Institutionen för datavetenskap
Department of Computer and Information Science

Final thesis

Automated Setup of Display Protocols

by

Patrik Bergström

LIU-IDA/LITH-EX-A–15/014–SE

2015-05-08

Linköpings universitet
SE-581 83 Linköping, Sweden

Linköpings universitet
581 83 Linköping

Linköpings universitet
Institutionen för datavetenskap

Final thesis

Automated Setup of Display Protocols

by

Patrik Bergström

LIU-IDA/LITH-EX-A–15/014–SE

2015-05-08

Supervisors: **Kristian Köpsén**
Sectra Imtec
Olov Andersson
IDA, Linköpings universitet

Examiner: **Cyrille Berger**
IDA, Linköpings universitet

Linköping University

Avdelning, Institution
Division, Department

Instutitionen för Datavetenskap,
Dept. of Computer and Information Science
581 83 Linköping

Datum
Date

2015-05-08

Språk
Language
☐ Svenska/Swedish
☒ Engelska/English
☐ _____

Rapporttyp
Report category
☐ Licentiatavhandling
☒ Examensarbete
☐ C-uppsats
☐ D-uppsats
☐ Övrig rapport
☐ _____

ISBN
ISBN

ISRN
LIU-IDA/LITH-EX-A–15/014–SE

Serietitel och serienummer ISSN
Title of series, numbering -

URL för elektronisk version
http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-117745

Titel
Title

Automated Setup of Display Protocols

Författare
Author

Patrik Bergström

Sammanfattning
Abstract

Radiologists' workload has been steadily increasing for decades. As digital technology matures it improves the workflow for radiology departments and decreases the time necessary to examine patients. Computer systems are widely used in health care and are for example used to view radiology images. To simplify this, display protocols based on examination data are used to automatically create a layout and hang images for the user. To cover a wide variety of examinations hundreds of protocols must be created, which is a time-consuming task and the system can still fail to hang series if strict requirements on the protocols are not met. To remove the need for this manual step we propose to use machine learning based on past manually corrected presentations. The classifiers are trained on the metadata in the examination and how the radiologist preferred to hang the series. The chosen approach was to create classifiers for different layout rules and then use these predictions in an algorithm for assigning series types to individual image slots according to categories based on metadata, similar to how display protocol works. The resulting presentations shows that the system is able to learn, but must increase its prediction accuracy if it is to be used commercially. Analyses of the different parts show that increased accuracy in early steps should improve overall success.

Nyckelord
Keywords machine learning, hanging protocol, image display, display protocol

Abstract

Radiologists' workload has been steadily increasing for decades. As digital technology matures it improves the workflow for radiology departments and decreases the time necessary to examine patients. Computer systems are widely used in health care and are for example used to view radiology images. To simplify this, display protocols based on examination data are used to automatically create a layout and hang images for the user. To cover a wide variety of examinations hundreds of protocols must be created, which is a time-consuming task and the system can still fail to hang series if strict requirements on the protocols are not met. To remove the need for this manual step we propose to use machine learning based on past manually corrected presentations. The classifiers are trained on the metadata in the examination and how the radiologist preferred to hang the series. The chosen approach was to create classifiers for different layout rules and then use these predictions in an algorithm for assigning series types to individual image slots according to categories based on metadata, similar to how display protocol works. The resulting presentations shows that the system is able to learn, but must increase its prediction accuracy if it is to be used commercially. Analyses of the different parts show that increased accuracy in early steps should improve overall success.

Sammanfattning

Röntgenläkares arbetsbördan har under flera årtioenden ökat. Den digitala sjukvårdsteknologin utvecklas ständigt vilket bidrar till ett förbättrat arbetsflöde och kortare undersökningstider i radiologiavdelningar. Datorsystem används idag överallt inom sjukvården och används bland annat för att visa bilder åt röntgenläkare. För att underlätta visningen används display protocol som automatiskt skapar layouts och hänger bilder åt användaren. För att täcka ett stort antal olika undersökningstyper krävs att användaren skapar hundratals protokoll vilket är en tidskrävande uppgift, och systemet kan ändå misslyckas med att hänga upp bilder om de strikta kraven protokollen ställer inte uppfylls. För att ta bort detta manuella steg föreslår vi att man använder maskininlärning baserat på tidigare sparade presentationer. Klassificerarna tränas på undersökningens metadata och radiologens preferenser på hängning av serier. Den valda metoden går ut på att skapa klassificerare för olika layout-regler och att sedan använda deras output i en algoritm som placerar ut series-typer till individuella bildplatser enligt kategorier baserade på metadata. Denna metod liknar den process de nuvarande display protokollen utför. De presentationer som skapats visar att systemet kan läras upp, men kräver högre precision om det ska användas kommersiellt. Analys av de olika delarna tyder på att ökad precision tidigt i systemet skulle öka den totala precision.

Acknowledgements

First I would like to thank Sectra Imtec for giving me this thesis opportunity, and especially the people of PD RADIT for their welcoming and enthusiastic attitude. I want to thank both my supervisor Kristian Köpsén and my examiner Cyrille Berger for their assistance in both theoretical and practical matters.

I would also like to thank Daniel Andersson and Ida Cervin for their inspiring company during my thesis. Last, I would like to thank Region Skåne for the dataset that I used during the thesis.

List of Abbreviations

CR	Computed Radiography
CT	Computed Tomography
MRI	Magnetic Resonance Imaging
DICOM	Digital Imaging and Communications in Medicine
HIS	Hospital Information System
PACS	Picture Archiving and Communication System
RIS	Radiology Information System

List of Figures

List of Tables

Contents

Chapter 1

Introduction

This chapter presents a short background to the thesis, motivates its purpose and provides an overview of the report.

1.1 Background

The use of digital workstations in hospitals has been increasing since the latter half of the last century. Hospital staff members are able to use digital versions of their old workplace to an increasing degree as the technology matures. One example is the digital version of the radiology film lightboxes; the back-lit tables that radiologists use to hang their film on. Instead of physically hanging the developed film on lightboxes they hang images virtually in predefined slots on monitors.

As the workload on radiologists continue to increase[2], developing tools which decrease the time spent on tasks unrelated to reviewing cases, such as manually hanging images, gives the radiologist more time to spend on examining the patients. A Picture Archiving and Communication System (PACS) is a distributed system which can handle the entire medical image workflow, storing and retrieving digital radiology images and presenting them to the radiologist reviewing cases. In order to help the radiology staff save time, Sectra's PACS has display protocols which automatically hang images relevant to the examination the radiologist is viewing. Images generated by radiography equipment are referred to as series.

Studies have shown that while reviewing examinations using soft-copy instead of film-based technology radiologists are able to examine more patients daily[3] and have a potentially higher chance of detecting cancer[4].

A display protocol is a hierarchical set of rules which decides how a set of series is placed, hanged, on a display device. The display protocol is a composite object made up of three different components, each containing zero or more of the class below it and is made up as follows: hanging, monitor and pane. Each component class is responsible for a different set of

rules, e.g. the monitor class decides how a display device will be partitioned. When a user opens a new examination the PACS will compare it with the existing display protocols and hang the series according to the best fitting protocol.

Many radiologists believe that easy-to-use display protocols are a priority when selecting PACS[5].

The current method of creating display protocols requires a technician to create a rigid set of logical rules for each step in the process. The technician will have to go through this process for each display protocol that is needed, which can be hundreds per hospital. This process is time-consuming and certain cases of examinations might still be overlooked due to the combinatorial explosion to catch all possible cases. The protocol might still fail to hang series and require manual corrections for individual cases when the series taken does not exactly match the protocol's expectations.

As the number of display protocol grows it also adds overhead costs, such as the maintenance burden of making sure hundreds of items are up to date and functional. It can also be difficult for someone without training to set up protocols.

The automatically selected display protocol for a given examination might not be able to match all the series taken in the examination and will then elect to not hang these at all, creating black panes where a user might still have been able to get information from another series. One goal of this thesis is to take advantage of the user's experience in correcting missing series, and to find rules which could be missed during the manual creation of display protocols.

In order to reduce the amount of resources spent on this manual setup process, this thesis presents a learning system which will automatically set up the display protocols. Further, it will attempt to match series taken in previous examinations with current ones even when there is not a perfect match, which might still be helpful for the radiologist reviewing the case.

