Driver Recognition and Driver Verification Using Data Mining Techniques

KRİSTİN SURPUHİ BENLİ

IŞIK UNIVERSITY
2007

Driver Recognition and Driver Verification Using Data Mining Techniques

KRİSTİN SURPUHİ BENLİ

Submitted to the Graduate School of Science and Engineering
in partial fulfillment of the requirements for the degree of
Master of Science in
Computer Engineering

IŞIK UNIVERSITY
2007

Driver Recognition and Driver Verification
Using Data Mining Techniques

APPROVED BY:

Assist. Prof. M. Taner ESKİL (Thesis Supervisor)	(Işık University)	_____________________
Prof. Dr. Selahattin KURU	(Işık University)	_____________________
Assist. Prof. Hasan F. ATEŞ	(Işık University)	_____________________

APPROVAL DATE : 25/09/2007

Abstract

In this thesis we present our research in driver recognition and driver verification. The goal of this study is to investigate the affect of different classifier fusion techniques on the performance of driver recognition and driver verification. We are using five different driving behavior signals for identifying the driver identities. Driving features were extracted from these signals and Gaussian Mixture Models were used for modeling the driver behavior. Gaussian Mixture Model training was performed using the well-known EM algorithm. In recognition study posterior probabilities of identities called scores were obtained with the given test data. These scores were combined using different fixed and trainable (adaptive) combination methods. In verification study we compared posterior probabilities with fixed threshold values for each classifier. For different thresholds, false-accept rate versus false-reject rate was plotted using the receiver operating characteristics curve. We observed lower error rates when we used trainable combiners. We conclude that combined multi-modal signal or classifier methods are very successful in biometric recognition and verification of a person in a car environment.

Veri Madenciliği Teknikleri kullanılarak Sürücü Tanıma ve Sürücü Doğrulama

Özet

Bu tez sürücü tanıma ve sürücü onaylama çalışmalarını içermektedir. Bu çalışmalar için sürücülerden toplanan beş değişik davranış işaretleri kullanılmıştır. Bu işaretler yardımıyla sürücülerin öznitelikleri çıkarılmış ve Karma Gauss Dağılım Modelleri kullanılarak sürücü davranışları modellenmiştir. Karma Gauss Dağılım Modellerinin eğitilmesi için Beklenti Enbüyütme algoritması kullanılmıştır. Sürücü tanıma çalışması için kimlikleri sınama verileri kullanılarak art olasılıklar elde edilmiş ve bu olasılıklar aynı zamanda her sınıf için puan olarak kabul edilmiştir. Bu puanların tümleştirilmesi için sabit kurallar ve eğitilebilir tümleştiriciler kullanılmıştır. Sürücü doğrulama çalışması için olabilirlik oranının bir eşikle karşılaştırılması yapılmıştır. Değişik eşik değerleri için yanlış kabul-yanlış red sıklıklarını grafiklemek için alıcı işletim eğrisi kullanılmıştır. Bu çalışmanın amacı değişik sınıflandırıcı tümleştirme yöntemlerinin sürücü tanıma ve sürücü doğrulama performanslarına etkilerinin incelenmesidir. Eğitilebilir tümleştirme yöntemleri ve sürücü davranış sinyalleri kullanılarak sürücü tanımasında ve doğrulamasında düşük hata oranları elde edilmiştir. Sonuçlarımız çok modlu sürüş sinyallerinin sınıflandırıcı tümleştirme yöntemleri ile kullanıldığında sürücünün araba içi şartlarda tanınması ve onaylanmasında çok etkili olduklarını göstermiştir.

Acknowledgements

I want to present my gratitude to Prof. Dr. Ersin Kalaycıoğlu, Prof. Dr. Yorgo İstefanopulos and Prof. Dr. Selahattin Kuru for giving me the chance for being a member of Işık University as a master student and a teaching assistant.

