AMERICAN UNIVERSITY OF BEIRUT

A MACHINE LEARNING APPROACH FOR KERATOCONUS DETECTION

by
MHD JAWAD NIZAR KAISANIA

A thesis
submitted in partial fulfillment of the requirements
for the degree of Master of Science
to the Department of Computer Science
of the Faculty of Arts and Sciences
at the American University of Beirut

Beirut, Lebanon
September 2021

AMERICAN UNIVERSITY OF BEIRUT

A MACHINE LEARNING APPROACH FOR KERATOCONUS DETECTION

by

MHD JAWAD NIZAR KAISANIA

Approved by:

Dr. Shady Elbassuoni, Assistant Professor Computer Science	Advisor
Dr. Haidar Safa, Chairperson and Professor Computer Science	Member of Committee
Dr. Shady Awwad, Associate Professor of Clinical Specialty Ophthalmology	Member of Committee

Date of thesis defense: September 1, 2021

AMERICAN UNIVERSITY OF BEIRUT

THESIS, DISSERTATION, PROJECT RELEASE FORM

Student Name: Kaisania Mhd Jawad Nizar

Last First Middle

☒ Master's Thesis ◯ Master's Project ◯ Doctoral Dissertation

☒ I authorize the American University of Beirut to: (a) reproduce hard or electronic copies of my thesis, dissertation, or project; (b) include such copies in the archives and digital repositories of the University; and (c) make freely available such copies to third parties for research or educational purposes.

☐ I authorize the American University of Beirut, to: (a) reproduce hard or electronic copies of it; (b) include such copies in the archives and digital repositories of the University; and (c) make freely available such copies to third parties for research or educational purposes after: **One ___ year from the date of submission of my thesis, dissertation or project.**
Two ___ years from the date of submission of my thesis , dissertation or project.
Three ___ years from the date of submission of my thesis , dissertation or project.

September 14, 2021

Signature Date

This form is signed when submitting the thesis, dissertation, or project to the University Libraries

Acknowledgements

I am deeply grateful to my thesis advisor Prof. Shady Elbassuoni for his exemplary guidance, valuable feedback and constant encouragement. Without him, this thesis would not have been possible. I was incredibly lucky to have him as my advisor. He taught me how to do research and how to write papers.
My sincere thanks must also go to the members of my thesis committee: Prof. Haidar Safa and Dr. Shady Awwad who accepted to serve on my committee. They provided helpful suggestions to improve my work.
Finally, I must express my very profound gratitude to my mother, my family, and my friends for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you.

An Abstract of the Thesis of

Mhd Jawad Nizar Kaisania for Master of Science

Major: Computer Science

Title: A Machine Learning Approach for Keratoconus Detection

Keratoconus is a disorder of the eye that results in progressive thinning of the cornea. It usually occurs in the second decade of life and affects both genders and all ethnicities. The estimated prevalence in the general population is 54 per 100,000. Detecting Keratoconus is typically done using corneal tomography with different imaging systems, such as the Pentacam HR. More recently, corneal biomechanics (the corneal response to stress, and the ability of the cornea to resist deformation/distortion), has become more and more used to diagnose patients with ecstatic corneal disorders such as keratoconus. However, all of these techniques rely on medical experts to manually detect keratoconus based on an inspection of the cornea tomographic images and biomechanical signals.

In this thesis, we propose to utilize machine learning to automatically detect Keratoconus based on markers extracted from tomographic and biomechanical inspections of the eye. To be able to do this, we rely on various (anonymized)

datasets that are manually labelled by medical experts from the American University of Beirut Medical Center (AUBMC). Given that our datasets are limited in size, we perform 5-fold cross-validation and train various state-of-the-art machine learning techniques to automatically detect keratoconus. Our models achieved cross-validation accuracies ranging from 85% to 100% depending on the dataset and the classification task.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Keratoconus is the most common primary ectasia. Ocular signs and symptoms vary depending on disease severity. Early forms normally go unnoticed unless corneal topography is performed. Disease progression is manifested with a loss of visual acuity, which cannot be compensated for with spectacles. Corneal thinning frequently precedes ectasia [1], The process for detecting keratoconus is an ongoing one, and is an integral part of the preoperative evaluation of any patient considering corneal refractive surgery (LASIK, PRK, SMILE), as well as certain patients for intraocular surgery who wish for optimal refractive outcomes (such as premium intraocular lenses for cataract surgery, secondary phakic intraocular lenses, etc.). Figure 1.1 shows the difference in the cornea shape between a normal eye and a keratoconus eye.

Traditionally, detecting Keratoconus was done using corneal topography, which is a measure of the anterior curvature of the eye only. Then corneal tomogra-

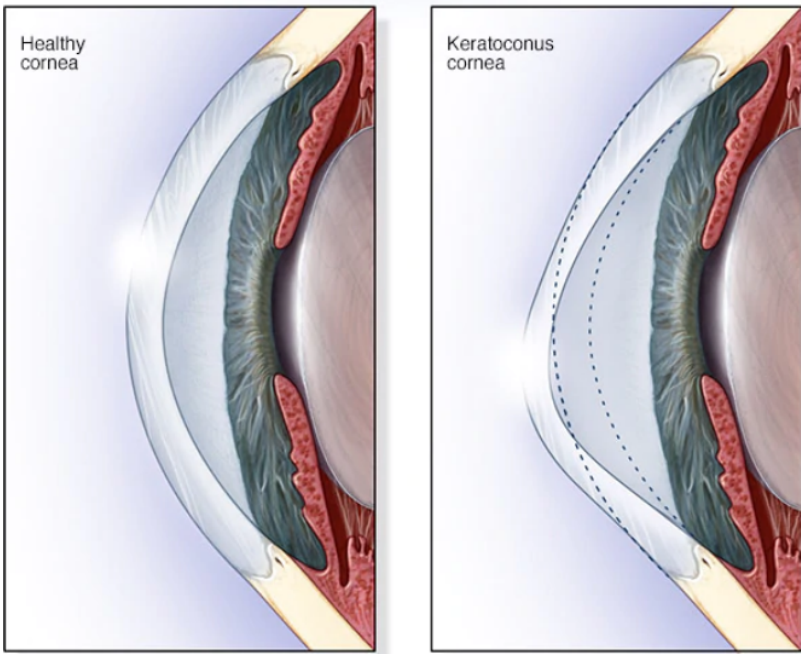

Figure 1.1: Healthy Cornea VS Keratoconus

phy was introduced, with different imaging systems such as the Pentacam HR, which provide insight into the anterior corneal surface, the posterior corneal surface, and what happens in between as well (such as the thickness of the cornea). More recently, corneal biomechanics (the corneal response to stress, and the ability of the cornea to resist deformation/distortion), has become more and more used to diagnose patients with corneal ecstatic disorders (such as keratoconus), because these present inherent weaknesses in the corneal biomechanical properties. Biomechanical and tomographic imaging results are used to determine and separate normal (no Keratoconus) from abnormal (Keratoconus) and borderline (cases where imaging is inconclusive) upon performing a refractive surgery.

Detecting Keratoconus has thus mostly relied on manual inspection of corneal tomography and biomechanics results. This can be very demanding and tedious work. On the other hand, the use of Artificial Intelligence (AI) in ophthalmology has drastically increased over the past decade with advances in machine learning and the proliferation of datasets that can be used to build such AI-based systems. In this thesis, we aim to use machine learning to build an automatic classifier to automatically detect Keratoconus based on corneal tomography and biomechanics signals.

1.2 Objectives and Contributions

In this thesis, we aim to train multiple classifiers to detect Keratoconus using two different datasets obtained from the American University of Beirut Medical Center (AUBMC) and that were manually labelled by medical experts. The first dataset consists of records of 202 patients that were collected using two different devices, namely Pentacam and Corvis. The second dataset consists of 277 patient records using the same two aforementioned devices.

We use each dataset to train a number of different classifiers. The first one is a 3-way classifier to distinguish between normal, subclinical keratoconus, and full keratoconus cases. The second classifier we train is a 2-way classifier where we consider the sub-clinical keratoconus and Keratoconus as one class and then train a binary classifier to distinguish between them and the normal class. Similarly, we train another 2-way classifier, where we consider the normal and subclinical cases as one class and the keratoconus cases as the second class. Finally, we also build two-stage classifiers that first use the 2-way classifiers to distinguish between the combined class and the other, and then use another binary 2-way classifier to distinguish between the combined classes.

Given the limited size of the datasets, we propose to use 5-fold cross-validation to train different machine learning models and choose the best-performing ones when training the various classifiers outlined above. Moreover, we also conduct various feature-selection experiments to identify the most influential features out of the set of available features in each dataset. To avoid over-fitting, we rely on medical intuition and expertise to guide the feature-selection process to avoid examining the space of all possible features. Finally, we also perform careful error analysis to bring humans back in the loop and provide interpretation of our trained machine learning models and their outputs.

1.3 Thesis Plan

This thesis is organized as follows. Chapter 2 gives an overview of related work that has employed machine learning techniques to detect subclinical keratoconus and keratoconus cases using corneal tomography and biomechanics signals. Chapter 3 describes the datasets we will use to train the various classifiers and their features. In Chapter 4, we describe how to train the various classifiers using the two datasets and report on their performances using 5-fold cross-validation. In the same chapter, we will also provide the results of our feature selection analyses, as well as error analyses for the various trained classifiers. Finally, we conclude and present future directions in Chapter 5.

Chapter 2

Literature Review

In this chapter, we review related work that utilizes Machine Learning for Keratoconus and Sub-Clinical Keratoconus cases detection based on corneal tomography and biomechanics signals.

In [2], the authors used four refractive maps (Sagittal map, Pachymetric map, Elevation map front and Elevation map back) taken by the Pentacam device to extract features. They first took each map and converted it from RGB to grey and then based on some specific diameters with the help of image processing, they extracted 12 features from all the maps. The next step was feeding the features to support vector machine classifier and building a 2-Way classifier (keratoconus vs normal). To train such a classifier, they built a dataset consisting of 40 patients in total and used 30 patients for training and 10 for testing and obtained a 90% test accuracy.

In [3], the authors also used support vector machine classifier in their study. 22 features were extracted from Pentacam and their dataset consisted of 860 patients. After processing the data, they built three classifiers; the first one to classify keratoconus vs normal, the second for subclinical keratoconus vs normal and the final one is a 5-way classifier (keratoconus, subclinical keratoconus, Astigmatic, After Refractive Surgery, and normal). They used 10-fold cross validation to train and validate their models and obtained cross-validation accuracies of 98.9%, 93.1% and 88.8% for the three classifiers, respectively.