Another aspect of the problem is that staff working with the display protocols does not follow a single standard and the system should be able to handle this issue by being able to adapt to different workplaces. An on-line learning system could be used to solve this, since an off-line system would not be able to adapt in the field.

1.2 Motivation

Application specialists who works in close co-operation with the company's customers were interviewed and asked what they perceived were the largest problems with the display protocols:

- To create a set of display protocols which cover most of the cases the technicians must add hundreds of different display protocol and series definitions.

- Maintaining the protocols takes time and effort.
- Radiologists can have different preferences on how to hang series and finding rules that suit everyone is very difficult for a human.
- The display protocols are very strict, if an examination has small differences from the protocols it is possible the system fails to hang the examination properly. The learning system should be able to adapt to these differences.

1.3 Goals

The main goal of this thesis is to create a learning system that is able to virtually hang series in a presentation which is similar to how radiologists would hang the same set of series. In order to do this a study to find information carrying features should be performed.

Two ways of defining the goals as questions would be:

- Can a learning system take an unseen examination's series and create a display protocol and hang the series according to the protocol?
- Is it possible to increase the effectiveness, defined as hanged series per image slot, of display protocol by using a learning system?

Once completed, the program should be able to take a new examination's series and hang them with minimal human guidance. A visualization tool should also be implemented in the solution. This tool shall be able to display the hangings created by the system and should confirm if sound results have been created.

1.4 Related work

There have been some uses of machine learning to hang series in a more user-friendly manner, but the area is not extensively studied. Boone *et al.*[6] uses neural networks to automatically label the orientation of the patient in chest examinations and hang the series according to already existing hanging protocol. Luo *et al.*[7] also uses neural networks directly on the chest series to rotate them correctly for the hanging. Both studies show that a learning system is able to find patterns in medical data and that radiologists save time when the hanging protocol can utilize these predictions. GE Healthcare[8] lets the user teach the system by manually hanging the series in a way the user finds useful and then remembering the hanging protocols for future uses. This thesis focuses more on the metadata in the examination and series rather than the raw image of a series, and also uses a large database containing past examination information to automatically teach the system.

Studies of radiologists' attitudes towards soft-copy and hanging protocols have shown that radiologists experience they need to spend less time on each image using digital workstations[9]. Moise *et al.* also proposed that 10-20 different hanging protocols would suffice for 80-90 % of examinations, but that the final percentages would be impractical to cover. A machine learning system might be able to study the available data to increase the coverage.

1.5 Contributions

This thesis aims to extend the capabilities of Sectra's PACS display protocols by allowing an automatic hanging based on previous use, using a novel approach based on machine learning and DICOM information. It also examines the usefulness of combining smaller learning systems in order to create more complex combinations of predictions.

1.6 Scope

Using image data from the series will not be considered in this thesis, so no computer vision related work will be performed. On-line learning was evaluated but not implemented in the system.

1.7 Outline

The two following chapters provides background to the thesis, chapter 2 presents the state of the art in digital radiology and chapter 3 offers an overview of machine learning techniques related to the thesis. Chapter 4 describes the method of the thesis, how machine learning was used and presents an algorithm for hanging series. Chapter 5 presents an evaluation of the implemented solution and the methods used for testing. Chapter 6 discusses the results of the thesis and the chosen approach. The thesis is concluded by chapter 7 with ideas for future work and a summary.

Chapter 2

Digital radiology

This chapter presents the state of the art in digital radiology.

2.1 Digital workflow

Digital radiology generates images via computed tomography (CT) by shooting x-ray or gamma ray radiation at the patient. Sensors pick up the different volumes of radiation and generate images, these machines are known as modalities. These series can then be stored on servers connected to the PACS and retrieved to the client workstations.

Radiology images can be generated from a variety of different methods. Magnetic Resonance Imaging using magnetic radiowaves, computed radiography (CR) and CT use x-rays and you can use ultra sound to get images. What they all have in common in the digital workflow is that the images generated are not saved on film but rather as computer files, some immediately from the modality and some are scanned from plates. The PACS stores these files on a server and retrieves them for the client workstations. During the examination the user can immediately see the results and act on this. The radiologist does not need to be in the same place as the modality, the series can be sent between hospitals via the Internet.

2.2 PACS

Nearly all hospitals in the Western world are using digital radiology instead of film-based analog technologies. This transition to digital media has led to using more IT in the hospitals, such as storing information on servers instead of journal storage rooms. With increasingly many new technologies, the digital workstation often aims to resemble the analog one, but with improvements made possible by IT. Examples of this are the use of

databases instead of journal storage rooms and computer monitors instead of lightboxes.

The PACS usually uses the format Digital Imaging and Communications in Medicine (DICOM, see next section) to store the series and examination information. A PACS will normally have an interface to a Hospital Information System (HIS), or a more specialized version for a single department. This means that information not directly related to the radiography procedure is also available to the system and its users.

The Sectra PACS has several different windows: the information, the image and the matrix window. In the information window the user can read about patient details, such as their medical history. The two other windows are more focused on image manipulation. The matrix window presents an overview of the layout and allows easy reorganization through drag-and-drop. The radiologist use the image window to review examinations. In this window the user can focus on the images, paning, zooming and in the case of multi-image stacks also zoom through the stack.

2.3 Display protocols

Sectra uses display protocols to represent a template for how the series should be hanged for a specific type of examination. A display protocol consists of three different components: hangings, monitors and panes. The hangings, not to be confused with hanged series, contain one or more monitors can be considered the visible layout displayed on all display devices. A monitor represents a display device and contains a set of panes and a partition setting which decides how the panes are partitioned on the screen, for example 2x2 panes on one monitor. The panes in turn are mapped one-to-one to the series types in the examinations, so a pane might be created specially to contain a transverse CT series. A pane can be thought of as a slot in which to put series and each can display one image at a time.

Display protocols are made by setting up logical rules and are often created by a technical administrator. Each of the three different components is set up separately. Each entity can also have several settings, such as zoom levels or to use prior examinations as part of the presentation.

The content of each pane is also a type of rule. The user decides on a type of image that should be shown at each position, and also implicitly the relation between the images' positions.

Figure 2.1 presents an example of a display protocol for a CT thorax examination. On the left side are the different hangings used in the presentation. The right side shows the selected hanging in greater detail. This part shows how the monitor partitioning rules have split the screen in half along the horizontal plane. The figure also shows how the panes are assigned different series types.

Figure 2.2 shows an example of how the protocol can look when a radiologist is interested in comparing two different examinations. The series types

Figure 2.1: Example of a display protocol for CT images of the thorax.

Figure 2.2: Monitor set up for comparison of two different examinations.

match each other along the vertical plane, but with the additional rule that images on the more lightly tinted bottom half must have been taken in a prior examination.

When users are not satisfied with the result of the automatic setup delivered by the display protocol they can re-arrange the position of each series and do changes to the layout, such as changing the partition. When they are content with how the series are hanged they can choose to save the presentation, so next time the same examination is opened the presentation is remembered.

One of the more common issues with the display protocols is that they require a specific series type for each pane, and if there are none which fulfill the requirements the pane is left blank. Another reason for this to occur is if there is a mismatch in the number of series taken and how many panes the protocol has reserved for each specific series type.

2.3.1 Quick protocols

For a quicker set up in Sectra's PACS, a user is able to create Quick Protocols, which do not need the same complex configuration as the advanced protocols. Instead, the user adds the condition for the quick protocol: modality, body part and description, whether to use previous examinations or not, how to place comparison images from prior examinations and finally a partitioning. Unlike the advanced display protocols, the quick protocol does not take series types into consideration and will continue to hang series in the order they were generated until there are no more available. This is sometimes enough, but it can be harder to cross-reference two series when they are not hanging together.

2.4 The DICOM format

DICOM, Digital Imaging and Communications in Medicine, is the standard format used in medical imaging technology and includes a file format for images. A DICOM image object consists of a large set of headers that contain information about the series, for example the modality used to generate the series or the examination code. In addition to the metadata the object stores image data. For many modalities this means only a single image, but for some modalities the image data can contain multiple images which are stored in a large stack of images that the user can scroll through. Most of the fields can be filled in automatically during the creation of the object with information from the modality and from the Radiology Information System (RIS), however some fields can still need some manual correction after an examination has been performed. The DICOM standard is used in much of the workflow related to an examination.