I want to thank Prof. Dr. Selahattin Kuru. It is very big chance and honor to me, working with him. I learnt a lot of valuable things from him as a student and a teaching assistant.

I am indebted to my advisor Assist. Prof. Taner Eskil for his patience, support and advices. He always shares his experiences with me. It is very big pleasure working with him.

Also, many thanks to the scientific and technological research council of Turkey (TÜBİTAK) for graduate scolarship.

One of the reason for choosing this topic is that, the courses “Signals and Systems”, “Digital Signal Processing” and “Advanced Data Warehousing and Data Mining”, which gives me pleasure when I deal with them. I want to thank the instructors, Assist. Prof. Ayşe Ufuk Ağar, Assist. Prof. Tamer Dağ and Prof. Dr. Hasan Dağ, who taught these courses to me.

I am also indebted to Prof. Dr. Aytül Erçil from Sabancı University, Emrecan Çökelek, Hakkı Can Koman from OTAM and Nagoya University for sharing the drivers data with me.

Besides all these people, I want to thank my dear friends Gülşen Aydoğan, Hilal Karatoy, Rina Barbut, Fethiye Erdem and Can Ulucan, for their close friendship and encourage. Also, I want to thank to Gülce Gül, who patiently made corrections of the thesis format and Sevgi Dikmen, who answers all my questions sincerely.
I want to thank all these people that provide my improvement in computer engineering. They help me, they care me, they work with me and they are always next to me. I hope I won't disappoint them depend on their all help.

Finally, I want to give my special thanks to my family, Janet-Garabet Benli, Mari-Sirop Özçulcu, Surpuhi-Agop Benli, Linda-Melissa Gülağacı, Belinda Özçulcu, Hilda-Nerses-Maral Yağar. Thanks a lot for their unconditional love and support.

To my parents...

Janet-Garabet Benli

Table of Contents

List of Tables

List of Figures

List of Abbreviations

CIAIR	Center for Integrated Acoustic Information Research
DET	Detection error- trade off
DPT	The State Planning Organization
EER	Equal Error Rate
EM	Expectation Maximization
FAR	False Accept Rate
FER	Failure to Enroll Rate
Fisher	Fisher Classifier
FMR	False Match Rate
FNMR	False Non-Match Rate
FRR	False Reject Rate
FTC	Failure to Capture Rate
GMM	Gaussian Mixture Model
KNN	K-Nearest Neighbor Clssifier
LDC	Linear Discriminant Combiner
Max	Maximum Rule
Mean	Mean Rule
Median	Median Rule
Min	Minimum Rule
MLP	Multi-layer Perceptions
NB	Naive Bayes Combiner
NLC	Normal Density Based Linear Classifier
NMC	Nearest Mean Combiner
NQC	Normal Density Based Quadratic Classifier
OTAM	Automotive Technology Research and Development Center
Parzen	Parzen Combiner
PCA	Principal Component Analysis
PDF	Probability Density Function
Perl	Perl Classifier
Prod	Product Rule
ROC	Receiver (or relative) Operating Characteristic
WNMC	Weighted Nearest Mean Classifier

Chapter 1
Introduction

Identification of a person in a vehicle using solely behavioral signals is a new research area in which few scientific studies were made. This research field has many application areas such as authentication, access control, keyless entry, secure communications and automated personalization of vehicle controls.

Person recognition within a vehicle provides the following benefits [1]:

1. Vehicle safety: Require authorization before or during the driving of the vehicle to make sure that the current driver is an authorized driver. An unauthorized driver is denied the control of the vehicle.
2. Vehicle personalization: Suit the vehicle according to the driver’s physical and behavioral characteristics. So, a safer, more comfortable and more efficient driving environment is obtained and the distraction is minimized. Consequently many accidents may be prevented.
3. Secure mobile transaction opportunities: Mobile banking using biometric authentication may be an example of these opportunities.

