): Bagging is a technique in which multiple subsets of the dataset are created using sampling with replacement, which means that datapoints can be sampled more than once [17]. In the context of RF, each decision tree is trained on a different subset. For instance, from an original dataset of $[1, 2, 3, 4, 5]$, a bootstrapped sample might look like $[5, 3, 5, 2, 5]$.

Feature Randomness: In addition to bagging, RF introduces the concept of feature randomness. Each decision tree in the forest is trained using a random subset of features, adding an additional layer of diversity to the model.

The final prediction of the RF model is determined by aggregating the predictions of all individual trees, typically through a majority vote for classification tasks. This ensemble approach, combined with bagging and feature randomness, helps reduce overfitting, making the model more generalizable to unseen data [25].

2.4.2. Feature Importance: Mean Decrease in Impurity

: In the context of decision trees, impurity is a measure of how mixed the classes are in a set. For classification tasks, measures like the Gini impurity or entropy are commonly used. When a decision tree makes a split on a feature, it aims to reduce this impurity, leading to purer child nodes.

The importance of a feature is then computed as the total reduction of the impurity brought about by that feature, averaged across all trees in the forest. In other words, if a feature consistently results in nodes with high purity across many trees, it will have a high MDI value, indicating the importance of the feature in the model.

However, it is worth noting that while MDI provides valuable insights, it might sometimes be biased towards features with a high number of unique values or high cardinality. This is because such features can create more complex decision boundaries, leading to more splits, and

consequently, a higher calculated importance [9]. Despite this potential bias, MDI remains a widely used and insightful metric of feature importance in RF models.

2.4.3. Model Selection

A random forest model was selected because it is a relatively straightforward model that has been extensively used in research on phase recognition using machine learning [24]. In addition, a simple model can provide an indication of the usefulness of the data which can lead to the use of more complex models to maximize performance. And lastly, it is a good fit with the knowledge base of the author of this thesis, who does not have a computer science background.

2.5. Literature Review: Machine learning in Coronary Angiography Procedures

This literature review is based on the findings of Weijenberg, 2023 [24]. Since the concept of surgical phase recognition (SPR) for coronary angiography procedures is unknown in literature, this review will focus on the uses of SPR for other procedures to paint a picture of the research field and discuss its relevance for our research purpose. The section includes the topics of granularity in phase recognition, the potential data sources for CAG procedures, and challenges and considerations in selecting appropriate ML algorithms.

2.5.1. Granularity in Phase Recognition

In surgical workflow modeling (SPM), Granularity refers to the degree of specificity in the workflow model of the procedure. Carly et al. (2020) noted that a procedure defined with a high degree of granularity may be more clinically relevant. For example, a CAG procedure segmented into broad phases might be less insightful than a procedure that further dissects the specific phases of coronary artery imaging. However, Jumah et al. (2022) cautioned that increased granularity could compromise model performance due to the challenges in distinguishing between closely related phases. Therefore, optimal granularity is based on the quality and distinctiveness of the available data.

2.5.2. Potential Data Sources for CAG Procedures

Several data sources can be used for phase recognition in CAG procedures:

- **System Data:** The Phillips Allura device logs a range of system settings, including c-arm movement, surgical table movement, and x-ray acquisition data [18]. Dipetro et al. (2015) and Stauder et al. (2014 & 2015) have explored the utility of surgical table position as a data source for phase recognition. Notably, Stauder et al. (2014) identified the slope of the table as a significant feature of its model, although it was found that this was due to the sensor noise occurring in specific phases of the procedure and not the value of the inclination [**Stauder-2014**]. The remaining data from the system, especially those related to the c-arm device, remain unexplored in the context of phase recognition.
- **Video Data:** Although the use of endoscopic or microscopic video data is the most common in SPR research, these data sources are not directly applicable to CAG procedures. However, bird's eye video data, especially privacy-preserving depth video, presents a promising avenue. Li et al. (2016 & 2017) have demonstrated the potential of Kinect depth video in medical procedures. Depth video offers privacy advantages by displaying only silhouettes, but the resolution is low and contains less context than regular video data.

- **Instrument Data:** While direct sensor placement on instruments used in CAG procedures is not feasible, video data can provide insights into instrument usage [Weede-2012]. Bardram et al. (2011), Dipetro et al. (2015), and Stauder et al. (2014 & 2017) have shown the potential of using sensors or RFID tags for phase recognition. However, for the purpose of CAG procedures, where instruments are inserted in the body of the patient, video data may be more suitable.
- **Audio Data:** Audio transcriptions, especially in the context of c-arm machine movements, could offer valuable insights. Li et al. (2016) demonstrated the utility of Mel-frequency spectral coefficients of sound for phase recognition in trauma resuscitation [Li-2016]. In a cathlab setting, audio transcriptions might provide more context, especially when combined with other data sources.

2.5.3. Machine Learning Algorithms for CAG Procedures

The choice of ML algorithms for phase recognition in CAG procedures is influenced by several factors:

- **Real-time Requirements:** The need for real-time classification, especially for dynamic planning and context-aware systems, requires algorithms with quick inference times. Although some algorithms, such as Decision Trees (DTs), inherently offer quick inference, others may not be suitable for real-time classification [2].
- **Temporal Information:** Incorporating temporal information can improve classification performance [27]. However, this might increase the complexity of the model and, consequently, the inference time.
- **Performance Requirements:** The intended use of the model dictates its required accuracy. For example, an assistance system demands higher accuracy than a dynamic planning system.
- **Interpretability:** The early stages of research could benefit from models that provide information on the importance of characteristics, which aids in the evaluation of the usability of the data [Stauder-2014].

In summary, the literature emphasizes the balance between classification performance, inference time, and model complexity in the context of CAG procedures. While the ideal algorithm for the task of phase recognition in CAG procedures remains to be researched, the literature suggests that, given the structured workflow of CAG procedures, simpler models might suffice. Future research directions include comparing models with temporal and nontemporal components and assessing the usability of various data sources.

3

Data

This chapter will give an overview of the data sources used for this thesis. This data has been made available through previous work in the cathlab workflow project. Video data was collected by 5 cameras that were placed at different angles as stated in Section 2.1. The system logging data has been collected from the C-arm device, as has also been stated in Section 2.1. Clinically relevant patient data were collected from the HiX platform of the Reinier de Graag Gasthuis. Lastly, annotation data was generated using the Noldus Observer XT software [4]. This chapter will give an overview of the retrieval and structure of these datasets.

3.1. Video Data

The dataset contains video recordings of coronary angiography procedures conducted at the cathlab of the Reinier de Graaf Gasthuis (RDGG) hospital. Each procedure was captured from 5 angles, as can be seen in Figure 2.1, a representative sample is presented in Figure 3.1. The angles are [south east (SE), wall south (WallS), south west (SW), wall west (WallW), north west (NW)]. Of these angles, the WallW video was filmed with a wide-angle camera. All videos were recorded at a frame rate of 25 fps, as documented by R. Dai, 2022 [7].



(a) North-West Angle



(b) West Wide Angle



(c) South-West Angle



(d) South Angle



(e) South-East Angle

Figure 3.1: Examples of all angles of the video data that have been gathered in the cathlab of the Reinier de Graaf Gasthuis hospital.

3.1.1. Object Detection Data

To extract information from the video data that is useful for use in a prediction model, we use an object detection algorithm which has been developed by R. Dai in his 2022 Msc Thesis [7] as part of the project of TU Delft, RDGG, and Phillips detailed in Section 2.1. The algorithm's application is further discussed in Section 4.3. Using the object detection algorithm, bounding boxes are drawn around specified objects of interest in the videos. For the purpose of this research, the south west camera angle was used with an FPS which was reduced to 1 frames per second; more on the reasoning behind this can also be seen in Section 4.3. The model

generated a text file for every frame of all the videos that were analyzed. The text files include information on the bounding boxes that were drawn over the detected objects, their size, and location. An overview of the data can be seen in Table 3.1.

The names of the text files correspond to the date, starting and ending time of the video, the angle of the video, and the frame number. In our case, the frame number is the same as the second in the video. For example, consider the file named `20201020 - 072554_20201020 - 080000_CornerSW_16.txt`. In this name, 2020-10-20 represents the date; 07:25:54 and 08:00:00 indicate the video's start and end times, respectively; CornerSW denotes the angle of the video; and 16 corresponds to both the frame number and the 16th second of the video. Therefore, the text file contains information specifically related to the 16th second of the video."

Column Name	Description	Unit
Object	The object which was identified. Indicated with a number.	-
x	The X-coordinate of the center of the bounding box expressed as a fraction of the total screen (0-1).	-
y	The Y-coordinate of the center of the bounding box expressed as a fraction of the total screen (0-1).	-
w	The width of the bounding box expressed as a fraction of the total screen (0-1).	-
h	The height of the bounding box expressed as a fraction of the total screen (0-1).	-

Table 3.1: Overview of the Object Detection Data

3.2. C-Arm Data

The logging data of the C-Arm device in the cathlab of Reinier de Graaf Hospital was provided by Phillips Healthcare. The data covers 439 coronary angiography procedures and includes 3 data sheets: exam, acquisitions, and movements.

3.2.1. Exam

The exam sheet provides information on all procedures (exams) from which the logging data was obtained. An overview of the data given in this set can be seen in Table 3.2

Feature Name	Description	Unit
ExamID	Unique identification code of the procedure	-
ExamStartTime	Start time of the procedure	yyyy-mm-dd hh:mm:ss
ExamEndTime	End time of the procedure	yyyy-mm-dd hh:mm:ss
ExamDurationSecs	The difference between the start and end times in seconds	Seconds
PatientNumber	Unique identification code of the patient	-
PhysicianNumber	Unique identification code of the physician	-

Table 3.2: Overview of the C-Arm Data Exam Sheet

3.2.2. Acquisitions

The acquisitions sheet provides information on all X-ray acquisitions that have been performed during the procedures from which the logging data was obtained. At each acquisition, information regarding the acquisition and a variety of other system settings, such as the positions of the different components of the device, are captured. An overview of the main components of the C-arm device is shown in Figure 3.2. The axes of the C-arm logged during each acquisition can be seen in Figure 3.3. An overview of a selection of the data given in the acquisitions sheet is given in Table 3.3. The full dataset description is visible in A.1

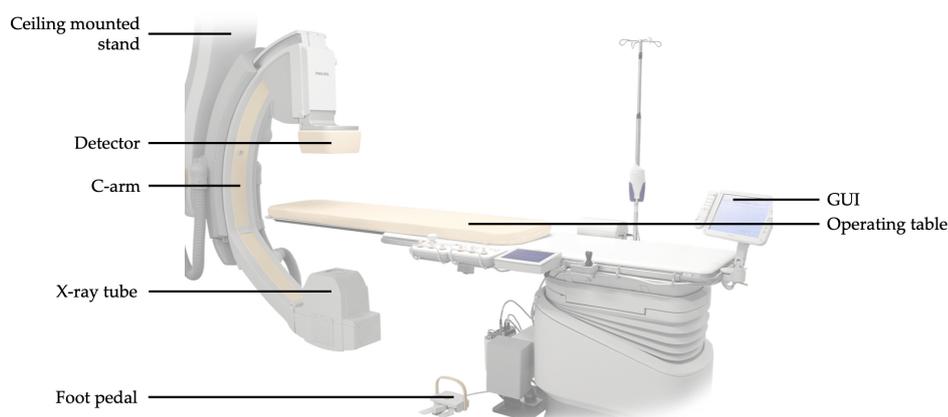


Figure 3.2: The main components of the C-arm, reprinted from the 2020 thesis of S. Imming [16].

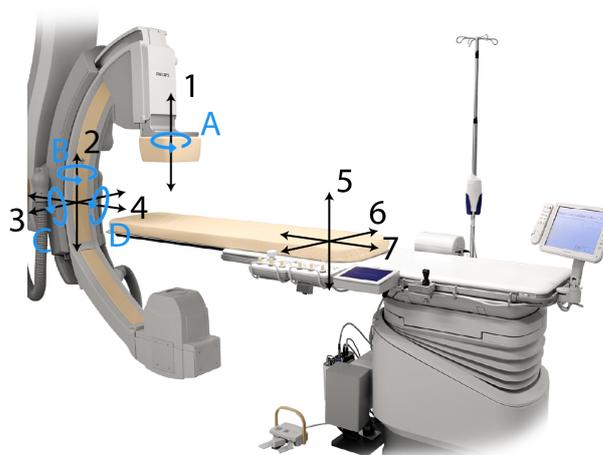


Figure 3.3: Axes of the C-arm that are logged during X-ray acquisitions, adapted from the 2020 thesis of S. Imming [16].

Table 3.3: An overview of the features in the acquisitions tab of the C-arm logging data used.

Feature Name	Description	Unit	Axes in Fig 3.3
ExamID	Unique identification code of the procedure	-	
AcquisitionID	Unique identification code of the X-ray acquisition	-	
AcqStartTime	Start time of the X-ray acquisition	dd-mm-yyyy hh:mm:ss	
AcqDuration	Duration of the X-ray acquisition	dd-mm-yyyy hh:mm:ss	
ShutterPositionLeft	The position of the left shutter.	$10^2 * \mu m$	
ShutterPositionTop	The position of the top shutter.	$10^2 * \mu m$	
ShutterPositionRight	The position of the right shutter.	$10^2 * \mu m$	
ShutterPositionBottom	The position of the bottom shutter.	$10^2 * \mu m$	
AngulationStart	The angulation of the C-arm at the start of the acquisition.	deg	C
AngulationEnd	The angulation of the C-arm at the end of the acquisition.	deg	C
RotationStart	The rotation of the C-arm at the start of the acquisition.	deg	D
RotationEnd	The rotation of the C-arm at the end of the acquisition.	deg	D
PositionCarm	The unfiltered angulation of the C-arm.	$10^{-2} * deg$	C

PositionDetector	The vertical position of the detector.	$10^2 * \mu m$	1
PositionPropellor	The unfiltered rotation of the C-arm.	$10^{-2} * deg$	D
FrontalBeamLongitudinal	The longitudinal position of the C-arm.	$10^2 * \mu m$	4
FrontalRotateDetector	The rotation of the detector.	$10^{-2} * deg$	A
FrontalZrotation	Swing of the C-arm.		B
TableHeight	The vertical position of the operating table.	$10^2 * \mu m$	5
TableLateral	The lateral position of the operating table.	$10^2 * \mu m$	6
TableLongitudinal	The longitudinal position of the operating table.	$10^2 * \mu m$	7

3.2.3. Movements

The movements sheet provides information on the movements of the C-arm device that occurred during the procedures from which the logging data was obtained. The logged information includes the name of the movement, the value and unit with which the movement occurred, and the duration of the movement. It is important to note that it is not entirely obvious from the movements and information logged during the acquisition which movement corresponds to the axis shown in Figure 3.3. How this is overcome is explained in Section 4.6.3. A selection of data from the clinical dataset can be seen in Table 3.4. A full view of the dataset can be seen in the Appendix A.1.

Feature Name	Description	Unit
EventTimestamp	The timestamp of the event.	dd-mm-yyyy hh:mm:ss
DurationSecs	The duration between the start and the end of the movement.	s
MovementName	The name of the movement.	-
Type	The type of movement.	-
Axis	The axis of the movement.	-
Value	The difference of the component setting at the start and end of the movement.	-
PositionUnit	The unit of the value in the 'value' column.	-

Table 3.4: An overview of the features in the movements tab of the C-arm logging data used.

3.3. Clinical Data

The clinical data set contains anonymized patient and procedure-specific information, gathered from the Reinier de Graaf electronic patient dossier. In addition to this data, the set includes a series of timestamps extracted from the clinical notes of the procedure. During each procedure, certain critical points are logged by the cath lab personnel. An overview of the features of the data set is given in Table 3.5.

Table 3.5: An overview of the features in the clinical database.

Feature Name	Description	Unit
Study Number	A unique identification code given to the procedure	-
Date	The date of the procedure	-
Gender	The gender of the patient	-
Age	The age of the patient	Years
Procedure	The type of the procedure	-
Cardiologist	Which cardiologist performed the procedure (anonymized)	-
NoNurses	Number of nurses that were present during the procedure	-
Start new procedure	Start of the procedure according to the clinical notes.	hh:mm:ss
Patient on table	Moment patient is on table according to the clinical notes.	hh:mm:ss
End procedure	End of the procedure according to clinical notes.	hh:mm:ss
Right catheters used	The number of catheters used for the right coronary artery	-
Left catheters used	The number of catheters used for the left coronary artery	-

3.4. Annotations Data

The annotations were generated using the Noldus Observer XT software [4]. This software provides an interface for the purpose of annotating video data with behaviours, or, in our case with phases. The workflow definition that was used during annotation are discussed in Section 4.1. More on the methodology of the annotation process can be read in Section 4.2. Each entry in the annotations data corresponds to either the start or the end of a phase for each phase in each procedure that was used. The annotation data is an export from the Noldus software, and an overview of the data is given in Table 3.6.