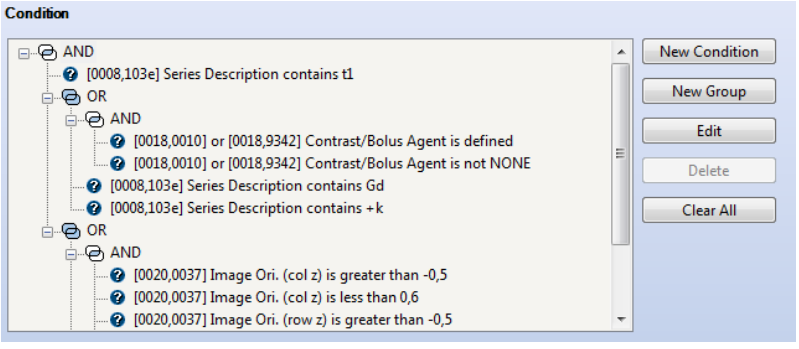

Figure 2.3: An example of a series type definition.

2.4.1 Series and series types

An image taken by a modality is called a series. A series can also contain a series of images, hence the name. A series with multiple images is known as a stack and is generally created by CT or MRI modalities so a radiologist can see slices of the organs under examination. The stacks can also be used to generate 3D images of the patient. In these types of examinations it is common to also generate a single image in another orientation, which is used in conjunction with the stack to help the radiologist navigate in the stack depth; these types of series are called scouts or scanograms. During examinations with CR it is more common to only take a single image per series, for example to locate a bone trauma.

A series contains DICOM headers and image data. Each series is unique, but can be part of several series types. Series types are rule-based category types based on the content of the DICOM headers. A sagittal series of the spine could be part of both the *sagittal spine* series type and of the *sagittal* type. Series are strongest connected to the type with the most matching rules, in this case that would be the *sagittal spine* type. When hanging images the series types are used to differentiate what the panes can contain in the hanging. In real applications there can be hundreds of different series types, ranging from very general with only one condition to more specific ones with several conditions and controls for different spelling. In this thesis, five general series types have been identified which can be used to categorize many series:

- Transverse: Images taken along the transverse or axial plane, which splits the body into an upper and lower part.
- Coronal: Images taken along the coronal plane, which splits the body in a front (ventral) and back (dorsal) part.
- Sagittal: Images taken along the sagittal plane, which divides the body into a left and right half.
- Contrast: Images taken along any plane using contrast medium. It is often used to enhance the visibility of internal body structures.
- Prior: Image taken along any plane in a previous examination. These series can used to observe how a patient has developed since the last examination or to use as a reference in case of injuries.

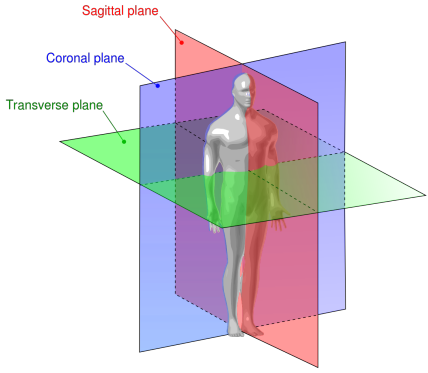

Figure 2.4: Planes of the human anatomy[1].

The image orientation can either be written down in a free-text description or require a calculation based on 3D coordinates from the headers. In this thesis a series is considered a prior if its examination ID differs from the current examination's, and has been generated earlier or within a short time - since a radiologist might not look at the examination immediately and there might be multiple radiology sessions in quick succession.

By using these general series types it could be possible to find relations between series in general cases. The series type matching works by first attempting to match as many rules as possible, and then try with decreasingly many restrictions. This means that many of the specialized ones can be subsumed into more general ones.

This also mean that the learning systems does not need to learn as many relations, which decreases how many comparisons need to be understood by the learners and increases at what speed they learn. Since the more specific ones subsumes into general ones it is also quite likely that the wider cases can also find relations between different series.

Chapter 3

Machine learning

This chapter presents the machine learning aspects of the work, and provides some background theory for the systems that were used.

3.1 General

Machine learning is a technique that allows a system to learn to predict outputs given an input. Machine learning is commonly used to find patterns in data and is very useful for large and complex data sets where a human would be bored or overwhelmed. One of the most important traits of a machine learning system is its ability to generalize from examples, that it can make qualified predictions based on input data it has never seen before. There are three different paradigms of training machine learning systems: supervised, unsupervised and reinforcement learning.

3.2 Ensemble learning and boosting

Ensemble learning is a supervised method of grouping together many weak learners, which could be considered a "rule of thumb", into an ensemble. The weak learners do not necessarily have to achieve high accuracy in their predictions, hence the name. For most ensemble methods it is sufficient that they perform slightly better than random guessing. The weak learners in the ensemble will each attempt to classify the output, but will have learned different things well. When they each have made a prediction, they vote for which class the ensemble should predict by combining the weight of each weak learner and its output and the ensemble outputs the class which won by majority. An ensemble of weak classifiers can often achieve a very high accuracy and is comparable to most state-of-the-art machine learning systems. One of the most commonly used techniques is AdaBoost (Adaptive Boosting) proposed by Freund and Shapire[10].

Boosting is seemingly immune to overfitting[11]. This is an important trait when the data can have noise and outliers.

One of the most commonly used types of weak learner is decision trees. Learning systems using decision trees are often good at handling categorical data. Categorical data often lack natural distance metrics. This can be solved by creating a specialized metric, but this can be a large and time consuming task and is not considered in this thesis. Decision trees used for classification are called classification trees.

One good aspect of decision trees is that they are easy to interpret by the user, which is important in medicine technology. Transparency increases the confidence in the system if it is possible to see how the decisions are reached.

AdaBoost trains its weak classifiers by generating a set of decision trees. For each iteration, AdaBoost chooses the weak classifier which performed best, and increases the value of misclassified data points. This way, the next time the classifiers are trained they will be better at finding the misclassified data points. This results in the ensemble consisting of classifiers which are good at finding different aspects of the classes.

Initialise $d = |N|$, where N is the number of data points
while $0 \leq error_t \leq 1/2$ **do**
train classifier on $\{S, w^{(t)}\}$, and get hypothesis $h_t(x_n)$ for datapoints x_n

An example of how the DT classifier is trained.
    for all potential features **do**
        for all potential thresholds **do**
            Find and select feature and threshold with highest accuracy.
        end for
    end for
compute training error $e_t = \sum_{n=1}^{N} w_n^{(t)} * I(y_n \neq h_t(x_n))$
set $\alpha_t = \log(\frac{1-e_t}{e_t})$
update weights: $w_n^{t+1)} * \exp(\alpha_t * I(y_n \neq h_t(x_n))/Z_t)$
end while

Output $f(x) = sign(\sum_{t=1}^{T} \alpha_t h_t(x))$
Example of how to train AdaBoost, taken from [12].

One way to handle missing input data is by using a technique called stubbing. When a value is missing, this technique searches for another split in the input data which closest matches the missing value. By using stubbing it is possible to achieve slightly higher accuracy in the presence of missing data.

3.2.1 Multi-class prediction

Boosting is generally able to only predict binary classes: either an object belongs to class x or it belongs to class y. It is however possible to extend this functionality to handle multi-class cases via several different methods. The most direct version is to test if an object belongs to class x or not. If the data point belongs to class x the algorithm finishes, but if it does not the classifier passes the object to the next classifier which in turn determines if it belongs to class y or not. This process is repeated until the object is found to belong to a class.

AdaBoost.M2

A more advanced algorithm to perform multi-class predictions is AdaBoost.M2, presented by Freund and Shapire[13]. This version allows the weak learners to assign plausibility to each class, and can return multiple plausible labels instead of a single one. Instead of minimizing the error rate, AdaBoost.M2 minimizes a pseudo-loss:

$$\epsilon_t = \frac{1}{2} \sum_{(i,y) \in B} D_t(i,y)(1 - h_t(x_i, y_i) + h_t(x_i, y_y))$$

Where D_t is a mislabel distribution of the training set where mislabeled data points are weighted more, y_i the correct label for data point x_i. It is similarly sufficient if a weak learner achieves a pseudo-loss slightly lower than randomly guessing.