In [4], the authors focused on detecting subclinical keratoconus. They extracted 11 parameters from a Pentacam Oculus topography device and their dataset consisted of 88 patients. They then applied 10-fold cross validation for 8 different models and chose the best ones (random forest, support vector machine and k-nearest neighbors). Next, they applied feature selection and reported the most important set of features for each of the selected models.Finally they reported that the highest AUC (Area under the Curve) was 0.97 for detecting subclinical keratoconus and it was achieved using five parameters by the random forest method. On the other hand, The highest sensitivity (0.94) and specificity (0.90) were obtained by support vector machine and the k-nearest neighbors model, respectively.

In [5], the authors proposed building a 2-way classifier to distinguish between Early Stage Keratoconus and Normal eyes and to do that they compared 25 machine learning models on different sets of features (443 in total) extracted from OCT-based topography instruments. Their best model was support vector machine classifier, which achieved a test accuracy of 94% using 8 features and a dataset of 3151 patients.

In [6], the authors experimented with various models such as support vector machine (SVM), Radial Basis Function (RBF) and a Multi-Layer Perceptron(MLP). They used a dataset consisting of 318 patients and used 11 features that were extracted from an Orbscan ii topography. They then performed data pre-processing and hyperparameter tuning using 10-fold cross validation and reported that the performances of the three classifiers were close and they relied on accuracy, sensitivity, specificity and Receiver Operating Characteristic(ROC) to evaluate them.

In [7], the authors proposed to use a Deep Learning approach to classify between Keratoconus and normal eyes. The dataset consisted of 304 Keratoconus eyes and 239 normal eyes and the extracted features were from six color-coded maps that have been taken from swept-source AS-OCT device. They reported an accuracy of 0.991 in classifying between Keratoconus and normal eyes.

Chapter 3

Datasets

This chapter describes the datasets we use to train the various machine learning models used in this thesis.

3.1 Data Acquisition

Our data was obtained through a retrospective case-control study of patients with keratoconus, subclinical keratoconus, and normal corneas that was conducted at the American University of Beirut Medical Center (AUBMC), Beirut, Lebanon. It was approved by the university's Institutional Review Board (Protocol: BIO-2018-0080), and adhered to the tenets of the Declaration of Helsinki.

3.1.1 PCBI Dataset

Our first dataset consists of data for a total of 202 patients who were examined and monitored by expert ophthalmologists at AUBMC. The patients were manually classified into three classes: patients with normal eyes (98), patients with subclinical keratoconus(49), i.e., those whose eyes have no symptoms or pain but if they did a refractive surgery then their eyes might develop the Keratoconus disease in other words these cases are borderline and the medical experts can not tell if there will be a disease or not later, and finally patients with Keratoconus disease (55). In total, each patient in this dataset is represented using 13 different features, which were collected from two different devices as follows.

Pentacam® HR tomography system (OCULUS Optikgeräte GmbH, Wetzlar, Germany): This is a tomography high-resolution camera system that measures the shape of the cornea, and the following seven features were obtained from it:

- Pentacam Kappa Chord Length: the chord length of the angle kappa, which is the angle between the patient's visual axis and the pupillary axis
- Pachymetric Progression Index Min: corneal thickness progression in the meridian with the smallest thickness progression from the thinnest point
- Pachymetric Progression Index Max: corneal thickness progression in the meridian with the greatest thickness progression from the thinnest point
- Ambrósio's Relational Thickness Min: the ratio of the thinnest point to PPI Min. (Pachy Prog Index Min.)
- Ambrósio's Relational Thickness Max: the ratio of the thinnest point to PPI Max. (Pachy Prog Index Max.)
- Pentacam Post. IS2mm: this parameter was derived from the instantaneous posterior curvature map of the cornea. It is the largest difference between

diametrically opposite points at a radius of 2mm from the center of the map, between the angles of 60 and 120 degrees.

- Pentacam Post. IS1mm: this parameter was derived using the same process of the previous one but the selected points were at a radius of 1mm from the center of the map.

Corvis® ST (OCULUS Optikgeräte GmbH, Wetzlar, Germany): This is a device that measures corneal biomechanics, and the following six features were derived using it:

- A1 Velocity: Velocity of the corneal deformation during applanation 1. When cornea is deformed, it goes through two positions called applanation 1, which occurs during the initial deformation, and applanation 2, which occurs when the corneas is returning to its original form.

- Deflection Amp. Max: The maximum deflection amplitude (the difference between the deflection and deformation is that deformation takes into account the whole eye movement as well based on the AUBMC experts)

- DA Ratio Max (2mm): Deformation at the apex over the deformation 2mm from the center (the deformation here is the maximum displacement of the cornea from its original position)

- DA Ratio Max (1mm): Deformation at the apex over the deformation 1mm from the center

- SPA1: Stiffness parameter at applanation 1 (derived from a couple of other Corvis parameters)

- ARTh: Ambrósio's Relational Thickness in the horizontal profile: the ratio of the thinnest point of the horizontal meridian to the PPI of the horizontal meridian.

3.1.2 STPI Dataset

The second dataset has a total of 277 patients and they are all different from the previous 202 patients in the first dataset. The patients in our second dataset were examined using a tomography machine called Galilei that has dual scheimpflug and placido imaging system instead of just scheimpflug (Pentacam HR). The patients were manually classified into three classes. 133 patients were classified as normal, 97 patients were classified as Keratoconus and 47 patients were classified as subclinical keratoconus. For each patient, seven features were obtained through some manipulation of the output of Galilei. For each point on the corneal, there is a value starting with "0" for the thinnest point and the rest of the points around that thinnest point have values that represent the speed of the thickness increase from the thinnest point to their respective locations. Using the image/map from Galilei, the medical experts averaged these points of speed for every 15-degree arc for radii 0.5, 1.0 and 2mm, and by measuring the greatest "15-degree arc averaged speed", they obtained the below three features:

- Max value for 0.5mm radius: The 15 degree arc with the maximum average speed of thickness progression at a radius of 0.5mm.

- Max value for 1.0mm radius: The 15 degree arc with the maximum average speed of thickness progression at a radius of 1.0mm.

- Max value for 2.0mm radius: The 15 degree arc with the maximum average speed of thickness progression at a radius of 2.0mm.

After that, they calculated the absolute difference between the averaged speed of diametrically opposite 15-degree arcs for every pair of diametrically opposite arcs, and the influence behind extracting these values is that keratoconus creates asymmetries in the cornea in a way that they would expect to have a high speed of thickness increase in one direction, and a low speed of thickness increase in the opposite direction, and again they reported the maximum value for these differences, obtaining the following three additional features:

- Abs.Diff max value for 0.5mm radius: Maximum difference of speed of thickness progression between diametrically opposite 15 degree arcs at a radius 0.5mm.

- Abs.Diff max value for 1.0mm radius: Maximum difference of speed of thickness progression between diametrically opposite 15 degree arcs at a radius 1.0mm.

- Abs.Diff max value for 2.0mm radius: Maximum difference of speed of thickness progression between diametrically opposite 15 degree arcs at a radius 2.0mm.

Finally, the last feature was the greatest acceleration from 0.5 to 1.0 mm, i.e.,

- Max value acc from 0.5-1.0: Maximum acceleration of thickness progression between 15 degree arcs at 0.5mm and 15 degree arcs at 1.0mm.

Chapter 4

Machine Learning Models

4.1 Setup

4.1.1 Base Models

We train four different machine learning models using each of the two datasets described in the previous chapter for various classification tasks as we explain later in this chapter. These four models are random forest, logistic regression, support vector machines and adaboost. Briefly,

(1) Support vector machine[8] translates data into another space where a plane ("hyperplane") maximally separates disparate data groups from itself.

(2) Logistic regression is a simple regression method that learns a mapping from input variables (X) to an output variable (Y) with Y = f(X).

(3) Ensemble method such as random forest that consists of many decision trees which were generated randomly using the features in our dataset, the output of random forest is the selected class by most trees.

(4) Adaboost algorithm, short for Adaptive Boosting, is a boosting technique that begins by fitting a classifier on the original dataset and then fits additional copies of the classifier on the same dataset but where the weights of incorrectly classified instances are adjusted such that subsequent classifiers focus more on difficult cases.

We only use these four models because they are the most common ones used in the context of in diagnosing diseases and biomedical studies. Moreover, since our datasets are limited in size, more complex models such as deep neural-network-based ones would overfit as they typically require relatively large training sets.

4.1.2 Data Normalization

Before training the different classifiers using the four machine learning models outlined above, we performed data normalization using two different techniques: MinMax and standardization. Our experimental results demonstrated that both techniques provide roughly the same results in most cases, with standardization performing slightly better in the remaining cases. Thus, in the rest of this chapter, we will only present results using standardization as a normalization technique.

4.1.3 Model Validation and Testing

Given that our datasets are limited in size, we perform 5-fold cross validation to validate and test the different machine learning models we train and to tune their hyperparameters. Briefly, 5-fold cross validation works by splitting a dataset into five folds, using four folds for training and the fifth for validation. This process is repeated five times, using a different fold for validation each time.

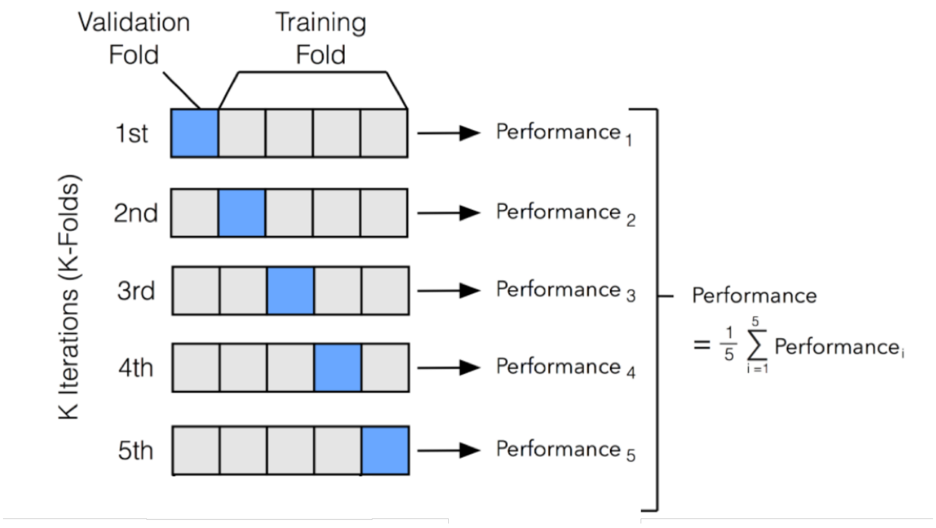

Figure 4.1: 5-Fold Cross Validation

Finally, evaluation metrics are then averaged over the five runs and are used to validate and test the trained models. Figure 4.1 shows an example of 5-fold cross validation.