Feature Name	Description	Unit
Time	The time of the observation.	dd-mm-yyyy hh:mm:ss
Observation	Name of the observation consisting of studynumber (corresponding with the clinical data) and the date.	
Behavior	The annotated phase.	
Event_Type	Whether the entry is the start or the end of the phase.	

Table 3.6: An overview of the features in the annotations dataset.

4

Methodology

4.1. Formalization of the Workflow

A coronary angiography procedure performed at the Reinier de Graaf Hospital (RDGG) in Delft generally consists of a chronological sequence of actions performed by a cardiologist and a team of lab assistants. To recognize the phases of the procedure, an annotated dataset is needed. For the annotation of the dataset a clear definition of procedural phases is warranted. The formalization of the coronary angiography procedure workflow in RDGG was researched in 2020 by K.M. van der Graaf. She concluded that the general workflow could be defined as [12]:

1. Lab preparation: the lab is prepared for the patient.
2. The procedure: the procedure is performed on the patient.
 - (a) Patient preparation phase: patient enters room and is prepared for CAG.
 - (b) Operative phase: the CAG is executed.
 - i. Realizing endovascular access.
 - ii. Insert the catheter and direct it to the upper aortic arch.
 - iii. Entering and recording of the first coronary artery.
 - iv. Entering and recording of the second coronary artery.
 - v. Removing the catheter and closing the entry wound.
 - (c) Post-care phase: patient procedure is finalized.
3. The turnaround: lab is made ready for the next patient.
4. Clean-up: the resources and supplies of the procedures are cleaned up.

This thesis builds on van der Graaf's work, adapting the defined workflow to suit the specific requirements of this research. Formalization of the workflow was based on several criteria.

1. **Visibility on Video:** To allow effective annotation, it is of importance that the annotated labels are clearly visible on video. With the availability of five camera angles, occlusion of the actions performed should not be an issue. The collected videos were analyzed to identify visible points that align with the workflow defined by K.M. van der Graaf.
2. **Clinical Relevance:** Our adaptation of the workflow should aim to ensure that it reflects the real-world practice of coronary angiography procedures at RDGG, aligning the academic study with the practical needs and priorities of clinicians. For this purpose, the adapted workflow was discussed with a cardiologist at the RDGG.

3. **Granularity:** Granularity refers to the level of detail in the workflow. We chose a granularity that is more specific, recognizing that we could always leave out certain annotations later to make the formalization less granular. This approach also provides flexibility in model performance optimization, where the assumption is that decreasing the granularity would increase the performance of the model.

The criteria stated above guided the creation of the CAG procedure timeline, shown in Figure 4.1. This figure outlines the key points within the procedure, which were subsequently annotated using the Noldus Observer XT software; see Section 4.2 for more information on the annotation process. Additionally, the figure defines the phases of the procedure, each encompassing a portion of the procedure between two key points. Note that in Figure 4.1, some key points and phases are colored yellow. These occur only in particular procedures where accessing the coronary arteries with a catheter may be difficult, necessitating the use of additional catheters.

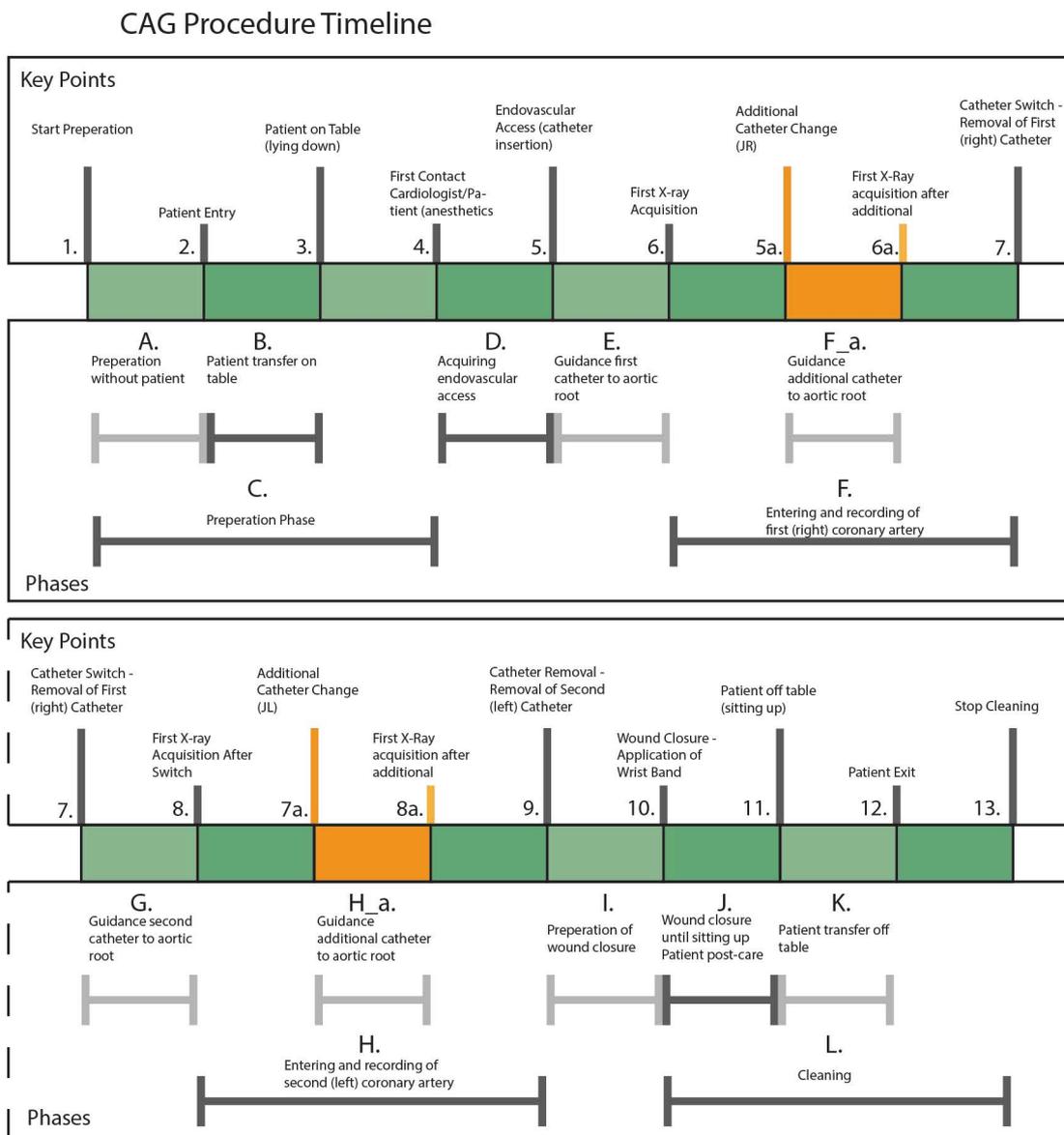


Figure 4.1: The full fo of the workflow of a coronary angiography procedure used for this thesis.

4.2. Annotation Process

A total of $N = 297$ coronary angiography procedures were annotated for this project, with the key points shown in Figure 4.1. The annotations were conducted by a medical student utilizing Noldus Observer XT software. The choice of a medical student for this task was driven by several factors. While the procedure is fairly simple, having some medical knowledge is beneficial for analyzing the video and identifying key points. Additionally, the time-consuming nature of the task, spanning several months, made it impractical for a medical specialist to undertake, given their demanding schedules.

Initially, one recorded video angle was visually analyzed for each of the 297 procedures. If the key point was unclear from this angle, other angles were examined. After the annotation process was finalized, the data was exported from the Noldus Observer XT software for use in model training and validation. An overview of the data can be found in Section 3.4

4.3. Video Data Object Detection

To use the video data collected from the recorded procedures in our ML model, some steps had to be taken. In this thesis, phase recognition is carried out using a random forest (RF) algorithm. An explanation for selecting this particular ML approach is provided in Section 2.4.3. When working with a random forest model, the data must be structured in tabular format. Therefore, the video data must be transformed into a usable form for the model. This transformation can be achieved through a process known as feature extraction. Feature extraction enables the extraction of various characteristics from the video data, and different methods can be used to achieve this. These techniques can capture various visual patterns and characteristics within the video. However, for the specific phase recognition task in our study, an adapted object detection algorithm was found to be the best approach. This algorithm is designed to detect clinically relevant objects, providing information on the coronary angiography procedure being performed.

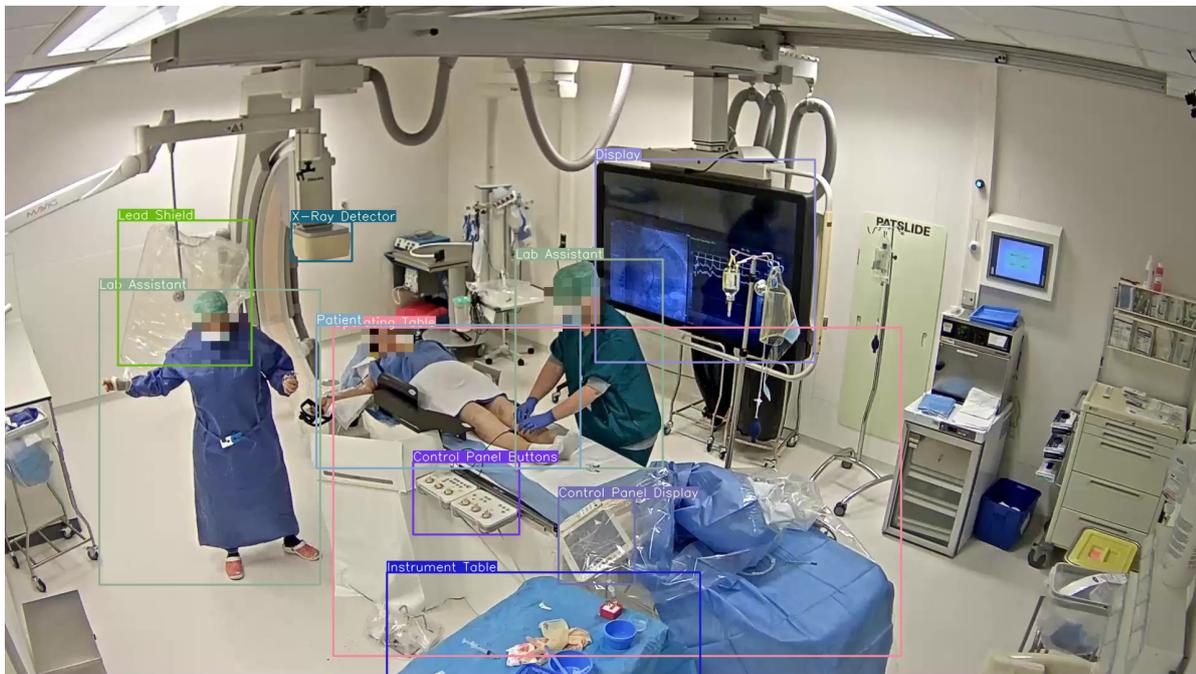


Figure 4.2: An example output frame of the bounding box algorithm which was created by R. Dai in 2022 [7]

In the master thesis conducted by R. Dai in 2022 [7], an adapted object detection algorithm was developed specifically to detect objects in coronary angiography videos from the Reinier de Graaf Hospital. In his thesis, Dai fine-tuned a Scaled-YOLOv4 based detection model for the detection of medical equipment and personnel [7]. The medical equipment and personnel that the model is capable of detecting can be seen in Figure 4.2.

To prevent long loading times for the large number of videos, measures were taken to prevent this. We focused on a single camera angle to extract bounding boxes. To find the best angle, we ran the algorithm on a short video segment from each angle, except the wide angle due to its distortion. Based on a visual interpretation of the detection accuracy, we chose the south-west angle for the algorithm. Using one angle speeds up processing but may compromise performance. Given our time limits and research goals, one angle was considered adequate. We also reduced the frame rate from 24 to 1 fps. Since we want to assess phase recognition for planning and remaining time predictions, a 1 second interval was deemed acceptable. However, when implementing phase recognition for more advanced purposes such as real time intraoperative assistance, this might not be enough.

The usefulness of the data for our random forest model is closely linked to the accuracy of the object detection algorithm. In practical terms, the more accurately the object detection model identifies medical equipment and personnel in the video, the more reliable the data becomes for the random forest model to recognize different phases of the coronary angiography procedure.

Given the strong performance of the object detection model, that performed with over 90% accuracy over all objects detected, we have confidence in the reliability of the features extracted for the random forest model. This suggests that our approach for phase recognition is likely to be effective in a clinical setting.

4.4. Criteria for Assessing Effectiveness

4.4.1. Performance Metric

To assess the performance of our machine learning model for classifying phases of CAG procedures, it is important to choose an evaluation metric that captures the model's performance accurately and is easily communicated to a wide audience. The most used and easy-to-understand metric of performance is accuracy, making it suitable for a broad audience and offering the ability to easily compare the performance of the model. However, using accuracy as a performance metric has downsides. For datasets with uneven class distributions, accuracy can cause a misleading image of a model's performance. In cases where one class dominates, a high overall accuracy might be attained even if the model poorly recognizes the minority class, leading to a misleading assessment of model performance. To address this limitation, we have decided to adopt per-class accuracy for this thesis. This approach is simple and easy to understand, like model accuracy, but also provides insight into the model's performance on each individual class.

4.4.2. Defining Performance Benchmarks

To create a suitable minimum accuracy benchmark for our model, it is important to understand the goal we want to achieve. In addition to assessing the usability of the data for phase recognition, we want to assess its ability to aid in monitoring and remaining time predictions of CAG procedures. For these purposes, very high accuracy-while ideal- is not as important as it might be for more critical applications such as intraoperative assistant systems, where a misclassification might lead to a misinformed decision leading to bad patient outcome. For monitoring, understanding the general phase of the procedure can already suffice. A broad

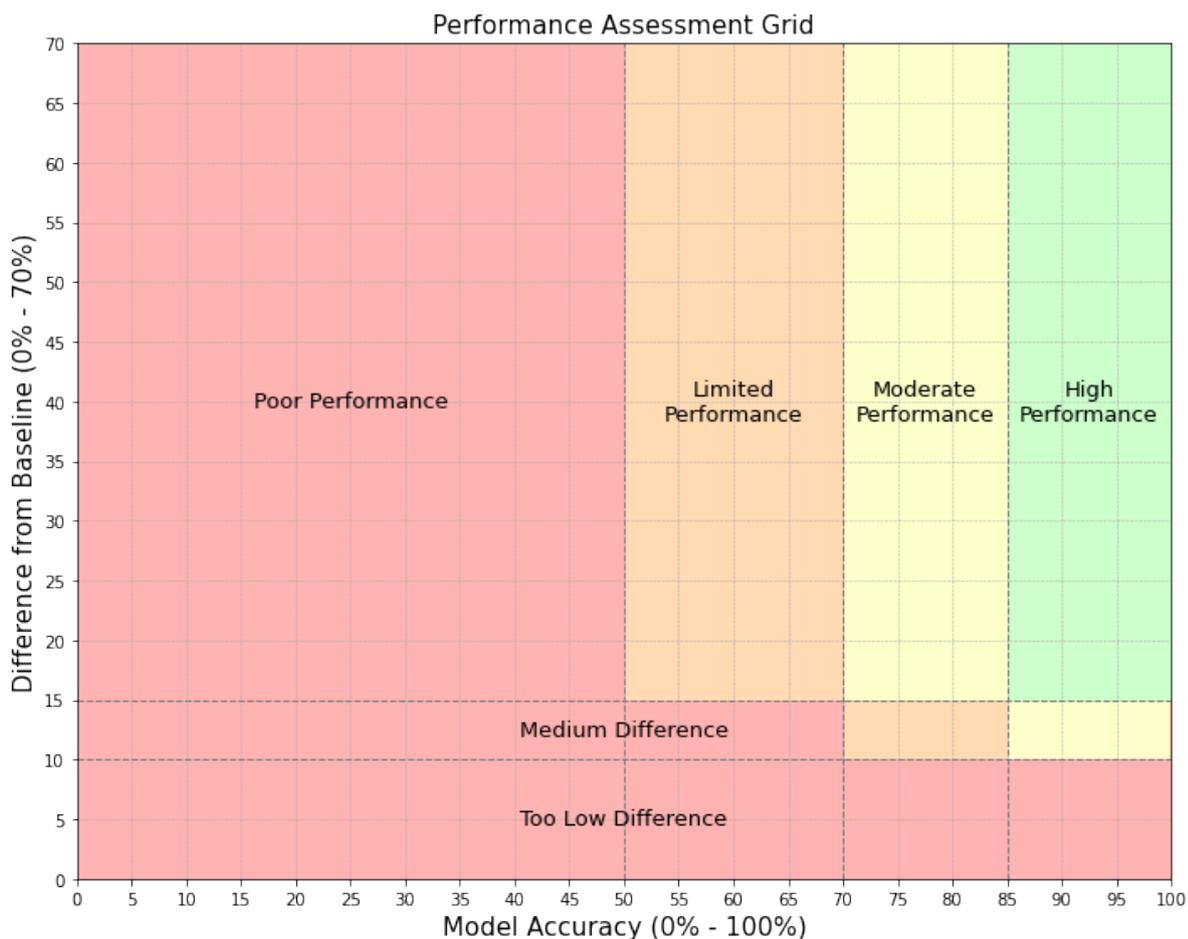


Figure 4.3: Caption

overview can already provide the necessary information to make informed planning decisions. Similarly, for predicting the remaining time of a procedure, recognizing the general phase can already yield a prediction of the remaining time of the procedure, which is useful for planning purposes, even if some granularity is sacrificed.