3.2.2 On-line learning

Normally most ensemble algorithms perform batch learning: they read and process the entire training set until the base model is good enough, and then stop. This makes it much harder to incrementally teach the system new behaviors while the user is operating it, since it would have to periodically perform a complete batch training which is very time-consuming.

Oza presents in his PhD thesis[11] ways to transform ensemble batch learning algorithm to use on-line learning instead. In his thesis he shows that on-line boosting's weak learners will converge with those of standard batch learning, thus achieving the same result. Of note is that during off-line learning all samples are used to update and select one weak classifier, but during on-line learning one sample is used to update all of the fixed number of weak classifiers.

Chapter 4

Method

The system collects data from a SQL server and processes it for training the different classifiers. To create layouts for the monitors several rules have been identified and are predicted by the system. The hanging system then begins to hang the series according to the completed layout. The first section describes the available dataset and is followed by the general architecture of the system. The system is then presented in greater detail, covering the selected inputs, the identified rules, how the layout is created and how the series are hanged.

4.1 Overview of method

The machine learning part of the project was created in MATLAB using off-the-shelf learners and functions. This had several benefits, the largest being that MATLAB provides an easy framework to quickly prototype ideas and measure effectiveness. Many of the existing machine learning methods today are very good at accurate predictions, and usually the best way to increase predictive powers is to work more on the features.

The learning solution consists of two major parts: the rules component and the hanging component. The rules use machine learning to estimate what the most likely display protocol-like rules should exist and feeds the result into the hanging component which use the rules' result as guidance while hanging the series onto the monitor. The rule component consists of several smaller systems which each focus on predicting a single value.

In the beginning the learning system was tested on general cases with mixed examinations, but this was changed to investigate if it could reach a high accuracy in more narrow cases and then to be extended into more general examination cases.

Since one of the aims of the thesis was to evaluate the possibility of training the system based on past user preferences, and the access to a database containing this data, the decision to use supervised learning was

made.

4.2 Dataset: Region Skåne

Training data was collected from an anonymized database containing the examination information of multiple hospitals in Scania which Sectra had collected for several years. This database contained a very large amount of information, with a subset relevant to this thesis. The data needed to be cleaned in order to be more useful. Cleaning means detecting and correcting erroneous records and harmonizing the entries. The database had already been scrubbed from any information that could be used to identify patients before its use in this thesis. The features in the database consist in many cases of categorical data.

To recreate DICOM objects and match them with the examinations in which they were acquired several tables had to be joined together. Only those DICOM headers that Sectra deemed important in order to create the existing display protocols were present in the database. Because the information is not stored as DICOM objects but as smaller classes in different tables it was not possible to use any existing software without also performing join operations to parse the DICOM objects. This prompted the creation of a specialized parser for the Region Skåne database.

Table 4.1 presents some examples of the data found in the database. Some of the fields contains values which need translating, such as the RowX and RowZ (and RowY) which together with their column counterparts which describe which orientation the series has. Another thing to note is that values in BodyPart contain the same body part in different languages, which also has to be processed so the system does not treat them differently. Note the difference between the valid values in table 4.1 compared to table 4.2 which presents the input values to the classifiers.

BodyPart	HJARNA, NECK, BRAIN, CHEST, ABDOMEN, ..., null
RowX	-1 ... 1
RowZ	-1 ... 1
Contrast	DOTAREM, APPLIED, 12ml Dotarem, ..., NULL
Modality	CT, MR, CR, ...

Table 4.1: Some examples of the DICOM headers and their possible values.

In the database the display protocols are stored in a format similar to most markup languages, which is how Sectra used to store protocols when the data was collected. The presentation is stored in a hierarchical manner very similar to how the display protocol works, with hangings containing monitors which in turn contains the panes. Each of the classes also contain class specific options, such as in which order to sort stacks of images for each pane class.

One issue that occurred was that the series and examinations are in a one-to-one relation, however since series can be reused this means that only getting information about series generated for one specific examination is not enough. This was solved by parsing through the display protocol description and patching in the missing information by cross-referencing with all existing series with the same examination code.

4.2.1 Free-text parsing

Most DICOM fields have a set of permitted values (enforced by the Sectra PACS interface). In order for the radiologist to add more information about a series or examination there are several free-text fields where it is possible to write information which would not fit well in the other fields. Such information can be what the purpose of the series was or status of the patient. Using this information it is possible to recreate some of the more specific series types, such as *coronal lung*, instead of the more general *coronal*.

A regular expression parsing tool was developed to parse these fields easier. The keywords to search for were determined by examining the existing display protocols to understand what radiologists normally used to distinguish different series.

The use of the free-text fields also made it possible to reconstruct "damaged" fields, as the free-text could contain the body part or orientation, which were often left empty if the free-text field has been used instead. This also served to reduce the amount of noise in the inputs, different doctors or modalities might call a brain series "brain", "hjärna", abbreviate it or misspell it. By using regular expressions many of the differently names for the same thing could be rewritten to the same value.

The free-text fields were not suitable for use as inputs since they are not structured and they often contain the same information which can be found in other headers.

4.3 System architecture

The solution uses several different ensemble systems to learn the rules of the display protocol. The underlying machine learning algorithms used for predicting the rules were chosen by studying the accuracy of each system for the rule, and selecting the highest scoring one. The criteria for highest score was based both on accuracy but also on the confusion matrix of the test, to make sure the algorithm did not simply select the largest class if the class balance was skewed, for the results please see table 5.2. By combining the output from the learning systems the hanging system gets enough rules to be able to create a basic protocol and hang the series accordingly.

Some of the learners would also receive parts of their input from the output of other learners, which made it important to achieve high accuracy or the final output would have errors from multiple sources as the error

propagated. Figure 4.1 shows how the system processes the input and the distinction between the layout rules prediction part and the hanging part. The boxes in rules are all classifiers, the multiple boxes signify that there can be multiple classifiers of that type. Each of these classifier is specially trained for a specific monitor in the presentation, so the presentation's third monitor to be created by the system would be predicted by the third classifier of each type. The uppermost box in hanging, neighbor prediction, is also a classifier but is trained to work on any monitor. The last two boxes are hanging algorithms that use the predictions to create presentations.

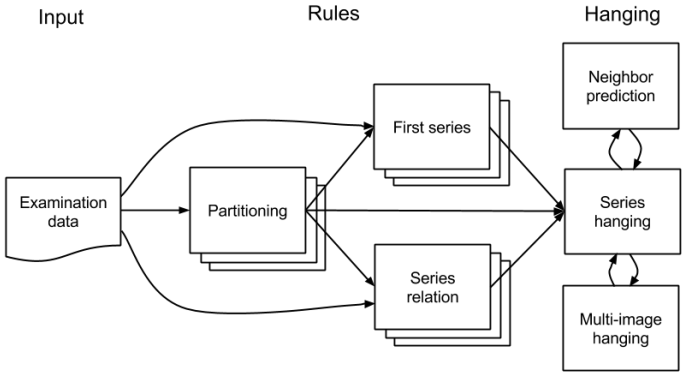

Figure 4.1: Overview of the learning system.

To generate the inputs and outputs for the learning system a parser according to 4.2 was built.

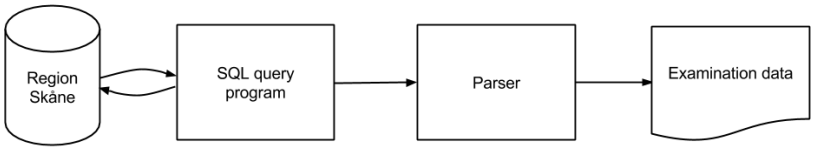

Figure 4.2: Overview of the parser.

4.3.1 Training the system

The system is trained from input generated by the parser, which harmonizes the inputs into fewer categories and attempts to remove errors. This data consists of both information about series and the user-created presentations.

The underlying learning algorithms differs between the learners for each rule, and were picked depending on how well the learner was able to fit the data. Normally the highest scoring algorithm was selected, but for the series' relation rule what was deemed important was to be able to accurately predict when there was a relation. It was an acceptable solution to sacrifice some of the general prediction power in order to get higher accuracy on existing relations. Because of the two-staged setup the false positives, assuming a non-existing relation, is not as damaging as false negatives. The series type assignment algorithm needs information to process and should be able to filter false positives to a degree by examining other relations.