Human beings recognize each other using biological characteristics like face, voice or gait. One of the well-known characteristic of the person is the fingerprint which is used more than hundred years for person identification. Biometrics is the study of methods for recognizing people according to one or more real physical or behavioral properties. In this thesis biometrics-based driver identification and verification is done.

Some of the limitations inherent in unimodal biometric systems can be overcome by using multiple biometric modalities like face and fingerprint of a person or multiple fingers of a person. These systems are known as *multimodal biometric systems* and they are expected to be more reliable due to the presence of multiple, independent pieces of evidence [2].

A biometric system is a pattern recognition system which consists of six components. The first component is data acquisition in which biometric data of the subject is collected. In data collection process sensors such as microphone; video camera are used. So, the format of the data is digital [3].

Data compression and decompression are the second and third components of the system and they are optional. The fourth component is the feature extraction algorithm which produces a feature vector. The components of the feature vector are numerical characterization of the biometrics. So, collected data of one subject at different times are similar. Also, these data are dissimilar or different from the other subject's data. The fifth component is the matcher. Matcher compares the feature vectors and obtains a similarity score. The similarity between two biometrics data is shown with this score. The final component of the system is decision maker and the result is the output of the system [3].

A biometric system may target either person recognition or person verification. The recognition process aims to find the answer of the question "Who is he/she?" In this process there exists biometric information of the subject in the database. At this time new information is taken from the subject and a comparison is done between this new biometric information and all the other stored biometric information. So, it is an example of 1-to-many comparison and this process is more difficult than the verification process [4].

The verification process aims to find the answer of the question "Is he/she the person he/she claims to be". In this process the subject claims that he/she is a person whose biometric information already exists or stored in the database. In this case again new biometric information is taken from the subject and a comparison is done between this new biometric information and claimed biometric information. If the new biometric information is matched with the stored biometric information, the verification process will end successfully and the subject is accepted otherwise he/she is rejected. So, it is an example of 1-to-1 comparison [4].

In this study we follow the steps of the biometric system which is explained in the above paragraphs. The first component of the system is the data acquisition. In our study we did not collect the data of the subjects. We use the CIAIR database of the Nagoya University for driver recognition and driver verification studies. So, our study did not involve the data acquisition stage. The second and the third components of a biometric system are data compression and decompression. These steps are optional also we skipped these steps. The fourth component is the feature extraction algorithm which is used for producing a feature vector. The components of the feature vector are numerical characterization of the biometrics. In this study Gaussian Mixture Model were used for modeling the driving behavior. The fifth component is the matcher which compares the feature vectors for obtaining a similarity score. In this step well-known Expectation Maximization algorithm is used for training the GMMs.

The final component of the system is the decision maker. In our study we make decision fusion and use two different combination methods such as fixed methods and trainable methods. Fixed methods have simple fixed rules to combine information from a set of classifiers. In this study five fixed rules were used. These rules are maximum rule, minimum rule, median rule, mean rule and product rule. Trainable methods have some free parameters that can be trained on a separate part of the training data. In this study seven trainable combiners were used. These combiners are fisher, linear discriminant, nearest mean, naïve bayes, perl, parzen and k-nearest neighbor.

In this thesis both recognition and verification experiments were carried out. The experiments were done with hundred subjects, fifty male and fifty female, and the results are reported. Also, a study was done for driver fatigue detection as a future work.

The goal of this study is to investigate the affect of different data mining techniques on the performance of driver recognition and verification. This thesis was done for detecting driver recognition and driver verification by using signal processing and data mining techniques. In this study the driving data of Center for Integrated Acoustic Information Research (CIAIR) and Automotive Technology Research and Development Center (OTAM) databases were used. Classifier fusion techniques were investigated for combining information from the different sources for improving the performance of a system.