Since there is no literature on the performance needs for phase recognition in coronary angiography procedure planning, the description in the previous paragraph was used to guide the accuracy ranges. Therefore, the accuracy of the model will be assessed by looking at three accuracy ranges for each phase: 0-50% indicates poor performance, 50-70% indicates limited performance that might be useful for our purpose, 70- 85% is considered suitable for our specific goals, and 85-100% represents optimal performance.

Another important step in evaluating our model is comparing it to a simpler baseline. As a baseline, the average contribution of each phase to the total duration of the procedures will be used to create a simple prediction model of the phases of the procedure. It is important to note that this method is not suitable for real-time prediction since it uses the total duration of the procedure. However, it can still give a baseline for our purpose. For our ML model to be useful, it should significantly outperform the baseline. We have set the margin at 10-15%. Achieving this difference in combination with achieving high accuracy would confirm the value of our model in monitoring CAG procedures and predicting their remaining duration.

4.5. Data Preprocessing

In this section, the data preprocessing steps will be discussed. This includes all transformations that were performed on the data before the different data sources were integrated.

4.5.1. Logging Data

Exam

The exam dataset of the logging data did not need preprocessing before the data could be integrated.

Acquisitions

The pre-processing steps for the acquisitions sheet of the logging data consisted of:

- **Splitting Durations Into Timestamps:** Each entry in the acquisition sheet corresponds to an X-ray acquisition and includes timestamps for the beginning of an acquisition and the duration of that acquisition. To put all the data on a timeline, each radiation row was converted into two rows that were labeled with either the start or the end of an the X-ray acquisition. Due to this, the *duration* feature became unnecessary.
- **Simplify Rotation and Angulation Columns:** Since each entry in the data set now consists of either a start or an end of an acquisition, the features on the start and end of the rotation and angulation can be simplified to just the rotation and angulation. For the other settings of the C-arm, only one value was defined for the entire acquisition, and as a result these were just duplicated.

These pre-processing steps are exemplified in Figure 4.4.

Before Pre-Processing Acquisitions					
ExamID	AcquisitionId	AngulationStart	AngulationEnd	PositionDetector	AcqDuration
5728...	1074...	5	6	10180	3
5728...	aead...	4	58	10670	5

↓

After Pre-Processing Acquisitions				
ExamID	AcquisitionId	Angulation	PositionDetector	Start/End
5728...	1074...	5	10180	Start
5728...	1074...	6	10180	End
5728...	aead...	4	10670	Start
5728...	aead...	58	10670	End

Figure 4.4: Before and After Pre-Processing Acquisitions

Movements

The pre-processing steps for the movements sheet of the logging data consisted of:

- **Splitting Durations Into Timestamps:** Each entry in the movements sheet corresponds to a movement of the C-arm. To put all the data on a timeline and get insight in overlapping movements, the movements with a duration were converted into two key points that were labeled with either the start or the end of the movement. Due to this, the *duration* feature became unnecessary.

Before Pre-Processing Movements				
EventTimestamp	DurationSecs	MovementName	Value	PositionUnit
09:01:14	6	ROTATEBEAMFRONTAL	-90.6	degree
09:01:19	2	SHIFTDETECTORFRONTAL	-37	mm

↓

After Pre-Processing Movements				
EventTimestamp	MovementName	Value	PositionUnit	StartEnd
2020-10-20 09:01:14	ROTATEBEAMFRONTAL	-90.6	degree	MovStartTime
2020-10-20 09:01:19	SHIFTDETECTORFRONTAL	-37	mm	MovStartTime
2020-10-20 09:01:20	ROTATEBEAMFRONTAL	-90.6	degree	MovEndTime
2020-10-20 09:01:21	SHIFTDETECTORFRONTAL	-37	mm	MovEndTime

Figure 4.5: Before and After Pre-Processing Movements

4.5.2. Clinical

The clinical dataset did not need preprocessing before the data could be integrated.

4.5.3. Object Detection Data

The preprocessing steps for the object detection data consisted of:

- **Concatenating Text Files:** As stated in Section 3.1.1, the object detection data consists of a large number of text files, one for each second of every video. The first step of the preprocessing process will be to add all the data from the separate text files into one dataset.
- **Adding Additional Features:** Since the information in the files only includes information on the bounding boxes, the names of the files were used to add additional useful information to the dataset, such as the date of the procedure and the timestamp of the object detection frame. The file names were in the format `startdate-starttime_enddate-endtime_angle_frame`, for example `20201020-072554_20201020-080000_CornerSW_1`. From this information in the file name, the following features were extracted:
 - **Start Video:** the beginning time of the video according to the computer that was used for recording the videos.
 - **End Video:** the end time of the video according to the computer that was used for recording.
 - **Angle:** the angle of the video. This corresponds with one of the angles in Figure 3.1.

- **Frame:** the number of the frame in the video that the object detection is performed on. This corresponds to the second of the video, since the video has a frame rate of 1fps.
- **Renaming Objects:** The objects column in the object detection data consisted of numbers. For clarity, the numbers were renamed to the corresponding object names: The

Original	Renamed	Original	Renamed
0	Cardiologist	6	Control Panel Buttons
1	Lab Assistant	7	X-Ray Detector
2	Patient	8	-
3	Instrument Table	9	Lead Shield
4	Operating Table	11	Display
5	Control Panel Display	12	-

Table 4.1

object names for numbers 8 and 12 were not included since these objects were not detected in any of the frames that were analyzed.

- **Correcting Timestamps**

A significant challenge with the object detection data was that the video titles were generated automatically after the system time of the computer that was used. The computer was not connected to the internet and because of this, the system time of the pc did not correspond to reality. Since the solution to this problem requires data integration steps, it will be discussed under the data integration part in Section 4.18.

4.5.4. Annotations

The preprocessing steps for the acquisitions sheet of the Annotation data consisted of:

- **Features Extracted from Observation Names:** As can be seen in the overview of the annotation data features in Section 3.4, the **Observations** column consists of the studynumber and the date of the procedure. During preprocessing, we extracted this information to create two new features: **Date** and **StudyNumber**.
- **Date and Time Combined:** The **Date** and **Time** columns are combined to a single column named **Date_Time**
- **Pivot the table:** The format of the dataset is described in Section 3.4. Each entry of the dataframe corresponds to a separate annotation. For a better overview of the dataset the table was pivoted. In the pivoted table, each entry is a procedure with a unique studynumber in the **StudyNumber** column. All the other columns correspond to the logged keypoints, where the values in the table are the dates and times of the specific keypoint for each procedure.

The general structure of the dataset before and after preprocessing can be seen in Figure 4.6

Before Pre-Processing Annotations			
Observation	Time	Behavior	
Patient.183.11.01.2021	14:42:50	Start Preparation	
Patient.183.11.01.2022	14:46:36	Patient entry	
...	
Patient.320.24.12.2021	15:28:32	Start Preparation	
Patient.320.24.12.2022	15:31:00	Start Preparation	
...	

↓

After Pre-Processing Annotations			
StudyNumber	Start Preparation	Patient Entry	...
183	2021-11-01 14:42:50	2021-11-01 14:46:36	...
320	2021-12-24 15:28:32	2021-12-24 15:31:00	...
...

Figure 4.6: Before and After Pre-Processing of Annotations Data

4.6. Data Integration

To evaluate a random forest model trained on the discussed sources of data including C-arm system data, object detection data, and clinical records, it is necessary to merge these separate datasets into a unified dataset. This chapter describes the methodical process of data integration, detailing how each data set was merged together. A visual representation of this process is provided in Figure 4.7, which helps to illustrate the these steps.

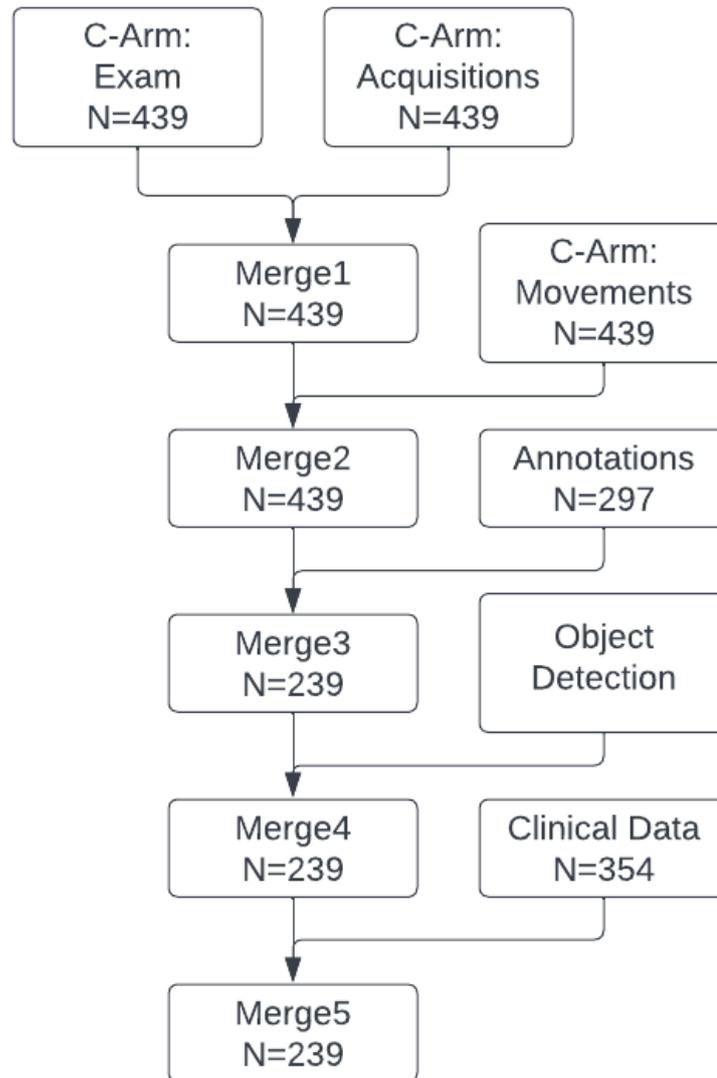


Figure 4.7: An overview of the data integration processing steps.

4.6.1. Merge 1: Logging: Acquisitions & Logging: Exam

Both the **acquisitions** and the **exam** data have an *ExamID* column. Using this common ID column, the data present in the exam sheet was straightforwardly added to the corresponding acquisitions. This process can be seen in Figure 4.8

Logging: Acquisitions					
ExamID	AcquisitionId	Timestamp	...		
f04c...	53c7...	2021-11-01 10:01:55	...		
f04c...	894f...	2021-11-01 10:01:58	...		

Logging: Exam		
ExamID	PatientNumber	PhysicianNumber
f04c...	G6xP...	oWVr...

↓

Resulting Joined Table					
ExamID	AcquisitionId	Timestamp	...	PatientNumber	PhysicianNumber
f04c...	53c7...	2021-11-01 10:01:55	...	G6xP...	oWVr...
f04c...	894f...	2021-11-01 10:01:58	...	G6xP...	oWVr...

Figure 4.8: Joining the tables *Logging: Acquisitions* and *Logging: Exam* on the *ExamID* column.

4.6.2. Merge 2: Add Movements to Merge 1

To merge the dataset of the first merge and the movements, they are concatenated together. This means that the rows of the two datasets are placed on top of each other, joining the common columns between the two tables. After the datasets are concatenated, the table is sorted on the **Timestamp** column. The functionality of the **StartEnd** column is expanded to also include information on whether an entry consists of an acquisition or a movement. A subset of the datasets which were used for the merge, and the resulting dataframe are shown in Figure 4.9.

Merge 1							
ExamID	StartEnd	Timestamp	...	PositionDetector			
5728...	AcqEnd	08:59:17	...	10180			
5728...	AcqStart	08:59:47	...	10670			

Logging: Movements							
Timestamp	StartEnd	MovementName	Value	Unit			
08:59:28	MovStart	SHIFTDETECTO...	190	mm			
08:59:33	MovStart	SHIFTDETECTO...	-141	mm			
08:59:33	MovEnd	SHIFTDETECTO...	190	mm			
08:59:36	MovEnd	SHIFTDETECTO...	-141	mm			

↓

Resulting Concatenated Table							
Timestamp	MovementName	Value	Unit	StartEnd	ExamID	...	PositionDetector
08:59:17				AcqEnd	5728...	...	101800
08:59:28	SHIFTDETECTO...	1900	mm	MovStart		...	
08:59:33	SHIFTDETECTO...	-1410	mm	MovStart		...	
08:59:33	SHIFTDETECTO...	1900	mm	MovEnd		...	
08:59:36	SHIFTDETECTO...	-1410	mm	MovEnd		...	
08:59:47				AcqStart	5728...	...	106700

Figure 4.9: Concatenating the tables *Merge 1* and *Logging: Movements*.

4.6.3. Merge 3: Add Annotations to Merge 2

The data set that results from merge 2 includes a column named **Timestamp**, see Figure 4.9. For the purpose of showing the concatenation process, only the times are shown but in reality this column includes information on the date and time. The date and time represent the system time of the C-arm machine, which corresponds to the actual time during a procedure. The annotation data also includes a column that specifies the date and time named **Date_Time**. However, since the Noldus software did not have any way to obtain the time during the procedure, the times do not correspond to the true time of the C-arm logging.

When thinking of the unaligned timelines of the two datasets, a comparison can be drawn to synchronizing audio and video tracks during film editing. In filmmaking, since audio and video tracks are typically recorded separately, they must be aligned during the postproduction phase. This is commonly achieved through the use of a clapperboard, whose sound and visual cue can be easily identified in both the audio and video data, facilitating their synchronization. Similarly, in our research, resolving the unaligned timelines of the datasets for each procedure could be accomplished if we had a unique moment that was recognizable in both the annotations and the C-arm logging data. This moment was identified as the first X-ray acquisition of a procedure. This moment is annotated in the annotations as the key point: *first X-ray acquisition*, and it is also identifiable in the system logging data as the first acquisition that occurred in a procedure. Namely, the first acquisition of each procedure with a unique **ExamID** occurs at the first **Timestamp** with the label 'AcqStart' in the **StartEnd** column.

Since the timelines of the annotations and the data from the C arm are not aligned, and multi-

ple procedures are conducted on most days, it is not clear which procedure in the C-arm data belongs to the annotations. Therefore, it becomes essential to identify a method to associate each procedure with the correct annotations. Thus, for each **ExamID** it is necessary to determine the corresponding **StudyNumber**. Figure 4.10 illustrates the need for this connection, showing the timestamps of the annotations for the start and end of eight procedures alongside the timestamps of the acquisitions logged by the C-arm machine.



Figure 4.10: A visualization of the timestamps of the start and end annotations of 8 procedures on november 2nd 2021, and the acquisitions logged by the C-arm machine. It is clearly visible that the times of all the annotations overlap severely and therefore it is not possible to identify what set of annotations corresponds to what procedure.