4.4 Feature selection

In the context of machine learning, a feature is some kind of measurement that describes an aspect of an object. The data from the databases used for training the learning system was often dirty, with columns that could have been left blank by the user, or that different users or hospitals had different standards regarding how to hang the series. This was mitigated by reconstructing the missing or wrong DICOM headers from other headers, most commonly using the free-text description of the series or the examination. Further, some outliers and unlikely cases that could affect the precision of the solution were removed, since the weak learners in the ensemble might try too hard to correctly classify these samples while training.

Many of the inputs were selected by using domain knowledge and intuition to save time. Many DICOM headers in the database were seldom used or not at all. Adding a new header to be used as an input was a somewhat time-consuming task and if the header had not been used by hospital staff it would not improve the solution at all.

Some of the outputs were not explicitly stated in the database, these had to be created from parsing the descriptions of the presentations. Most notable was the position comparison between series types where the parser had to recreate the monitor in order to compare the relations. Because of this, some presentations could not be used due to not fulfilling enough requirements for the parser to make sense of.

Unlike neural networks and support vector machines, ensemble learning will not attempt to use features which will not improve the predictive power of the model. This means that training with poor features will not diminish the result, which in turn means that each model does not need to have specialized input features, but rather that each general type of classifier can have the same feature vector. This saves development time when attempting

to add new features. This does increase the training time for the ensemble since there are more features to evaluate.

Since the solution uses several different classifiers for the rules, not all input are important to every classifier, but can be very important to a subset of the classifiers. MATLAB offer tools for weighting the importance of features.

4.4.1 Selected inputs

This is an overview of the inputs which were used and how they affected the solution. Not all inputs are used for each classifier, some does not add any accuracy while increasing the training time and complexity and are thus removed. The letter after the name of the feature is which set it is part of.

- **Number of series present A** A larger number of series might indicate the need to split the partitioning more to accommodate more series.
- **Body part A** Which part of the body that was examined.
- **Contrast A** If contrast was used in the examination, also a flag for individual series.
- **Modality A** The type of modality used in the examination.
- **Exam code A** This header is an aggregate of the body part under observation, the examination method and an administrative code, so it had to be split into parts. The administrative code has not been stored in the database.
- **Station name A** Since different hospitals and even staff members might have different work procedures, one way to differentiate between them was to use the name of the workstation connected to the modality. The same staff members are more likely to work in a similar manner when using the same modality.
- **Used prior As** If the radiologist has used prior examinations when creating the presentation.
- **Number of displays A** How many display devices was used by the radiologist when creating the presentation.
- **Number of each series type present B** The number of series with a subtype: liver, soft tissue, artery, mediastinum and lungs were the selected subtypes.
- **Role and user B** The user or role ID the radiologist saved the presentation as. Since only one field can be used at a time the two are merged.

- **Institution B** At what institution or hospital the examination was performed.
- **Partition of monitor A** How many rows and columns the monitor was split into.
- **Scanogram present B** Checks if the examination includes a scanogram or scout, which are generally used as a way to navigate through stacks of images.
- **Thin slices present B** Checks if any of the stacks have used thin slices as representation.
- **Prior exam code B** Imposes some limitations on the data because the prior data is not available in the current examination information, so the examinations must be joined in the database. Has the same format as the current exam code. Not at all examinations have a prior exam code.

Input variable	Valid values
Number of series	1 ... n
BodyPart	head, arm, thorax, ...
Contrast used	0, 1
Modality	CT, CR, MRI, DT, ...
Exam code	100-1099, followed by 00-99
Station name	Sum of ASCII characters in name
Used priors	0, 1
Number of displays	1, 2, 3
Each number of subtypes	1 ... n
Role and user	The role and user IDs concatenated
Institution	Sum of ASCII values in name
Partition	1x1, 1x2, ... (See 4.5.1)
Scanogram present	0, 1
Thin slices present	0, 1
Prior exam code	0, or same as exam code.

Table 4.2: Ranges of valid values for the input, all inputs except number of series are categorical.

One of the advantages of CT and MRI scans is that the modality is rigid which makes it easier for the user to assign exact values to the image orientation. Since the procedure is quite standardized the data is less likely to contain noisy elements, or hard to classify orientations such as a CR of a twisting patient.

To further attempt to capture the different user behavior both at different sites as well as between persons both the user and role ID of the person who created the presentation were also included and so was the institution name.

One issue with the first two is that users tend to use the system level user when they create their presentations. Another issue is that users either use their personal user ID or their role ID, meaning whenever one is used the other field is left empty, creating a large amount of empty data. These two headers were encoded into a singular header to decrease the empty data.

4.5 Layout rules

Existing display protocols in Sectra's PACS are made up of several explicit and implicit rules to create the layout, as well as series specific rules such as how to sort image stacks or zoom levels. The goal of this thesis is to hang series, so the focus was to predict rules similar to those that the manually created protocols would have used. These rules are generally associated with one learning system which specializes in predicting the class of the rule.

It is from the stored presentations in the dataset that the ground truth is generated for both training and validation. These stored presentations contain all data required to create a perfect reconstruction of the presentation the radiologist used on the examination, such as series type placements, number of monitors and which series were mapped to each pane. The monitors contain the partitioning and the panes contain which type is used in the upper left corner. The series type relations can be found by comparing each pane and its neighbors. The inputs can similarly be created from the metadata of the examination and its series.

Since the correct output is generally string variables they are encoded into integers as enumerated types.

4.5.1 Partitioning

This rule is used for deciding how many panes are supposed to be used on one monitor, and how many rows and columns the matrix window should be split into, e.g. 4 panes split into 2x2 or 3 panes into 3x1. To predict this rule an AdaBoost.M2[13] learning system was used. Unlike the other layout rules this rule is predicted only once per monitor and is already done by the time the series hanging algorithm is executed. These classifiers use input sets **A** and **B** and can produce the following outputs:

1,1	1,2
2,1	2,2
2,3	3,2
3,1	

Where the first number is number of rows and the second is number of columns. By counting the numbers of occurrences these partitions were the by far most used ones.

4.5.2 Series' relation

This rule is used to determine the relative positioning between two series types. The series types can be either between two different types or between the same series type, since it is not unlikely to include two of the same type in one monitor. The relation is based on neighboring panes along the horizontal or vertical axes. Compared to other learning systems used in this thesis this system needs a larger volume in input since it needs to learn the relation between many objects. To decrease the amounts of comparisons necessary the series types were only compared in one direction, e.g. sagittal series were compared to coronal ones, but the coronal to sagittal case was not considered.

When creating the output labels for this rule the parser recreates the monitor and counts the most often occurring relations. This means that relations between panes on different monitors could not be measured.

These classifiers uses input from only set **A**. The output can be described as follows:

- Type 1 placed on left side of type 2.
- Type 1 placed on right side of type 2.
- Type 1 placed above type 2.
- Type 1 placed below type 2.
- Type 1 and type 2 have no relation.

A series type can have several neighboring types in each direction. This was to handle the issue that appears in the case in a 2-by-2 grid where one half of the grid is the same type and the other side is mixed. Without allowing several values in each direction the side in majority would only be able to store one of the neighbors.

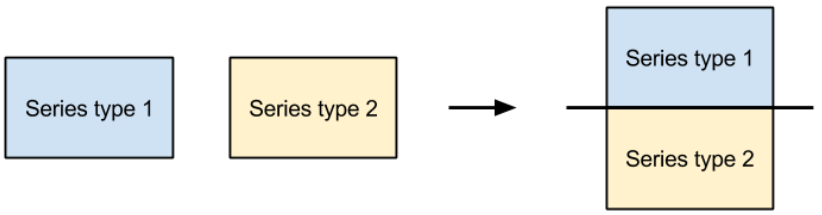

Figure 4.3: Example of how the system determines the most likely placement for two series relative to each other.

4.5.3 First series

This rule is used to help find the most likely series placed in 1,1. This is the only slot which is guaranteed to exist since a monitor can have only one row or column, or both. The first series usually have less information available than the subsequent series so this is a way to increase the guidance for the hanging algorithm. The outputs from this rule are the three orientations and use input features from set **A** and **B**.