4.1.4 Grid Search parameters Tuning

Grid Search was used during the training time to tune the hyperparameter for each classifier, we reported the optimal parameters for the best performance model in Appendix A.

4.2 PCBI Dataset Models

We trained various classifiers using the PCBI dataset. As mentioned in Chapter 3, this dataset consists of records belonging to three different classes, namely: normal which is denoted by class 0 and represents the healthy patients, keratoconus which is class 3 and represents the patients will full Keratoconus, and subclinical keratoconus which represents the patients with no clear determination, meaning the experts are not sure if the eye would develop Keratoconus after doing the LASIK/LAZIR surgery or not and this class is denoted by class 1.

4.2.1 Data Cleaning and Analysis

Before we set out to train the different classifiers using the PCBI dataset, we first performed some data cleaning and analysis. First, we augmented the dataset with an additional feature TBI, which is derived from both the Pentacam and Corvis devices based on the medical team recommendation. Second, we had two missing values for two features, which are ARTH and TBI (we refer the reader to Chapter 3 for a description of the first feature. We alerted the medical team about the missing values, and they decided to exclude one record from the dataset and to use the mean of values for the other one. Moreover, the medical team excluded 7 more patients from the dataset for technical reasons, and thus we ended up with a total of 202 records (98 normal cases, 49 subclinical keratoconus and 55 keratoconus) in the dataset.

Figure 4.2 shows the distribution of the records in the dataset over the three different classes and as can be seen from the figure, the dataset is relatively imbalanced with the majority of the records belonging to the "Normal" class.

4.2.2 Classifiers

We used the PCBI to build two different classifiers. The first was a 3-way classifier that classifies a patient record into one of the three classes we have, i.e., normal

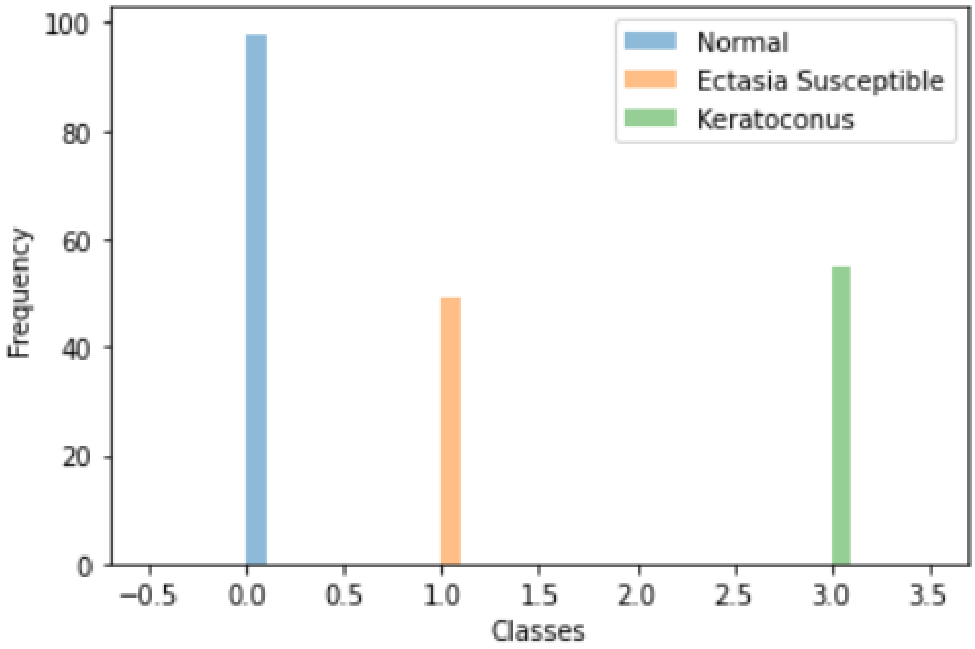

Figure 4.2: PCBI Classes Distribution

vs subclinical keratoconus vs keratoconus. The second was a 2-way classifier where we combined subclinical keratoconus cases with keratoconus cases and considered them as one class and the classifier was trained was to classify a patient record as either normal or (keratoconus/subclinical keratoconus). We trained different versions of each of the two classifiers using a different subset of features, guided by the expertise and insight of the medical team. For each version of each classifier, we used four different machine learning models as explained earlier. In the following, we present the results for each version of each classifier using the different subsets of features separately.

Original Features

The first version of our two classifiers (i.e., the 3-way and the 2-way) were trained using all 12 original features in the PCBI dataset (i.e., Pentacam Post IS2mm, Corvis SP A1, Pentacam Kappa Chord Length, Pachy Prog Index Min, Pachy Prog Index Max, ART Min, ART Max, Corvis A1 Velocity [mm], Corvis Deflection Amp. Max [mm], Corvis DA Ratio Max (2mm), Corvis DA Ratio Max (1mm) and Corvis ARTh).

As mentioned earlier, for each classifier, we trained four machine learning models using 5-fold cross validation. Table 4.1 shows the performance of each model using various metrics for the 3-way classifier whereas Table 4.2 shows the performance of each models for the 2-way classifier.

Model	Precision	Recall	F1-Score	Accuracy
SVM	**0.89**	**0.89**	**0.89**	**0.886**
Random Forest	0.88	0.88	0.88	0.876
Logistic Regression	0.87	0.87	0.87	0.866
AdaBoost	0.81	0.69	0.71	0.693

Table 4.1: Performance of 3-way Classifier using Original Features

Model	Precision	Recall	F1-Score	Accuracy
Random Forest	**0.93**	**0.93**	**0.93**	**0.930**
SVM	0.92	0.92	0.92	0.920
Logistic Regression	0.92	0.92	0.92	0.915
AdaBoost	0.81	0.81	0.81	0.811

Table 4.2: Performance of the 2-Way Classifier using Original Features

Support vector machine performed the best in classifying between our 3 classes, the weighted average accuracy is 0.886% and when looking at normal vs (subclinical keratoconus/keratoconus) random forest achieved an accuracy of 0.93% among the other 2-way classifiers.

Original Features Augmneted with TBI Feature

The second version of our two classifiers were trained using all 12 original features in the PCBI dataset in addition to the TBI feature, which was derived from both the Pentacam and Corvis devices together. Again, for each classifier, we trained four machine learning models using 5-fold cross validation. Table 4.3 shows the performance of each model using various metrics for the 3-way classifier whereas Table 4.4 shows the performance of each models for the 2-way classifier.

Model	Precision	Recall	F1-Score	Accuracy
Logistic Regression	**0.89**	**0.89**	**0.89**	**0.891**
Random Forest	0.89	0.89	0.89	0.886
SVM	0.88	0.88	0.88	0.876
AdaBoost	0.80	0.73	0.75	0.727

Table 4.3: Performance of the 3-way Classifier using Original Features Augmented with TBI

Model	Precision	Recall	F1-Score	Accuracy
SVM	**0.92**	**0.92**	**0.92**	**0.920**
Random Forest	0.92	0.92	0.92	0.915
Logistic Regression	0.92	0.92	0.92	0.915
AdaBoost	0.86	0.86	0.86	0.856

Table 4.4: Performance of the 2-way Classifier using Original Features Augmented with TBI

For this subset of our dataset, logistic regression performed the best as a 3-way classifier with an accuracy of 0.891%, which means adding TBI to our original features gave us a better performance. As for the 2-way classifier, support vector machine performed the best with 0.92% accuracy.

Original Features Without Pentacam Post 2mm Feature

The third version of our two classifiers were trained using 11 original features in the PCBI dataset after excluding Pentacam Post 2mm feature. Again, for each classifier, we trained four machine learning models using 5-fold cross validation. Table 4.5 shows the performance of each model using various metrics for the 3-way classifier whereas Table 4.6 shows the performance of each models for the 2-way classifier.

Model	Precision	Recall	F1-Score	Accuracy
Logistic Regression	**0.84**	**0.84**	**0.84**	**0.836**
SVM	0.83	0.83	0.83	0.826
Random Forest	0.82	0.82	0.82	0.821
AdaBoost	0.54	0.71	0.61	0.707

Table 4.5: Performance of the 3-way Classifier using Original Features without Pentacam Post 2mm

Model	Precision	Recall	F1-Score	Accuracy
SVM	**0.92**	**0.91**	**0.91**	**0.910**
Random Forest	0.90	0.90	0.90	0.896
Logistic Regression	0.91	0.90	0.90	0.900
AdaBoost	0.84	0.84	0.84	0.836

Table 4.6: Performance of the 2-way Classifier using Original Features without Pentacam Post 2mm

Logistic regression performed the best in classifying between the 3 different classes among the rest of the models, it achieved an accuracy of 0.836%, while support vector machine performed the best as a 2-way classifier with 0.91% accuracy. We can notice that both the 3-way and the 2-way classifiers performances dropped from the previous experiment which assures that having Pentacam Post 2mm as a feature made our classifiers learn more on how to classify between the classes.

Original Features Without Corvis SPA1 Feature

The fourth version of our two classifiers were trained using 11 original features in the PCBI dataset after excluding Corvis SPA1 feature. Again, for each classifier, we trained four machine learning models using 5-fold cross validation. Table 4.7 shows the performance of each model using various metrics for the 3-way classifier whereas Table 4.8 shows the performance of each models for the 2-way classifier.

Model	Precision	Recall	F1-Score	Accuracy
Random Forest	**0.87**	**0.87**	**0.87**	**0.871**
Logistic Regression	0.87	0.87	0.87	0.871
SVM	0.87	0.87	0.87	0.866
AdaBoost	0.75	0.77	0.74	0.767

Table 4.7: Performance of the 3-way Classifier using Original Features without Corvis SPA1

Model	Precision	Recall	F1-Score	Accuracy
Random Forest	**0.92**	**0.92**	**0.92**	**0.920**
SVM	0.92	0.92	0.92	0.920
Logistic Regression	0.92	0.92	0.92	0.915
AdaBoost	0.81	0.81	0.81	0.806

Table 4.8: Performance of the 2-way Classifier using Original Features without Corvis SPA1

Random forest performed the best as a 3-way classifier with 0.871% accuracy and as a 2-way classifier with 0.92% accuracy. Again eliminating Corvis SPA1 made our 3-way and 2-way classifier performed poorly than when we included in which gave us an insight of the importance of it as the medical team proposed.

