Smartphone App Concept for Medical Applications.

Alexander Gustafson
University of Applied Sciences,
Zürich,
Switzerland,
alex_gustafson@yahoo.de

September 12, 2016

Dozent: Reto Knaack (knaa@zhaw.ch)

School of Engineering, Abteilung Zürich
Studiengang Informatik

Abstract

The market for smartphone based medical applications is a relatively new and growing quickly. The majority of medical apps are relatively simple health management and tracking applications that might remind a user to take his or her medicine or monitor blood pressure and heart rate data provided by accompanying devices. However, more sophisticated apps can directly provide diagnostic information by capturing and analysing data directly. Several apps exist that can asses the risk of skin cancer by tracking changes in the growth of skin lesions over time.

Contents

Chapter 1

Introduction

TODO: Medical Apps / future promise and benefits / communication with experts / fast diagnosis

TODO: Fields that can benefit from Medical Apps on smart phones

TODO: What is cancerous melanoma? danger? importance of early detection.

1.1 Project Description

This bachelor project will investigate the following questions:

- What smartphone based medical apps are available that can track and assess the risk of skin lesions?
- What methods and algorithms can be employed by mobile phones to calculate the risk?
- What is necessary to build a mobile app that can provide a risk assessment of skin cancer melanoma from captured images?

1.2 Goals

1. Research the medical app market to gain an overview of areas of development, what apps are available, what trends are discernible. Of special interest are apps that use internal sensors (e.g., camera) and mathematical algorithms to detect skin cancer melanoma.
2. Gain familiarity with machine learning and image recognition algorithms. Describe the basic principles of these algorithms.
3. Research and compare available data and algorithms (e.g. in computer vision / machine learning). Define the relevant attributes that can be used to train the algorithm. This can also include changes in size over time.

4. Based on the research and comparison choose a specific algorithm and data with which to proceed. The algorithm must be able to
   (a) identify the skin lesion correctly,
   (b) compute the relevant attributes,
   (c) use these attributes to classify the skin lesions whether it is possibly cancerous or not
5. Implement the algorithm as a prototype (using Python for example). Using images of a skin lesion as input, the algorithm will provide a risk assessment as output. Implement tests that can calculate the accuracy of the algorithm. Compare the implemented algorithm with at least one other algorithm.
6. Create a concept for a medical mobile app that can utilise the features of modern smartphones to provide a risk assessment of skin cancer melanoma based on captured images. This includes a full requirements analysis, i.e. use cases, functional / non-functional requirements, architecture.
7. Implement as proof of concept specific aspects of the medical mobile app.

1.3 Expected Results

1. Results of the medical app market research including graphical presentation of trends and statistics.
2. Documentation of the available data and algorithms including high level excursion into the theory behind machine learning and image recognition algorithms.
3. Comparison of available data and algorithms. Definition of relevant attributes.
4. Decision, which algorithm to use.
5. Documentation of the implementation of the algorithm, including tests, evaluation and comparison with at least one other algorithm.
6. Documentation of a concept for a medial app, including defined use cases, requirements, architecture and implemented design patterns.
7. Proof of Concept: implementation of specific aspects of the medical mobile app.

Chapter 2

Market Research

2.1 Data Gathering

2.1.1 Gathering Data from the Apple iTunes Store

Searching the Apple iTunes store is typically done manually via the iTunes Application from which text and data cannot be automatically extracted. Therefore, searching for and gathering data about IOS Applications is not easy. However, Apple does provide an rss feed that can be used to list Apps in specific categories and ordered according to how new, or how popular they are and if they are free or not. The rss feed is limited to 100 items per category. The data provided by the rss feed is minimal, not much more than title and a text description of the app. There are no sub-genres or tags than can be used to further differentiate the apps.

Using a python script data was gathered from the following rss feeds:

Top 100 Free Medical Apps Top 100 Grossing Medical Apps Top 100 Paid Medical Apps This combined results included data about 255 IOS apps. The title and description fields were imported into a database. Other information from the data such as price, right, or image link were ignored.

2.1.2 Gathering App Data from the Google Play Store

In order to gather data from the Goole app store a script was programmed that could extract lists of apps from a specific url. The following urls were scanned:

- Top Paid Medical Apps : https://play.google.com/store/apps/category/MEDICAL/collection/topselling_paid
- Top Free Medical Apps : https://play.google.com/store/apps/category/MEDICAL/collection/topselling_free

For each app listed the script would extract the url of the app's detail page. From the detail page more imformation would be gathered and stored in a database. The data set is similar to that of the itunes rss feed. The title and description text were imported, other fields such as pricing and copyright were ignored.

Data on 480 Medical Apps for Android was imported. However a significant percentage of the apps could not be classified because the description text was in a language other than English, German, or French.

2.2 Categorization

The term ”Medical App” is broad and neither the iTunes nor Google app stores offer any kind of sub categorization. In order to get a better overview of what sort of Medical Apps are available it was necessary to manually browse the gathered data and assign categories to the apps.

A database management tool was created using the python based Django Web Framework. Django provides many tools that makes constructing and interacting with databases very easy. The built-in backend administration tool can be configured to browse, edit, and filter data.

In order to quickly browse through and categorize over 700 apps. The Django backend admin was configured so apps could be categorized one after the other with a minimum of clicks or scrolling. The user was presented a list of uncategorized apps. The first one is clicked. The user is then presented with a page displaying the title and summary text of the app and a field from which a category can be selected. Once saved, the app is no longer presented on the list, the user can select the next app at the top of the list.

2.2.1 Description of Categories

- **Community** - provides some soft of social networking service through which the user can share data with her family or with a network of people suffering from similar disorders.
- **Fun / Entertainment** - These apps have to real medical purpose. They are for enjoyment only.
- **Alert / First Response** - Apps that assist first responders or that help users alert first responders that help is needed.
- **Health / Lifestyle** - Relaxation and meditation apps, or ovulation and fertility reminders
- **Resource Finder** - Apps that locate resources in the vicinitly, nearby pharmacies or care providers.
- **Reminder** - Apps with timer or calendar functionality that might remind a user of an appointment or manage medince consumption.
- **Algorithmic / Diagnostic** - These are apps that provide some sort of diagnositc information based on data that as been gathered by sensors or entered by the user. Examples are seizure detection apps, or stroke severity evalutaion apps.

- **Learning / Educational / Reference** - By far the largest category, this includes apps that provide reference information about diseases or education material like anatomy apps for example.
- **Organisational** - Apps in this category might help a user or practitioner organise, share or track data and documents. Examples are apps that help users track the status of their blood pressure or blood sugar levels, create health diaries, or manage clinical data and images.

2.3 Results

2.3.1 IOS Medical Apps

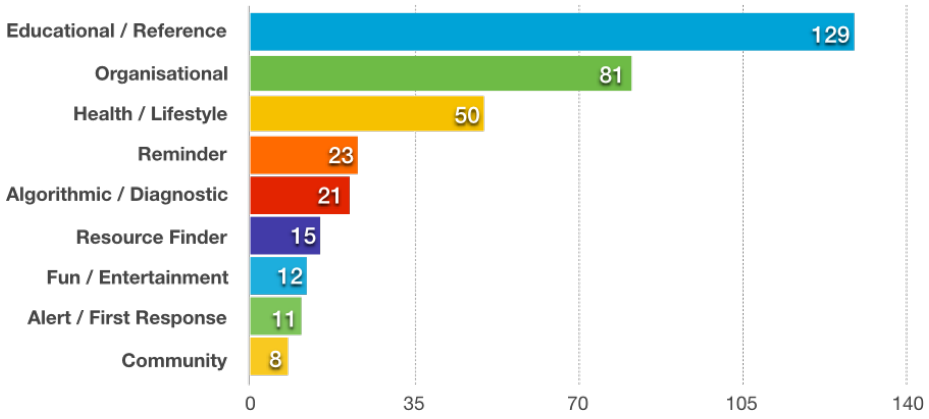

Figure 2.1: Medical Apps on the iTunes Apple Store, Search conducted on 17.05.2016

2.3.2 Android Medical Apps

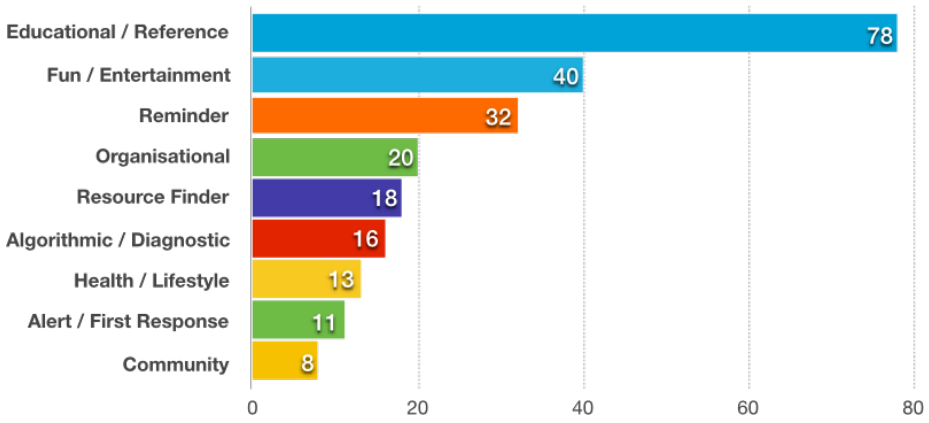

Figure 2.2: Medical Apps on the Goolge Play Store, Search conducted on 17.05.2016

2.3.3 Dermatology Apps 2013 vs 2016

Mobile apps are an especially good fit for dermatology-related care. Most dermatological conditions are by nature visible. The initial diagnosis and follow up monitoring is mostly done visually. A mobile device with a camera can aid patients and practitioners in the diagnosis of a dermatological condition and tracking it's development. The article Mobile Applications in Dermatology [6] in 2013 identified 229 dermatology-related apps across 5 app platforms (Android, Apple, Blackberry, Nokia, and Windows). These were grouped into categories based on their primary functionality. The "Self-surveillance/diagnosis" category was the second largest on the Android and Apple platforms, with 13 and 24 apps respectively.

Table 1. Total Applications

Category	Android	Apple	Blackberry	Nokia	Windows	Total, No. (%)
Reference	22	35	3	0	1	61 (26.6)
Self-surveillance/diagnosis	13	24	1	0	3	41 (17.9)
Disease guide	20	10	7	0	2	39 (17.0)
Educational aid	7	11	2	0	0	20 (8.7)
Sunscreen/UV recommendation	7	12	0	0	0	19 (8.3)
Calculator	2	9	1	0	0	12 (5.2)
Teledermatology	1	7[a]	0	0	0	8 (3.5)
Conference	2	3	0	1	0	6 (2.6)
Journal	2	4	0	0	0	6 (2.6)
Photograph storage/sharing	1	4	0	0	0	5 (2.2)
Dermoscopy	0	2	0	0	0	2 (0.9)
Pathology	0	2	0	0	0	2 (0.9)
Other	1	6	0	1	0	8 (3.5)
Total applications, No. (%)	78 (34.1)	129 (56.3)	14 (6.1)	2 (0.1)	6 (2.6)	229 (100.0)

[a] Two additional Apple iOS teledermatology applications were identified in a query of the term *teledermatology* during March 2013: HIV-Derm Algo Study (launched October 20, 2012) and SFGH Teledermatology pilot (launched November 12, 2012).

Figure 2.3: Total Applications, Brewer 2013

Using the same search criteria and categories from the article above indicates that the availability of dermatological apps is growing. Today there are 33 dermatological apps on the Apple platform that can be identified as having ”Self-surveillance/diagnostic” features. With the exception of the ”Reference” category, all other categories show significantly higher numbers of available apps.

Category	Apple 2013	Apple 2016	Android 2013	Android 2016
Reference	35	29	22	31
Self-surveillance/diagnosis	24	33	13	23
Disease guide	10	24	20	46
Educational aid	11	23	7	24
Sunscreen/UV recommendation	12	20	7	19
Calculator	9	4	2	10
Teledermatology	7	19	1	12
Conference	3	11	2	17
Journal	4	19	2	10
Photograph storage/sharing	4	3	1	1
Dermoscopy	2	7	0	3
Pathology	2	0	0	0
Other	6	8	1	4
Total	129	200	78	200

Figure 2.4: Demotological Apps by Category, July 2013(Brewer 2013) vs May 2016

It is important to note that the categories listed above are not defined in the stores. The apps must be manually assigned to a category based on an interpretation of the description text in the store and information obtainable on related websites. It is possible therefore that apps that were originally designated to the “Reference” category

might have been interpreted in this paper as a “Educational aid” app for example. The interpretation of the functionality of an app is fuzzy in many cases, and many apps have some crossover functionality. An app developed for self-surveillance will often contain information pertaining to symptoms and treatment (reference).