4.6 Neighbor prediction based on existing structure

In order to take advantage of the idea of building the hanging by placing one series at a time a sort of semi-structured prediction component was implemented. This classifier predicts the most likely type of series orientation based on its neighbors. Since different numbers of neighbors require different amount of input parameters this learning component is split into several learning systems, each specialized in the amount of neighbors that are available at the time. The two most important features that it considers are from which direction the neighbor is connected from, and what kind of orientation they have. Additionally, the pane in the monitor that is being currently assigned to is used as an input.

Input	
Neighbor type	tra, cor, sag, other
Target row	1, 2, 3
Target column	1, 2, 3
Direction	West, north, east, south
Partition	See set **A**
Exam code	See set **A**
Station name	See set **A**
User and role	See set **B**
Output	
Most likely neighbor	tra, cor, sag

Table 4.3: Input and output from the structure based classifier.

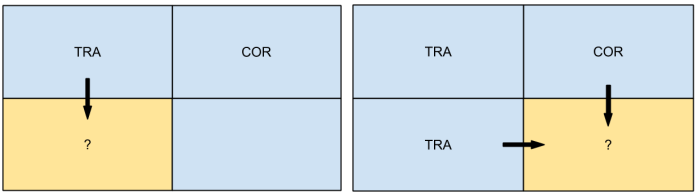

Figure 4.4: Predictions based on one and two available neighbors.

4.7 Creating the display protocol

Once the rules have been predicted the program is ready to create a display protocol to use as a template for series hanging. A greedy algorithm was developed to create the display protocol and is presented in algorithm 1. In each iteration of the algorithm it finds the highest scoring series type for all panes in the monitor. The slot with the highest score is then assigned to that series type and removed from future iterations. The series which is being placed first does not have as much information available as the following ones. They can also check if their neighboring types corresponds to the neighboring panes' assigned types. In the algorithm, SR is the series' relation matrix, SR_{td} is the array corresponding to type t and d is the cell which stores the neighbor's most likely orientation in each direction.

Series types have neighbors based on the predictions performed by the series' relation system. When testing the match potential of a possible series type to a slot the type's neighbors are compared to the neighbors of the slot to see how well the two matches. Points are awarded based on the importance of the match. The possible neighbors are empty, matching neighbor - which consists of *any* (wild card) or a specific type - and possible match. If a series type has no neighbor in one direction it is a good fit to be placed next to the edge of the monitor with the empty neighbor being the monitor frame. If a series type has a specific neighbor type and is placed next to a pane which satisfied the relation they are both likely placed together. The prior and contrast type also allow for wild cards where they can be placed next to any orientation series type and are assigned the correct orientation during the actual hanging. Finally, if the placed series type has specific neighbor types but the neighboring panes are not yet assigned a type, the assignment is considered non-blocking since a good fit is likely to be found in the upcoming iterations.

The amounts of point for each satisfied condition were based on the importance of the condition, with matching neighbors being the most important. The score was also based on making sure that one condition could be "beat" by several smaller ones.

Once the algorithm has allocated series types to every pane in the presentation it begins to place the series. Each series is placed in a bucket for each unique series type and are removed one by one as they are assigned to

Neighbor	Tra	Cor
West	None	Tra
North	None	None
East	Cor	None
South	Any	Any

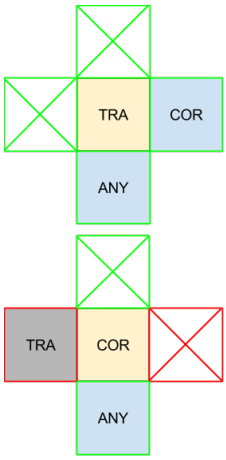

Table 4.4: Example of how two different series types are evaluated for the upper-left position of the monitor. The *TRA* type's four relations are satisfied while only two of *COR*'s are.

different slots. An issue with this method is that the series are sometimes used several times. However, in order to reuse series there needs to be some metric to decide each available series potential for each assignment. The algorithm creates one monitor at a time and continues to do so until there are no more series to place.

4.7.1 Merging different hanging components

Similar to how there are several components for finding rules there are multiple systems for finding the correct hanging sequences. Using the greedy algorithm mentioned above, it is possible to generate input for the other components, and using these in turn to influence the allocation of series types to the panes. One example of this is the structured learning system cooperating with the hanging algorithm. When the hanging algorithm examines each slot it notes how many neighboring slots have already been filled, if there are at least one it can call on the structured prediction. Since the neighbor structure component is split into several learning systems based on number of available neighbors it needs to count all possible matches and then call the correct one. The resulting prediction will increase the score for one of the series types.

Multi-image hanging

Another way to influence the hanging is the choice of using series with single or multiple images. This component uses the same greedy algorithm as algorithm 1, but instead of finding the best orientations it only use *single*

Algorithm 1 Greedy series hanging algorithm

```
for all panes in monitor do
    for all potential panes do
        if p == first pane then
            Assign score to orientation t predicted by first series' classifier.
        end if
        Assign score to orientation t predicted by structured classifier.
        for all t in potential series types do
            for all d in directions do
                if SR_td is empty and pane_d is outside monitor then
                    Assign some points to orientation t.
                end if
                if SR_td is non-empty and pane_d is same type then
                    Assign more points to orientation t.
                end if
                if SR_td is non-empty and pane_d is empty then
                    Assign some points to orientation t.
                end if
            end for
        end for
        Find max score of type and pane combination.
        Assign pane type t and remove pane from potential set.
    end for
end for
```

or *stack*. Similar series relation classifiers are also trained to find normal relations between these two types and are used instead of the orientations.

Sectra's PACS has a feature for displaying a picture-in-picture of a scanogram in panes displaying stacks which could reduce the need to use a pane as an overview for its neighbors.

Chapter 5

Evaluation

This chapter presents the evaluation of the system and how it was tested.

5.1 Evaluation metrics

Some of the issues with the existing display protocols were that panes were left blank or that series were not hanged when the protocol was finished. Thus, two of the metrics used is how often either of these cases occurs. However, in some cases having empty panes are preferred. An example of this is a presentation with two monitors partitioned into a 2x2 setup with two different examinations. It could be easier for the radiologist to have each prior series placed in the corresponding slot on the second monitor instead of mashing together the two examinations to maximize screen usage.

The results of the hanging were compared to how well the system did compared to the ground truth - how the radiologist hanged that specific examination. It should also be noted that sometimes the actual hanging is not always worthwhile to pursue, since they can be overly specific or not following standards.

5.2 Testing

For the majority of the project, only a subset of the available data was used for speed purposes. The results presented in the following sections are from looking specifically at CT examinations of the thorax region. This data set includes 714 examinations with manually saved presentations along with 7216 series.

5.2.1 K-fold cross-validation

Cross-validation is a common technique used to validate the model created by machine learning algorithms. Because machine learning often reports

an overly optimistic result on training data it is important to evaluate it on unseen data to get a more accurate and realistic result. K-fold cross-validation splits the data into K evenly large sets and trains on $K-1$ sets and evaluates on the remaining part. This process is the repeated until each subsample has been evaluated. Cross-fold validation is a useful technique to deploy when further samples are difficult to acquire.

5.3 Error analysis

This section presents the different errors in prediction by each component. Worth repeating is that the earlier errors occur they more they affect the final result. **Unless specified the input is correct, i.e. has not been predicted by another component but is supplied directly from the database.** To give a clearer picture of the accuracy of the predictions the scores have been split into separate scores for each monitor. The accuracy was measured using 5-fold cross-validation. To verify the result each rule was also tested by using the oobError function of the TreeBagger data structure in MATLAB. TreeBagger creates an ensemble of decision trees by bagging the results, which means it is a different method than AdaBoost which were used to create the classifiers used in the actual solution.

Number of monitors	1	2	3	4	5	6	7
Partition	56	74	71	84	76	87	78
Series relation	89	83	91	85	94	86	
First series	86	72	84	72	81	75	81
Neighbor matching (one neighbor)	83						
Neighbor matching (two neighbors)	81						

Table 5.1: Error analysis results for the individual classifiers based on which monitor they are predicting. The same neighbor matching classifier is used on all monitors. Results are in correct % accuracy.