For linking **ExamID** with the **StudyNumber** column, the clinical data set is used. More on the content of this data set can be found in Section 3.3, specifically in Table 3.5. The data set contains, in addition to a variety of patient and procedure data, a few key points that were logged during the procedure by a cath-lab assistant. The keypoints that correspond to the start and end of the procedure in the clinical data set have been visualized together with the acquisitions logged by the C-arm machine in Figure 4.11. In the Figure, we can see that while the timestamps logged by clinical personnel is subjected to human error, they correspond roughly to the true timestamps which were logged by the C-arm machine. Since the clinical data set contains the **StudyNumbers** as well as these times, we can link the **StudyNumbers** with the **ExamIDs** using the overlap between the time interval of the procedures recorded by the cathlab assistant and the procedures recorded by the C-arm machine.



Figure 4.11: A visualization of the timestamps of the start and end as logged by cath-lab personnel of 8 procedures on november 2nd 2021, and the acquisitions logged by the C-arm machine.

When all **StudyNumbers** are connected to their corresponding **ExamIDs**, the moment of the first X-ray acquisition can be used to correct the timestamps of the annotations. In the annota-

tions data this moment corresponds to the timestamp of the key point **First X-Ray Acquisition** while in the C-arm log this moment corresponds to the timestamp of the first recorded X-ray acquisition in a procedure. After this process, the annotation timestamps correspond to the real time during the procedure. In Figure 4.12 an example can be seen of what this looks like for November 2nd 2021.

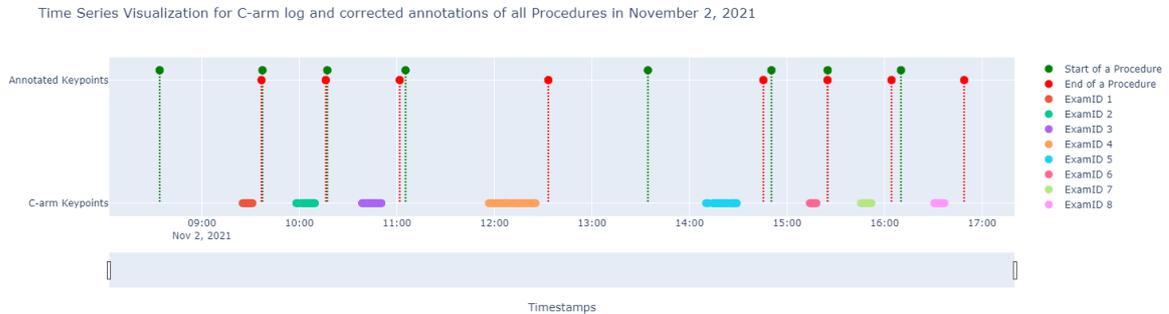


Figure 4.12: A visualization of the corrected timestamps of the annotations of 8 procedures on november 2nd 2021.

The movements recorded by the C-arm during the procedures do not include an **ExamID**, as can be seen in Figure 4.9. As we now have the correct timestamps of the annotations, we know exactly when each procedure begins and ends. We can use this information to label all movements with the correct **ExamID**. All the recorded movements or acquisitions that happened outside the annotated procedures can be removed from the data set as these will not have any predictive value for the procedures that we are using.

The corrected annotation keypoints were used to define the phases of the procedure following the definitions in Figure 4.1. Each timestamp in the dataset that lies between two keypoints gets a label that shows the steps that occur between those points. The resulting phase annotations for the C-arm data for an example procedure during November 2nd 2021 can be seen in Figure 4.13.



Figure 4.13: C-arm log data that has been color coded according to phase from an example procedure that occurred on the 2nd of November 2021

Processing After Merge 3

When looking at Figure 4.9 we can observe the movements and acquisitions after merge 3. Since we concatenated the two different entry types, there are gaps of unknown values in the

table. Because movements in the table lead to a change in the settings that are logged during acquisitions, they can be used to enrich the data set. The example in Figure 4.9 illustrates this. If we add the values of the movements ($1900 - 1410 = 490$), we can see that this corresponds to the difference in the **PositionDetector** values over the period of time when movements occurred ($106700 - 101800 = 4900$). By manually checking which movement names correspond to the recorded settings during acquisitions, Table 4.2 was created.

Recorded C-Arm Setting (A)	Movement Name (B)	Scaling Factor (B/A)
PositionDetector	SHIFTDETECTORFRONTAL	1/10
PositionPropellor	ROTATEBEAMFRONTAL	1/100
TableHeight	CHANGEPATIENTSUPPORTHEIGHT	1/10
TableLateral	MOVEPATIENTSUPPORTLATERAL	1
TableLongitudinal	MOVEPATIENTSUPPORTLONGITUDINAL	1/10
Rotation	ROTATEBEAMFRONTAL	1
Angulation	ANGULATEBEAMFRONTAL	1
PositionCarm	RESETGEO, axis='C-arm (ROLL)'	1/100
FrontalBeamLongitudinal	MOVEBEAMLONGITUDINALFRONTAL	1/10

Table 4.2: An overview of the recorded C-arm settings and their corresponding movement name. The scaling factor column shows the difference in scale of the two columns.

Based on the table provided, the information derived from the movements was incorporated into the acquisition data. This integration was achieved by cumulatively adding the associated movements to the acquisitions whenever they took place. A deliberate choice was made to add the movement at its initiation point, specifically where StartEnd is labeled MovStart. An alternative approach to populating the values would be to assume that the C-arm movements occurred linearly. Under this assumption, the values of the recorded settings could be interpolated throughout the movement's duration. Although this method might provide more information to the model, its computational cost is substantial.

For instance, calculating a single column for the "merge 3" data set (comprising approximately 100,000 rows) takes about 1 minute and 31 seconds. Multiplying this duration by nine (for each column) results in approximately 13.6 minutes. However, the challenge is amplified when considering integration with the video data set. This dataset records every second of each video, equating to approximately 4.3 million rows. Assuming a linear time complexity for the computation, the estimated calculation time would be:

$$\text{Time} = 1.52 \times 9 \times \frac{43}{1} \approx 586 \text{ minutes} = 9.76 \text{ hours}$$

Given the project's primary objective, to validate data sources for phase recognition, such an extensive computational effort was deemed unnecessary. Therefore, the initial approach was adopted. Based on the provided table, the information derived from the movements was incorporated into the acquisition data. This integration was achieved by cumulatively adding the associated movements to the acquisitions whenever they took place.

Since not all movements of the system settings were included in the movements data, the values for these settings were filled in using the last known value. The input datasets consisting of the data from merge 2 and the annotations, and the output of the third merge are shown in Figure 4.14

Merge 3						
Timestamp	MovementName	Value	Unit	StartEnd	ExamID	PositionDetector
08:59:17				AcqEnd	5728...	101800
08:59:28	SHIFTDETECTO...	1900	mm	MovStart		
08:59:33	SHIFTDETECTO...	-1410	mm	MovStart		
08:59:33	SHIFTDETECTO...	1900	mm	MovEnd		
08:59:36	SHIFTDETECTO...	-1410	mm	MovEnd		
08:59:47				AcqStart	5728...	106700

Annotations				
StudyNumber	Start Preparation	Patient Entry	Patient on Table	...
100	2020-12-11 16:45:17	2020-12-11 17:16:01	2020-12-11 17:16:15	...
...

↓

Resulting Merged Table						
EventTimestamp	StartEnd	ExamID	...	PositionDetector	Phase	
2020-10-20 08:59:17	AcqEndTime	5728...	...	101800	F	
2020-10-20 08:59:28	MovStartTime	5728...	...	120800	F	
2020-10-20 08:59:33	MovStartTime	5728...	...	106700	F	
2020-10-20 08:59:33	MovEndTime	5728...	...	106700	F	
2020-10-20 08:59:36	MovEndTime	5728...	...	106700	F	
2020-10-20 08:59:47	AcqStartTime	5728...	...	106700	F	
...

Figure 4.14: Merging the tables *Merge 2* and *Acquisitions*.

Resulting Number of Procedures in Dataset after Merge 3

A total of three datasets were used for adding the annotations to the data of merge 2: Clinical, C-Arm System, and Annotations Data. The datasets each have data a unique but overlapping set of procedures. The clinical database has data on $N = 354$ procedures, and the c-arm system data has data on $N = 439$ procedures. When the clinical database is used to link the **ExamID** and **StudyNumber** columns as described earlier in this section, this can only be done for the procedures that are in both datasets. The number of procedures that are left after using these datasets is $N = 251$ as shown in Figure 4.15. The same holds for adding the annotations to the data, where the annotations dataset has a number of procedures of $N = 297$ and adding them to the dataset with $N = 251$ procedures results in a dataset of $N = 239$ procedures as is shown in 4.16.

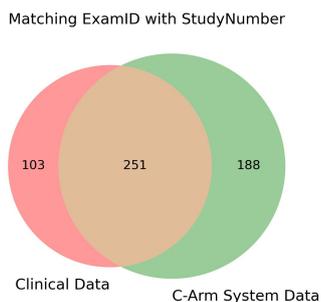


Figure 4.15: Number of Procedures in Both Clinical and C-Arm System Data

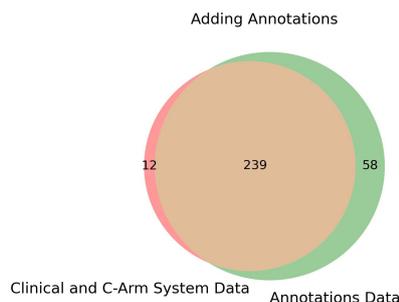


Figure 4.16: Number of Procedures in Both Clinical and C-Arm System Data with Annotations

4.6.4. Merge 4: Add Video Data to Merge 3

Correcting the Timestamps of the Video Data

As highlighted in Section 4.5.3, a significant challenge with our video data was the inconsistency of timestamps relative to the actual time of the procedure. To address this, the following formula, which was derived from Figure 4.17, was used:

$$FileStart = ProcedureStart - (File\Delta PE - Proc\Delta PE)$$

In this equation:

- *FileStart* signifies the accurate beginning time of the recording.
- *ProcedureStart* denotes the initiation of the procedure as per the data from merge 3.
- *File Δ PE* represents the interval between the video's start and the patient's entry.
- *Proc Δ PE* indicates the gap between the procedure's start and the patient's entry.

The newly calculated value for *FileStart* can then be used to shift all the timestamps for that procedure in the object detection data to the correct time. The values for *File Δ PE* were gathered by reviewing all the videos and notating the time of the patient entry. The reason the patient entry was chosen as a point to correct the timestamps is that it is a moment in the video that is easily found.

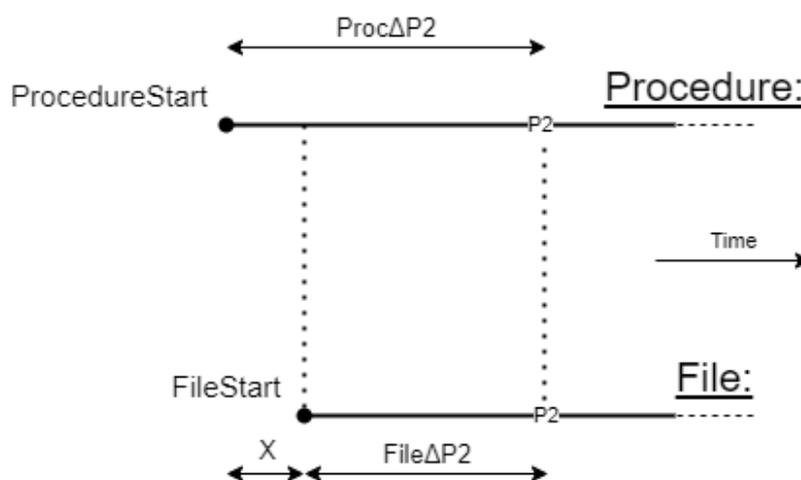


Figure 4.17: The figure describes a problem that occurs in the object detection data. The timelines of the actual procedure and the time defined by the file are not equal. This problem can be solved if the patient entry (PE) key point is found in both the file and the procedure.

Merging the Data with Merge3

To incorporate the object detection dataset into the previously established table from the third merge, we use concatenation. Given that the video data introduces new information, and thus new columns, the rows of the two datasets are vertically appended and subsequently sorted based on the **EventTimestamp**. After this, the appropriate **StudyNumber**, **ExamID**, and **Phase** were integrated into the video data, using the annotation data set, as has been done in Section 4.6.3. Lastly, the system configurations, as set forth in merge 3, were expanded to span the video data. Using our knowledge of the exact timestamps at which these settings were modified, we were able to adapt this to the video dataset. An overview of this process is depicted in Figure 4.18.

4.6.5. Merge 5: Add Clinical Data to Merge 4

Since the clinical data set includes the **StudyNumber**, the data could easily be merged using this column. This process is illustrated in Figure 4.19.

4.7. Feature Engineering

In this section, our aim is to derive features from our data that enhance the performance of the specified machine learning model. This process is often referred to as feature engineering [26].

4.7.1. Temporal Features

As noted in Section 2.4, a random forest model does not use temporal information for prediction. However, in multiple articles on phase recognition in a variety of medical procedures, the use of temporal information has been shown to improve classification performance [27, 2, 14]. Temporal information can be added to the information the model receives by engineering features that are dependent on values at other timestamps. To allow real-time classification, it is important that only past timestamps should be used to engineer these temporal features.

Counters

A method for implementing a time element feature is to count the occurrence of specific events during a procedure. The events that we used to create counters are:

Merge 3						
EventTimestamp	StartEnd	ExamID	...	PositionDetector	Phase	
...	
2020-10-20 08:59:33	MovStartTime	5728...	...	106700	F	
...	

Object Detection						
EventTimestamp	Object	X-center	Y-center	Width	Height	
...	
2020-10-20 08:59:33	X-Ray Detector	0.411719	0.305556	0.065104	0.068518	
2020-10-20 08:59:33	Display	0.645833	0.272685	0.157292	0.284259	
2020-10-20 08:59:33	Cardiologist	0.329688	0.550463	0.095833	0.523148	
2020-10-20 08:59:33	Control Panel Buttons	0.458854	0.640278	0.051042	0.065741	
2020-10-20 08:59:33	Lead Shield	0.378906	0.39213	0.079687	0.232407	
2020-10-20 08:59:33	Operating Table	0.58724	0.6625	0.415104	0.510185	
...	

↓

Resulting merged Table						
EventTimestamp	Object	X-center	...	PositionDetector	...	Phase
...
2020-10-20 08:59:33	X-Ray Detector	0.411719	...	106700	...	F
2020-10-20 08:59:33	Display	0.645833	...	106700	...	F
2020-10-20 08:59:33	Cardiologist	0.329688	...	106700	...	F
2020-10-20 08:59:33	Control Panel Buttons	0.458854	...	106700	...	F
2020-10-20 08:59:33	Lead Shield	0.378906	...	106700	...	F
2020-10-20 08:59:33	Operating Table	0.58724	...	106700	...	F
...

Figure 4.18: Merging the tables *Merge 3* and *Object Detection*.

- Acquisitions counter
 - Pulsed Fluoroscopy counter
 - Cineangiography counter
- Movements counter

Cumulative Values

Another method we used to add temporal information to our data was to cumulatively count the values together from the start of the procedure. This was done for the following information:

- **Time:** The time that has passed since the start of the procedure.
- **Acquisition Time:** The duration of all acquisitions together since the start of the procedure. This was also done for the following acquisition types:
 - [Pulsed Fluoroscopy, Cineangiography]

Merge 4					
EventTimestamp	StudyNumber	PositionDetector	...	Phase	
...
2020-10-20 08:59:33	101	106700	...	F	
...

Clinical Data		
StudyNumber	Gender	...
101	Female	...
...

↓

Resulting Merged Table					
EventTimestamp	StudyNumber	PositionDetector	...	Phase	Gender
...
2020-10-20 08:59:33	101	106700	...	F	Female
...

Figure 4.19: Merging the tables *Merge 4* and *Clinical*.