2.3.4 Dermatological Apps with Automatic Risk Assesment

Of the 55 apps identified belonging to the ”Self-surveillance/diagnosis” category, only 2 provided risk assesment features based on automatic analysis of captured images.

- SkinVision : https://skinvision.com
- mSkin Doctor https://play.google.com/store/apps/details?id=com.maleemtaufiq.mSkinDoctor

Chapter 3

Image Data Sources

3.1 Dermofit

High quality clinical images with corresponding masks. Great for training and testing.

Image Type	Description	Amount	Comment
Actinic Keratosis	Pre-cancerous patches of flakey or crusty skin, can develop into Squamous Cell Carcinoma	45	Not useful as comparison against melanoma
Basal Cell Carcinoma	Abnormal, uncontrolled growths of the skin's basal cells.	239	Not useful as comparison against melanoma
Dermatofibroma	Common and benign skin tumour.	65	Useful as comparison, some extreme cases might have to be left out of the training set.
Haemangioma	A collection of small blood vessels that form a lump under the skin.	97	Useful as comparison, some extreme cases might have to be left out of the training set
Intraepithelial Carcinoma	A type of squamous cell skin cancer limited to the upper layer of the skin.	97	Not useful
Malignant Melanoma	A type of cancer that develops from the pigment-containing cells known as melanocytes.	76	This is our baseline set, half will be used for training, the other half for testing
Melanocytic Nevus	Typical mole, benign.	331	Very usefull as comparison to Melanoma
Pyogenic Granuloma	Common skin growth, small, round and red in color due to large number of blood vessels.	24	Not usefull
Seborrhoeic Keratosis	Common non-cancerous skin growth.	257	Maybe usefull
Squamous Cell Carcinoma	Abnormal and uncontrolled growth of squamous cells in the epidermis.	88	Not usefull

Table 3.1: Dermofit Image Categories

3.2 DermQuest

Online teaching and learning resource with large image database.

Image were selected based on their usefullness for this project. Usefullness is based on quality and type of image. Dermoscopic images were not selected. Instead images were taken that appeared to be taken with a standard camera. The images only contained the mole to be examined and the surrounding skin area. No other details like eyelids, ears, or dark shadows.

Subgroups of Melanocytic Nevus were chosen. Dysplastic and Intradermal Nevus for the non-cancerous cases, and Malignant Melanoma as the cancerous cases.

Image Type	Description	Amount	Comment
Benign Keratosis		5	
Malignant Melanoma	A type of cancer that develops from the pigment-containing cells known as melanocytes.	39	This is our baseline set, half will be used for training, the other half for testing
Melanocytic Nevus	Typical mole, benign.	51	Very usefull as comparison to Melanoma

Table 3.2: DermQuest Image Categories

3.3 PH2Dataset

Demascopic Images

Many of the mole images extend almost to or even beyond the image border. This makes some of the processing difficult. However the images include an excel spreadsheet which clearly designates what type of mole, including some scoring info such as Asymmetry and Color information. Includes border mask images.

Image Type	Description	Amount	Comment
Common Nevus		79	
Atypical Nevus		79	
Melanoma		39	Baseline set for training

Table 3.3: PH2 Image Categories

Chapter 4

Image Feature Extraction

A digital image is a matrix of pixels that contain color information, typically comprised of the 3 color channels red, green, and blue. A person can look at an image and quickly make statements about it's content. For instance, someone might look at an image of a street scene and be able to easily say "This is an image of a street in a town, there is one automobile, three people and a building in this scene". A person would easily be able to draw outlines around the objects and be able to differentiate between areas in each object.

For a computer to be able to "make statements" about the contents of an image it must use algorithms to group together and differentiate between objects in an image. The grouping together and differentiating of areas in an image is known as *segmentation*. The statements that a computer can make about the content of an image are usually of statistical nature and are refered to as *feature extraction*. Before segmentation an image is goes through a *preprocessing* stage in order to remove or reduce irrelevant information or noise.

There are no universal algorithms for preprocessing, segmentation, or feature extraction. The algorithms employed depend on the context of the images to be analyed and the type of information that one is looking for.

4.1 Preprocessing

4.1.1 Equalization

4.1.2 Median Blur

4.1.3 Gaussian Blur

4.2 Segmentation

4.2.1 Iterative Thresholding

4.2.2 Region Selection

4.3 Image Feature Extraction

Chapter 5

TDS Algorithm

5.1 Description of the Algorithm

Early detection of melanoma greatly increases the chances of successful treatment. A biopsy can be performed in order to gain a definitive diagnosis. However, biopsies are invasive, painful and take time. There are also visual markers Dermatologists look for in order to make a risk assessment. The ABCD Rule, also known as Stokes or TDS Calculation, looks for 4 sets of features. Based on the features the Total Dermoscopy Score (TDS) is calculated.

The 4 sets of features are Asymmetry (A), Border Irregularity (B), Color (C) and Differential Structure or Diameter (D).

Asymmetry can have a value of 0, 1 or 2 depending on the symmetry of the lesion. Where 0 is symmetric and 2 is asymmetric. A value of 1 indicates at least one axis was found across which symmetry exists. The Border score is an integer value from 0 to 8 indicating the presence of border irregularities in 8 regions. Color is an integer value from 1 to 6 indicating the presence of one to six specific colors. Similarly, the value for D indicates the presence of one to five distinct structures or textures. Alternatively, in some literature [13] D is defined as Diameter, where a diameter greater than 6mm results in a value of 5, otherwise 1.

The final TDS Score is the weighted sum of the ABCD Values and is in the range 1.0 to 8.9.

$$TDS = A * 1.3 + B * 0.1 + C * 0.5 + D * 0.5 \tag{5.1}$$

A diagnosis can be made based on the TDS Score according to the following table:

Evaluation	TDS Score
Benign	$<$4.75
Suspicious	4.75 to 5.45
Malignant	$>$5.45

Table 5.1: TDS Evaluation [16]

5.1.1 Adaptation for use in a smart phone application

The ABCD Rule, also known as the Total ***Dermoscopy*** Score, is really only applicable to images captured using a dermatoscope. Especially the differential structures are only visible on dermatoscopic images where the subsurface structures are made visible [4].

Since the D component is not applicable to images captured with a smart phone this project will use a modified TDS without the D component. The original TDS formula 5.1 can have values ranging from 1 to 8.9 and D can be 0.5 to 2.5. Without D the TDS can have values between 0.5 and 6.4.

$$TDS_{mod} = A * 1.3 + B * 0.1 + C * 0.5 \tag{5.2}$$

This results in a bening cuttoff score of 3.2 and malignagnt cuttoff of 3.7.

Evaluation	Score
Benign	$<$3.20
Suspicious	3.20 to 3.7
Malignant	$>$3.7

Table 5.2: Adapted TDS Evaluation

5.2 Automatic Calculation of ABCD Values

The ABCD Rule is used by Dermatologists to differentiate benign from malignant melanocytic tumors. It is a clearly defined rule based on easily recognizable visual features. Clinicians with limited dermoscopy experience achieve better results using the ABCD rule than other methods [16]. The following sections detail methods to automate the calculation of the ABCD scores.

5.2.1 Asymmetry

The ”Center of mass” method of measuring asymmetry is relatively easy to calculate and is more accurate compared to other complex algorithms [11].

This algorithm begins by calculating an array of radii from the lesion's center of mass to border for each of 360 degrees. The coordinates for the center of mass are calculated from the sum of all x and y coordinates of pixels within the lesion's border each divided by the sum of pixels.

For each of the 360 radii r_i a score is calculated by comparing the lengths of pairs of radii that are symmetric across r_i. If the lengths of the pair of symmetric radii have a difference of less than 10% then a point is given. The sum of points is the SFA_i (Score For Axis) for r_i.

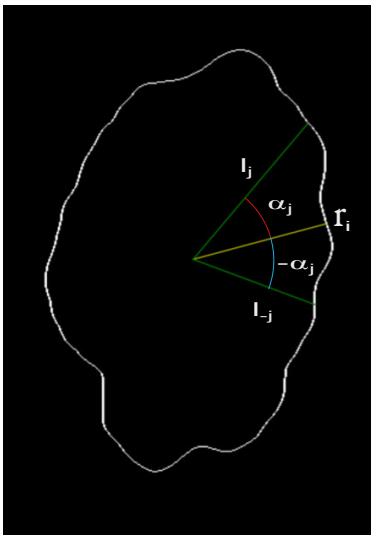

Figure 5.1: Calculate SFA for r_i

The radius with the maximum SFA score is defined as the major axis of symmetry. The SFA of the major axis as well as the perpendicular are stored. The Asymmetry score is evaluated as follows:

SFA Results	Description	Asymmetry Score
major axis $\geq$ 140 minor axis $\geq$ 140	Symmetric across both axis	0
major axis $\geq$ 140 minor axis $<$ 140	Symmetric across one axis	1
major axis $<$140 minor axis $<$140	Asymmetric	2

Table 5.3: TDS Evaluation [16]

5.2.2 Border

An image is generated where only the pixels at the border or the lesion are visible, the rest of the image is black. The pixels are gathered sequentially into an array starting at any arbitrary border pixel and traversing the border continually. It's important that the sequential order of the pixels be maintained since the change in angle and distance to the lesions center will be measured.

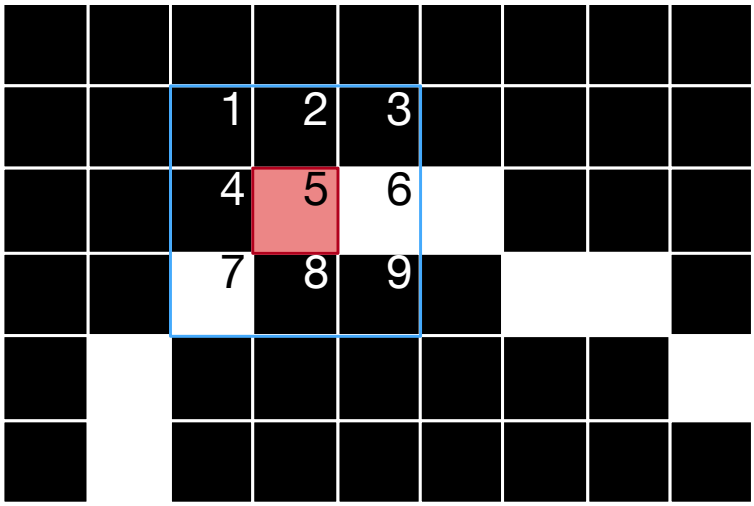

Figure 5.2: Traverse the border sequentially

An algorithm was developed that finds an arbitrary border pixel in an image that is otherwise black by iterating though pixels starting at the top left corner. When it has found a pixel that is not black it tests neighboring pixels, starting with the upper left neighbor and traversing the pixels row by row starting from the left most pixel in a row, as illustrated by the numbered pixels in figure 5.2. The center pixel, pixel number 5, is skipped.

When a non black pixel is detected it is compared to a a list or previously detected pixels. If it has not been previously detected, it is added to the list and becomes the new center. The algorithm is repeated.

If no new non-black pixel is detected the outter range is expanded by one pixel at each edge and the pixels at the edge of a 5 x 5 area are tested. This is repeated until a new pixel is found. This prevents the algorithm from halting if there are discontinuities in the border.

When the algorithm reaches the starting pixels it ends. The list of pixels now contains a sequential list of pixels that are in sequential order.

For each pixel the distance and angle from the lesion's center of mass is calculated. If a lesion's border is irregular the measure of distance and angle from the center will be erratic, whereas is the border is relatively circular and smooth the measure of distance and the difference in angle will not change abruptly between neighboring pixels.

The distance function is averaged around the distance of the start pixel (set to zero) in order to reduce discontinueities. A gaussian lowpass filter is used to reduce noise and aliasing artifacts due to pixelization.