5.4 Learning system accuracy

To determine which kind of learning system should be used as the underlying machine learning algorithm for the different rules they were each evaluated on the same data and are presented in table 5.2. The series relation requires two different metrics, one for all possible relations and one without counting the *no relation* outcome, since this is often much larger than the other classes put together. If this is the case, the classifier could simply always predict this larger class and still receive a high accuracy. The results are averages

after evaluating and using 3-fold cross-validation. The test results from bagging on partition and neighbor matching is likely over-optimistic due to using another test method, as they did not show the same good results when used in the full system.

While AdaBoost.M2 is described in some detail in chapter 3, the other classifiers are not. RUSBoost uses an undersampling technique in order to help mitigate the problem of skewed classes [14]. TotalBoost [15] uses optimization algorithms to maximize the minimal margin in the training set and can produce ensembles where it is easy to remove low-weight weak learners and save memory. Bagging uses bootstrap aggregation instead of boosting, where the weights of each sample is 1 or 0, as opposed to boosting where the weights change to fit earlier learners.

Classifier	Partition	Series relation	Neighbor matching
AdaBoost.M2	70	70 & 21	74 & 77
RUSBoost	66	65 & 25	73 & 76
TotalBoost	68	70 & 17	70 & 75
Bagging	73	55 & 44	83 & 81

Table 5.2: Average % correct for different learning system. Series relation is split into all relations and without *no relation*. Neighbor is split into one and two neighbors.

5.5 Hanging accuracy

This section presents the result of the entire hanging algorithm. Due to the way it keeps adding series to each monitor until they are full, the only time it does not fill a pane is if it has run out of potential series. The test was performed on a different max number of monitors in order to more clearly show the behavior of the system as the output increases in volume and complexity.

Number of monitors	2	4	6
Correctly placed series (% correct)	33	31	30
Wrong partition (% of cases)	44	42	42
Series remaining (% unhanged)	66	39	21

Table 5.3: Along x-axis is the number of monitors used.

The worst performing classifier is the first partition learner, which is

problematic as it is used as an input to so many other learners. As a test, we substituted the first learner's output with the correct value, the results can be seen in table 5.4.

Number of monitors	2	4	6
Correctly placed series	58	49	45
Wrong partition	17	19	23

Table 5.4: Accuracy of the hanging algorithm when supplying the correct partition setting for the first monitor.

Chapter 6

Discussion

This chapter discuss the results from previous chapters and the design choices made in the method chapter.

6.1 Design choices

The solution was inspired by the existing solution. In the old version the hanging process actually works well, the problem were the inflexible display protocols. Thus our goal was to create flexible display protocols. One "early" goal would also be to replace the quick protocols rather than the advances display protocols. This also provided us with a pretty clear-cut structure to follow, instead of creating something completely new. A much more complicated solution could be required otherwise.

Since the features in the data are mainly categorical it rules out several simple alternative solutions using k nearest neighbors since the distance is hard to define. One such solution would be to create a set of good presentations and use k-NN to classify incoming examinations. This would be a relatively similar solution to what exists today, and could have the same problem with more exotic examinations.

6.1.1 Series relations

An alternative solution to the series relation labeling would have been to create a canvas of all panes in the entire presentation instead of separate for each monitor. This solution would require that there always is a relation between every pane in the presentation, there is likely a relation between series in the same hanging, but comparing series across hangings is not as likely to be useful. The implementation of this system would also have taken more time and possibly introduced bugs or unforeseen consequences.

Table 5.2 shows that many systems have a bit of problem to properly predict when there are existing relations, possibly because of large variance

and that *no relation* is very common. The same table however shows that the neighbor prediction is able to predict correct quite often in comparison and thus is worth keeping even if it has a similar function to series relation classifiers. Its drawback is of course that it needs an existing structure in the monitor to be able to predict, which the series relation classifier does not.

6.2 Evaluating the results

It is possible that there are several different hangings that the user would find degrees of usefulness from; however the method used for evaluation only considers the case which was saved to be the only acceptable answer. This could mean that the hanging accuracy based on the saved ground truths is not the best evaluation criterion. Another way to evaluate could be to use a usefulness test with three levels: requires no fixes, requires few fixes, and not useful. To perform this test an expert would have to evaluate the proposed presentations.

Mammography examinations were not present in the database, which could have been interesting to test on since the presentations are often very similar as the procedure is heavily standardized. This could have been used to study if the learning system quickly could learn a standard protocol and use it for hanging series.

It would have been useful to use a larger sample to evaluate instead of only looking at CT thorax examinations, but the computer would regularly run out of memory when evaluating the system in parallel. The same issue happened when getting data from the database, the computer would run extremely slow and require manual supervision between the different phases of data retrieval. Another issue with the data from the database was that this time consuming process would have to be repeated if more DICOM input values were considered necessary.

6.2.1 Error analysis

The earliest and one of the most important classifiers, the partitioning for the first monitor, is also the classifier with the worst accuracy in the entire system. From figure 5.1 it is possible to see how the subsequent classifiers all have significantly higher score given that the previous monitors are also correct. This error also puts an upper bound on the hanging accuracy since a wrongly classified partition is considered a complete failure for that monitor, which heavily penalizes the final result as correctly placed series are not considered.

What is also interesting is how the accuracy of the prediction increase the more previous monitors are added. This could be because the learner understands the structure better. However, the partition of the monitors

with high numbers, monitors which are late in the presentation, are often skewed towards one class, so the accuracy is likely overly optimistic.

The series' relation is also interesting as it is possible to score very well by only guessing *no relation* - the largest class. By using undersampling techniques this effect can be mitigated, but can reduce the accuracy of detecting *no relation*. This can still increase the hanging accuracy however. Some attempts of using RUSBoost were made but AdaBoost.M2 often outperformed the undersampling ensemble.

Overall, the accuracy could be higher, since there are multiple predictions and only one result which is considered correct when in reality several configurations could be useful for the user.

6.2.2 Hanging accuracy

The accuracy is likely too low to use in any kind of real system. The system still beats random guessing, which proves that the system is able to generalize and is not simply guessing. The system seems to be able to learn structures in the hanging since the accuracy does not drop off sharply.

One error source is how the correct placement is calculated. It compares the placed orientation with what orientation the user saved. However, from the test case's data 1431 out of 11632, approximately 12% series does not have an orientation which mean they are guaranteed to fail.

Another source of errors is the hanging algorithm itself. Not as much evaluation has been done to it, but when supplied with correct values for partitioning and series relation it is not able to correctly hang all series. Additional tuning of the score assignment is likely necessary, but it is possible that this greedy approach is not a good fit for the problem.

6.3 Issues with labeling

There exist two fundamental issues with the labeling: one with the input and one concerning the correct output. The first issue arises from the difficulty in making distinctions between two series and is compounded by the assumption that using fewer series types is enough. The second issue is because of the user's ability to re-arrange the series as they see fit. If two radiologists look at examinations with the same input - not an impossible scenario - they might, for example, put the prior series on different sides of the monitor. The system would then see two identical inputs but see two different outputs, which will lead to confusion. In Region Skåne it was more common for the staff to use the highest level, *system*, instead of their own role ID when using protocols. Many regional health authorities in Sweden attempt to standardize at least how each hospital in the region work with hanging series which makes this approach easier and more worthwhile to implement.

Classification noise in the training phase of the classifier has been shown[16] to decrease boosting classifiers ability to correctly predict classes. Sources for these types are not random as in much of the literature, but can come from errors in parsing or when two radiologists given the same input choose differently. Using a more advanced version of boosting, such as one presented by [17] could possibly improve the accuracy of the system. However, during testing no other off-the-shelf algorithm provided by MATLAB could perform as well as ensemble learning.

Through discussions with specialists the amount of display protocols which needed manual correction were estimated at 20-30% for CT and CR, while MR could require corrections at a higher rate, potentially as high as 80-90% of the examinations. By comparing the examinations with a presentation to those without a ratio of 3% was found. This could have several explanations: users might not save after correcting, and not all hospitals used Sectra's PACS during the time frame that the data was collected from, which meant the system only imported the examination information and nothing about how it was presented.

6.4 Noise in input

Several studies[9], as well as experience working the data set, has shown that the headers in the DICOM objects include errors, both from human errors and wrongly configured modalities. A radiologist is less likely to be affected by this as they can inspect the series and examinations and draw conclusions that something must have gone wrong. The learning systems however will assume that the input values are correct which will lead to increased confusion for the learning system. The parser works to reduce these kind of errors, but is not able to completely find all these errors. This can cause issues with outliers and must be taken into consideration when selecting classifiers.