- **C-arm Settings:** The differences of the C-arm settings between each consecutive timestamp added together from the start of the procedure. This was done for the following settings:
 - ['ShutterPositionX', 'ShutterPositionY', 'WedgeLeftDistance', 'WedgeLeftAngle', 'WedgeRightDistance', 'WedgeRightAngle', 'PositionCarm', 'PositionDetector', 'PositionPropellor', 'FrontalBeamLongitudinal', 'FrontalBeamTransversal', 'FrontalRotateDetector', 'FrontalSwing', 'FrontalZrotation', 'TableHeight', 'TableLateral', 'TableLongitudinal']

Frequency

Additional temporal features were engineered using multiple other temporal features:

- **Acquisition Frequency:** The frequency of acquisitions from the start of a procedure was calculated using the *acquisition counter* divided by the *cumulative time*.
- **Movement Frequency:** The frequency of movements from the start of a procedure was calculated using the *movement counter* divided by the *cumulative time*.

4.7.2. Object Detection Features

Additional features were also extracted from the bounding boxes in the object detection data. The features that were extracted are listed here:

- **The Number of Objects in Frame:** The total number of objects that are detected at every frame and thus every second of each procedure. In addition the following objects in frame were counted:
 - [cardiologists, lab assistants, patients, people (= cardiologists + lab assistants + patients)]

- **Whether a Person is in Frame:** A binary value for whether a person is or is not in the frame for each second of the procedure. The detection was done for:
 - [cardiologist, patient, lab assistant]

Rolling Mean

From the number of objects in the frame, additional features were extracted. With the aim of including temporal information in the features, a rolling window mean was implemented. A rolling window mean takes the mean over x seconds, automatically moving forward with the passage of time. To allow real-time recognition, the mean was taken over only seconds that have passed. The following rolling window means were calculated:

- **Objects in Frame** ($2[s]$, $5[s]$, $10[s]$): a rolling window mean of 2, 5, and 10 seconds was added to assess the average number of objects over these durations for each frame.
- **Whether a Patient is in Frame:** a 5 second rolling window mean was applied to the binary value for whether a patient is in frame. This could add temporal information and also smooth out any erroneous detection.

Grid Based Object Count

Because a random forest does not handle information between rows such as temporal information and each row in our data set consists of a separate object, information on the location of only one object at a certain timestamp is included in the prediction. Thus in each row we would like to include spatial information about all the object. To achieve this, a grid-based object counter was made and each cell was added as a feature. The counter consists of a grid of 5×5 cells. The number of objects in each cell is counted.

4.8. Feature Matrix

This chapter will give an overview of the feature matrix which is the result of the steps described in Sections 4.5 to 4.7. Subsets of this data set are generated to assess the performance of phase recognition for separate data sources. The feature matrix and the data used for the subsets of this feature matrix are described in Table 4.4.

To gain a better understanding of how features change over time during a procedure, the visualization in Figure 4.20 was created. This figure shows the value of a sample of features from the feature matrix shown in Table 4.4 together with the phases of the procedure during an example procedure on 2 November 2021. The figure is a first indication that the data that we are using could be predictive of phase.

Index	Feature Name	Description
1	Phase	The phase of the procedure.
2	{C-arm Settings}	The recorded settings of the C-arm device. A full list can be found in Table 3.3.
3	{C-arm Settings}DcumulCount	A counter of the number of times the C-arm settings changed.
4	{C-arm Settings}DcumulSum	The cumulative value of all the changes made to the C-arm settings.
5	AcqCount, MovCount, FluoCount, CineCount	A counter for the number of movements, acquisitions, and acquisitions of type 'pulsed fluoroscopy' and 'cineangiography'.
6	TimeSinceAcq, TimeSinceMov	The time that has passed since the last movement and acquisition.
7	CumulTime	The cumulative time passed since the start of the procedure.
8	AcqFreq, MovFreq	The frequency of the number of acquisitions and movements over time.
9	CumulAcqTime, CumulMovTime, CumulCineTime, CumulFluoTime	The cumulative duration of seconds that acquisitions, movements, cineangiography, or pulsed fluoroscopy have been performed.
10	IsMovement, IsAcquisition	Whether the timestamp is during a movement or acquisition.
11	PhysicianNumber	Unique identification code of the physician.
12	IsCardiologist, IsPatient, IsAssistant	Whether the timestamp is during the detection of a cardiologist, patient, or assistant.
13	#ObjectsInFrame, #PeopleInFrame, #CardiologistInFrame, #PatientInFrame, #AssistantInFrame	The number of objects, people, cardiologists, patients, and assistants detected in the frame.
14	Rolling2MeanObjects, Rolling5MeanObjects, Rolling10MeanObjects	The rolling window mean with a window size of 2, 5, and 10 of the number of objects detected.
15	Rolling5MeanPatient, Rolling5MeanCardiologist, Rolling5MeanAssistant	The rolling window mean with a window size of 5 for the number of patients, cardiologists, and assistants detected in the frame.
16	Cell 0_0, ..., Cell 4, 4	The number of objects present in each cell of a 5x5 grid laid over the video data.
17	Number of Nurses	The number of nurses that were present during the procedure according to clinical notes.
18	Gender	The gender of the patient.
19	Age	The age of the patient.
20	Part of Day	The part of the day in which the procedure was executed.
21	Procedure Type	The type of procedure.

Table 4.4: The feature matrix which is the culmination of the processing steps described in Sections 4.5 to 4.7. The matrix includes all the features that are available to train the random forest model for the phase recognition task.

Sample of 4 recorded Features over Time for a Procedure in November 2 2021

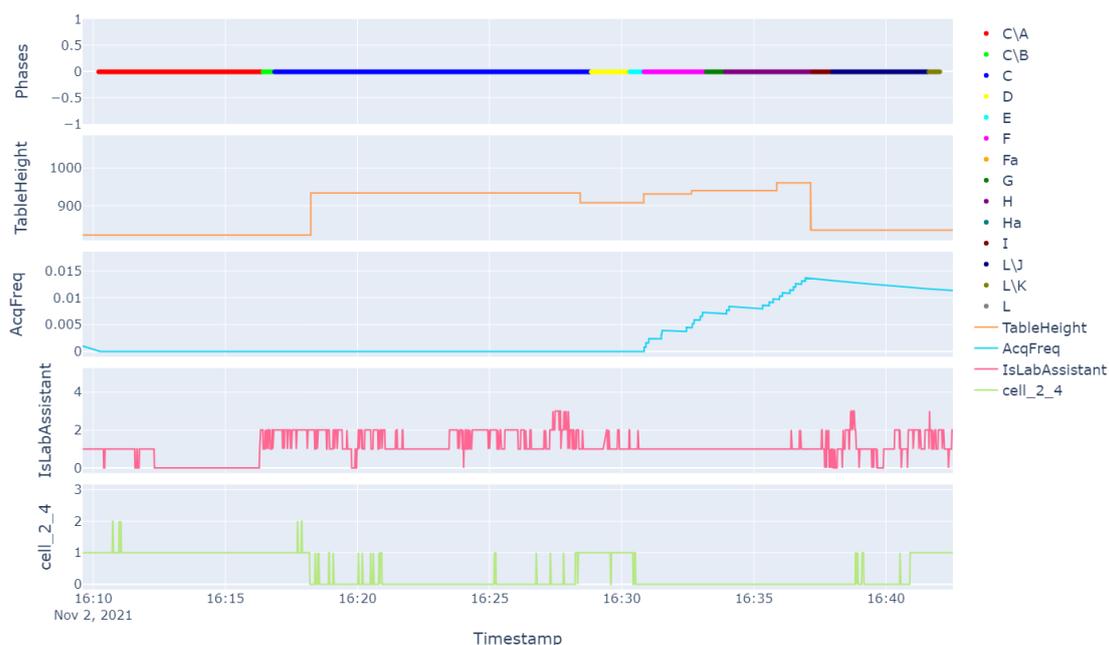


Figure 4.20: A sample of features from the feature matrix shown in Table 4.4 of a procedure on November 2, 2021.

4.8.1. Splitting Data Sources For Recognition

All the features that are available for use in the phase recognition process are shown in Table 4.4. To assess the usability of the C-arm log data and the video data separately, the data for these sources will be split off from the feature matrix. The features that can be used for phase recognition when only the C-arm is used as a data source consist of the features in the feature matrix with an index of 2-12. The object detection features that were gathered using the recorded videos of the procedures consist of the index numbered 13-16. The features that were introduced using the clinical data set are indexed with numbers 17-21. It is important to note that the features of the clinical data were included in all models, however, since they did not add any benefits for phase recognition, they have not been included in the results. The clinical data has been important for correcting the timestamps of the annotations by linking the columns **ExamID** and **StudyNumber**, as described in Section 4.6.2.

The data is split not only on the feature dimensions but also on the timestamp dimension. While the video data and the full data set have data for every second of the procedure, the data set of the logging data only consists of data during the start or end of a movement or acquisition. A visualization of this is shown in Figure 4.21

4.9. One-Hot Encoding

Before using the data from the feature matrix depicted in Figure 4.4 for training the random forest model, it is important to address the presence of categorical variables. Given that the random forest algorithm is incapable of processing categorical values directly, these must be transformed into numerical equivalents. The categorical features, along with their respective categories, are presented in Table 4.6. Since the categories are nominal in nature, a technique known as one-hot encoding is used for the transformation. One-hot encoding is a method that



Figure 4.21: A scatter plot of the the object detection and C-arm data plotted against time.

converts categorical features into numerical form by dividing each feature into separate binary columns corresponding to each category. Since a random forest is not able to interpret ordinal data, the category were separated into their own binary columns, leading to an increase in its dimensionality. The dimensions of the full data set after the described processing are 615860 rows \times 109 columns.

Feature Name	Categories
StartEnd	Acquisition Start Time, Acquisition End Time, Movement Start Time, Movement End Time, Bounding Box Data
Side	Left, Right
Gender	Male, Female
Part of Day	Morning, Afternoon
Procedure Type	CAG wrist, CAG groin, CAG bypass wrist, CAG bypass groin

Table 4.6: Categories associated with the categorical features in the matrix.

4.10. Random Forest Classification

In this section the procedure of phase recognition using a random forest model is explained. For the implementation of the model, the Python library scikit-learn v is used. Because we want to find the best possible performance for different feature sets, we will optimize the model separately for each one. The optimization method and a basic explanation of the hyper-parameters that were optimized are described in this section.

4.10.1. hyper-parameter Tuning: Random Search and Grid Search

The method of optimization used for each model was a random search. This is a method of hyper-parameter tuning using a trial-and-error approach. For each hyper-parameter that will be tuned, a range of values is chosen, and subsequently, the random search will try out random combinations of these values for a set number of iterations. An alternative method for hyper-parameter tuning is a grid search, which will try out each combination of hyper-parameters in the search. While a grid search offers a more extensive tuning of the hyper parameters, the downside of a grid search is that the number of iterations, and therefore the computing time, increases exponentially with the increasing number of hyper-parameters that are tuned. The benefit of a random search is that the number of iterations can be chosen, limiting the computing time that will be spent on searching for the best parameters. The total number of combinations of hyper-parameters values was 2400, the random search was set to randomly select 60 sets of parameters from these combinations. The value of 60 iterations was chosen to limit the computing time to less than 3 hours. A description of the hyper-parameters and the value sets per parameter are shown in Table 4.7. //

The result of the random search over a wide range of hyper-parameter values results in a set

Hyper Parameter	Description	Compromise
Number of Trees	The number of trees that are trained and used in the RF model.	More trees can result in better results but a more computationally expensive model.
Max Features	The number of features that are used to train each tree in the RF model.	More features per tree most likely results in better results, but also in a more computationally expensive model. auto = no restrictions on number, sqrt = square root of total feature number, log2 = log2 of total feature number
Max Depth	The maximum depth of each tree in the RF model, limiting the number of splits.	Higher max depth will most likely result in better results, but also a more computationally expensive model. A high max depth can also lead to over-fitting.
Minimum Samples Split	The minimum number of samples required to allow for a split.	A high value might miss details that are specific to the sample data for a tree. With a low value, the model might lead to overfitting.
n_estimators	Number of trees to be used in the forest.	Increasing the number of trees will improve the accuracy but make the model slower.
min_samples_leaf	The minimum number of samples required to be at a leaf node.	A smaller leaf makes the model more prone to capturing noise in train data.
bootstrap	Whether bootstrap samples are used when building trees.	If False, the whole dataset is used to build each tree.

Table 4.7: A description of the hyper-parameters that were tuned using a random search and the compromises that are inherent to each of them.

Hyper Parameter	Random Search	Grid Search: Full Dataset	Grid Search: C-Arm Data	Grid Search: Video Data
Number of Trees	[50,100,150]	[80,100,120]	[80,100,120]	[90,100,110]
Max Features	[auto,sqrt]	[sqrt]	[sqrt]	[sqrt]
Max Depth	[10,20,30,None]	[8,10,12]	[None,2,4]	[8,10,12]
Min Samples Split	[2,3,...,9,10]	[2,3,4]	[6,7,8]	[6,7,8]
Min Samples Leaves	[1,2,3,4,5]	[3,4,5]	[1,2,3]	[3,4,5]
Bootstrap	[True, False]	[True]	[False]	[False]

Table 4.8

4.10.2. Validation

To assess the performance of the model, the data is divided into training and testing subsets at the observation level. This ensures that procedures are entirely assigned to training or testing data, preventing data leakage. This separation reflects a real-world scenario in which the model is unfamiliar with the procedure on which classification is applied. Data separation is performed using a validation technique known as k -fold cross-validation. In this method, the dataset is partitioned into k equal parts, referred to as folds, with each fold containing an approximately equal number of procedures. These k folds are then used in the cross-validation

process, where the model is created and validated k times. During each iteration, the training set consists of $k-1$ folds, and each fold is used as a test set exactly once. The final performance assessment is based on the average of the performance metrics across all iterations.

The main reason this method was chosen over simple hold-out validation is to decrease the chance of misleading results due to the chance that the random set of procedures chosen to train the model during this method would generalize very well to the test set that was chosen by chance. In essence, the cross-validation technique is more robust. A downside of this method is that the computation time is increased. Because of the high computation time, the k in the k -fold during the grid search was set to 3.

4.11. Reducing Granularity

A secondary random forest classification is conducted following the initial analysis with C-arm data and video-derived data. The aim of this subsequent classification is to predict procedural phases based on a less granular, yet clinically relevant workflow. For simpler applications in the catheterization laboratory, such as monitoring, planning, and rough time estimation, a less detailed workflow definition may be adequate. As a result, this study also evaluates a workflow model that bundles multiple operative phases into a single intervention phase, named Phase M. Furthermore, the classification performance will be assessed for two distinct intervention phases: one for the first coronary artery and another for the second coronary artery. The new formalizations of the workflow with the bundled intervention phase can be seen in Figure 4.22. The clinical relevance of the phases for planning purposes and remaining time predictions have been verified by a cardiologist of the Reinier de Graaf Hospital.

4.11.1. Random Forest Model Training

The models were trained on the combined C-arm and video-derived dataset. Due to time constraints, we did not perform hyperparameter tuning for this secondary classification. Instead, we used the same hyperparameter settings that were applied to the full granularity workflow. The results of the performance for both the model trained on the bundled phase M and the model trained on the bundled phases M_First, and M_Second can be seen in the results section.