From the smoothed distance function the first derivative is calucalated. From the angle function the difference from neighboring pixels is calculated. The resulting two functions are split into 8 equal segments. Each segment is analysed for strong variations in distance and sign changes in the difference of the angles. A sign change would

indicate a very strong change in direction of the border function. For each segment in which strong variations are detected a point in the B score is added.

A lesion with strong variations in one segment, but otherwise smooth border function would have a B score of 1, whereas a lesion with variation is each of the 8 segments would have a B score of 8.

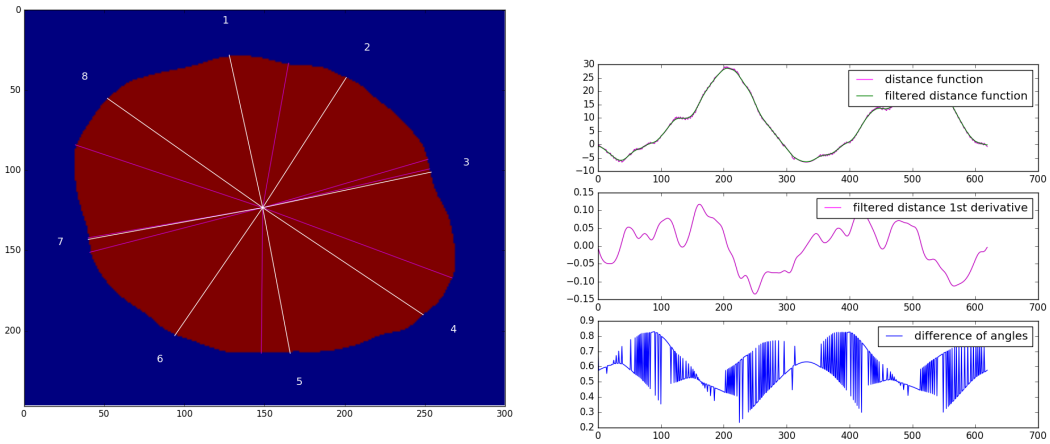

Figure 5.3: Example of border score 0

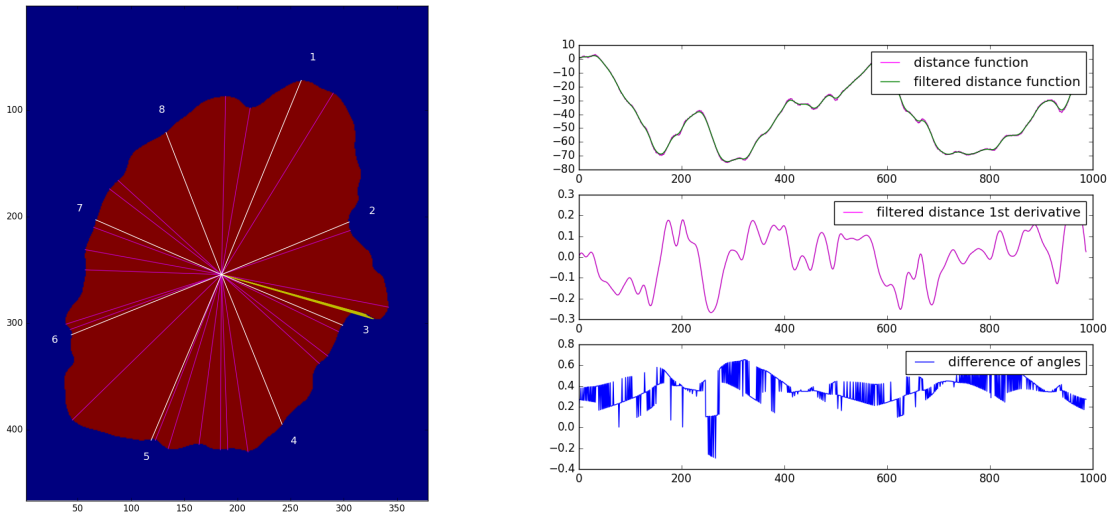

Figure 5.4: Example of border score 4

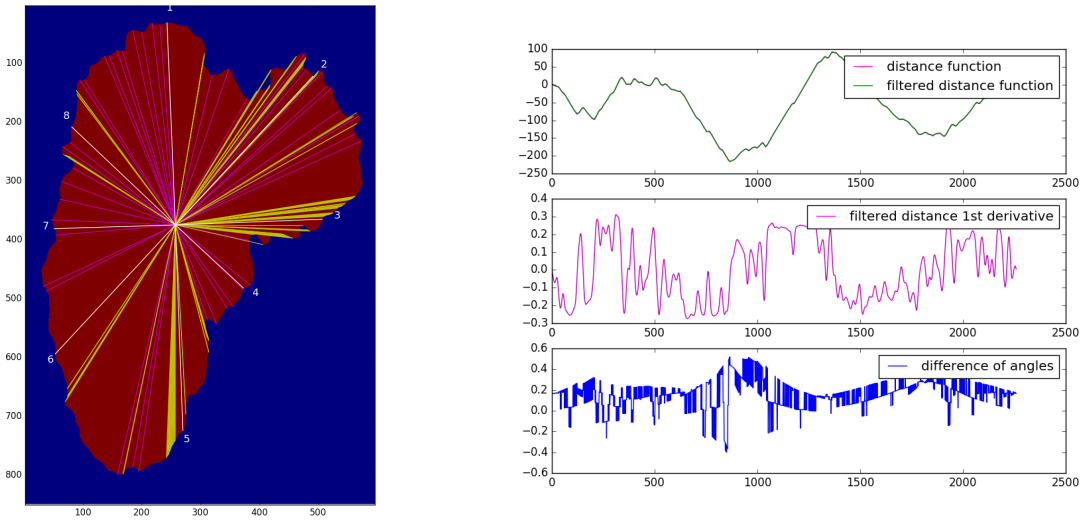

Figure 5.5: Example of border score 8

5.2.3 Color

In order to calculate the color score the *Quantification of Color* [4] algorithm was implemented in which the distance of each pixel to a defined set of colors is evaluated.

The original image is first filtered using a low pass Gaussian filter in order to reduce noise. Then for for each pixel with the lesion area the euclidean distance from 6 RGB values is calculated. The 6 RGB values, as shown in table 5.4, represent the colors that are relevent to the TDS calculation.

Color	R	G	B
White	255	255	255
Red	204	51	51
Light Brown	153	102	0
Dark Brown	51	0	0
Blue Gray	51	153	255
Black	0	0	0

Table 5.4: RGB values of the six possible TDS-relevent colors [4]

For each color a counter is incremented when a pixel is found that is closest to it. The end results are divided by the total number of pixels within the lesion's boundaries. For each color with a value greater than 0.01 a point is given to the TDS C score.

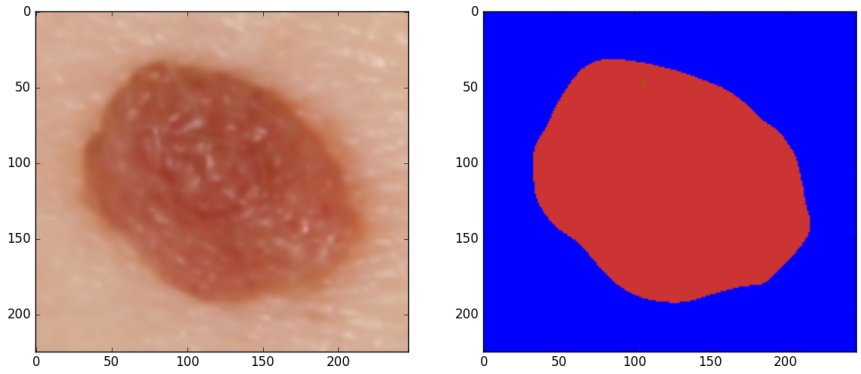

Figure 5.6: Example of color score 1

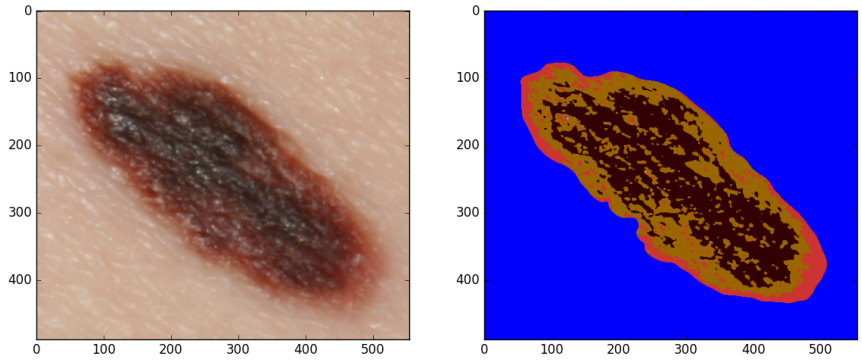

Figure 5.7: Example of color score 3

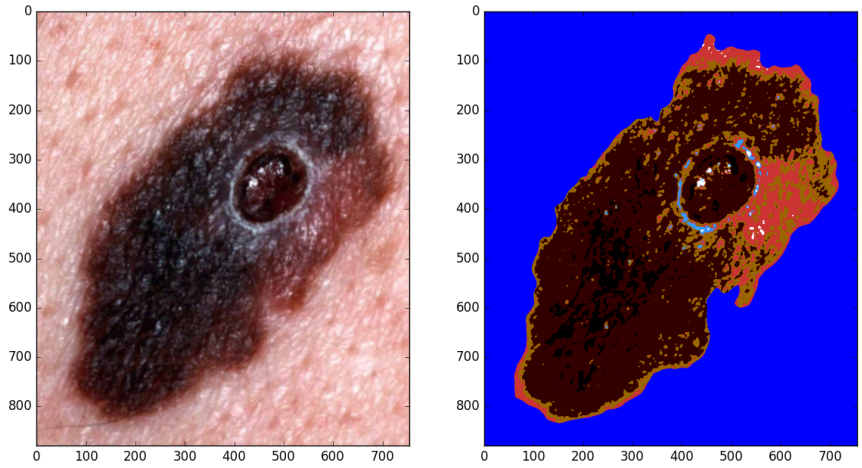

Figure 5.8: Example of color score 4

5.2.4 Differential

5.3 Two Examples

5.3.1 Postitiv Example

5.3.2 False Example

TDS Values were calculated but classification result was false.

5.4 Results of Algorithm

Source	False Positives	False Negatives	Correct	Total
Dermofit	10	8	115	133
Dermquest	10	4	32	46
PH2	13	0	29	42

Table 5.5: Results of TDS calculation

Source	Malignagt	Malignant Correct	Total
Dermofit	16	8	133
Dermquest	16	12	46
PH2	0	0	42

Table 5.6: Correctly Diagnosed Malignant Melanoma Images

5.5 Performance Evaluation

Chapters 11.1, 11.2, maybe 11.3

Chapter 6

Machine Learning

6.1 Section Title

6.2 Results of Algorithm

Performance Evaluation

Chapters from book 11.1, 11.2, 11.3, 11.5, 11.6

Chapter 7

Implementaion of the Algorithm

7.1 Section Title

Chapter 8

Mobile App Requirements

This chapter will describe the requirements of the application as a whole. The structure of this chapter roughly follows the SRS Template as proposed by Wiegers in Chapter 11 and Appendix D of [17].

8.1 Vision and Scope

8.1.1 Background

Melanoma is a type malignant skin tumour that develops from benign melanocytic nevi. Although less frequent than other skin cancer types, it causes more deaths [3], and it's incidence is increasing dramatically, especially in the young white population [14]. Early diagnosis and treatment is vital.

Several non-invasive techniques exist which dermatologist can employ to visually make a diagnosis. The most common are pattern analysis, the ABCD rule, the 7-point checklist and the Menzies method, among others [9]. These methods generally use a combination of defined visual clues and indicators to assess the risk of a skin lesion. CAD (Computer Aided Diagnosis) systems exists that can assist a dermatologist, and even provide automatic diagnosis with an accuracy comparable to that of doctors trained in dermoscopic techniques [7].

8.1.2 Business Case

A smart phone based assistant would provide a valuable service by making it easy, painless, inexpensive, and fast to get a preliminary risk assessment of a skin lesion. It is too easy to postpone making an appointment with a dermatologist to have a suspect lesion examined. Visiting an dermatologist is not on most people's list of fun things to do in the spare time, it requires time and effort. By the time a lesion is visually compelling enough, it might be too late.