We will report the detailed stages of our study and results. In the first part of the report a brief introduction will be given in the research domain. In the second part of this report, biometric identification is explained including the applications recognition and verification. In the third part a detailed discussion of the data acquisition stage and the "Uyanık" vehicle will be given. In the fourth part the classifier theory is explained with Gaussian Mixture Model and Expectation-Maximization Algorithm. In the fifth part fusion techniques are introduced and detailed descriptions are given for Fixed Rules and Trainable Combiners. In the sixth part brief an introduction to driver recognition and driver verification fields is given. In the seventh part the experiments are explained and the results are presented. Finally, a conclusion is done about the studies done in the context of this study, including future research directions.

Chapter 2
Biometric Identification

In Latin, the word bios has the meaning of "life" and the word metric has the meaning of "measure". So, Biometrics is the study of methods for recognizing *people* according to one or more real physical or behavioral traits.

2.1 Why Biometrics

The developments in the areas of networking, communication and mobility increased the need of reliable ways for verification of the identity of user. Identity verification is done in two ways such as possession-based and knowledge-based.

- **Possession-based:** In this type of identity verification the user has a token like a credit card or a document and all the security measures are satisfied with this token. If the token is lost, somebody else can use it to assume the owner's identity.

- **Knowledge-based:** In this type of identity verification the security is satisfied with a password. In this case the property of the password becomes very important. If the password is too short, then it becomes very easy to guess it by making several attempts. On the other hand if it is too complex than remembering it will become very difficult so the user may be need to write the password somewhere. At this time the risk of lost or stolen password will occur.

If the person's "self" becomes the key, the disadvantages of the standard validation systems can be removed. It is generally accepted that biometrics adds complexity to the identification systems. If a comparison done with the biometrics and standard systems, the following advantages of the biometrics can be listed.

- Biometric property can not be forgotten or misled. It can be lost, if person undergoes a trauma. In such a situation the person could also forget the passwords and lost tokens.
- It is difficult to copy, share and distribute biometric property of a person.
- Biometric property is done at a “point of authentication” and the person must be present at the time of authentication.

Biometric identification is generally not considered as a replacement to current systems. Used in combination, biometric and conventional security systems can be improved [4].

2.2 Common Biometric Characteristics

Biometric characteristics can be divided into two main category such as Physiological and Behavioral.

- **Physiological:** This category is related to the shape of the body. One of the most well-known physiological properties of humans is fingerprints. Fingerprint identification technique is used for longer than 100 years. Other examples of this category are face recognition, hand geometry and iris recognition.

- **Behavioral:** This category is related to the behavior of the person. The signature of a person was the first behavioral characteristic that was used. It is still used in many applications. The more modern examples are keystroke dynamics and voice.

Voice is also a physiological property; every person has a different pitch depending on the vocal track. Since voice recognition is the study of the way of a person speaks, as a result it is generally classified under the behavioral category.

Moreover there are many other examples of biometric identification such as gait, retina, hand veins, ear, facial thermogram, DNA, odor and palm prints [4].