CAG Procedure Timeline with Intervention Phase

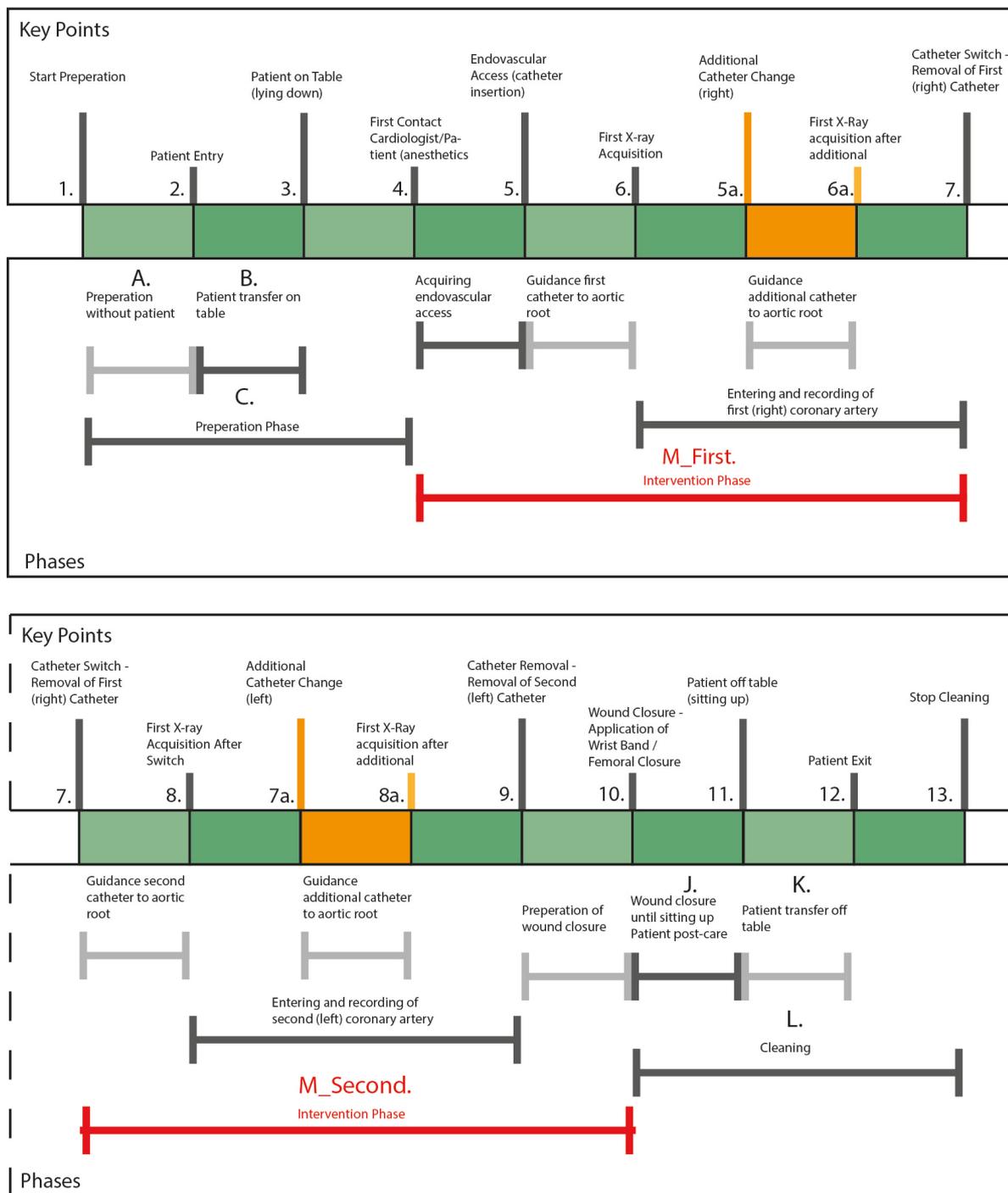


Figure 4.22: The formalization of the procedure as described in Section 4.1 with the bundled phase M which is separated in a first coronary artery and second coronary artery section.

4.12. Phase Recognition for Predicting Procedure Delay

To evaluate the clinical applicability of the predicted phases, a classification task is performed to predict procedural delays. A "delay" in this context is defined as any procedure that exceeds the 45-minute time block currently allocated for coronary angiography (CAG) procedures in the Reinier de Graaf Hospital. The 45-minute threshold is based on hospital scheduling. A more nuanced three-class model is also considered, introducing a 'severely delayed' category for procedures with a duration over 60 minutes. This added class aims to capture extreme cases where delays might significantly impact hospital resources.

4.12.1. Data

The dataset for this classification task comprises two variables: the predicted phases as described in Section 4.10, and the elapsed time since the procedure began, measured in seconds. Using this data set can give us an idea of the performance of a potential use case for classifying phases in coronary angiography procedures. The expectation is that the model will, to some extent, be able to assess whether a phase occurs earlier or later in the procedure than usual and is therefore able to predict whether the whole procedure will have delay or not.

4.12.2. Annotations

For the classification task, annotation of the data is necessary. Procedures are labeled as 'on-time' if their duration is less than or equal to 45 minutes, which is the standard time allocation in the Reinier de Graaf Hospital for CAG procedures. Procedures that exceed this duration are labeled 'delayed'. In the second model, a class was added for procedures with a duration over 60 minutes which was classified as 'severely delayed'.

4.12.3. Random Forest Classification

Due to the exploratory nature of this research, Random Forest classification was performed using the standard hyperparameters as detailed in the sklearn documentation [9]. Unlike the full granular phase prediction described in Section 4.11, these standard hyperparameters were employed as this classification problem was considered too distinct to generalize hyperparameters from previous models.

4.12.4. Model Performance

This subsection outlines the methods used to evaluate performance of the models. To improve accuracy, a majority vote approach was employed, displaying the predicted classes over 5-minute intervals for each procedure. For example, the majority vote for the first 5 minutes combines the initial $5 * 60 = 300$ predictions. The accuracy of each interval was then taken over all procedures and plotted on a graph. This can be seen in the results section.

5

Results

The objective of this study is to validate the usability of video and C-arm system log data collected from a catheterization laboratory to automatically recognize the phases of coronary angiography procedures for use of monitoring and estimated remaining time predictions. The reasoning behind the performance metrics used and the performance benchmarks defined are detailed in Chapter 4.4. This chapter will give an overview of the results of the models that were developed for the purpose of this thesis. These consist of the results of the models:

- Baseline model using average contribution of phases to procedure length.
- Random forest model using only the C-arm system logging data.
- Random forest model using only the object detection data obtained from the videos.
- Random forest model combining C-arm system logging data and object detection data.

5.1. Performance

The performance of the models has been calculated as described in the methodology in Section 4.4. The per class accuracy of the baseline model is shown in Table 5.5. The model had an overall accuracy of 45.80%. The best predicted phase was phase A: preparation of the cathlab before the patient enters, with an accuracy of 65.45%, making up 18.02% of the procedure on average. The worst predicted phase was phase Ha: catheter guidance to the aortic root during recording of the second coronary artery, with an accuracy of 2.54%. On average, this phase constitutes 2.53% of the procedure.

The C-arm model, shown in Figure 5.6, achieved a higher accuracy than the baseline model, with a total accuracy of 80.73%. Each of the 47,284 seconds recorded by the C-arm machine log contained either a movement or an acquisition. The phase with the highest accuracy was phase F: the entry and recording of the right coronary artery, attaining 85.32% and generally making up 65.54% of the procedure. The model performed the worst on classifying phases A, B, Fa, Ha, K, and L with an accuracy of 0.00%. These phases combined consisted of 2.46% of the dataset.

The overall accuracy of the model using only the object detection data reached 63.80%. It was best at predicting phase A, with an accuracy of 92.15%, and the lowest accuracy was reached by predicting phases E and G at 0.00%. Phases E and G culminate to 3.06% of the time of all procedures.

The combined data set achieved a performance of 79.46%. The highest precision was obtained classifying phase A, with 95.22%, and the lowest accuracy was obtained in phase Fa and Ha, at 0.00%. Phases Fa and Ha contribute for 2.7% to the dataset.

Table 5.1: Per Phase Accuracy Baseline Model

Phase	Accuracy	N	N%
A	65.45%	110,959	18.02%
B	2.73%	6,454	1.05%
C-(A&B)	52.78%	132,833	21.58%
D	32.12%	78,129	12.69%
E	2.63%	5,131	0.83%
F	30.02%	70,541	11.46%
Fa	2.95%	7,247	1.18%
G	7.90%	13,701	2.23%
H	49.39%	87,006	14.13%
Ha	2.54%	9,389	1.52%
I	30.34%	24,016	3.90%
J	61.55%	45,698	7.42%
K	20.03%	8,454	1.37%
L-(J&K)	63.25%	16,301	2.65%
Total	45.80%	615,859	100%

Table 5.2: Per Phase Accuracy C-arm Model

Phase	Accuracy	N	N%
A	0.00%	14	0.03%
B	0.00%	6	0.01%
C-(A&B)	53.19%	610	1.29%
D	69.67%	1,922	4.07%
E	60.85%	435	0.92%
F	85.32%	12,551	26.54%
Fa	0.00%	318	0.67%
G	81.87%	1,329	2.81%
H	80.46%	23,852	50.45%
Ha	0.00%	658	1.39%
I	77.92%	4,251	8.99%
J	81.05%	1,191	2.52%
K	0.00%	40	0.08%
L-(J&K)	0.00%	127	0.27%
Total	80.73%	47,284	100%

Table 5.3: Per Phase Accuracy Object Detection Model

Phase	Accuracy	N	N%
A	92.15%	110,959	18.02%
B	67.79%	6,454	1.05%
C-(A&B)	72.16%	132,833	21.58%
D	58.15%	78,129	12.69%
E	0.00%	5,131	0.83%
F	47.20%	70,541	11.46%
Fa	7.42%	7,247	1.18%
G	0.00%	13,701	2.23%
H	47.34%	87,006	14.13%
Ha	3.14%	9,389	1.52%
I	25.37%	24,016	3.90%
J	52.24%	45,698	7.42%
K	62.36%	8,454	1.37%
L-(J&K)	66.72%	16,301	2.65%
Total	63.80%	615,859	100%

Table 5.4: Per Phase Accuracy Combined Data Model

Phase	Accuracy	N	N%
A	95.22%	110,959	18.02%
B	83.33%	6,454	1.05%
C-(A&B)	81.36%	132,833	21.58%
D	76.16%	78,129	12.69%
E	28.16%	5,131	0.83%
F	79.65%	70,541	11.46%
Fa	0.00%	7,247	1.18%
G	80.84%	13,701	2.23%
H	70.00%	87,006	14.13%
Ha	0.00%	9,389	1.52%
I	74.84%	24,016	3.90%
J	70.11%	45,698	7.42%
K	82.81%	8,454	1.37%
L-(J&K)	80.36%	16,301	2.65%
Total	79.46%	615,859	100%

Figure 5.1: Comparison of Accuracy across Different Models

5.2. Feature Importance

The importance of features has been calculated for the random forest models. These are the fractions of contribution of each feature to the total accuracy of the model. The *time since the last acquisition* is the most important feature for both the model using the C-arm system log data and the model that uses the combined data, as can be seen in 5.2 and 5.4, respectively. The most important feature for the model that uses the object detection data obtained from the videos is the *cumulative time since the start of the procedure*, as can be seen in 5.3. This feature is the 17th and 12th most important feature of the C-arm and combined models, respectively. *Cell 1_1* is the second most important feature of the model using the object detection data. However, this cell is not present in the analysis of the importance of features of the combined data model. The second most important feature of that model is cell *1_3*, which is the 4th most important feature of the model that uses object detection data. For the model in Figure 5.2, the 5 most important features amount to a total importance of 28.3%. For Figure 5.3 this is 56.5%, and for Figure 5.4 this is 17.35%.

5.3. Example Procedure Visualization

An example CAG procedure that occurred on November 2 in 2021 was used to visually compare the performance of the models in Figure 5.6. In Figure 5.6-a the baseline model is shown. In Figure 5.6-b, we can see the model using the C-arm data. As we can see, most of the movements and acquisitions are concentrated in a few phases. In Figure 5.6-c, we can see the object detection-based model, which does include each second of the procedure. The model using the combined data can be seen in Figure 5.6-d. In the visualization it is visible that smaller phases, like phase B, E, G, K, and L do not get picked up by the model during prediction.

5.4. Performance of Model with Reduced Granularity

The performance of the models which were trained on the combined data set but with a decreased granularity, bundling the procedural phase to phase M, or to phase M_First and M_Second as described in 4.11 are shown in the same manner as the previously trained models. The per phase accuracy can be seen in Figure 5.7.

The accuracy of the model that used a single phase M as the procedural phase had a total accuracy of 88.23%, which is 8.77% more than the model that used full granularity. The accuracy of the model which split the phase M into the procedural phase during the entering and recording of the first and second coronary artery was 85.29%, which is 2.94% less accurate than the model with a unified phase M.

5.5. Using Phase Recognition for Predicting Procedure Delay

The results of the models that aim to predict whether the procedure has a delay is shown in Fig 5.10. Two figures are shown, the left figure is the result of the binary prediction, where the procedures were classified as either 'on-time' or 'delayed', with a threshold of 45m. The right figure is the result of the prediction that classified 3 classes: on-time [$<45m$], delayed [$45m < x < 55m$], and severely delayed [$>55m$].

In both models, the accuracy generally seems to increase over time. This means that later in the procedure, we are more certain of whether the procedure will be delayed or not.

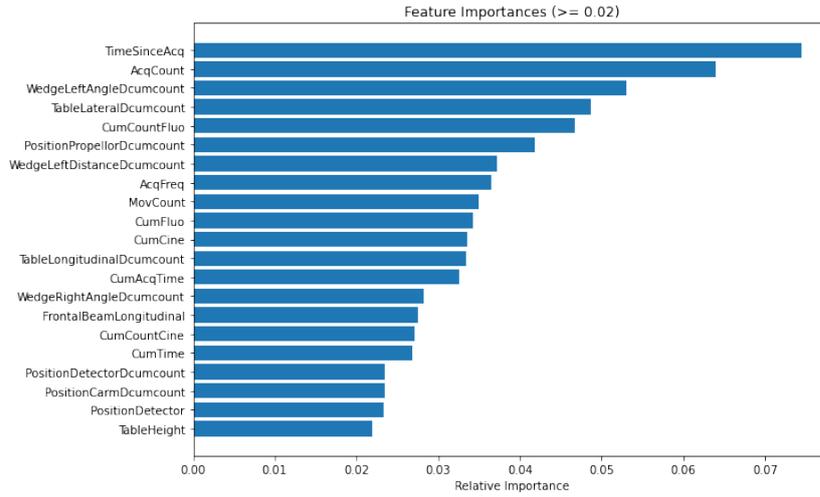


Figure 5.2: Feature Importances of C-Arm System Log

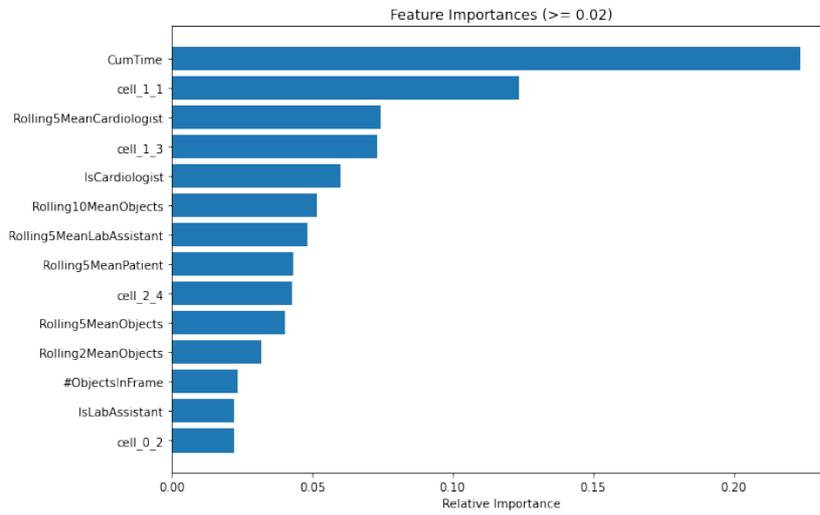


Figure 5.3: Feature Importances of Object Detection

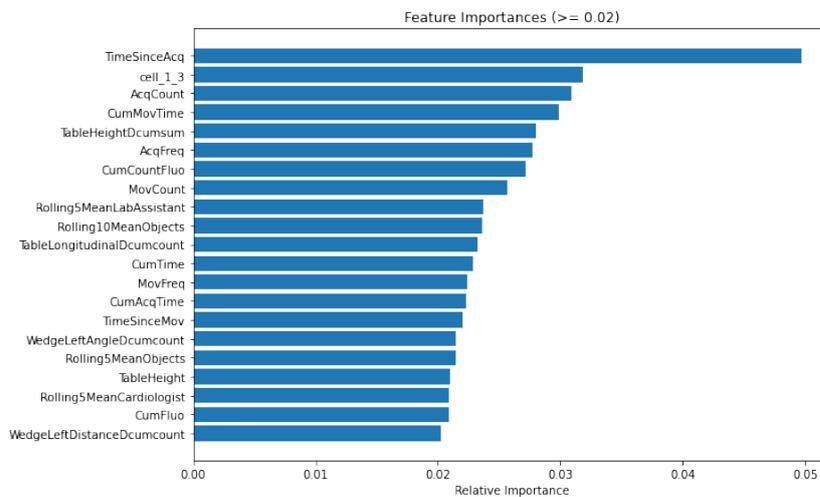


Figure 5.4: Feature Importances of Combined Data

Figure 5.5: The feature importances of the models. Relative importance can be seen as the percentage of contribution of the feature to the outcome of the model