A smart phone based application that could make a preliminary diagnosis in near real time would be a great time save and offer compelling enough information to actually make the appointment with a dermatologist.

8.1.3 Scope and Limitations

8.1.3.1 Major Features

FE-1 : The smart phone app allows the user to capture and analyse an image of a skin lesion and provide a risk assessment to the user of the lesion being a malignant melanoma.

FE-2 : The app allows the user to create, view, edit or delete metadata that is associated with an image and corresponding risk assessment.

FE-3 : The app allows the user to save or archive the image and corresponding assessment for future comparison and review.

FE-4 : The user can browse archived images, assessments, and associated metadata.

FE-5 : The user can send a set of images with associated assessment and metadata via email.

FE-6 : The app will run on all popular smart phone devices and systems.

FE-7 : The analysis algorithms employed by the app should be easily updatable and extendable.

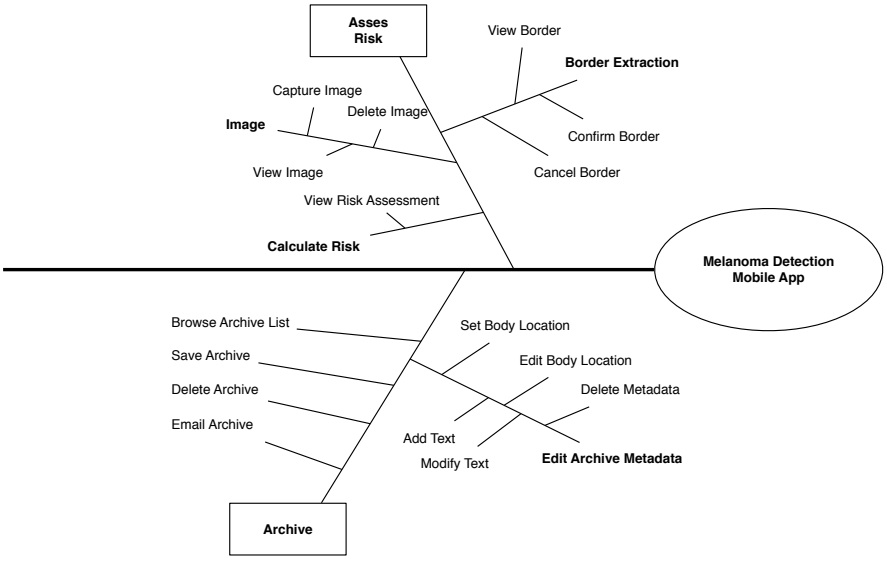

Figure 8.1: Partial Feature Tree for the Melanoma Detection App

8.1.3.2 Stakeholder Profile

8.1.4 Use Cases

ID and Name:	**UC-1: Capture Image**		
Primary Actor:	Smartphone User	**Secondary Actor:**	Smartphone Camera System
Description:	The User is shown a real time preview of the camera's input. With the help of visual indicators layered over the real time preview the User can optimise the distance and angle of the Smartphone. When the real time preview appears to show an optimal image the User can activate the image capture process.		
Trigger:	The User has navigated to the Image Capture view of the App.		
Preconditions:	None		
Postconditions:	POST - 1 : Image file is saved on the Smartphone file storage system POST - 2 : A reference to the image is saved in the MDApp		
Normal Flow:	**1.0 Capture an Image** 1. User is presented with a real time preview of the camera's input. 2. User adjusts the camera position until the real time preview is optimal. 3. The User activates the image capture process 4. The MDApp acquires the image from the Smartphone Camera System 5. The MDApp stores the image and a reference		
Alternative Flow:	None		
Exceptions:	1.0.E1 : Camera hardware is not available		
Priority:	High		

Figure 8.2: UC-1

ID and Name:	**UC-2: Confirm Border**		
Primary Actor:	Smartphone User	**Secondary Actor:**	
Description:	The MDApp initiates the border extraction calculation. When the calculation is complete the User is presented with a visual rendering of the extracted border. The MDApp prompts the user to confirm or reject that the border was calculated precisely.		
Trigger:	The User captured an image.		
Preconditions:	PRE - 1 MDApp saved and image and a reference		
Postconditions:	POST - 1: A datafile of the extracted border is saved to the Smartphone's file storage system. POST - 2 : A reference to the Datafile is saved in the MDApp		
Normal Flow:	**1.0 Confirm Border** 1. The MDApp automatically initiates the border extraction process. 2. When the border extraction process is finished the User is presented with a visual rendering of the border over the image of the skin lesion for comparison. 3. The MDApp prompts the User to confirm that the border precisely outlines the area of the lesion. 4. If confirmed, the MDApp stores the border datafile and a reference.		
Alternative Flow:	**1.3 Reject Border** 3. The User rejects the border because is does not precisely outline the lesion area. 4. The User is returned to step 1 of UC-1		
Exceptions:	2.0.E1 : border calculation service is not responding. 2.0.E2 : a border could not successfully be calculated.		
Priority:	High		

Figure 8.3: UC-2

ID and Name:	**UC-3: Set Body Location**		
Primary Actor:	Smartphone User	**Secondary Actor:**	
Description:	The User is presented with a visual representation of a human body. The User selects a location on the body. The MDApp saves the body location data as metadata associated with captured image.		
Trigger:	The User navigates to the metadata view of the MDApp		
Preconditions:	PRE - 1 MDApp saved and image and a reference		
Postconditions:	POST - 1: Metadata describing the location of the lesion is stored with a reference to the image of the lesion.		
Normal Flow:	**1.0 Set Body Location** 1. The user navigates the the metadata view of the MDApp. 2. The MDApp presents the user with a visual representation of the human body. 3. The User selects an area of the body. 4. The MDApp stores the body location as metadata associated with the captured image.		
Priority:	Medium		

Figure 8.4: UC-3

ID and Name:	**UC-4: Set Text Metadata**		
Primary Actor:	Smartphone User	**Secondary Actor:**	
Description:	The User is presented with text area where she can enter any text that should be associated with the captured image		
Trigger:	The User navigates to the metadata view of the MDApp and scrolls to the text area.		
Preconditions:	PRE - 1 MDApp saved an image and a reference		
Postconditions:	POST - 1: Metadata test is stored with a reference to the image of the lesion.		
Normal Flow:	**1.0 Set Text Metadata** 1. The MDApp presents the user with a text area 2. The User enters text in the text area 3. The MDApp stores the text as metadata associated with the captured image		
Priority:	Medium		

Figure 8.5: UC-4

ID and Name:	**UC-5: Save Archive**		
Primary Actor:	Smartphone User	**Secondary Actor:**	Smartphone File System
Description:	The User may store an image and associated metadata for review and comparison at a later date.		
Trigger:	The User clicks the “Save” button		
Preconditions:	PRE - 1 MDApp saved an image and it’s reference.		
Postconditions:	POST - 1 The MDApp saves the image and associated metadata as an archive to the Smartphone’s file system.		
Normal Flow:	**1.0 Save Archive** 1. The User clicks the “Save” button. 2. The MDApp gathers last captured image together with the associated metadata (including the calculated border and risk assessment if available) and saves this to the Smartphone File System.		
Priority:	Medium		

Figure 8.6: UC-5

ID and Name:	**UC-6: Browse Archive List**		
Primary Actor:	Smartphone User	**Secondary Actor:**	Smartphone File System
Description:	The User may scroll through a list of previously archived images and metadata. The list provides a preview or the image as well as the datae of when the image was captured. The list is ordered by the date of capture.		
Trigger:	The User navigates to the archive-list view.		
Preconditions:	PRE - 1 : At lease one image as previously been archived.		
Postconditions:	None		
Normal Flow:	**1.0 Browse Archive List** 1. The MDApp presents the user with a navigable list of previously saved archived images. 2. The User can navigate or scroll through the list.		
Priority:	Medium		

Figure 8.7: UC-6

ID and Name:	**UC-7: View Archive Detail**		
Primary Actor:	Smartphone User	**Secondary Actor:**	Smartphone File System
Description:	The User may select an archive from the archive list and the MDApp with provide the user with a detail view of the selected archive.		
Trigger:	The User selects an archive from the archive-list view.		
Preconditions:	PRE - 1 : At lease one image as previously been archived.		
Postconditions:	POS - 1 : A detail view of a previously archived image and associated metadata to visible to the user. POS - 2 : The selected archive is designated by the MDApp as the active archive.		
Normal Flow:	**1.0 View Archive Detail** 1. The User selects an item in the list of archived images 2. The MDApp designates the selected item as the active archive. 3. The MDApp presents a view showing image and associated metadata to the user.		
Priority:	Medium		

Figure 8.8: UC-7

ID and Name:	**UC-8: Email Archive**		
Primary Actor:	Smartphone User	**Secondary Actor:**	Smartphone Email System
Description:	The User may send an archived image and associated metadata by email to a dermatologist or doctor for review. A subject and message text can be added by the user before submitting via one of the email accounts registered in the mobile phone's system.		
Trigger:	The User touches the "send" icon.		
Preconditions:	PRE - 1 : The Smartphone OS is configured with a valid email account. PRE - 2 : An archive exists that is the current active archive. PRE - 3 : The User is in the Archive Detail View.		
Postconditions:	None		
Normal Flow:	**1.0 Email Archive** 1. The user clicks the "send" button. 2. The MDApp provides the user with the smartphone systems standard email composer. 3. The MDApp sends a request to the smartphone's email system with the archive data as an attachment. 4. The MDA		
Priority:	Medium		

Figure 8.9: UC-8

8.2 Software Requirements Specification

8.2.1 Introduction

8.2.1.1 Purpose

The Software Requirements Specification will describe the functional and nonfunctional requirements of the Melanoma Detection App. It is meant to be a guideline for developers who will be implementing the mobile application.

8.2.2 Overall Description

8.2.2.1 Product Perspective

The Melanoma Detect App will provide the user with a risk assessment of a skin lesion. The App will guide the user through the process of capturing an image and confirming the correct recognition of the boundary of the lesion. Once confirmed the App will analyse the visual features of the lesion based on dermatological rules and provide the user with an initial risk assessment.

The App should help motivate the user to make an appointment with a dermatologist when a significant risk has been detected. It is not meant to replace the need to visit a dermatologist though.

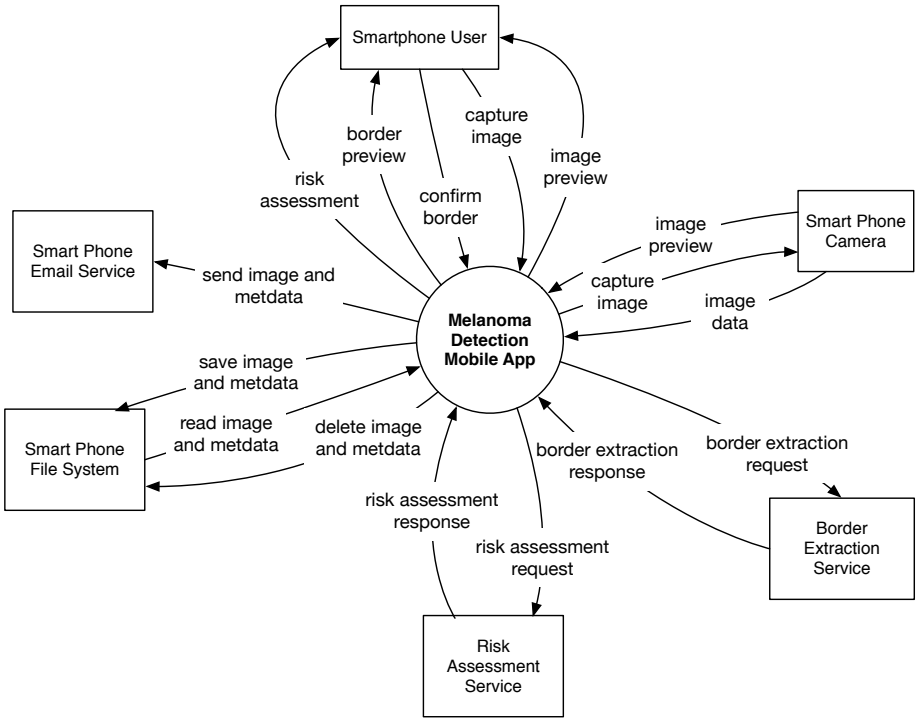

Figure 8.10: Context Diagram of the Melanoma Detection App

8.2.2.2 User Classes and Characteristics

The Melanoma Detection App only has one class of user, the Smartphone User. The Smartphone User has one or several skin lesions that she or he is worried about. The Smartphone User is not expected to have any technical or dermatological expertise.