Original Features Without Pentacam Post 2mm and Corvis SPA1 Features

The fifth version of our two classifiers were trained using 10 original features in the PCBI dataset after excluding Pentacam Post 2mm and Corvis SPA1 features. Again, for each classifier, we trained four machine learning models using 5-fold cross validation. Table 4.9 shows the performance of each model using various metrics for the 3-way classifier whereas Table 4.10 shows the performance of each models for the 2-way classifier.

Model	Precision	Recall	F1-Score	Accuracy
Logistic Regression	**0.84**	**0.84**	**0.84**	**0.836**
SVM	0.84	0.84	0.84	0.836
Random Forest	0.83	0.83	0.83	0.826
AdaBoost	0.81	0.82	0.81	0.816

Table 4.9: Performance of the 3-way Classifier using Original Features without Pentacam Post 2mm and Corvis SPA1

Model	Precision	Recall	F1-Score	Accuracy
SVM	**0.89**	**0.89**	**0.89**	**0.891**
Random Forest	0.88	0.88	0.88	0.881
Logistic Regression	0.90	0.89	0.89	0.891
AdaBoost	0.84	0.84	0.84	0.836

Table 4.10: Performance of the 2-way Classifier using Original Features without Pentacam Post 2mm and Corvis SPA1

Logistic regression performed the best in classifying between the 3 classes with 0.836% accuracy, while support vector machine achieved an accuracy of 0.891% as a 2-way classifier. Again the performance of both the 3-way and the 2-way classifier dropped after eliminating Pentacam Post 2mm and Corvis SPA1 from the original features.

Only Pentacam Post 2mm and Corvis SPA1 Features

The sixth version of our two classifiers were trained using only two of the original features summarized as Pentacam Post 2mm and Corvis SPA1 features. Again, for each classifier, we trained four machine learning models using 5-fold cross validation. Table 4.11 shows the performance of each model using various metrics for the 3-way classifier whereas Table 4.12 shows the performance of each models for the 2-way classifier.

Model	Precision	Recall	F1-Score	Accuracy
Logistic Regression	**0.86**	**0.86**	**0.86**	**0.861**
Random Forest	0.85	0.85	0.85	0.846
Random Forest	0.86	0.85	0.85	0.846
AdaBoost	0.56	0.72	0.62	0.717

Table 4.11: Performance of the 3-way Classifier using only Pentacam Post 2mm and Corvis SPA1

Model	Precision	Recall	F1-Score	Accuracy
Logistic Regression	**0.90**	**0.90**	**0.90**	**0.896**
Random Forest	0.90	0.90	0.90	0.896
SVM	0.90	0.89	0.89	0.891
AdaBoost	0.85	0.85	0.85	0.851

Table 4.12: Performance of the 2-way Classifier using only Pentacam Post 2mm and Corvis SPA1

Logistic regression performed the best as a 3-way classifier using only two features (Pentacam Post 2mm and Corvis SPA1) with an accuracy of 0.861%, while when looking at normal vs (subclinical keratoconus/keratoconus) logistic regresson also performed the best with 0.896% accuracy. We noticed having these two features alone gave us a result at somehow close to the one when using all 12 original features, but still having all the features performed better.

Only TBI Feature

The seventh version of our two classifiers were trained using only TBI feature which was constructed from features from both Pentacam and Corvis devices. Again, for each classifier, we trained four machine learning models using 5-fold cross validation. Table 4.13 shows the performance of each model using various metrics for the 3-way classifier whereas Table 4.14 shows the performance of each models for the 2-way classifier.

Model	Precision	Recall	F1-Score	Accuracy
SVM	**0.83**	**0.84**	**0.83**	**0.836**
Logistic Regression	0.82	0.83	0.82	0.831
Random Forest	0.82	0.82	0.82	0.821
AdaBoost	0.56	0.74	0.64	0.737

Table 4.13: Performance of the 3-way Classifier using Only TBI

Model	Precision	Recall	F1-Score	Accuracy
Logistic Regression	**0.91**	**0.91**	**0.91**	**0.910**
SVM	0.92	0.91	0.91	0.910
Random Forest	0.90	0.89	0.89	0.891
AdaBoost	0.90	0.89	0.89	0.891

Table 4.14: Performance of the 2-way Classifier using Only TBI

We found that support vector machine achieved an accuracy of 0.836% and be the best machine learning model when looking at normal vs subclinical keratoconus vs keratoconus, while logistic regression performed the best when looking at normal vs (subclinical keratoconus/keratoconus) with 0.91% accuracy. TBI alone gave us a decent accuracy, yet using the 12 original features performed better in both 3-way and 2-way classifier.

Tow Stages Classification

Given that the 2-way classifier consistently performed better than the 3-way classifier in all previous experiments, we also experiment with a two-stage classifier. To train the first classifier, we combine the normal and subclinical keratoconus records and train a binary classifier to distinguish between them and keratoconus records. We also train another binary classifier to distinguish between normal and subclinical keratoconus cases. We then use a pipelined approach to perform 3-way classification using those two binary classifiers. Similarly, we also train another two-stage classifier and use it exactly the same as just described, but combining the subclinical keratoconus and keratoconus cases into one class. We describe each such classifier separately next.

First two-stage classifier: Normal/Subclinical keratoconus vs Keratoconus

Our first binary classifier in the first two-stage classifier was trained using the whole dataset (merging normal and subclinical keratoconus cases into one class) and the second was trained on the subset of the dataset that contains only normal and subclinical keratoconus cases.

1. **C1: (Normal/Subclinical Keratoconus) VS Keratoconus**

   To train this classifier, we merged the normal cases (98) and the subclinical keratoconus cases (49) and ended up with 147 cases as normal/subclinical keratoconus. On the other hand, we had 55 keratoconus cases. The classifier was trained using 5-fold cross validation with Standard-Deviation normalization, and four different models were explored. Grid search was used to train the hyperparameters of all models. Table 4.15 shows the 5-fold cross-validation results for the four different models we explored. As can be seen from the table, Logistic Regression did the best performance with 0.975% accuracy for that we reported its confusion matrix in table 4.16, in the confusion matrix we denoted keratoconus as KC and subclinical keratoconus as SUSKC.

Model	Precision	Recall	F1-Score	Accuracy
Logistic Regression	**0.98**	**0.98**	**0.98**	**0.975**
Random Forest	0.97	0.97	0.97	0.970
AdaBoost	0.96	0.96	0.96	0.960
SVM	0.96	0.96	0.96	0.960

Table 4.15: PCBI- First two-stage classifier cross validation result for C1

		Predicted	
		Normal+SUSKC	**KC**
Actual	**Normal+SUSKC**	**143**	4
	KC	1	**54**

Table 4.16: PCBI- First two-stage classifier confusion matrix for C1

2. **C2: Normal VS Subclinical Keratoconus**

We used the 147 cases that are labeled as either normal or subclinical keratoconus to train this classifier. Again, the classifier was trained using 5-fold cross validation with Standard-Deviation normalization and four different models were explored. Grid search was used to train the hyperparameters of all models. Table 4.17 shows the 5-fold cross-validation results for the four different models we explored. Logistic regression and support vector machine had the same accuracy, but we selected logistic regression as the best model because it misclassified 5 normal cases as subclinical keratoconus while support vector machine misclassified 10. Again, table 4.18 shows the confusion matrix of logistic regression.

Model	Precision	Recall	F1-Score	Accuracy
Logistic Regression	**0.90**	**0.90**	**0.90**	**0.897**
SVM	0.90	0.90	0.90	0.897
Random Forest	0.86	0.86	0.86	0.857
AdaBoost	0.86	0.85	0.85	0.857

Table 4.17: PCBI- First two-stage classifier cross validation result for C2

		Predicted	
		Normal	**SUSKC**
Actual	**Normal**	**93**	5
	SUSKC	10	**39**

Table 4.18: PCBI- First two-stage classifier confusion matrix for C2

3. **Overall Approach**

   To evaluate the whole approach (i.e., two-stage classifier), another round of cross-validation was done as follows. The dataset was split into 5 folds (p1, p2, p3, p4 and p5). Next:

   (a) We used 4 folds, say "p1 + p2 + p3 + p4" to train C1 (normal/subclinical keratoconus VS keratoconus) using the best model with the best hyperparameters identified from training on the whole dataset.

   (b) We used the same 4 folds, again say "p1 + p2 + p3 + p4", to train C2 (normal VS subclinical keratoconus).

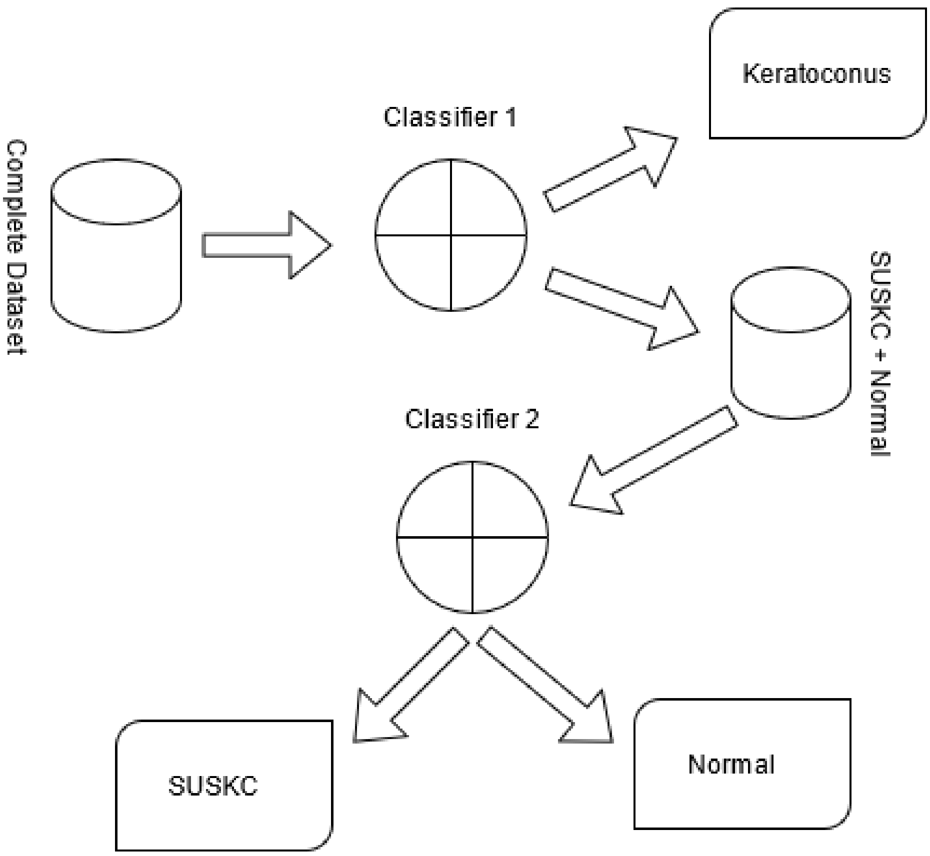

Figure 4.3: Two Stages Classification SUSKC+Normal VS KC

(c) We used the remaining fold, i.e., p5, to evaluate the two-stage classifier by passing each example in p5 through first C1 and if it predicts normal/subclinical keratoconus, we pass it to C2.