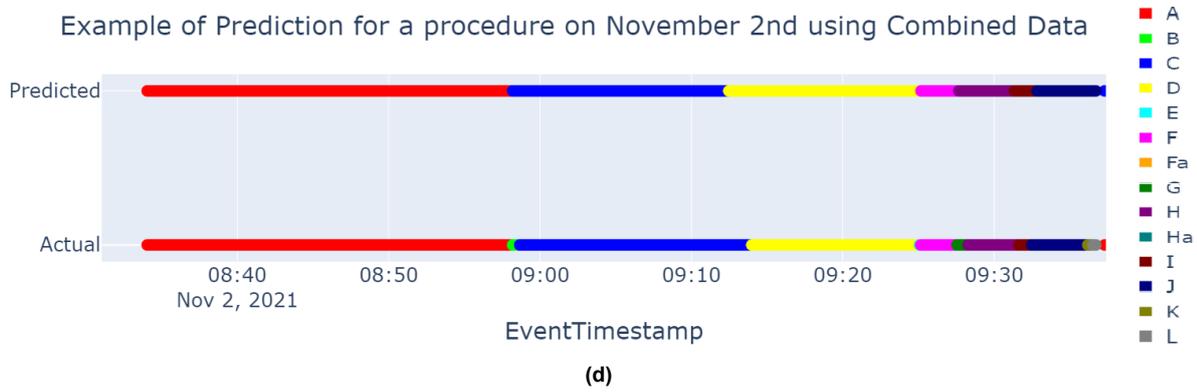
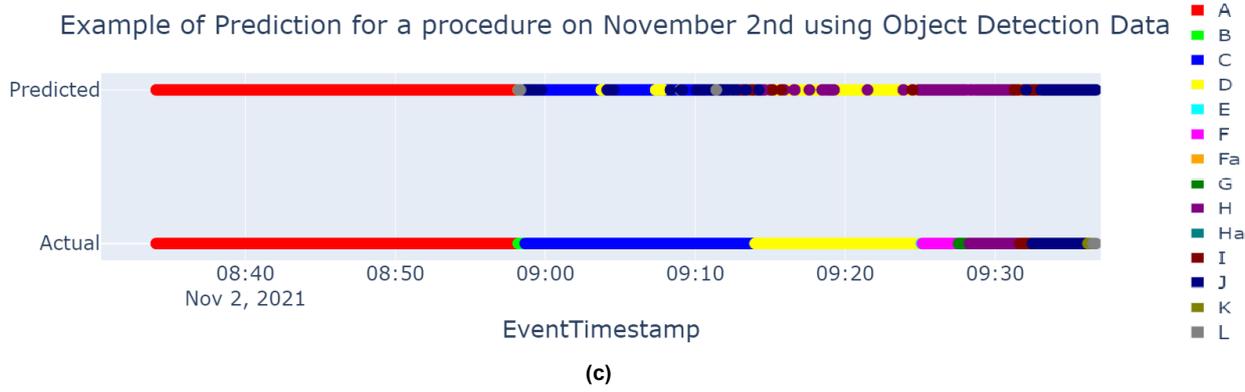
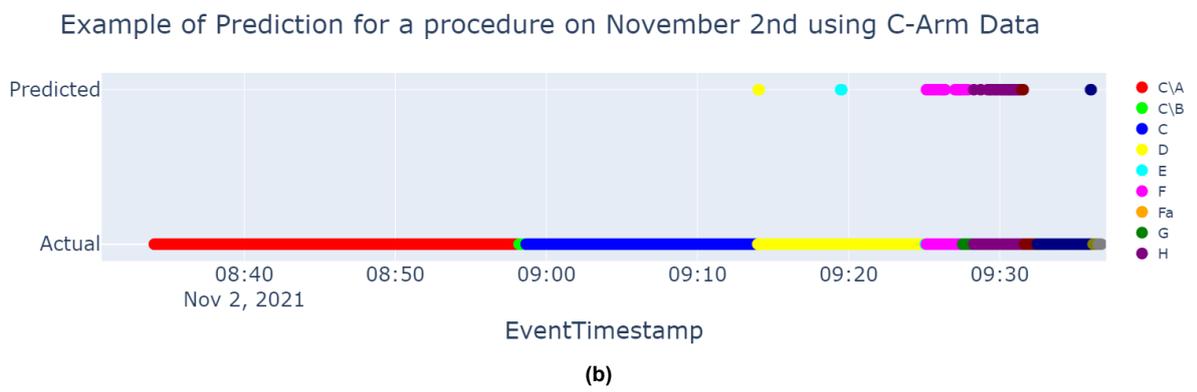
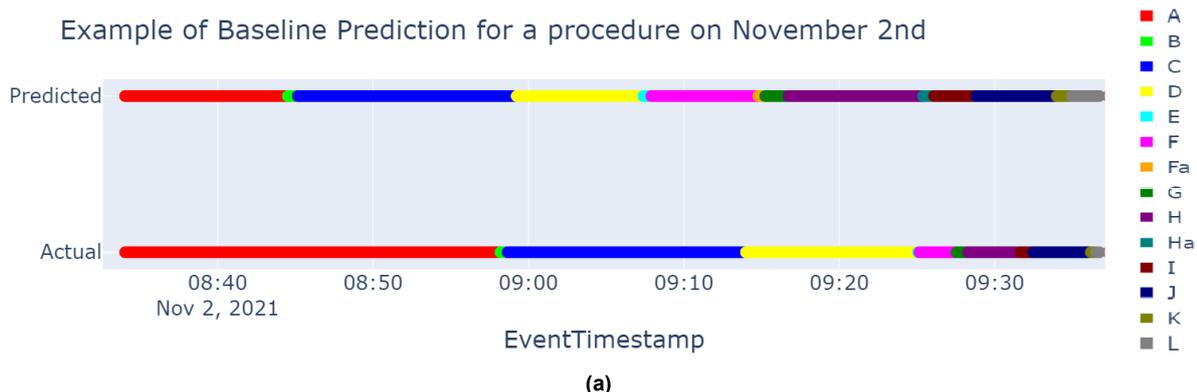


Figure 5.6: A comparison of the prediction of the models for an example procedure on November 2nd.

Table 5.5: Per Phase Accuracy with Phase M

Phase	Accuracy	N	N%
A	95.02%	110,959	18.02%
B	82.35%	6,454	1.05%
C-(A&B)	81.27%	132,833	21.58%
M	92.67%	297,160	48.3%
J	70.51%	45,698	7.42%
K	76.32%	8,454	1.37%
L-(J&K)	76.29%	16,301	2.65%
Total	88.23%	615,859	100%

Table 5.6: Per Phase Accuracy With Phase M Split into First and Second Coronary Artery

Phase	Accuracy	N	N%
A	95.02%	110,959	18.02%
B	85.71 %	6,454	1.05%
C-(A&B)	81.51%	132,833	21.58%
M_First	85.22%	161,048	26.15%
M_Second	81.87%	134,112	21.78%
J	71.14%	1,191	2.52%
K	80.90%	8,454	1.37%
L-(J&K)	80.92%	16,301	0.27%
Total	85.29%	615,859	100%

Figure 5.7: Comparison of Accuracy across Reduced Granularity Phase Recognition

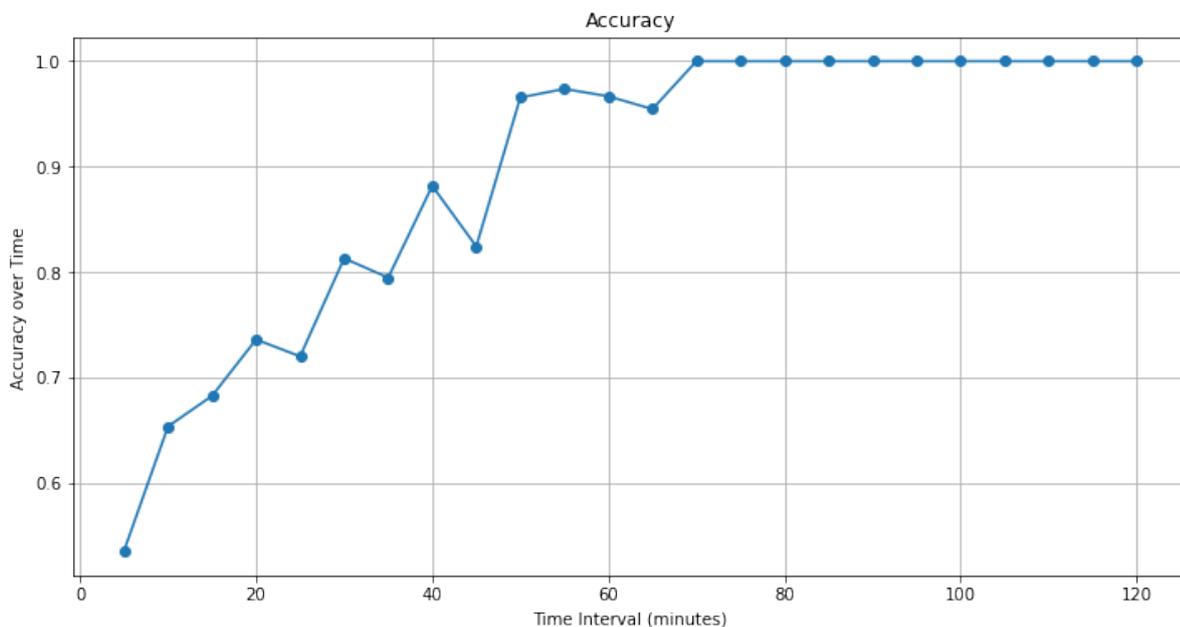


Figure 5.8: Binary Prediction of Delay with 45m as Threshold

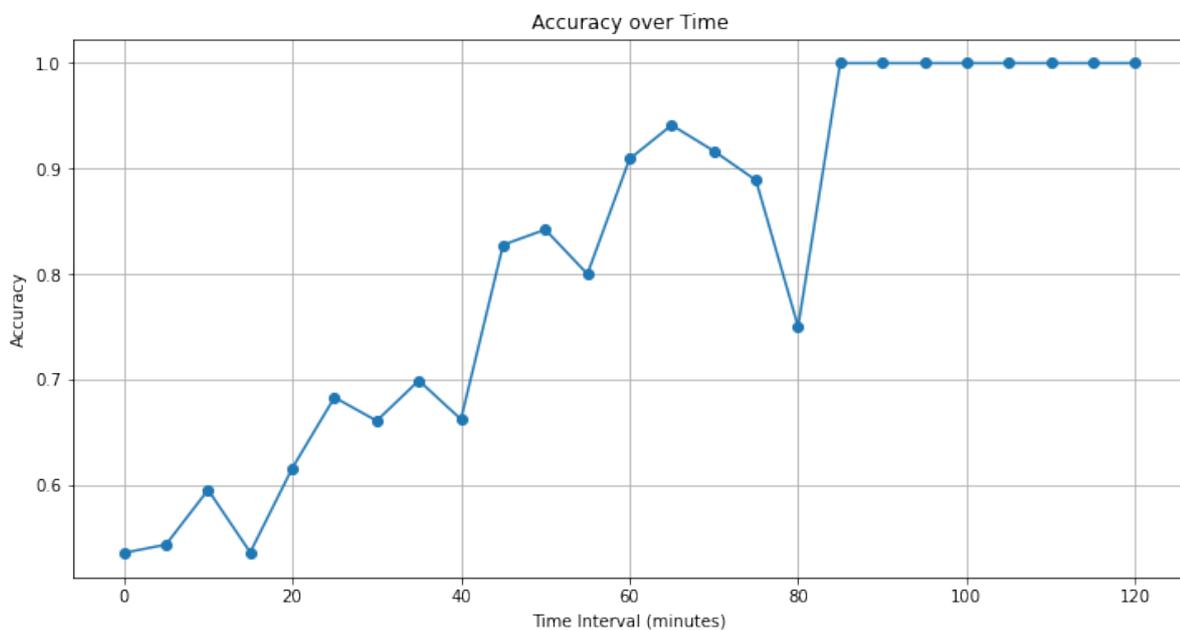


Figure 5.9: 3-Clas Prediction of Delay with 45m and 55m as Thresholds

Figure 5.10: Graphs of the prediction accuracy for predicting whether a procedure has a delay over time. The prediction was done using a random forest model after which the majority vote over 5 minute segments was taken. The performance of this majority vote over time can be seen in this figure. Both the models using 2 classes and 3 classes can be seen.

6

Discussion

6.1. Interpretation of the Classification Results

6.1.1. Baseline Model

The baseline model achieved a total accuracy of 45% by using only the average contribution of each phase to the total duration of the procedures. This outperforms random phase selection, which would lead to a performance of $\frac{1}{14\text{phases}} * 100\% = 7.14\%$. However, the model is limited in its predictive capabilities. Furthermore, it requires the total duration of the procedure as input, making it unsuitable for real-time phase prediction. Lastly, it only offers insights into an average procedure and relies solely on duration for output variation, limiting its utility for planning and postoperative analysis.

When examining per-phase accuracy, the model is less accurate for shorter phases. This is likely due to the higher chance of duration overlap in longer phases.

6.1.2. C-arm Model

When using the C-arm data to train a random forest model, the total accuracy reached was 80.73%. This initially suggests that the data is useful for phase recognition. However, when looking at the per class accuracy, it is observed that only a selection of phases actually perform well, and the total accuracy is influenced by class imbalance. This situation highlights the drawback of using total accuracy as a performance metric. While it provides a convenient and comparable representation of the performed classifications, it fails to accurately depict the performance of the overall task, which is to classify each phase in the procedure. Therefore, examining the per class accuracy is also necessary.

The class imbalance in the C-arm model is due to the C-arm data encompassing only information on the machine's movements and X-ray acquisitions. As expected, acquisitions and movements mainly take place during the entering and recording of the first and second coronary arteries (phases F and H), constituting 77.0% of all the movements and acquisitions, but only 25.6% of the procedure. The abundance of data for these phases, along with high accuracies (85.32% for phase F and 80.46% for phase H), demonstrates the usability of this data for the recognition of these phases. The data was also valuable for other phases with frequent acquisitions or movements, such as the guidance of the catheter to the first (60.85%), and second (81.87%) coronary artery, the preparation of the wound closure (77.92%), and the patient post-care (81.05%). However, the data is less useful for phases with minimal acquisitions and movements, restricting the model's utility for planning and post-procedural analysis.

6.1.3. Object Detection Model

The object detection data was obtained using an object detection model that was developed by R. Dai in 2022 to detect objects in the video data that was collected in the cathlab. This data includes the position and size of bounding boxes around detected objects. A random forest model trained on this data achieved a total accuracy of 63.8%, a notable decrease compared to the C-arm model. It is again important to realize that the total accuracy was influenced by class-imbalance. However, the videos span the entire procedure and each row corresponds to a second in the procedure, so the class imbalance is a result of the variety in duration of the phases and not due to the frequency of events in specific phases. Taking into account class imbalance, it is important to look at the per phase accuracy and assess the usability of the data for the phases separately.

The object detection model excels at the classification of the first phase, the preparation of the cathlab before the patient enters, with an accuracy of 92.15%. This can be explained by the fact that the phase is characterized by the absence of a 'patient' object, allowing the model to correctly classify this phase using the patient object and the cumulative time. The cleaning phase after the patient exits is another phase defined by the absence of a specific person, but it has a lower accuracy of 66.72%. This may be due to the variability in cumulative time around the last phase compared to the consistent start time of the first phase.

Examining per class accuracies suggests that combining this data with the C-arm data could be beneficial. The object detection model excels in the early and late stages, while the C-arm model is more accurate in the operational phase. The object detection data could be clinically useful, particularly in detecting delays in the early stages. However, its lower reliability during operational phases limits its applicability for postoperative analysis and detailed procedure duration predictions.

One limitation of using the object detection model is that it introduces a layer of abstraction over the raw video data, which may result in a loss of information. On the other hand, a notable advantage is that the abstraction process inherently anonymizes the data, removing any personally identifiable information and making it more suitable for wider research and application purposes.

6.1.4. Combined Data Model

Combining the C-arm data and object detection data to train a random forest model results in an accuracy of 79.46%, similar to using only the C-arm data. However, since the data of the C-arm model has been extended and enriched with that of the object-detection model, the combined model can identify the phase in every second of the procedure. While the total accuracy is high, analyzing the per phase accuracy reveals shortcomings in the model. It achieves 0% accuracy in identifying phases Fa and Ha, which involve guiding additional catheters to the aortic root during the overarching phase of entering and recording the coronary arteries. These phases occur only when the catheter guided to the coronary artery fails to enter and an additional catheter has to be used. The model's inability to recognize these phases may stem from the data being insufficiently distinct from the overarching phases F and H, or from the infrequency of additional catheters being used. Consequently, the model may have learned to overlook these phases to optimize performance.