8.2.2.3 Operating Environment

OE-1 : The Mobile App shall operate on the following platforms: Android OS (v.5.2.2) and iOS (v. 9.3)

8.2.2.4 Design and Implementation Constraints

CO-1 : The Border Extraction and Risk Assessment Services shall be implemented on a server in oder to leverage the effort already made with python and python based libraries for image processing and scientific calculations. Wifi access is therefore a requirement for the app to run.

CO-2 : The method for archiving images and risk assessment data will use whatever standard is the most accepted for the relevant platform. i.e. iCloud on iOS.

8.2.2.5 Assumptions and Dependencies

AS-1 : It is assumed that the user of the software is also the user of the mobile phone. The software might need to request permission from the user to access the camera or network.

8.2.3 System Features

8.2.3.1 Capture Image

Description:	The MDApp allows the Smartphone User to capture an image of a skin lesion.

Functional Requirements:

Image.Capture	**Capture Image using Camera**
.Preview	The App must provide the User with a realtime preview of the camera input. It will include some visual indicators that help the user capture an optimal image.
.Trigger	The User will trigger the capture when he or she decides the realtime preview is optimal.
.Response	The App must be able to capture a still image from the camera when the User pushed the capture button. The image will be saved in a standard image format on the phone's file system.
.Display	The App will present the image to the user on the device's screen.

8.2.3.2 Extract Border

Description:	In oder to be able to properly analyse the skin lesion image, the MDApp must be able to distinguish pixels that belong to the lesion from pixels that belong to the healthy skin area surrounding the lesion.
Functional Requirements:	
Border.Extract	**Calculate Border of Skin Lesion**
.Calculate	The App sends a captured image as a request to the Border Extraction Service. When the service has completed the calculation the results of the border calculation are provided by the service to the App.
.Display	The App will present the results or the border calculation to the User on the device's screen.
.Promt	The App will prompt the User to confirm that the border was precisely calculated.
.Response	The User can confirm or cancel the border.

8.2.4 Data Requirements

8.2.4.1 Logical Data Model

8.2.4.2 Data Dictionary

8.2.4.3 Reports

Chapter 9

Architecture Design

In section 2.2.1 categories of medical smart phone apps were defined. Apps in the same categories might have a similar feature set and capabilities. An app in the *Alert / First Response* category might have a database that stores important contacts in the user's area and a user interface that allows the user to quickly alert a contact in an emergency situation. An app in the *Learning / Educational / Reference* category might be modelled as a quiz app which presents the user with a question and several possible answers to choose from. Answers from several questions will can be evaluated and a score is presented to the user at the end of a session. The app would be implemented with a database of questions and weighted answers, as well as an interface that presents the questions to the user sequentially.

One of the goals of this project is to define requirements for an app that provides a risk assessment of a lesion being benign or malignant. A high level description of the primary function of the app is the following:

> The smart phone app allows the user to capture and analyse an image of a skin lesion and provide a risk assessment to the user of the lesion being a malignant melanoma.

The user might also like to save the image and results in oder to be able to compare it with other assessments in the future, or to review the assessment with a dermatologist. For comparison it might be useful to save associated metadata, such as the date and location on the body of the lesion. For convenience it might be nice to send the assessment and image via email to a specialist for review. Secondary functionality can be described as follows:

> The app allows the user to save or archive the image and corresponding assessment for future comparison and review.
>
> The user can add, edit, and save metadata associated with the image of the lesion.
>
> The user can browse archived images, assessments, and associated metadata.

The user can send a set of images with associated assessment and metadata via email.

From this high level description some basic assumptions about the app's architecture can already be made. The app will require access to a database that can store information about an image, results of the analysis, and metadata. The app will require a user interface that allows a user to browse and edit data associated with an image. The app requires access to the smart phone's camera api in order to capture images. Another consideration that will impact design decisions is the investment already made in developing the image processing and analysis algorithms. Choosing python as the basis for the numerical algorithms and leveraging python based scientific computing libraries has implications because python code is not easily portable to iOS or Android devices. In order to efficiently utilise the algorithms as they are, and to continue to update and improve them, the image processing and analysis algorithms should be implemented as online services.

The image processing and risk assessment algorithms will be implemented as online services.

In order to formally elicit the requirements the hight level description will be broken down into structured use cases. A set of requirements will be extracted from the use cases.

9.1 User Interface

Figure 9.1: Image Capture View

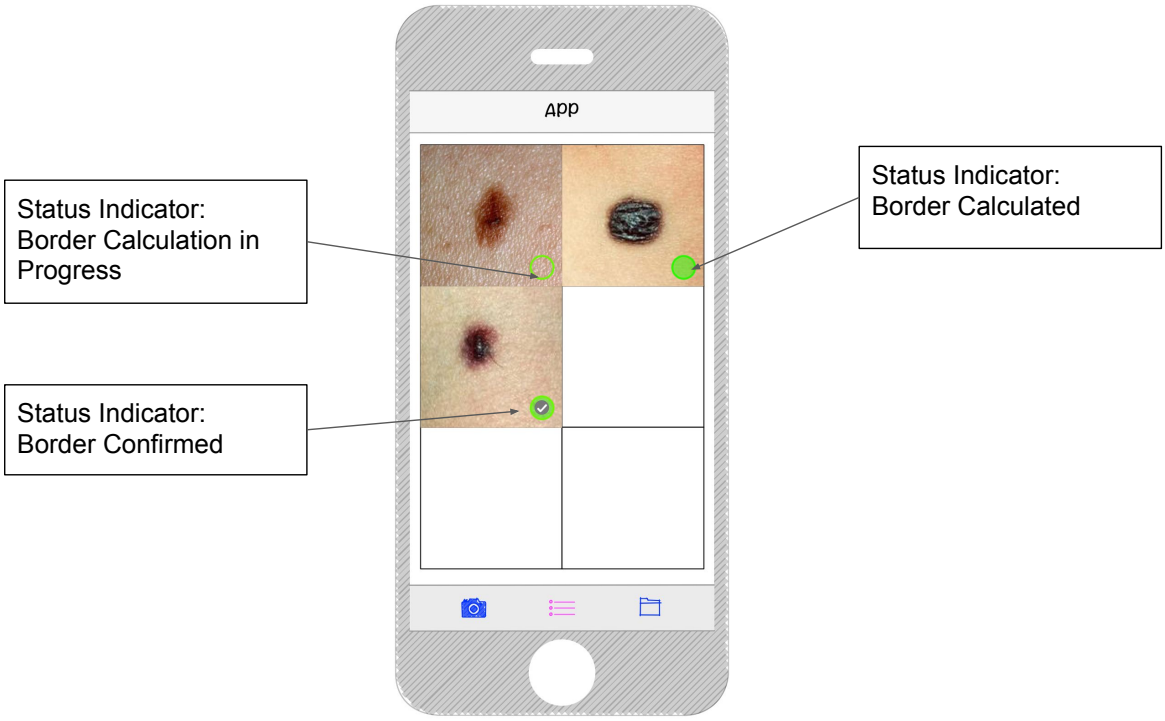

Figure 9.2: Image List View

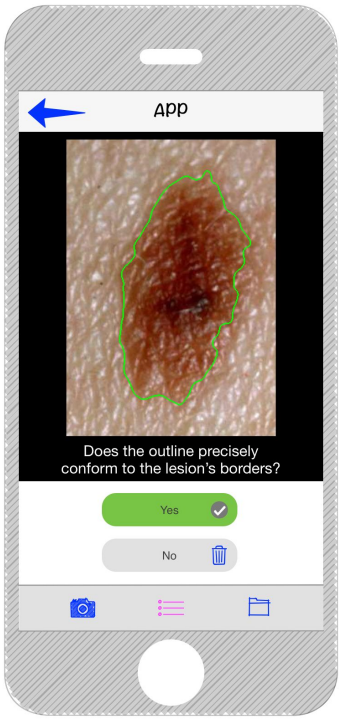

Figure 9.3: Border Confirm View

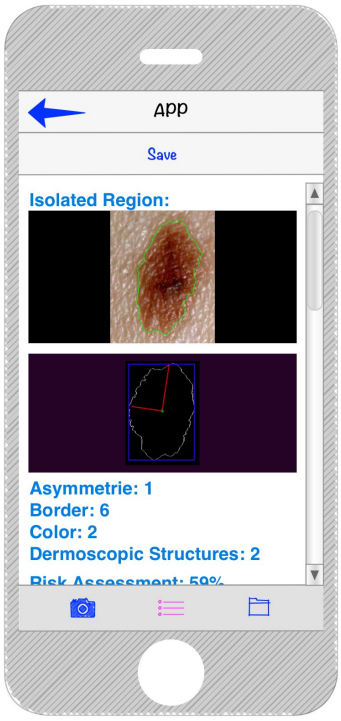

Figure 9.4: Image Detail View

Figure 9.5: Archive View

Figure 9.6: Archive Detail View

9.2 Data Structure

9.2.1 Server Data Structure

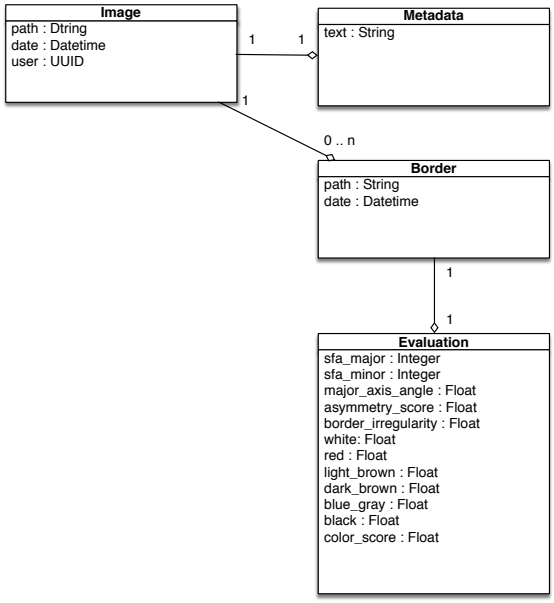

Figure 9.7: Serverside Data Structure

9.2.2 Client Data Structure

9.3 Software Architecture

9.3.1 MVC

Since the 1970s the Model View Controller (MVC) pattern is the standard architectural design pattern for applications that present the user with a graphical user interface. It was developed out of a need for modularity, to encapsulate responsibility of specific concepts to separate program modules, or Separation of Concerns. MVC identifies three main components that program code should be grouped into, namely [15]:

Model: The representation of some object of knowledge, encapsulates code managing the associated data and behaviour (business-logic).

View: The visual representation of the model. The view can feature or hide aspects of the model and thus act as a presentation filter. The view observe the model for changes and update the presentation accordingly.

Controller: The controller allows the user to interact with the model. It allows the user to trigger behaviours implemented in the Model.

In the classic MVC pattern the model does not ”know” about the view or the controller. And the controller does not effect the view. Instead, both the view and controller monitor the model using an observer mechanism and synchronise themselves when updates to the model occur.

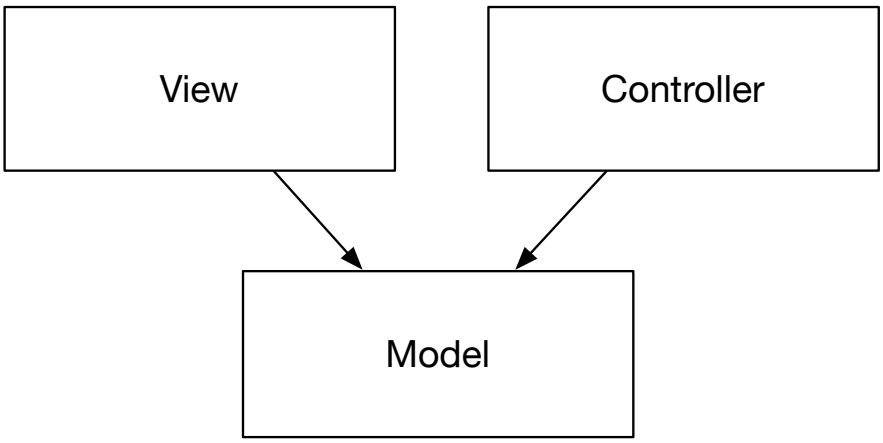

Figure 9.8: Classic MVC

9.3.2 Modern MVC

Modern MVC has evolved from being a software design pattern which handles components of an application to an architectural design pattern that defines the structure of an application itself. It has many similarities to the Layer Architecture. The responsibilities of each layer are slightly different to the typical presentation, business, and persistence layer definitions.