(d) We repeated the previous 4 steps 5 times using different folds for training and testing. Figure 4.3 is an overview of the whole process

The 5-fold cross-validation results and the confusion matrix of the overall approach are shown in table 4.19as and table 4.20respectively.

Logistic Regression	Precision	Recall	F1-Score	Accuracy
Weighted Average	**0.87**	**0.86**	**0.86**	**0.861**

Table 4.19: PCBI- First two-stage classifier cross validation result

		Predicted		
		Normal	**SUSKC**	**KC**
Actual	**Normal**	**85**	13	0
	SUSKC	8	**37**	4
	KC	0	3	**52**

Table 4.20: PCBI- First two-stage classifier confusion matrix

We can notice that using this approach, our two-stage classification did correctly classify 85 normal and misclassified 13 as subclinical keratoconus, when looking at subclinical cases we had 37 correctly classified cases, 8 cases were misclassified as normal and 4 cases were misclassified as keratoconus. Finally we had 52 keratoconus cases classified as keratoconus and 3 misclassified cases as subclinical cases.

Second two-stage classifier: Keratoconus/Subclinical keratoconus vs Normal

Our first binary classifier in the second two-stage classifier was trained using the whole dataset (merging keratoconus and subclinical keratoconus cases into one class) and the second was trained on the subset of the dataset that contains only keratoconus and subclinical keratoconus cases.

1. **C1: (Keratoconus/Subclinical Keratoconus) VS Normal**

   To train this classifier, we merged the keratoconus cases (55) and the subclinical keratoconus cases (49) and ended up with 104 cases as keratoconus/subclinical keratoconus. On the other hand, we had 98 normal cases. The classifier was trained using 5-fold cross validation with Standard-Deviation normalization, and four different models were explored. Grid search was used to train the hyperparameters of all models. Table 4.21 shows the 5-fold cross-validation results for the four different models we explored. As can be seen from the table, Random Forest had the best performance with 0.935% accuracy and its confusion matrix is reported in table 4.22.

Model	Precision	Recall	F1-Score	Accuracy
Random Forest	**0.94**	**0.94**	**0.94**	**0.935**
Logistic Regression	0.92	0.92	0.92	0.915
SVM	0.91	0.91	0.91	0.910
AdaBoost	0.91	0.91	0.91	0.910

Table 4.21: PCBI- Second two-stage classifier cross validation result for C1

		Predicted	
		Normal	**SUSKC+KC**
Actual	**Normal**	**93**	5
	SUSKC+KC	8	**96**

Table 4.22: PCBI- Second two-stage classifier confusion matrix for C1

2. **C2: Keratoconus vs Subclinical Keratoconus**

   We used the 104 cases that are labeled as either keratoconus or subclinical keratoconus to train this classifier. Again, the classifier was trained using 5-fold cross validation with Standard-Deviation normalization and four different models were explored. Grid search was used to train the hyperparameters of all models. Table 4.23 shows the 5-fold cross-validation results for the four different models we explored. Adaboost had the best performance with 0.961% accuracy and its confusion matrix is reported in table 4.24.

Model	Precision	Recall	F1-Score	Accuracy
Adaboost	**0.96**	**0.96**	**0.96**	**0.961**
Logistic Regression	0.95	0.95	0.95	0.951
SVM	0.93	0.93	0.93	0.932
Random Forest	0.92	0.92	0.92	0.921

Table 4.23: PCBI- Second two-stage classifier cross validation result for C2

		Predicted	
		KC	**SUSKC**
Actual	**KC**	**53**	2
	SUSKC	2	**47**

Table 4.24: PCBI- Second two-stage classifier confusion matrix for C2

3. **Overall Approach**

   To evaluate the whole approach (i.e., two-stage classifier), another round of cross-validation was done as follows. The dataset was split into 5 folds (p1, p2, p3, p4 and p5). Next:

   (a) We used 4 folds, say "p1 + p2 + p3 + p4" to train C1 (normal/subclinical keratoconus VS keratoconus) using the best model with the best hyperparameters identified from training on the whole dataset.

   (b) We used the same 4 folds, again say "p1 + p2 + p3 + p4", to train C2 (normal VS subclinical keratoconus).

   (c) We used the remaining fold, i.e., p5, to evaluate the two-stage classifier by passing each example in p5 through first C1 and if it predicts normal/subclinical keratoconus, we pass it to C2.

   (d) We repeated the previous 4 steps 5 times using different folds for training and testing. Figure 4.4 is an overview of the whole process

The 5-fold cross-validation results and the confusion matrix of the overall approach are shown in table 4.25as and table 4.26respectively.

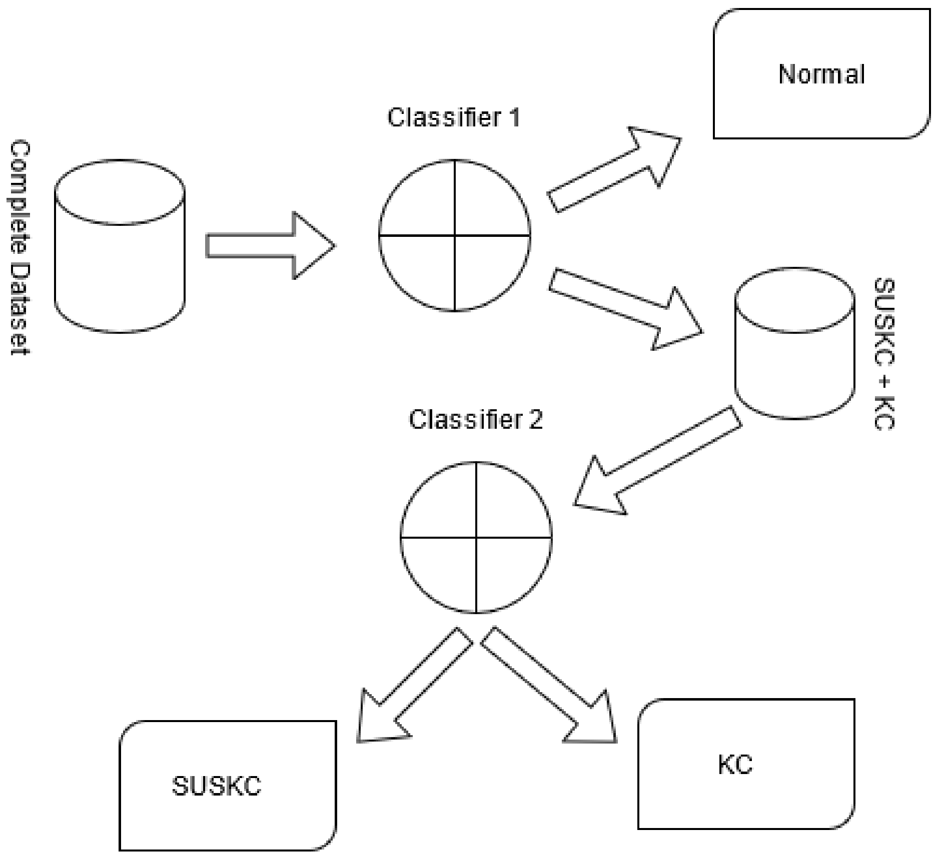

Figure 4.4: Two Stages Classification SUSKC+KC VS Normal

AdaBoost	Precision	Recall	F1-Score	Accuracy
Weighted Average	**0.92**	**0.91**	**0.91**	**0.910**

Table 4.25: PCBI- Second two-stage classifier cross validation result

		Predicted		
		Normal	**SUSKC**	**KC**
Actual	**Normal**	**90**	8	0
	SUSKC	3	**44**	2
	KC	0	5	**50**

Table 4.26: PCBI- Second two-stage classifier confusion matrix

We found that combining keratoconus with subclinical keratoconus in one class then classify them vs normal cases in C1 to finally make C2 classify keratoconus vs subclinical keratoconus preformed better than combining normal with subclinical keratoconus vs keratoconus in C1, also the confusion matrix shows less misclassified normal cases as subclinical keratoconus, more correctly classified subclinical keratoconus cases and when looking at keratoconus cases this approach misclassified more keratoconus cases as subclinical keratoconus.

4.2.3 Summary of Results and Error Analysis

Model performance using the 12-parameter set

Figure 4.5 and figure 4.6 shows the performance of each of the four machine learning methods for the set of 12 variables for each of the two and three-way classifiers respectively. Random forest performed best for when looking at normal compared to keratoconus with very good sensitivity (0.952), specificity (0.908), and overall accuracy (0.931%). While the other models' performances were inferior to random forest, they were still good, with overall accuracies between 0.812% for adaboost to 0.921% for support vector machine. However, when looking at normal vs subclinical keratoconus vs keratoconus, support vector machine performs best with overall accuracy of 0.886%, sensitivity of 0.926, and specificity of 0.929. While random forest and logistic regression perform well (accuracies of 0.876% and 0.866%, respectively), Adaboost performs poorly (accuracy = 0.693%).

Selecting the parameter combinations with the best performance

We confirmed that adaboost performs poorly compared to the other models when using all 12 variables, and thus focused on the three others. When running the same analysis on all biomechanical parameters but without including the (Pentacam Post 2mm), the highest accuracies of random forest in the two-way classifier and SVM in the three-way classifier both drop (0.931 to 0.896, and 0.886 to 0.827, respectively). In fact, when removing one feature at a time from the 12-variable model for each machine learning method, the biggest drop in accuracy happens when the (Pentacam Post 2mm) is excluded. We therefore tested all possible parameter combinations using the three methods, while keeping (Pentacam Post 2mm) present in all permutations.

1. **Three-way classifier:**

   For the three-way classifier, the following models had the best performances

   (a) The highest accuracy was obtained with logistic regression (0.896), followed by support vector machine (0.891), both using the same minimal parameter set of Pentacam Post 2mm, Corvis SPA1, DA Ratio Max (1mm), ART Min, ART Max, and ARTH.

(b) The highest sensitivity was obtained with both support vector machine and logistic regression (0.929) using the same combination of parameters as 1.a.

(c) The highest specificity was obtained with support vector machine (0.939) using the same minimal combination of parameters as 1.a.