Another problem area is that the available database only contains presentations that the radiologists have manually saved, when the existing display protocols have not worked satisfactory. This makes it harder to know when the pre-built ones were good enough, and might make the learning biased towards special cases which do not occur as often.

6.5 Using the image as input

A case could be made for using the image portion of the DICOM objects as inputs as well as the headers. However, much of the information which could be gained from this, such as body part under inspection, should also exist as information in the headers. Using the images would likely contain less mislabeled headers however, but due to time constraints it was easier to work under the assumption that the provided data was correct after some

cleaning. Using machine learning and image processing on the images as a pre-processing step to increase the chance of correct data would probably be a good idea.

6.6 Training input

One potential problem with the input used for training the system is that it in some cases contain poorly constructed presentations. An example of when this occurs is when the radiologist uses the option to automatically hang series which did not fit the selected display protocol, which often results in empty panes in the presentation. This main presentation created by the display protocol is then followed by monitors containing only one series each, often in the order they were generated. This is likely not the best way of displaying the series, but it is good enough for the radiologist studying the case. The machine learning will however be trained on this data and create similar presentations instead of the optimize version which likely exists. It would have been interesting to either filter the training data more or to use an expert to create a training set of very good presentations - for example the ones created by Sectra's application specialists. This would likely produce a lower accuracy against the ground truth, but might create more useful presentations.

Using experts to create presentations however is what this thesis project attempted to solve, but the machine learning would likely be less static and be able to handle noise in the data better than the current display protocols, especially if on-line learning was implemented.

Chapter 7

Conclusions and future work

This chapter offers conclusions from the project and some ideas for future development.

7.1 Conclusions

This thesis presents an approach for using machine learning to fully automatically configure display protocols and hang series in appropriate panes. To do this, several necessary rules to build the layout of a display protocol were identified and different learning algorithms were evaluated for predicting them. Input for the machine learning was generated by a parsing system which gathered presentation data from a database from hospitals in Region Skåne. The thesis proposes a simple algorithm for hanging the series according to the layout rules, and utilizes additional machine learning to increase the likelihood of selecting the correct series. The resulting presentations correspond in roughly 25-33% of the cases to correct presentations according to the ground truth from user-created presentations. However, it is possible that several different presentations could be useful to the user which would mean that the used evaluation criterion is not flexible enough to consider these cases. For the system to be useful, the prediction accuracy has to be increased.

7.2 Future work

This section presents some ideas for future work in the area.

7.2.1 Increase accuracy

If this system is to be of use in a commercial setting the accuracy must be increased. One way of doing this is to increase the amount of inputs that are used in the system to separate the classes more, which some of the classifiers struggle with. When increasing the number of trees in the ensemble the accuracy did not continue to increase after a while, which is a sign more input is needed.

It that would be of interest to store more of the presentation during the examination: a radiologist might be swapping series while working and the final, saved, presentation can end up looking differently from the start. Currently only the final state is available, having more of these presentation states could help understand how the radiologist reasons and their preferences for hanging better.

Another approach would be to mix the existing hard coded solution with the one presented in this thesis, replacing some of the machine learning predictions with hard coded milestones from domain knowledge. This could reduce the output space, and steer the system into more useful configurations.

The greedy hanging algorithm will also require additional work, either more tuning or an entirely different approach could be necessary.

7.2.2 Finding opened exams

An additional rule which would be interesting would be determining which prior examinations would be of value to open and included in the display protocol. An opened examination is an examination that has been loaded into short storage memory and can quickly be accessed from the workstation. This allows the radiologist to manage the presentation without waiting for series to load. A patient can have had several past examinations, however not all are relevant to the current examination.

7.2.3 Adding on-line learning

One of the main challenges with this thesis was that given similar or exactly the same inputs, radiologists might still decide to hang series differently. A solution to this would be to train the system to a general default setting, and then let the user incrementally teach the system simply by working. This could then by done explicitly like GE's solution[8] to avoid training the system outliers at the cost of requiring more user input, or to train the system in the background without the need to explicitly teach the system each time.

7.2.4 Structures in the presentation

Further investigate if structures are present in the reports, and automatically add these to the full presentation. An example of a structure could be an overview of the series, which could contain small versions the series, and the full-sized series would follow.

7.2.5 Taking hardware into account

An additional set of inputs which could not be used was the hardware configuration of the workstation that the radiologist used to view the examinations. While the number of monitors was taken into consideration, there was no information regarding the screen resolution and similar saved in the database.

7.2.6 Feedback from presentation

Currently the system will only feed its results forward, but it could be interesting to use some form of feedback loop from the resulting presentation to use as input to the rules in order to let the system fine-tune itself.

7.2.7 Evaluate on users

The system has been evaluated against a database containing ground truths for specific examinations. However, it is possible that users could have several "ground truths" for what they would consider an acceptable hanging. Letting users evaluate the system could provide qualitative results not possible in this thesis evaluation. Another perspective of such a study would be from the technician who has to set up the display protocols, and compare how fast it is to teach the system rather than setting up the protocols manually.

Bibliography

[1] cc-by-3.0 Juan Pablo Bouza, 2012.

[2] Mythreyi Bhargavan, Adam H. Kaye, Howard P. Forman, and Jonathan H. Sunshine. Workload of radiologists in united states in 2006-2007 and trends since 1991-1992. *Radiology*, 252(2):458–467, Aug. 2009.

[3] Bruce I. Reiner, Eliot L. Siegel, Charles Flagle, Frank J. Hooper, Robert E. Cox, and Mary Scanlon. Effect of filmless imaging on the utilization of radiologic services. *Radiology*, 215:163–167, 2000.

[4] J. Nederend, L. E. M. Duijm, M. W. J.Louwman, J. H. Groenewoud, A. B. Donkers van Rossum, and A. C. Voogd. Impact of transition from analog screening mammography to digital screening mammography on screening outcome in the netherlands: a population-based study. *Annals of Oncology*, 23(12):3098–3103, Dec. 2012.

[5] Vivek Joshi, Kyootai Lee, David Melson, and Vamsi R. Narra. Empirical investigation of radiologists' priorities for pacs selection: An analytical hierarchy process approach. *Journal of Digital Imaging*, 24(4]):700–708, Aug. 2011.

[6] John M. Boone, Greg S. Hurlock, Anthony Seibert, and Richard L. Kennedy. Automated recognition of lateral from pa chest radiographs: Saving seconds in a pacs environment. *Journal of Digital Imaging*, 16(4):345–349, Dec. 2003.

[7] Hui Luo, Wei Hao, David H. Foos, and Craig W. Cornelius. Automatic image hanging protocol for chest radiographs in pacs. *IEEE Transactions on Information Technology in Biomedicine*, 10(2):302–311, Apr. 2006.

[8] Tianyi Wang and Alexandre Iankoulski. Intelligent tools for a productive radiologist workflow: How machine learning enriches hanging protocols. White paper, GE Healthcare, July 2013.

[9] Adrian Moise and M. Stella Atkins. Workflow oriented hanging protocols for radiology workstation. In *Proc of SPIE*, pages 189–199, 2002.

[10] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, Aug. 1997.

[11] Nikunj Chandrakant Oza. *Online Ensemble Learning*. PhD thesis, University of California, 2001.

[12] Stephen Marsland. *Machine learning: an algorithmic perspective*. CRC Press, 2011.

[13] Yoav Freund and Robert E. Schapire. Experiments with a new boosting algorithm. In *ICML*, volume 96, pages 148–156, 1996.

[14] Chris Seiffert, Taghi M Khoshgoftaar, Jason Van Hulse, and Amri Napolitano. Rusboost: improving classification performance when training data is skewed. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–4. IEEE, 2008.

[15] Manfred K. Warmuth, Jun Liao, and Gunnar Rätsch. Totally corrective boosting algorithms that maximize the margin. In *Proceedings of the 23rd international conference on Machine learning*, pages 1001–1008. ACM, 2006.

[16] Philip M. Long and Rocco A Servedio. Random classification noise defeats all convex potential boosters. *Machine Learning*, 78(3):287–304, 2010.

[17] Adam Kalai and Rocco A. Servedio. Boosting in the presence of noise. In *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, pages 195–205. ACM, 2003.

På svenska

In English