Most modern web frameworks such as Ruby on Rails, Symfony or the IOS environment refer to themselves as MVC based frameworks. The modern MVC concept has changed slightly. The component definitions are the same, but some responsibilities have shifted. Modern MVC strictly separates the model from the view. All modern frameworks state that the view should have as little logic as possible. Any logic implemented in the view should only be relevant to presentation. The view components should not directly reference the model components [5] [1].

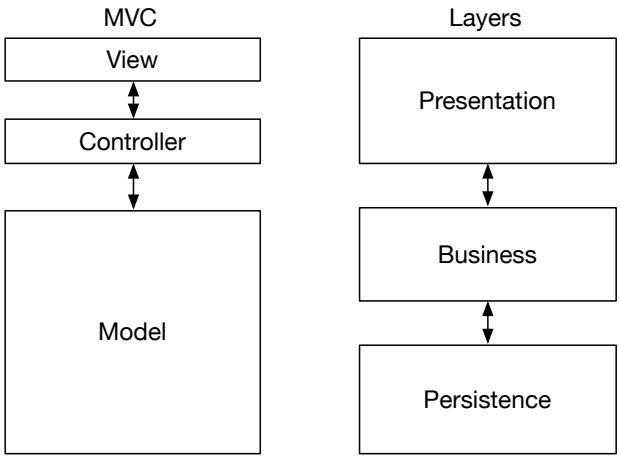

Figure 9.9: Modern MVC vs 3-Tier Layered Architecture

This division of responsibility has the added benefit of increased testability. GUIs are difficult to test, by removing as much logic as possible from the user interface there is less necessity to test it. The controller and model components can more easily be tested in isolation using normal unit tests [8].

9.3.3 MVC Derivatives

Many MVC derivatives exist. The main differences are where the division of responsibility is made and how it is labeled. MVVM defines a view model instead of a controller. The view model acts as a facade around the model and introduces a data binder element that is responsible for keeping the view and view model synchronised. The MVT, calls the view a template and the controller a view. The slight difference is that the template is basically a static file, with no logic, and placeholders for the data. The Django web framework uses this model, but there is little difference to the other MVC derivatives such as MVP (model view presenter). The AngularJS web framework attempts to end the discussion of which design to follow by labelling itself a MVW (Model View Whatever) framework [10].

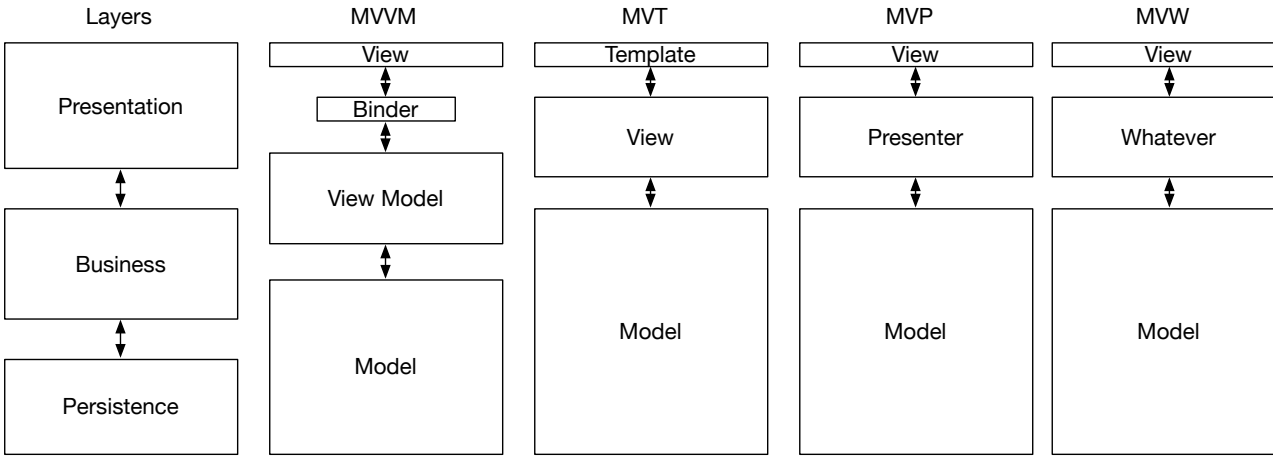

Figure 9.10: MVC Derivatives in relation to 3-Tier Layer

9.3.4 MVC and Smart Phone Applications

Although most frameworks and environments targeted toward mobile application development use concepts from MVC, it can be difficult to define strict devision of responsibility, especially in conjunction with services provided from an online server. Often the responsibilities of the controller are reimplemented server side, some model management might occur in the app. Server and application code are developed using different languages and often most likely different developers. One solution is is to develop the app as a thin client, where it basically becomes the view component of MVC and the controllers and model are implemented on the server. Taken to the extreme, the app runs in a standard web browser and server provides the view components that emulate a native mobile user interface. This is referred to as a web app. Any logic in the view is implemented using javascript. Hardware support is limited to what the mobile browser provides access to. In order to extend hardware support a hybrid approach can be used in which the app implements a custom browser view which is extended with native capabilities. Here the division of responsibilities as defined by MVC become fuzzy. A fully native app working in conjunction with a web server will not conform well to the definitions of MVC.

9.3.5 Considerations

From the prioritised requirements defined above two important issues have an impact on architectural design decisions. Cross platform integration and extendability. The image processing and risk assessment algorithms are a complex combination of existing python based libraries and custom code. The rational behind choosing python as the basis is explained in detail in previous chapters. The disadvantage of choosing python for the basis of a mobile application is that is it not easily deployable on

mobile hardware. Regardless of the decision to use python, the complexity of these algorithms would make it difficult and costly to implement in a cross platform manner in any language.

The solution is to offload the image processing and risk assessment algorithms to an online server running a python environment. The mobile application can interface with the online server in such a way that the process is transparent to the user.

The requirement for extendability and maintenance (updatability) becomes much easier as well. Updating the complex algorithms in multiple languages on multiple platforms and deploying updates to all existing users via the iTunes and Android shop platforms is not necessary. Instead only the server code needs to be updated. In the best case the phone-side app will not have to be updated at all.

Still the question remains, what architecture design pattern best covers the requirements of a mobile app with online server integration.

9.3.6 VIPER

VIPER is a new software architecture pattern which extends MVC with some addition concepts that make it more adaptable to mobile applications, especially when they are extended with online server based services. The classic controller is split into a presenter and a controller, the model component is split into a central data manager communicating with many services and entities [12].

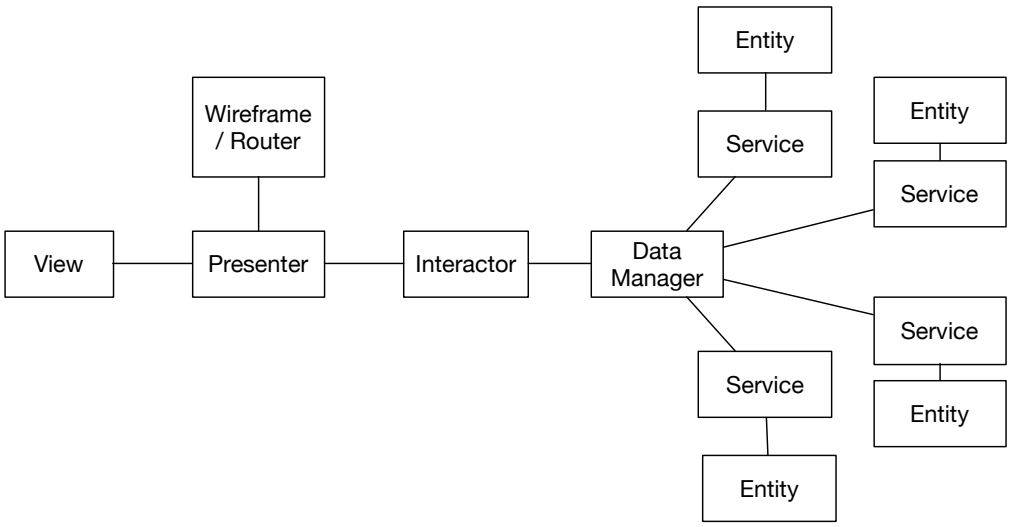

Figure 9.11: VIPER Overview

View: The view corresponds to the view as defined in modern MVC definitions, it is a slim component with as little logic as possible. It presents the current state of the models and provides “widgets” with which the user can interact.

Presenter: The presenter passes data to the view and handles events from the view. The presenter might perform some basic validation, in the case of a user sign-up scenario, for example, the presenter might validate that a user's email is indeed formatted as a correct email, it will not validate if the email has already been used by someone else.

Interactor: Business logic is handled by the interactor. Most a what the model component in MVC was responsible for is handled here. The interactor however does not know anything about data storage, databases, or persistence. It does not know if data is local or accessible via a network.

Data Manager: To the Interactor the data manager looks like a database, with the difference that the it knows how to notify the interactor when data is available. The data manager will receive requests from the interactor, check it's local cache, perform external service requests if necessary, then pass data back to the interactor once available.

Service: Services are encapsulated processes that can be online or local. The service knows how to fetch or store data, it might be a local sqlite database, or it can encapsulate the communication to an online database via a REST-API.

Entity: The representation of the actual data. As the entity get passed through the system it might be in the form of a database entry, a json object, or data object.

Wireframe / Router: The wireframe initialises all the other classes and handles the transitions to other views in the app. The wireframe handles the current "state" of the app and might implement a "history" allowing the user to transition back to a previous view with the associated other components and data.

9.4 Mobile Development Strategies

9.4.1 Pure Native vs Hybrid Native vs Hybrid vs Web Apps

The mobile app market is segmented into several platforms. Each platform has it's own framework, API, and language specifications and requirements. Android apps are written in java and access the Android apis, iOS apps in Objective-C or Swift and interface with the iOS frameworks. Android and iOS cover about 94% of the worldwide mobile market [2]. A developer must de decide which platforms are priorities, and how might programming effort be consolidated across multiple platforms. Several strategies exist, and there are many tool kits and frameworks that can help accelerate development.

Web Apps: The simplest development strategy is to develop an HTML based web page that is disguised to appear and behave like a mobile app. The web app runs in a normal browser, is deployed from a standard web server. Model mobile web browsers offer javascript APIs that can access various other hardware components of the smart phone such as the compass, GPS/geolocation services, accelerometer, and the camera. Other services will remain inaccessible.

Hybrid: A similar strategy is to develop using javascript and html, but to package these elements into a "hollow" native app that is little else than a native view component with an embedded web browser. The web browser url is locked to the internal html files and appears to the user as a mobile app. though the user interface might slightly different that a pure native app. The advantage of a hybrid strategy is that the app can be distributed via the standard app stores increasing it's marketability. It can also be extended with native components that might otherwise not be accessible to to a web app such as internal file storage and native data storage mechanisms.

Pure Native: The developer uses the development environment as provided by the phone manufacturer and has full access to all of the native hardware APIs, development kits and tools. The performance and responsiveness of a pure native app is much better compared to web and hybrid solutions. The user interface is build with the standard UI tools allowing a better user experience because the user is presented with familiar UI elements.

Hybrid Native: Recently several cross platform application frameworks and emerged that allow the developer in a single language that can be compiled across several platforms. Qt and Juce are examples of application frameworks that allow cross platform development in C++, Xamarin uses C#.

The criteria to compare these strategies are the following:

Development Effort : Judging the development effort is subjective. It depends greatly on the experience of the developers involved. An expert Objective-C developer with years of experience developing IOS applications will have no problem developing a native IOS app, but might be stumped when it comes to hybrid development. For this project we will assume Javascript and HTML development requires the least amount of effort and thus give high scores to Web Apps and Hybrid development.