2. **Two-way Classifier:** For the two-way classifier, the following models had the best performances

   (a) The highest accuracy was obtained with random forest (0.931) using the minimal parameter set of Pentacam Post 2mm, Corvis SPA1, DA Ratio Max (1mm), ART Min, ART Max, and ARTH, as well as with the 12-variable parameter set.

   (b) The highest sensitivity was obtained with random forest (0.952) using all 12 variables, as well as a minimal parameter set of Pentacam Post 2mm, Corvis SPA1, DA Ratio Max (1mm), ART Min, and ART Max.

   (c) The highest specificity was obtained with SVM (0.959) using the same minimal combination of parameters as 2.a.

Confusion Matrices

From the previous steps, we decided that for the three-way classifier, support vector machine with the minimal parameter set (Pentacam Post 2mm, Corvis SPA1, DA Ratio Max (1mm), ART Min, ART Max, and ARTH) provides the best bal-

Variables	Classifier	Model	Accuracy	Recall/Sensitivity	Precision/PPV	F1	Specificity	NPV
12 variables	2-way	Random Forest	0.931	0.952	0.917	0.934	0.908	0.947
12 variables	2-way	SVM	0.921	0.894	0.949	0.921	0.949	0.894
12 variables	2-way	Logistic Regression	0.916	0.894	0.939	0.916	0.939	0.893
12 variables	2-way	AdaBoost	0.812	0.808	0.824	0.816	0.816	0.800
12 variables + TBI	2-way	SVM	0.921	0.913	0.931	0.922	0.929	0.910
12 variables + TBI	2-way	Random Forest	0.916	0.913	0.922	0.918	0.918	0.909
12 variables + TBI	2-way	Logistic Regression	0.916	0.875	0.958	0.915	0.959	0.879
12 variables + TBI	2-way	AdaBoost	0.856	0.817	0.895	0.854	0.898	0.822
12 variables without Post. Curv	2-way	SVM	0.911	0.865	0.957	0.909	0.959	0.870
12 variables without Post. Curv	2-way	Logistic Regression	0.901	0.856	0.947	0.899	0.949	0.861
12 variables without Post. Curv	2-way	Random Forest	0.896	0.865	0.928	0.896	0.929	0.867
12 variables without Post. Curv	2-way	AdaBoost	0.837	0.837	0.845	0.841	0.837	0.828
12 variables without Post. Curv & SP A1	2-way	SVM	0.891	0.865	0.918	0.891	0.918	0.865
12 variables without Post. Curv & SP A1	2-way	Logistic Regression	0.891	0.846	0.936	0.889	0.939	0.852
12 variables without Post. Curv & SP A1	2-way	Random Forest	0.881	0.865	0.900	0.882	0.898	0.863
12 variables without Post. Curv & SP A1	2-way	AdaBoost	0.837	0.837	0.845	0.841	0.837	0.828
12 variables without SP A1	2-way	Random Forest	0.926	0.942	0.916	0.929	0.908	0.937
12 variables without SP A1	2-way	SVM	0.921	0.894	0.949	0.921	0.949	0.894
12 variables without SP A1	2-way	Logistic Regression	0.916	0.875	0.958	0.915	0.959	0.879
12 variables without SP A1	2-way	AdaBoost	0.822	0.837	0.821	0.829	0.806	0.823
6 variables	2-way	Random Forest	0.921	0.933	0.915	0.924	0.908	0.927
6 variables	2-way	Logistic Regression	0.921	0.894	0.949	0.921	0.949	0.894
6 variables	2-way	SVM	0.916	0.875	0.958	0.915	0.959	0.879
6 variables	2-way	AdaBoost	0.832	0.798	0.865	0.830	0.867	0.802
Post. Curv & SP A1	2-way	Logistic Regression	0.896	0.885	0.911	0.898	0.908	0.881
Post. Curv & SP A2	2-way	Random Forest	0.896	0.865	0.928	0.896	0.929	0.867
Post. Curv & SP A3	2-way	SVM	0.891	0.865	0.918	0.891	0.918	0.865
Post. Curv & SP A4	2-way	AdaBoost	0.851	0.865	0.849	0.857	0.837	0.854
TBI only	2-way	Logistic Regression	0.911	0.875	0.948	0.910	0.949	0.877
TBI only	2-way	SVM	0.911	0.865	0.957	0.909	0.959	0.870
TBI only	2-way	Random Forest	0.891	0.846	0.936	0.889	0.939	0.852
TBI only	2-way	AdaBoost	0.891	0.846	0.936	0.889	0.939	0.852

Figure 4.5: PCBI: All Two-Way Classifiers

ance of accuracy (0.891), sensitivity (0.917), and particularly specificity (0.936), in discriminating between normal, subclinical keratoconus and keratoconus eyes. Similarly, for the two-way classifier where the main purpose is discriminating between normal and abnormal corneas – and so ruling out an ecstatic process — random forest with all 12 parameters provides the highest sensitivity and accuracy (0.952 and 0.931, respectively), and acceptable specificity (0.908).

Looking at the confusion matrix of the two-way random forest model in table 4.27, 14/202 cases (6.93%) were misclassified. 9 of these cases were normal corneas misclassified as keratoconus or subclinical keratoconus, and 5 were (subclinical keratoconus/keratoconus) corneas misclassified as normal. Looking closer at the latter group, we notice that all cases were subclinical keratoconus eyes, and no eye with overt keratoconus was misclassified as normal.

Variables	Classifier	Model	Accuracy	Recall/Sensitivity	Precision/PPV	F1	Specificity	NPV
12 variables	3-way	Random Forest	0.876	0.907	0.907	0.907	0.908	0.908
12 variables	3-way	SVM	0.886	0.926	0.926	0.926	0.929	0.929
12 variables	3-way	Logistic Regression	0.866	0.926	0.889	0.907	0.888	0.926
12 variables	3-way	AdaBoost	0.693	0.922	0.710	0.802	0.704	0.920
12 variables + TBI	3-way	SVM	0.876	0.937	0.899	0.918	0.898	0.936
12 variables + TBI	3-way	Random Forest	0.886	0.918	0.909	0.914	0.908	0.918
12 variables + TBI	3-way	Logistic Regression	0.891	0.907	0.936	0.921	0.939	0.911
12 variables + TBI	3-way	AdaBoost	0.728	0.694	0.855	0.766	0.898	0.772
12 variables without Post. Curv	3-way	SVM	0.827	0.890	0.871	0.880	0.878	0.896
12 variables without Post. Curv	3-way	Logistic Regression	0.837	0.915	0.851	0.882	0.847	0.912
12 variables without Post. Curv	3-way	Random Forest	0.822	0.857	0.886	0.872	0.898	0.871
12 variables without Post. Curv	3-way	AdaBoost	0.708	0.578	0.881	0.698	0.929	0.705
12 variables without Post. Curv & SP A1	3-way	SVM	0.837	0.901	0.882	0.891	0.888	0.906
12 variables without Post. Curv & SP A1	3-way	Logistic Regression	0.837	0.874	0.874	0.874	0.878	0.878
12 variables without Post. Curv & SP A1	3-way	Random Forest	0.827	0.860	0.879	0.870	0.888	0.870
12 variables without Post. Curv & SP A1	3-way	AdaBoost	0.817	0.817	0.894	0.854	0.908	0.840
12 variables without SP A1	3-way	Random Forest	0.871	0.917	0.898	0.907	0.898	0.917
12 variables without SP A1	3-way	SVM	0.866	0.905	0.905	0.905	0.908	0.908
12 variables without SP A1	3-way	Logistic Regression	0.871	0.867	0.924	0.895	0.929	0.875
12 variables without SP A1	3-way	AdaBoost	0.767	0.660	0.901	0.762	0.929	0.734
6 variables	3-way	Random Forest	0.866	0.906	0.897	0.902	0.898	0.907
6 variables	3-way	Logistic Regression	0.896	0.929	0.919	0.924	0.918	0.928
6 variables	3-way	SVM	0.891	0.917	0.936	0.926	0.939	0.920
6 variables	3-way	AdaBoost	0.748	0.842	0.853	0.848	0.888	0.879
Post. Curv & SP A1	3-way	Logistic Regression	0.861	0.896	0.896	0.896	0.898	0.898
Post. Curv & SP A2	3-way	Random Forest	0.847	0.906	0.861	0.883	0.857	0.903
Post. Curv & SP A3	3-way	SVM	0.847	0.905	0.869	0.887	0.867	0.904
Post. Curv & SP A4	3-way	AdaBoost	0.718	0.490	1.000	0.657	1.000	0.667
TBI only	3-way	Logistic Regression	0.832	0.879	0.889	0.884	0.898	0.889
TBI only	3-way	SVM	0.837	0.891	0.882	0.886	0.888	0.897
TBI only	3-way	Random Forest	0.822	0.865	0.895	0.880	0.908	0.881
TBI only	3-way	AdaBoost	0.738	0.667	0.947	0.783	0.969	0.779

Figure 4.6: PCBI: All Three-Way Classifiers

		Predicted	
		Normal	**SUSKC+KC**
Actual	**Norma**	**89**	9
	SUSKC+KC	5	**99**

Table 4.27: PCBI- 2-Way Random Forest using 12 variables Confusion Matrix

For the three-way classifier using the minimal parameter set and SVM method, 22/202 cases (10.89%) were misclassified (Table 4.28). Most of the incorrect classifications were in misclassifying the SUSKC eyes as either normal (8) or keratoconus (3). No normal eyes were misclassified as keratoconus, and conversely no eye with keratoconus was classified as normal.

		Predicted		
		Normal	SUSKC	KC
Actual	Normal	**92**	6	0
	SUSKC	8	**38**	3
	KC	0	5	**50**

Table 4.28: PCBI- 3-Way SVM using 6 variables Confusion Matrix

4.3 STPI Dataset Models

In this section, we trained various classifiers using the STPI dataset. As mentioned in chapter 3, this dataset consists of records belong sign to three different classes, namely: normal which is denoted by 0 and represent the healthy patients, keratoconus which is class 3 and represents the patients will full keratoconus and subclinical keratoconus which represents the patients with no clear determination if the eye would develop keratoconus after doing refractive eyes surgeries or not and this class will be denoted by class 2.

4.3.1 Data Cleaning and Analysis

Before we set out to train the different classifiers using the STPI dataset, we first performed data exploration and analysis, there were no missing values in the columns we did not have to perform any data augmentation processes

Figure 4.7 shows us the distribution of the records in the dataset over the three different classes and as can be seen from the figure, the dataset is relatively imbalanced with the majority of the records belonging to the normal class.