An interesting observation is that certain phases show increased accuracy compared to the object detection model, even when the C-arm model provides no additional useful information with an accuracy of 0%. These phases include the transfer of the patient on the table, the transfer of the patient off the table, and the cleaning of the cathlab after the patient has left. Figure 5.6 also demonstrates this phenomenon during the preparation while the patient is lying down. The reason for this is not immediately clear but could be attributed to the interplay of

feature importance assigned to the data. The C-arm data, stable during early and late phases when the system is idle, might have a noise-reducing effect, enhancing accuracy. Alternatively, this could result from a random interplay of features that isn't evident from the results.

6.1.5. Reduced Granularity Model

As stated in methodology, the granularity of the workflow was chosen to be high for the purpose of our research objectives, with the understanding that it's easier to decrease granularity than increase it. For the model to be useful for medical planning and intraoperative time predictions, a trade-off must be made between model performance and model granularity. Figure 5.7 indicates that when the operative phases are consolidated into a single phase M, the combined data model's total accuracy rises to 88.23%, with phase M achieving an accuracy of 92.67%. Notably, especially the early phases and the operative phase reached an accuracy of well over 80%. If the phase M is split into two operative phases for the first and second coronary artery, the accuracy decreases slightly to 85.29%. A decrease of around 2% could be a worthwhile trade-off for the increased granularity of the operative phase when looking at the clinical implementation. Introducing a second phase that divides phase M approximately in half could offer a more detailed view of the procedure's progress and improve remaining time prediction performance. Finding the optimal balance between granularity and performance could be a valuable direction for future research.

6.1.6. Predictions of Delay Model

When using only the phases and cumulative time to predict whether a procedure will experience delay and doing this at 5 minute intervals, the results are promising. It is important to note that this prediction is merely an avenue to show the use of phase prediction in practice. The results can be interpreted as a proof that the phases as predicted by our model can in fact be used to predict whether a procedure has a delay. However, for actual clinical use, a numerical prediction based on all the data from the C-arm and from videos could be of greater use. The accuracy of the model increases over time, aligning with expectations, as the progress of a procedure makes it clearer whether a phase occurs earlier or later than anticipated. The model probably uses the timing of a phase within a procedure to gauge whether the total duration will surpass 45 minutes. An interesting observation is that the model does not reach an accuracy of 100% when it passes the threshold of 45 minutes. One might expect the model to recognize that any timestamp beyond 45 minutes indicates a delayed procedure. Investigating the reason for this is beyond the scope of this thesis, as the result is presented merely as an illustrative example of data usage for predicting procedure delays.

6.1.7. Clinical Implementation and Research Objectives

Model's Suitability for CAG Phase Recognition

The combined data model, which integrates C-arm and object detection data, achieved an accuracy of 79.46%. When the granularity of the workflow is reduced, the accuracy improves to 88.23%. This indicates that the data can be effectively used for phase recognition in coronary angiography procedures leading to basic applications like predicting the remaining time. However, since the model that was implemented is relatively simple and is only able to take into account temporal data through the use of feature engineering which doesn't guarantee that all useful temporal data is extracted, it could be beneficial to implement a model that is able to use the temporal data to its fullest. In addition, the extraction of useful data from the video data is limited and there is more potential still in the video data of all angles.

The model proves the usability of the data for the purpose of phase recognition of clinically

relevant phases. However, before the data and model or any improved model can be used in practice steps have to be taken in automatically extracting and processing the data in real-time, running the model continuously during procedures, storing the data and creating automatic visualizations for insights.

Comparison with Baseline Prediction

The baseline model, relying on average phase duration, achieved an accuracy of 45%. In comparison, the combined data model demonstrates a significant improvement of nearly 35% indicating the usefulness of the data for the task of phase recognition.

Advantages of Integrating C-arm and Video Data

Integrating data from C-arm logs and video recordings has several advantages:

- **Comprehensive Coverage:** The combined model can identify phases throughout the entire procedure which is an improvement from only using the C-arm data.
- **Improved Accuracy:** Almost all phases show an increase in accuracy compared to the individual models.
- **Balance between Granularity and Performance:** Reducing granularity slightly still yields high accuracy, allowing for practical trade-offs in real-time applications.

However, the combined model faces limitations such as being unable to accurately predict certain specific phases, like guidance of additional catheters.

6.2. Limitations and Future Research Directions

While the model using C-arm and video data shows promise, it's important to consider the limitations in the data, data processing, and model. The main limitations are discussed in this chapter:

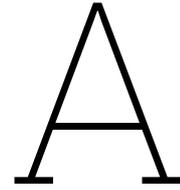
- **Random Forest is not a Temporal Model:** The random forest model used is not inherently capable of assessing temporal data. To address this, temporal features relying on current and past data were added. However, engineering these features is a human task and may not capture the full range of available temporal information. Analyzing the feature importances in Figure 5.5, we observe that time-dependent features like counts and cumulative values are deemed important by the models, suggesting that temporal data can be crucial for achieving high performance. Consequently, future research could explore implementing a model better suited to integrate all available temporal data into the decision-making process.
- **Dataset of 239 Procedures:** While a dataset of 239 procedures is high in comparison to similar research for other medical procedures, it is still a limited amount. To increasingly improve model performance and to generalize better to unseen data, the dataset size could be improved even further.
- **The Object Detection Data is Limited:** The object detection model used introduces an abstraction over the video data, potentially leading to information loss. However, this abstraction layer has the benefit of anonymizing the data. Additionally, information loss occurred by utilizing only one of the five cameras available in the cathlab. The selected camera angle selected by visual evaluation, however this does not guarantee that this was the optimal angle for our objective. There is significant room for improvement in providing the classification model with additional data from the videos, making further research in this area worthwhile.

- **Optimizing Granularity vs. Performance:** In this thesis, the performance of models on three different granularities are assessed, demonstrating that reducing the granularity in these cases increases performance. However, the optimal balance between performance and granularity for optimal clinical use remains unclear. If a temporal model can achieve higher accuracies, a more granular approach may also attain clinically relevant accuracy. Exploring the balance between granularity and performance, and grounding this exploration in real-world application, could be a valuable direction for future research.

References

- [1] *Artificial intelligence - Search results - PubMed*. URL: <https://pubmed.ncbi.nlm.nih.gov/?term=artificial%20intelligence&timeline=expanded>.
- [2] Jakob E. Bardram et al. "Phase recognition during surgical procedures using embedded and body-worn sensors". In: *2011 IEEE International Conference on Pervasive Computing and Communications (PerCom)* (Mar. 2011). DOI: 10.1109/percom.2011.5767594. URL: <http://dx.doi.org/10.1109/percom.2011.5767594>.
- [3] Brett K. Beaulieu-Jones et al. "Trends and focus of Machine learning applications for health Research". In: *JAMA network open* 2.10 (Oct. 2019), e1914051. DOI: 10.1001/jamanetworkopen.2019.14051. URL: <https://doi.org/10.1001/jamanetworkopen.2019.14051>.
- [4] *Behavioral Coding - Event logging Software | The Observer XT*. URL: <https://www.noldus.com/observer-xt>.
- [5] *Cardiovascular diseases*. June 2019. URL: <https://www.who.int/health-topics/cardiovascular-diseases>.
- [6] Ke Cheng et al. "Artificial intelligence-based automated laparoscopic cholecystectomy surgical phase recognition and analysis". In: *Surgical Endoscopy* 36.5 (July 2021), pp. 3160–3168. DOI: 10.1007/s00464-021-08619-3. URL: <http://dx.doi.org/10.1007/s00464-021-08619-3>.
- [7] R. Dai. *Detecting Medical Equipment in the Catheterization Laboratory using Computer Vision*. 2019.
- [8] Mahtab J. Fard et al. "Distance-based time series classification approach for task recognition with application in surgical robot autonomy". In: *The International Journal of Medical Robotics and Computer Assisted Surgery* 13.3 (Aug. 2016), e1766. DOI: 10.1002/rcs.1766. URL: <http://dx.doi.org/10.1002/rcs.1766>.
- [9] *Feature importances with a forest of trees*. URL: https://scikit-learn.org/stable/auto_examples/ensemble/plot_forest_importances.html.
- [10] Stefan Franke et al. "The intelligent OR: design and validation of a context-aware surgical working environment". In: *International Journal of Computer Assisted Radiology and Surgery* 13.8 (May 2018), pp. 1301–1308. DOI: 10.1007/s11548-018-1791-x. URL: <http://dx.doi.org/10.1007/s11548-018-1791-x>.
- [11] Carly R. Garrow et al. "Machine Learning for Surgical Phase Recognition". In: *Annals of Surgery* 273.4 (Nov. 2020), pp. 684–693. DOI: 10.1097/sla.0000000000004425. URL: <http://dx.doi.org/10.1097/sla.0000000000004425>.
- [12] K. van der Graaf. *Formalization and Quantification of the workflow in the Catheterization Laboratory*. 2020.
- [13] Joel Grus. *Data science from scratch*. Jan. 2015. URL: <https://ci.nii.ac.jp/ncid/BB18672167>.
- [14] Yue Gu et al. "Language-Based Process Phase Detection in the Trauma Resuscitation". In: *2017 IEEE International Conference on Healthcare Informatics (ICHI)* (Aug. 2017). DOI: 10.1109/ichi.2017.50. URL: <http://dx.doi.org/10.1109/ichi.2017.50>.

- [15] Yue Gu et al. “Multimodal Attention Network for Trauma Activity Recognition from Spoken Language and Environmental Sound”. In: *2019 IEEE International Conference on Healthcare Informatics (ICHI)* (June 2019). DOI: 10.1109/ichi.2019.8904713. URL: <http://dx.doi.org/10.1109/ichi.2019.8904713>.
- [16] S. Imming. *The usability of interventional X-ray data for intraoperative prediction of coronary angiography procedure duration*. 2020.
- [17] Corporate Finance Institute. “Bagging (Bootstrap aggregation)”. In: *Corporate Finance Institute* (May 2023). URL: <https://corporatefinanceinstitute.com/resources/data-science/bagging-bootstrap-aggregation/>.
- [18] C. Kensen. *Applicability of Video-based workflow monitoring of interventional cardiac procedures performed in Reinier de Graaf Hospital*. 2019.
- [19] M.L. Knapik. *Cardiac Catheterization Laboratory Facility Design and Equipment Selection*. 2002. URL: <https://www.hmpgloballearningnetwork.com/site/cathlab/articles/cardiac-catheterization-laboratory-facility-design-and-equipment-selection>.
- [20] Florent Lalys and Pierre Jannin. “Surgical process modelling: a review”. In: *International Journal of Computer Assisted Radiology and Surgery* 9.3 (Sept. 2013), pp. 495–511. DOI: 10.1007/s11548-013-0940-5. URL: <http://dx.doi.org/10.1007/s11548-013-0940-5>.
- [21] Bo Liu et al. “When machine learning meets privacy”. In: *ACM Computing Surveys* 54.2 (Mar. 2021), pp. 1–36. DOI: 10.1145/3436755. URL: <https://doi.org/10.1145/3436755>.
- [22] Kimimasa Sasaki et al. “Automated surgical workflow identification by artificial intelligence in laparoscopic hepatectomy: Experimental research”. In: *International Journal of Surgery* 105 (Sept. 2022), p. 106856. DOI: 10.1016/j.ijso.2022.106856. URL: <http://dx.doi.org/10.1016/j.ijso.2022.106856>.
- [23] *VZinfo | Volksgezondheid en Zorg*. 2021. URL: <https://vzinfo.nl/>.
- [24] Y. Weijenberg. “The Unexplored Application of Clinical Phase Recognition in the Catheterization Laboratory: a Literature Review”. In: (2022).
- [25] *What is Random Forest? | IBM*. URL: <https://www.ibm.com/topics/random-forest#:~:text=Random%20forest%20is%20a%20commonly,both%20classification%20and%20regression%20problems..>
- [26] Alice Zheng and Amanda Casari. *Feature engineering for machine learning*. ”O’Reilly Media, Inc.”, Mar. 2018.
- [27] Aneeq Zia et al. “Novel evaluation of surgical activity recognition models using task-based efficiency metrics”. In: *International Journal of Computer Assisted Radiology and Surgery* 14.12 (July 2019), pp. 2155–2163. DOI: 10.1007/s11548-019-02025-w. URL: <http://dx.doi.org/10.1007/s11548-019-02025-w>.



Full Dataset Tables

A.1. C-Arm Logging Data

Table A.1: An overview of the features in the acquisitions tab of the C-arm logging data used.

Feature Name	Description	Unit	Axes in Fig 3.3
ExamID	Unique identification code of the procedure	-	
AcquisitionID	Unique identification code of the X-ray acquisition	-	
AcqStartTime	Start time of the X-ray acquisition	dd-mm-yyyy hh:mm:ss	
AcqDuration	Duration of the X-ray acquisition	dd-mm-yyyy hh:mm:ss	
AirKerma	Measure to describe the amount of energy released by ionizing radiation in a unit mass of air.	J/KG = Gy	
DAP	Quantification of radiation exposure by multiplying the radiation dose by the irradiated area.	Gy*cm ²	
NoOfFrames	Number of frames that were generated during the acquisition	-	
Technique	Technique used for the acquisition.	-	
ShutterPositionLeft	The position of the left shutter.	10 ² * μm	
ShutterPositionTop	The position of the top shutter.	10 ² * μm	
ShutterPositionRight	The position of the right shutter.	10 ² * μm	
ShutterPositionBottom	The position of the bottom shutter.	10 ² * μm	
WedgeLeftDistance	?	?	

WedgeLeftAngle	?	?	
WedgeRightDistance	?	?	
WedgeRightAngle	?	?	
AngulationStart	The angulation of the C-arm at the start of the acquisition.	<i>deg</i>	C
AngulationEnd	The angulation of the C-arm at the end of the acquisition.	<i>deg</i>	C
RotationStart	The rotation of the C-arm at the start of the acquisition.	<i>deg</i>	D
RotationEnd	The rotation of the C-arm at the end of the acquisition.	<i>deg</i>	D
SourceImageDistance	Distance between the X-ray tube and the detector.	m	
PositionCarm	The unfiltered angulation of the C-arm.	$10^{-2} * deg$	C
PositionDetector	The vertical position of the detector.	$10^2 * \mu m$	1
PositionPropellor	The unfiltered rotation of the C-arm.	$10^{-2} * deg$	D
FrontalBeamLongitudinal	The longitudinal position of the C-arm.	$10^2 * \mu m$	4
FrontalRotateDetector	The rotation of the detector.	$10^{-2} * deg$	A
FrontalZrotation	Swing of the C-arm.		B
TableHeight	The vertical position of the operating table.	$10^2 * \mu m$	5
TableLateral	The lateral position of the operating table.	$10^2 * \mu m$	6
TableLongitudinal	The longitudinal position of the operating table.	$10^2 * \mu m$	7

A.1.1. Clinical Data

Table A.2: An overview of the features in the clinical database.

Feature Name	Description	Unit
Study Number	A unique identification code given to the procedure	-
Date	The date of the procedure	-
Gender	The gender of the patient	-
Age	The age of the patient	Years
Procedure	The type of the procedure	-
Side	Which coronary artery was entered first.	-
Cardiologist	Which cardiologist performed the procedure (anonymized)	-
NoNurses	Number of nurses that were present during the procedure	-

Planned start time	Planned start time of the procedure according to electronic planning system	hh:mm:ss
Start new procedure	Start of the procedure according to the clinical notes.	hh:mm:ss
Patient on table	Moment patient is on table according to the clinical notes.	hh:mm:ss
Lidocaine	Moment of lidocaine injection according to the clinical notes.	hh:mm:ss
Cocktail	Moment of injection of a cocktail of medicin administered to aid catheter access through vessels.	hh:mm:ss
Right catheter	Start of catheterization for the first catheter used for the right coronary artery according to clinical notes.	hh:mm:ss
Left catheter	Start of catheterization for the first catheter used for the left coronary artery according to clinical notes.	hh:mm:ss
TR-band	Moment of application TransRadial band for assisting haemostasis according to clinical notes.	hh:mm:ss
End procedure	End of the procedure according to clinical notes.	hh:mm:ss
Right catheters used	The number of catheters used for the right coronary artery	-
Left catheters used	The number of catheters used for the left coronary artery	-