Code Portability: How much of the code developed can be reused on each target platform without any extra effort.

Camera Access: Does a strategy allow access to all the device's camera,? is the camera view customisable with guidelines and information that can help the user take a high quality picture?

Look and Feel: Does a strategy offer pure native UI elements that conform with user expectations for a particular device?

Deployment: What effort is required to deploy the application to a device?

Performance: What is the performance impact of a chosen strategy?

For comparison we rank each criteria per strategy either 2 = benefit, 1 = no benefit, 0 = negative benefit.

	Weight	Web App	Hybrid	Native	Hybrid Native
Development Effort	0,3	2	2	0	1
Code Portability	0,3	2	2	0	1
Look & Feel	0,025	1	1	2	2
Deployment	0,05	2	0	0	0
Performance	0,025	0	1	2	2
Camera Access	0,3	1	2	2	1
Total	1	1,625	1,85	0,7	1

Figure 9.12: Mobile Development Strategies Comparison

Chapter 10

Testing

10.1 Section Title

Chapter 11

Application Implementation

11.1 Language and Libraries

11.1.1 Server

11.1.2 Mobile Client

Chapter 12

Conclusions

12.1 Section Title

Bibliography

[1] Symfony Book: Chapter 2 - Exploring Symfony's Code, The MVC Pattern, 2015. [Online; accessed 2016-8-27], link.

[2] Mobile/Tablet Operating System Market Share, 2016. [Online; accessed 2016-8-28], link.

[3] Skin Cancer Overview — Patient Version, 2016. [Online; accessed 2016-9-2], link.

[4] Jose Fernandez Alcon, Calina Ciuhu, Warner ten Kate, Adrienne Heinrich, Natallia Uzunbajakava, Gertruud Krekels, Denny Siem, and Gerard de Haan. Automatic Imaging System With Decision Support for Inspection of Pigmented Skin Lesions and Melanoma Diagnosis. *IEEE*, 3(1):14–25, feb 2009.

[5] Apple. Model-View-Controller, 2015. [Online; accessed 2016-8-27], link.

[6] Ann Chang Brewer, Dawnielle C. Endly, Jill Henley, Mahsa Amir, Blake P. Sampson, Jacqueline F. Moreau, and Robert P. Dellavalle. Mobile Applications in Dermatology. *JAMA Dermatol*, 149(11):1300, nov 2013.

[7] Mercedes Filho, Zhen Ma, and João Manuel R. S. Tavares. A review of the quantification and classification of pigmented skin lesions: From dedicated to hand-held devices. *Journal of Medical Systems*, 39(11):1–12, 2015.

[8] Martin Fowler. GUI Architectures, Humble View, 2006. [Online; accessed 2016-8-28], link.

[9] A.A. Marghoob, J. Malvehy, R.P. Braun, and A.W. Kopf. *An Atlas of Dermoscopy*. Encyclopedia of Visual Medicine. CRC Press, 2004.

[10] Igor Minar. MVC vs MVVM vs MVP, 2012. [Online; accessed 2016-8-28], link.

[11] J. Premaladha and K.S. Ravichandr. Asymmetry Analysis of Malignant Melanoma Using Image Processing: A Survey. *J. of Artificial Intelligence*, 7(2):45–53, feb 2014.

[12] Ryan Quan. Brigade's Experience Using an MVC Alternative, 2014. [Online; accessed 2016-8-28], link.

[13] Sophia.M Mallikarjun Mundas Vidya.R Siddiq Iqbal, Divyashree.J.A. Implementation of Stolz's Algorithm for Melanoma Detection. *International Advanced Research Journal in Science, Engineering and Technology IARJSET*, 2(6), jun 2015.

[14] Aurora Sáez, Begoña Acha, and Carmen Serrano. Pattern Analysis in Dermoscopic Images. In *Series in BioEngineering*, pages 23–48. Springer Science & Business Media, sep 2013.

[15] Stephen Walther. The Evolution of MVC, 2016. [Online; accessed 2016-8-27], link.

[16] Ulrike Weigert, Walter H.C. Burgdorf, and Wilhelm Stolz. ABCD Rule. In *Atlas of Dermoscopy*, page 116. Informa Healthcare, aug 2012.

[17] Karl Wiegers. *Software requirements*. Microsoft, Redmond, Washington, 2013.

Chapter 13

Appendix

13.1 Optimizations

13.1.1 Border Extraction

13.1.2 SFA Threshold

13.1.3 TDS Evaluation

Name	Source	Category	Asymmetry	Border	Color	TDS
B1052.png	Dermofit	Malignant Melanoma	2	4	3.0	4.50
B287.png	Dermofit	Malignant Melanoma	2	5	2.0	4.10
B302.png	Dermofit	Malignant Melanoma	0	3	2.0	1.30
B314.png	Dermofit	Malignant Melanoma	2	6	2.0	4.20
B65.png	Dermofit	Malignant Melanoma	2	4	3.0	4.50
C158.png	Dermofit	Malignant Melanoma	0	2	3.0	1.70
C201.png	Dermofit	Malignant Melanoma	2	4	3.0	4.50
C263a.png	Dermofit	Malignant Melanoma	2	4	3.0	4.50
C311b.png	Dermofit	Malignant Melanoma	2	8	3.0	4.90
C359.png	Dermofit	Malignant Melanoma	1	3	3.0	3.10
D143.png	Dermofit	Malignant Melanoma	0	1	3.0	1.60
D39.png	Dermofit	Malignant Melanoma	2	3	3.0	4.40
D630.png	Dermofit	Malignant Melanoma	0	2	1.0	0.70
D678.png	Dermofit	Malignant Melanoma	0	0	3.0	1.50
T233a.png	Dermofit	Malignant Melanoma	1	8	2.0	3.10
T86b.png	Dermofit	Malignant Melanoma	1	3	3.0	3.10
A121a.png	Dermofit	Melanocytic Nevus	0	3	3.0	1.80
A21b.png	Dermofit	Melanocytic Nevus	0	2	2.0	1.20
A2b.png	Dermofit	Melanocytic Nevus	1	2	2.0	2.50
A8c.png	Dermofit	Melanocytic Nevus	2	4	2.0	4.00
A8d.png	Dermofit	Melanocytic Nevus	1	2	1.0	2.00
A8e.png	Dermofit	Melanocytic Nevus	0	1	2.0	1.10

B125c.png	Dermofit	Melanocytic Nevus	1	2	2.0	2.50
B157.png	Dermofit	Melanocytic Nevus	1	2	3.0	3.00
B17a.png	Dermofit	Melanocytic Nevus	1	2	3.0	3.00
B17b.png	Dermofit	Melanocytic Nevus	1	1	2.0	2.40
B17c.png	Dermofit	Melanocytic Nevus	1	2	1.0	2.00
B17f.png	Dermofit	Melanocytic Nevus	1	3	2.0	2.60
B197d.png	Dermofit	Melanocytic Nevus	1	0	1.0	1.80
B202b.png	Dermofit	Melanocytic Nevus	1	0	3.0	2.80
B293b.png	Dermofit	Melanocytic Nevus	0	1	1.0	0.60
B293d.png	Dermofit	Melanocytic Nevus	0	1	1.0	0.60
B293e.png	Dermofit	Melanocytic Nevus	0	2	2.0	1.20
B311c.png	Dermofit	Melanocytic Nevus	1	4	1.0	2.20
B350a.png	Dermofit	Melanocytic Nevus	0	0	1.0	0.50
B355b.png	Dermofit	Melanocytic Nevus	2	0	1.0	3.10
B356.png	Dermofit	Melanocytic Nevus	1	2	2.0	2.50
B361a.png	Dermofit	Melanocytic Nevus	0	1	1.0	0.60
B379a.png	Dermofit	Melanocytic Nevus	0	0	2.0	1.00
B379b.png	Dermofit	Melanocytic Nevus	1	1	2.0	2.40
B447a.png	Dermofit	Melanocytic Nevus	2	8	2.0	4.40
B447b.png	Dermofit	Melanocytic Nevus	1	3	2.0	2.60
B472b.png	Dermofit	Melanocytic Nevus	0	2	1.0	0.70
B508.png	Dermofit	Melanocytic Nevus	2	5	2.0	4.10
B522b.png	Dermofit	Melanocytic Nevus	0	0	1.0	0.50
B52a.png	Dermofit	Melanocytic Nevus	1	2	2.0	2.50
B52c.png	Dermofit	Melanocytic Nevus	0	4	2.0	1.40
B543.png	Dermofit	Melanocytic Nevus	0	0	2.0	1.00
B549c.png	Dermofit	Melanocytic Nevus	1	1	2.0	2.40
B574.png	Dermofit	Melanocytic Nevus	1	0	2.0	2.30
B598a.png	Dermofit	Melanocytic Nevus	1	1	2.0	2.40
B612b.png	Dermofit	Melanocytic Nevus	1	2	1.0	2.00
B654c.png	Dermofit	Melanocytic Nevus	0	1	2.0	1.10
B66c.png	Dermofit	Melanocytic Nevus	0	3	2.0	1.30
B676a.png	Dermofit	Melanocytic Nevus	0	3	2.0	1.30
B676b.png	Dermofit	Melanocytic Nevus	1	4	3.0	3.20
B69a.png	Dermofit	Melanocytic Nevus	1	2	3.0	3.00
B69b.png	Dermofit	Melanocytic Nevus	2	2	2.0	3.80
B89e.png	Dermofit	Melanocytic Nevus	2	1	1.0	3.20
B91a.png	Dermofit	Melanocytic Nevus	0	0	2.0	1.00
B91b.png	Dermofit	Melanocytic Nevus	1	6	3.0	3.40
B91c.png	Dermofit	Melanocytic Nevus	2	0	2.0	3.60
D144.png	Dermofit	Melanocytic Nevus	2	3	3.0	4.40
D176b.png	Dermofit	Melanocytic Nevus	0	0	2.0	1.00
D239a.png	Dermofit	Melanocytic Nevus	1	1	2.0	2.40
D239b.png	Dermofit	Melanocytic Nevus	0	0	1.0	0.50
D271a.png	Dermofit	Melanocytic Nevus	1	1	2.0	2.40
D291b.png	Dermofit	Melanocytic Nevus	0	2	2.0	1.20

D339.png	Dermofit	Melanocytic Nevus	1	2	2.0	2.50
D374.png	Dermofit	Melanocytic Nevus	0	0	2.0	1.00
D384.png	Dermofit	Melanocytic Nevus	0	7	2.0	1.70
D395.png	Dermofit	Melanocytic Nevus	0	0	3.0	1.50
D404.png	Dermofit	Melanocytic Nevus	0	0	2.0	1.00
D426.png	Dermofit	Melanocytic Nevus	0	0	2.0	1.00
D427b.png	Dermofit	Melanocytic Nevus	0	1	2.0	1.10
D492.png	Dermofit	Melanocytic Nevus	2	1	2.0	3.70
D526b.png	Dermofit	Melanocytic Nevus	0	2	2.0	1.20
D567b.png	Dermofit	Melanocytic Nevus	2	2	1.0	3.30
D626.png	Dermofit	Melanocytic Nevus	0	2	3.0	1.70
D715.png	Dermofit	Melanocytic Nevus	0	0	1.0	0.50
D722.png	Dermofit	Melanocytic Nevus	2	6	2.0	4.20
D723a.png	Dermofit	Melanocytic Nevus	1	2	2.0	2.50
D726d.png	Dermofit	Melanocytic Nevus	0	7	2.0	1.70
D726e.png	Dermofit	Melanocytic Nevus	1	3	2.0	2.60
P103a.png	Dermofit	Melanocytic Nevus	0	3	2.0	1.30
P126b.png	Dermofit	Melanocytic Nevus	2	1	1.0	3.20
P144.png	Dermofit	Melanocytic Nevus	0	1	1.0	0.60
P18.png	Dermofit	Melanocytic Nevus	1	1	2.0	2.40
P196.png	Dermofit	Melanocytic Nevus	2	3	2.0	3.90
P199.png	Dermofit	Melanocytic Nevus	1	2	1.0	2.00
P2.png	Dermofit	Melanocytic Nevus	0	3	1.0	0.80
P237b.png	Dermofit	Melanocytic Nevus	1	6	2.0	2.90
P256a.png	Dermofit	Melanocytic Nevus	0	0	2.0	1.00
P271f.png	Dermofit	Melanocytic Nevus	2	2	2.0	3.80
P277b.png	Dermofit	Melanocytic Nevus	1	1	2.0	2.40
P291.png	Dermofit	Melanocytic Nevus	0	0	3.0	1.50
P304a.png	Dermofit	Melanocytic Nevus	1	0	2.0	2.30
P306a.png	Dermofit	Melanocytic Nevus	1	0	3.0	2.80
P306c.png	Dermofit	Melanocytic Nevus	0	0	3.0	1.50
P337a.png	Dermofit	Melanocytic Nevus	0	0	3.0	1.50
P337b.png	Dermofit	Melanocytic Nevus	0	1	3.0	1.60
P337d.png	Dermofit	Melanocytic Nevus	0	0	2.0	1.00
P337e.png	Dermofit	Melanocytic Nevus	1	1	1.0	1.90
P354a.png	Dermofit	Melanocytic Nevus	0	1	2.0	1.10
P359b.png	Dermofit	Melanocytic Nevus	0	1	2.0	1.10
P365d.png	Dermofit	Melanocytic Nevus	0	1	2.0	1.10
P365e.png	Dermofit	Melanocytic Nevus	1	1	2.0	2.40
P376b.png	Dermofit	Melanocytic Nevus	1	1	2.0	2.40
P376d.png	Dermofit	Melanocytic Nevus	2	5	1.0	3.60
P382a.png	Dermofit	Melanocytic Nevus	0	1	2.0	1.10
P384b.png	Dermofit	Melanocytic Nevus	1	2	2.0	2.50
P384c.png	Dermofit	Melanocytic Nevus	1	7	2.0	3.00
P392.png	Dermofit	Melanocytic Nevus	0	1	3.0	1.60
P399.png	Dermofit	Melanocytic Nevus	1	1	2.0	2.40