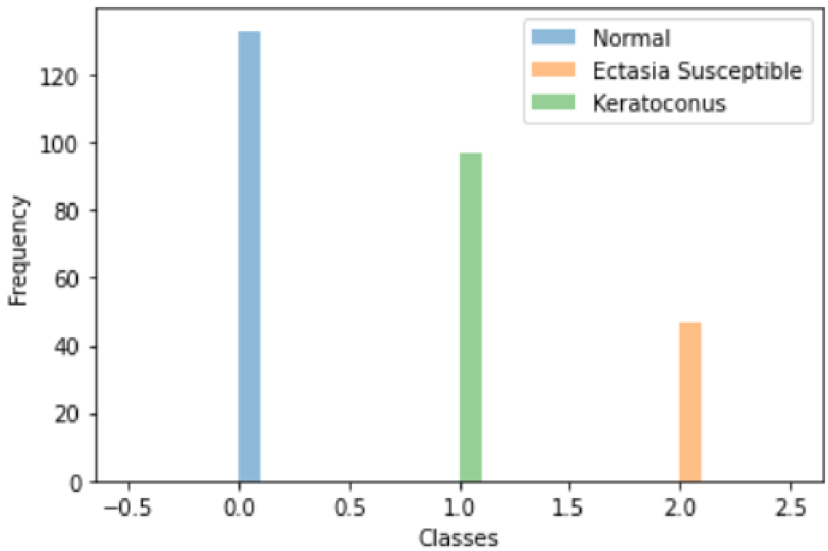

Figure 4.7: STPI Classes Distribution

4.3.2 Classifiers

We used the STPI to build various classifiers, the first was a 2-way classifier that classifies a patient record into either normal class or keratoconus class, The second was a 3-way classifier to classify between our 3 different classes. The third was another 2-way classifier but this was to classify if a patient is normal or (subclinical keratoconus/keratoconus) after combining the last two classes in one. Our fourth classifier was a 2-way classifier to classify between normal and subclinical keratoconus cases. We also performed a two-stage classification as we did in PCBI dataset.

2-Way Classifier: Normal VS Keratoconus

The first classifier was trained to classify between normal and keratoconus cases, we had a total of 230 patients divided into 133 labeled normal and 97 labeled keratoconus . After shuffling the data and normalizing the columns using standard scaler we trained our four machine learning models using 5-fold cross validation and Grid Search was used to tune the different hyperparameters for each single model.

For each model we reported the weighted average precision, weighted average recall, weighted average F1-score and the accuracy. Table 4.29 shows the cross-validation result using the four models:

Model	Precision	Recall	F1-Score	Accuracy
Logistic Regression	**1.0**	**1.0**	**1.0**	**1.0**
SVM	1.0	1.0	1.0	1.0
Random Forest	0.99	0.99	0.99	0.991
AdaBoost	0.99	0.99	0.99	0.991

Table 4.29: Performance of 2-way classifier: Normal VS Keratoconus

Logistic regression and support vector machine predicted both classes correctly and misclassified none, table 4.30 shows the confusion matrix of them.

		Predicted	
		Normal	**KC**
Actual	**Norma**	**133**	0
	KC	0	**97**

Table 4.30: SVM/Logistic Regression Confusion Matrix

Despite the fact that random forest and adaboost performed 0.991% accuracy, but they both misclassified 2 KC cases as normal which is what the medical team is trying to minimize. Table 4.31 shows their confusion matrix.

		Predicted	
		Normal	**KC**
Actual	**Norma**	**133**	0
	KC	2	**95**

Table 4.31: AdaBoost/Random Forest Confusion Matrix

3-Way Classifier: Normal VS Keratoconus VS Subclinical Keratoconus

Our 3-way classifier was to distinguish between the 3 classes we have, and in total we had 277 patients to run our experiment on. Again we trained our four machine learning models using 5-fold cross validation and Grid Search was used to tune the different hyper parameters for each single model.

Table 4.32 shows the cross validation result for our different models.

Model	Precision	Recall	F1-Score	Accuracy
Random Forest	**0.86**	**0.87**	**0.86**	**0.866**
SVM	0.85	0.85	0.85	0.862
AdaBoost	0.83	0.85	0.83	0.851
Logistic Regression	0.83	0.85	0.83	0.848

Table 4.32: Performance of 3-way classifier: Normal VS Keratoconus VS Subclinical Keratoconus

Random forest performed the best with 0.866% accuracy, more details on its classification process can be found in the confusion matrix in table 4.33

		Predicted		
		Normal	**SUSKC**	**KC**
Actual	**Normal**	**128**	5	0
	SUSKC	22	**23**	2
	KC	0	8	**89**

Table 4.33: STPI- Random Forest Confusion Matrix

2-Way Classifier: Normal VS (Keratoconus/Subclinical Keratoconus)

In this section we combined subclinical keratoconus cases (47) with keratoconus cases (97) in one class and train our four models on a 2-way classifier to classify between the combined class and normal class. Table 4.34 shows that random forest performed the best based on the cross validation results for the four models, thus its confusion matrix can be found in table 4.35.

Model	Precision	Recall	F1-Score	Accuracy
Random Forest	**0.92**	**0.91**	**0.91**	**0.906**
Logistic Regression	0.92	0.91	0.92	0.906
SVM	0.92	0.90	0.90	0.898
AdaBoost	0.89	0.89	0.89	0.888

Table 4.34: Performance of 2-way classifier: Normal VS Keratoconus/Subclinical Keratoconus

		Predicted	
		Normal	**SUSKC+KC**
Actual	**Norma**	**131**	2
	SUSKC+KC	24	**120**

Table 4.35: STPI- Random Forest Confusion Matrix

2-Way Classifier: Normal VS Subclinical Keratoconus

In this section we excluded keratoconus cases from our study and kept normal (133) and subclinical (47) cases, then we train our models using cross validation to classify between the two classes we have. Table 4.36 shows that random forest and support vector machine performed the best based on the cross validation results for the four models, thus their confusion matrices can be found in table 4.37 and table 4.38 respectively.

Model	Precision	Recall	F1-Score	Accuracy
Random Forest	**0.87**	**0.87**	**0.85**	**0.866**
SVM	0.87	0.87	0.85	0.866
Logistic Regression	0.85	0.84	0.82	0.844
AdaBoost	0.85	0.84	0.82	0.844

Table 4.36: Performance of 2-way classifier: Normal VS Subclinical Keratoconus

		Predicted	
		Normal	**SUSKC**
Actual	**Norma**	**131**	2
	SUSKC	22	**25**

Table 4.37: STPI- Random Forest Confusion Matrix

		Predicted	
		Normal	**SUSKC**
Actual	**Norma**	**132**	1
	SUSKC	23	**24**

Table 4.38: STPI- SVM Confusion Matrix

Tow Stages Classification

Since the 2-way classifier performed better than the 3-way classifier we decided to map our three classes into two 2-way classifiers and for that we tested two classifiers:

First two-stage classifier: Normal/Subclinical keratoconus vs Keratoconus

Our first binary classifier in the first two-stage classifier was trained using the whole dataset (merging normal and subclinical keratoconus cases into one class)and the second was trained on the subset of the dataset that contains only normal and subclinical keratoconus cases.

1. **C1: (Normal/Subclinical Keratoconus) VS Keratoconus**

To train this classifier, we merged the normal cases (133) and the subclinical keratoconus cases (47) and ended up with 180 cases as (normal/subclinical keratoconus). On the other hand, we had 97 keratoconus cases. The classifier was trained using 5-fold cross validation with StandardScaler normalization, and four different models were explored. Grid search was used to train the hyperparameters of all models. Table 4.40 shows the 5fold cross-validation results for the four different models we explored. As can be seen from the table, support vector machine, logistic regression and random forest results were the best, but random forest misclassified 4 keratoconus cases as (normal/subclinical keratoconus) while the others misclassified 6. Table 4.40 shows the confusion matrix of random forest.

Model	Precision	Recall	F1-Score	Accuracy
Random Forest	**0.96**	**0.96**	**0.96**	**0.963**
SVM	0.96	0.96	0.96	0.963
Logistic Regression	0.96	0.96	0.96	0.963
AdaBoost	0.95	0.95	0.95	0.945

Table 4.39: STPI- First two-stage classifier cross validation result for C1

		Predicted	
		Normal+SUSKC	**SUSKC**
Actual	**Normal+SUSKC**	**174**	6
	SUSKC	4	**93**

Table 4.40: STPI- First two-stage classifier confusion matrix for C1

2. **C2: Normal vs Subclinical Keratoconus**

   We used the 180 cases that are labeled as either normal or subclinical keratoconus to train this classifier. Again, the classifier was trained using 5-fold cross validation with StandardScaler normalization and four different models were explored. Grid search was used to train the hyperparameters of all models. Table 4.41 shows the 5-fold cross-validation results for the four different models we explored. Support vector machine had the best performance and table 4.42 shows its confusion matrix.

Model	Precision	Recall	F1-Score	Accuracy
SVM	**0.88**	**0.87**	**0.85**	**0.866**
Random Forest	0.85	0.85	0.83	0.850
Logistic Regression	0.85	0.84	0.82	0.844
AdaBoost	0.84	0.84	0.82	0.838

Table 4.41: STPI- First two-stage classifier cross validation result for C2

		Predicted	
		Normal	**SUSKC**
Actual	**Normal**	**132**	1
	SUSKC	23	**24**

Table 4.42: STPI- First two-stage classifier confusion matrix for C1

3. **Overall Approach**

   To evaluate the whole approach (i.e., two-stage classifier), another round of cross-validation was done manually as discussed and shown in figure 4.3, the 5-fold cross-validation results and the confusion matrix of the overall approach are shown in table 4.43as and table 4.44 respectively.

RF/SVM	Precision	Recall	F1-Score	Accuracy
Weighted Average	**0.84**	**0.85**	**0.84**	**0.851**

Table 4.43: STPI- First two-stage classifier cross validation result

		Predicted		
		Normal	**SUSKC**	**KC**
Actual	**Normal**	**125**	8	0
	SUSKC	23	**18**	6
	KC	1	3	**93**

Table 4.44: STPI- First two-stage classifier confusion matrix

Second two-stage classifier: Keratoconus/Subclinical Keratoconus vs Normal

Our first binary classifier in the second two-stage classifier was trained using the whole dataset (merging keratoconus and subclinical keratoconus cases into one class) and the second was trained on the subset of the dataset that contains only keratoconus and subclinical keratoconus cases.

1. **C1: (Keratoconus/Subclinical Keratoconus) VS Normal**

   To train this classifier, we merged the keratoconus cases (97) and the subclinical keratoconus cases (47) and ended up with 144 cases as (keratoconus/subclinical keratoconus). On the other hand, we had 133 normal cases. The classifier was trained using 5-fold cross validation with StandardScaler normalization, and four different models were explored. Grid search was used to train the hyperparameters of all models. Table 4.45 shows the 5-fold cross-validation results for the four different models we explored. As can be seen from the table, random forest had the best performance and its confusion matrix is reported in table 4.46.

Model	Precision	Recall	F1-Score	Accuracy
Random Forest	**0.92**	**0.91**	**0.91**	**0.909**
Logistic Regression	0.92	0.90	0.90	0.898
SVM	0.91	0.89	0.89	0.891
AdaBoost	0.90	0.88	0.88	0.880

Table 4.45: STPI- Second two-stage classifier cross validation result for C1

		Predicted	
		Normal	**SUSKC+KC**
Actual	**Normal**	**132**	1
	SUSKC+KC	23	**121**

Table 4.46: STPI- Second two-stage classifier confusion matrix for C1

2. **C2: Keratoconus vs Subclinical Keratoconus**

We used the 144 cases that are labeled as either KC or Suspect to train this classifier. Again, the classifier was trained using 5-fold cross validation with StandardScaler normalization and four different models were explored. Grid search was used to train the hyperparameters of all models. Table 4.47 shows the 5-fold cross-validation results for the four different models we explored. Random Forest had the best performance and made less number of misclassifying KC cases as suspect cases, its confusion matrix is reported in table 4.48.

Model	Precision	Recall	F1-Score	Accuracy
Random Forest	**0.94**	**0.93**	**0.93**	**0.930**
SVM	0.94	0.94	0.94	0.937
Logistic Regression	0.94	0.93	0.93	0.930
AdaBoost	0.92	0.92	0.91	0.916

Table 4.47: STPI- Second two-stage classifier cross validation result for C2

		Predicted	
		KC	**SUSKC**
Actual	**KC**	**90**	7
	SUSKC	2	**45**

Table 4.48: STPI- Second two-stage classifier confusion matrix for C1

3. **Overall Approach**

To evaluate the whole approach (i.e., two-stage classifier), another round of cross-validation was done manually as discussed and shown in figure 4.4, the 5-fold cross-validation results and the confusion matrix of the overall approach are shown in table 4.49as and table 4.50 respectively.

RF	Precision	Recall	F1-Score	Accuracy
Weighted Average	**0.82**	**0.81**	**0.81**	**0.812**

Table 4.49: STPI- Second two-stage classifier cross validation result

		Predicted		
		Normal	**SUSKC**	**KC**
Actual	**Normal**	**129**	4	0
	SUSKC	24	**20**	3
	KC	1	20	**76**

Table 4.50: STPI- Second two-stage classifier confusion matrix

By comparing the overall confusion matrices for both experiments, we found out that combining normal and subclinical keratoconus cases together and classify them vs keratoconus gave us better results than combining subclinical keratoconus with fully keratoconus eyes.

4.3.3 Results

Model performance as 2-Way Classifier

Table 4.29 shows the performance of each model of the four machine learning methods in classifying between normal and keratoconus cases. Logistic regression and support vector machine performed the best and got an accuracy of 1.0

followed by random forest and adaboost with 0.991 accuracy. However, when looking at normal vs subclinical keratoconus, random forest and support vector machine performed the best with overall accuracy of 0.87 while logistic regression and adaboost got an accuracy of 0.85. See table 4.36

Also, after we combined subclinical keratoconus and keratoconus together in one class and classified them vs normal cases. Table 4.34 shows that random forest and logistic regression performed the best with 0.906 as overall accuracy, followed by support vector machine and adaboost with 0.898 and 0.888 respectively.

Model performance as 3-Way Classifier

As for the 3-way classifier and classifying normal vs KC vs SUSKC. Using the two-stages classification process where the first classifier is (subclinical keratoconus/keratoconus) vs normal and the second one is subclinical keratoconus vs keratoconus performed poorly with 0.812 accuracy, on the other hand having (normal/subclinical keratoconus) vs keratoconus as a first classifier then normal vs subclinical keratoconus as the second one did better performance with an overall accuracy of 0.851. While the best performance was done by classifying the 3 classes at one stage using random forest with overall accuracy of 0.866, 0.86 precision, 0.87 recall and 0.86 f1-score. See table 4.32.

Chapter 5

Conclusion

In this thesis, we have explored the feasibility of different machine learning algorithms in detecting two-phases keratoconus (subclinical and full keratoconus) from control eyes based on combined features from tomographic and biomechanical eyes imaging systems, the dataset was collected from the American University of Beirut Medical Center (AUBMC) and labeled by medical experts. Due to our limited data size and to avoid overfitting we applied 5-fold cross validation to train various machine learning models named(random forest, logistic regression, support vector machine and adaboost), we also performed feature-selection experiments to identify the most influential features in detecting keratoconus.

Since we had 3 different classes (normal, keratoconus and subclnical keratoconus), using the machine learning models above we trained different classifiers summarized as; 3-way classifier to distinguish between the 3 classes, 2-way classifiers that looked at the different combination of our classes (normal vs keratoconus, normal vs subclinical keratoconus, keratoconus/subclinical keratoconus vs normal, normal/subclinical keratoconus vs keratoconus), and finally two-stage classifiers that first use the 2-way classifiers to distinguish between the combined class and the other, and then use another binary 2-way classifier to distinguish between the combined classes. As a result using the trained classifiers with different features combination, the confusion matrix of logistic regression when looking at normal vs keratoconus using STPI dataset shows that this classifier has predicted the keratoconus cornea with an excellent accuracy of 100%. As for the PCBI dataset, support vector machine was able to classify 89% of normal, subclinical keratoconus and keratoconus cases correctly giving the fact that our 3 classes are imbalanced and all the confusion happened when looking at subclinical keratoconus which mimics clinical reality.

In comparison with other machine learning algorithms and approaches that exist in specialized literature, the novelty factor of the tested algorithm is consisted by combining eyes features from different devices which is huge due to its possible contribution in detection an early keratoconus, thus saving lives. This work can be extended to include different advanced machine learning techniques such as boosting, as well as combining the two dataset from (Pentacam, Corvis and Galilli) together, increasing the dataset size specially the abnormal eyes would increase the performance. It also can be compared with other benchmark algorithms to reflect the performance gains achieved for each algorithm.

Appendix A

PCBI: Winning Models Tuned Parameters

The final results in PCBI section showed us that support vector machine using 6 variables performed the best as a 3-way classifier, table 5.1 shows the optimal parameters of this model using GridSearch. As for the 2-way classifier, random forest won using 12 variables and again table 5.2 shows its tuned parameters.

Model	SVM
C	10
Class-Weight	balanced
Gamma	0.1
Kernel	rbf

Table 5.1: PCBI- SVM Hyper Parameters Normal VS SUSKC VS KC

Model	Random Forest
Bootstrap	True
Class-Weight	balanced
criterion	gini
max_depth	5
max_features	log2
n_estimators	100

Table 5.2: PCBI- Random Forest Hyper Parameters Normal VS (KC+SUSKC)

STPI: Winning Models Tuned Parameters

The final results in STPI section showed us that logistic regression performed the best as a 2-way classifier (keratoconus vs normal) and table 5.3 shows the optimal parameters of this model using GridSearch. And when looking at (subclinical keratoconus vs normal) we found out that support vector machine won, table 5.4 shows its optimal parameters. While the best 3-way classifier was random forest and its parameters reported in table 5.5

Model	Logistic Regression
C	0.001
Class-Weight	balanced
Penalty	None
Solver	Newton-cg

Table 5.3: STPI- Logistic Regression Hyper Parameters Normal VS Keratoconus

Model	SVM
C	10
Class-Weight	balanced
Gamma	1
Kernel	poly

Table 5.4: STPI- Support Vector Machine Hyper Parameters Normal VS Subclinical Keratoconus

Model	Random Forest
Bootstrap	False
Class-Weight	balanced
criterion	entropy
max_depth	2
max_features	sqrt
n_estimators	100

Table 5.5: STPI- Random Forest Hyper Parameters Normal VS Subclinical Keratoconus VS Keratoconus

Abbreviations

KC	Keratoconus
SUSKC	Subclinical Keratoconus
AUBMC	American Universirt of Beirut Medical Center
RF	Random Forest
SVM	Support Vector Machine
PCBI	Posterior Curvature Biomechanical Index
LASIK	Laser-Assisted In Situ Keratomileusis
PRK	Photo-Refractive Keratectomy
SMILE	Small Incision Lenticule Extraction

Bibliography

[1] M. Romero-Jiménez, J. Santodomingo-Rubido, and J. Wolffsohn, "Keratoconus: a review," *Contact Lens and Anterior Eye*, vol. 33, pp. 157–166, Aug. 2010.

[2] A. H. Ali, N. Hussein, and Z. M. Musa, "Support vector machine for keratoconus detection by using topographic maps with the help of image processing techniques," *IOSR Journal of Pharmacy and Biological Sciences (IOSR-JPBS)*, vol. 12, no. 6, 2017.

[3] I. R. Hidalgo, P. Rodriguez, J. J. Rozema, S. N. Dhubhghaill, N. Zakaria, M.-J. Tassignon, and C. Koppen, "Evaluation of a machine learning classifier for keratoconus detection based on scheimpflug tomography," *Cornea*, vol. 35, no. 6, 2016.

[4] K. Cao, K. Verspoor, S. Sahebjada, and P. N. Baird, "Evaluating the performance various machine learning algorithms to detect subclinical keratoconus," *Translational Vision Science Technology*, vol. 9, no. 2, 2020.

[5] A. Lavric, V. Popa, H. Takahashi, and S. Yousefi, "Detecting keratoconus from corneal imaging data using machine learning," *IEEE Access*, vol. 8, 2020.

[6] M. B. Souza, F. W. Medeiros, D. B. Souza, R. Garcia, and M. R. Alves, "Evaluation of machine learning classifiers in keratoconus detection from orbscan ii examinations," *Clinics*, vol. 65, no. 12, 2010.

[7] K. Kamiya, Y. Ayatsuka, Y. Kato, F. Fujimura, M. Takahashi, N. Shoji, Y. Mori, and K. Miyata, "Keratoconus detection using deep learning of colour-coded maps with anterior segment optical coherence tomography: A diagnostic accuracy study," *BMJ Open*, vol. 9, no. 9, 2019.

[8] M. Hearst, S. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intelligent Systems and their Applications*, vol. 13, no. 4, pp. 18–28, 1998.