P404a.png	Dermofit	Melanocytic Nevus	1	0	2.0	2.30
P404c.png	Dermofit	Melanocytic Nevus	0	0	2.0	1.00
P407b.png	Dermofit	Melanocytic Nevus	0	2	2.0	1.20
P407c.png	Dermofit	Melanocytic Nevus	0	1	2.0	1.10
P432.png	Dermofit	Melanocytic Nevus	2	4	2.0	4.00
P435b.png	Dermofit	Melanocytic Nevus	0	3	3.0	1.80
P454a.png	Dermofit	Melanocytic Nevus	1	1	3.0	2.90
P454b.png	Dermofit	Melanocytic Nevus	0	0	1.0	0.50
P45a.png	Dermofit	Melanocytic Nevus	2	1	3.0	4.20
P45b.png	Dermofit	Melanocytic Nevus	1	5	2.0	2.80
P49.png	Dermofit	Melanocytic Nevus	0	0	3.0	1.50
P505c.png	Dermofit	Melanocytic Nevus	2	1	2.0	3.70
P505e.png	Dermofit	Melanocytic Nevus	1	7	2.0	3.00
P509c.png	Dermofit	Melanocytic Nevus	1	1	2.0	2.40
P53b.png	Dermofit	Melanocytic Nevus	0	6	1.0	1.10
P56b.png	Dermofit	Melanocytic Nevus	0	2	3.0	1.70
P58.png	Dermofit	Melanocytic Nevus	0	3	2.0	1.30
P63.png	Dermofit	Melanocytic Nevus	1	6	1.0	2.40
P88b.png	Dermofit	Melanocytic Nevus	0	0	2.0	1.00

Table 13.1: Results of TDS calculation for Dermfit

Name	Source	Category	Asymmetry	Border	Color	TDS
012383HB.jpeg	DermQuest	Benign Keratosis	2	5	3.0	4.60
012824HB.JPG	DermQuest	Benign Keratosis	0	2	3.0	1.70
019462HB.JPG	DermQuest	Malignant Melanoma	2	4	3.0	4.50
019475HB.JPG	DermQuest	Malignant Melanoma	1	2	3.0	3.00
019523HB.JPG	DermQuest	Malignant Melanoma	2	2	6.0	5.80
019563HB.JPG	DermQuest	Malignant Melanoma	2	1	4.0	4.70
019578HB.JPG	DermQuest	Malignant Melanoma	2	2	2.0	3.80
019681HB.JPG	DermQuest	Malignant Melanoma	1	4	3.0	3.20
019682HB.JPG	DermQuest	Malignant Melanoma	1	5	3.0	3.30
019725HB.JPG	DermQuest	Malignant Melanoma	0	2	4.0	2.20
019736HB.JPG	DermQuest	Malignant Melanoma	1	1	3.0	2.90
020192HB.JPG	DermQuest	Malignant Melanoma	1	5	4.0	3.80
020250HB.JPG	DermQuest	Malignant Melanoma	2	3	3.0	4.40
020266HB.JPG	DermQuest	Malignant Melanoma	1	0	3.0	2.80
020268HB.JPG	DermQuest	Malignant Melanoma	1	1	4.0	3.40
020302HB.JPG	DermQuest	Malignant Melanoma	2	8	4.0	5.40
020319HB.JPG	DermQuest	Malignant Melanoma	1	2	5.0	4.00
044936HB.JPG	DermQuest	Malignant Melanoma	2	3	4.0	4.90
005613HB.jpeg	DermQuest	Melanocytic Nevus	0	3	3.0	1.80
005728HB.jpeg	DermQuest	Melanocytic Nevus	2	3	4.0	4.90
005859HB.jpeg	DermQuest	Melanocytic Nevus	0	0	3.0	1.50
005900HB.jpeg	DermQuest	Melanocytic Nevus	0	3	5.0	2.80

006027HB.jpeg	DermQuest	Melanocytic Nevus	2	8	3.0	4.90
006074HB.jpeg	DermQuest	Melanocytic Nevus	1	6	3.0	3.40
006095HB.jpeg	DermQuest	Melanocytic Nevus	0	1	4.0	2.10
010769HB.jpeg	DermQuest	Melanocytic Nevus	2	6	3.0	4.70
010992VB.jpeg	DermQuest	Melanocytic Nevus	0	0	3.0	1.50
011031HB.jpeg	DermQuest	Melanocytic Nevus	0	0	4.0	2.00
011040VB.jpeg	DermQuest	Melanocytic Nevus	0	0	3.0	1.50
011201HB.jpeg	DermQuest	Melanocytic Nevus	1	0	3.0	2.80
011548HB.jpeg	DermQuest	Melanocytic Nevus	2	1	3.0	4.20
011697HB.jpeg	DermQuest	Melanocytic Nevus	1	4	4.0	3.70
011697HB.JPG	DermQuest	Melanocytic Nevus	1	4	4.0	3.70
011936HB.jpeg	DermQuest	Melanocytic Nevus	1	6	3.0	3.40
011936HB.JPG	DermQuest	Melanocytic Nevus	1	6	3.0	3.40
021733HB.jpeg	DermQuest	Melanocytic Nevus	1	3	1.0	2.10
021835HB.jpeg	DermQuest	Melanocytic Nevus	2	2	4.0	4.80
021837HB.jpeg	DermQuest	Melanocytic Nevus	1	0	4.0	3.30
021848HB.jpeg	DermQuest	Melanocytic Nevus	2	7	3.0	4.80
021852HB.jpeg	DermQuest	Melanocytic Nevus	0	0	3.0	1.50
021869HB.jpeg	DermQuest	Melanocytic Nevus	1	0	4.0	3.30
021878HB.jpeg	DermQuest	Melanocytic Nevus	2	1	3.0	4.20
021902HB.jpeg	DermQuest	Melanocytic Nevus	0	4	3.0	1.90
045720HB.jpeg	DermQuest	Melanocytic Nevus	0	1	3.0	1.60
046342HB.jpeg	DermQuest	Melanocytic Nevus	2	1	3.0	4.20
046343HB.JPG	DermQuest	Melanocytic Nevus	1	6	4.0	3.90

Table 13.2: Results of TDS calculation for DermQuest

Name	Source	Category	Asymmetry	Border	Color	TDS
IMD015.bmp	PH2Dataset	Melanocytic Nevus	1	5	2.0	2.80
IMD016.bmp	PH2Dataset	Melanocytic Nevus	1	2	2.0	2.50
IMD018.bmp	PH2Dataset	Melanocytic Nevus	1	1	2.0	2.40
IMD020.bmp	PH2Dataset	Melanocytic Nevus	0	1	2.0	1.10
IMD027.bmp	PH2Dataset	Melanocytic Nevus	2	7	2.0	4.30
IMD038.bmp	PH2Dataset	Melanocytic Nevus	1	2	3.0	3.00
IMD039.bmp	PH2Dataset	Melanocytic Nevus	1	3	2.0	2.60
IMD043.bmp	PH2Dataset	Melanocytic Nevus	2	7	3.0	4.80
IMD045.bmp	PH2Dataset	Melanocytic Nevus	1	1	2.0	2.40
IMD050.bmp	PH2Dataset	Melanocytic Nevus	1	6	2.0	2.90
IMD078.bmp	PH2Dataset	Melanocytic Nevus	2	8	3.0	4.90
IMD103.bmp	PH2Dataset	Melanocytic Nevus	1	5	2.0	2.80
IMD105.bmp	PH2Dataset	Melanocytic Nevus	1	4	2.0	2.70
IMD107.bmp	PH2Dataset	Melanocytic Nevus	2	3	2.0	3.90
IMD132.bmp	PH2Dataset	Melanocytic Nevus	2	5	3.0	4.60
IMD133.bmp	PH2Dataset	Melanocytic Nevus	1	2	2.0	2.50
IMD134.bmp	PH2Dataset	Melanocytic Nevus	2	4	2.0	4.00

IMD137.bmp	PH2Dataset	Melanocytic Nevus	0	5	2.0	1.50
IMD139.bmp	PH2Dataset	Melanocytic Nevus	1	3	2.0	2.60
IMD140.bmp	PH2Dataset	Melanocytic Nevus	1	5	3.0	3.30
IMD142.bmp	PH2Dataset	Melanocytic Nevus	1	5	3.0	3.30
IMD143.bmp	PH2Dataset	Melanocytic Nevus	1	2	3.0	3.00
IMD144.bmp	PH2Dataset	Melanocytic Nevus	1	2	3.0	3.00
IMD156.bmp	PH2Dataset	Melanocytic Nevus	1	3	2.0	2.60
IMD171.bmp	PH2Dataset	Melanocytic Nevus	1	4	2.0	2.70
IMD173.bmp	PH2Dataset	Melanocytic Nevus	0	7	2.0	1.70
IMD204.bmp	PH2Dataset	Melanocytic Nevus	0	1	2.0	1.10
IMD243.bmp	PH2Dataset	Melanocytic Nevus	1	8	3.0	3.60
IMD256.bmp	PH2Dataset	Melanocytic Nevus	2	7	2.0	4.30
IMD280.bmp	PH2Dataset	Melanocytic Nevus	1	2	2.0	2.50
IMD328.bmp	PH2Dataset	Melanocytic Nevus	2	1	3.0	4.20
IMD331.bmp	PH2Dataset	Melanocytic Nevus	1	5	3.0	3.30
IMD356.bmp	PH2Dataset	Melanocytic Nevus	2	2	3.0	4.30
IMD360.bmp	PH2Dataset	Melanocytic Nevus	1	1	3.0	2.90
IMD369.bmp	PH2Dataset	Melanocytic Nevus	2	5	3.0	4.60
IMD380.bmp	PH2Dataset	Melanocytic Nevus	1	3	2.0	2.60
IMD382.bmp	PH2Dataset	Melanocytic Nevus	2	4	4.0	5.00
IMD383.bmp	PH2Dataset	Melanocytic Nevus	0	3	2.0	1.30
IMD384.bmp	PH2Dataset	Melanocytic Nevus	2	5	2.0	4.10
IMD392.bmp	PH2Dataset	Melanocytic Nevus	1	6	2.0	2.90
IMD430.bmp	PH2Dataset	Melanocytic Nevus	1	2	3.0	3.00
IMD433.bmp	PH2Dataset	Melanocytic Nevus	2	4	3.0	4.50

Table 13.3: Results of TDS calculation for PH2Dataset