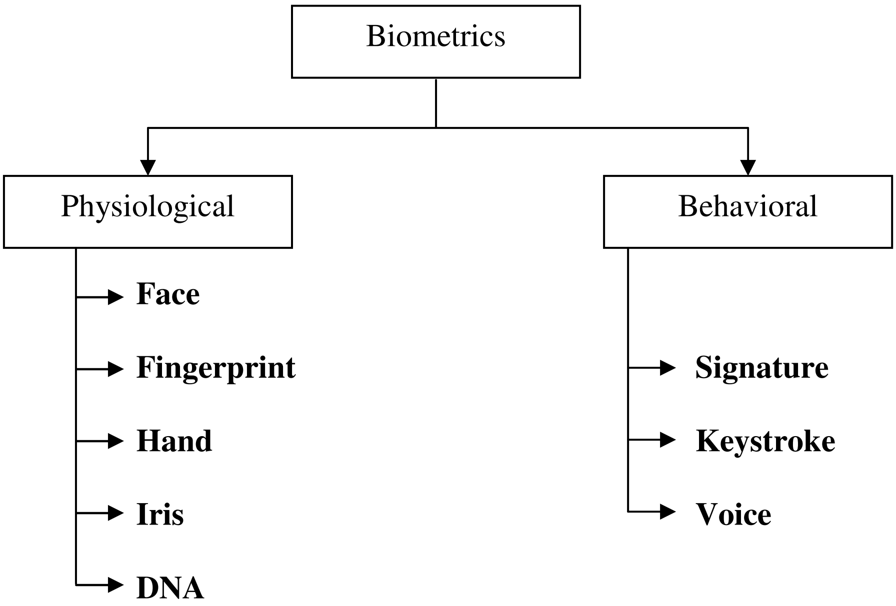

Figure 2.1 – Classification of some biometric properties [4]

2.3 Biometric Systems

Figure 2.2 depicts the block diagram of common biometric systems. Two main operations in a biometric system are enrollment and test. In the enrollment operation biometric information of the person is processed and stored. In the test operation biometric information is extracted and compared with the other stored biometric samples.

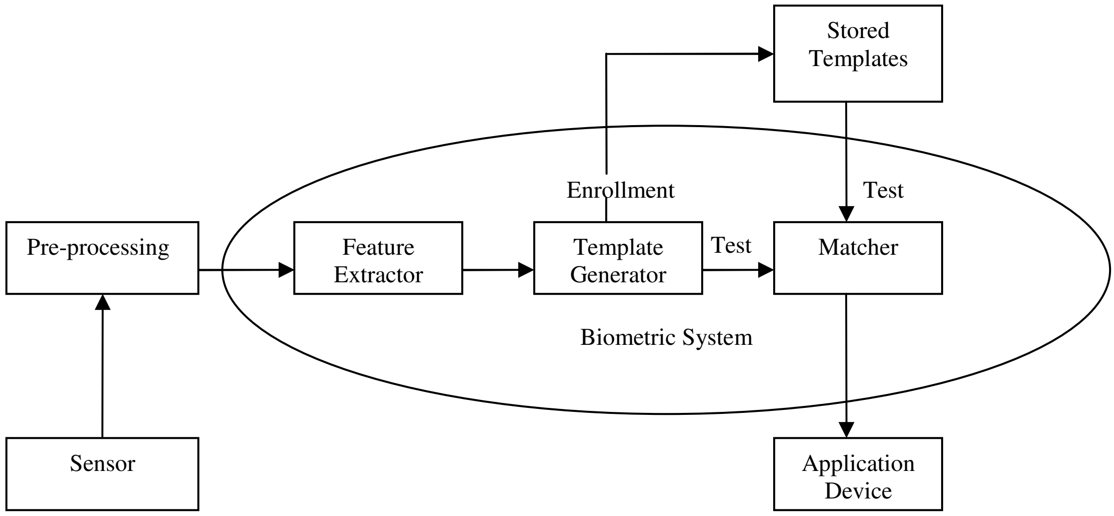

Figure 2.2 – The Block diagram of biometric systems [4]

In the first part of the block diagram, a sensor or a set of sensors is used for collecting all necessary data. It is the interface between the real world and the system. The property of the collected data can be changed according to the characteristics that are planned to consider.

In the second part of the block diagram all pre-processing operations are done. During the pre-processing operation the artifacts of the sensor are removed, input is enhanced by removing some noise; some kind of normalization is done, etc.

In the third part of the block diagram the needed features are extracted. The important questions are which features to extract and how to choose.

In the fourth part of the block diagram a template is created by data. These data may be a vector of numbers or an image with particular properties. The template is the combination of the all characteristics which are extracted from the source. Template must be as short as possible but must not eliminate too much information. Behavior of the system is changed according to what is requested like recognition or verification.

For performing the enrollment operation, the template must be stored somewhere like in a card or in a database. In the matching phase, obtained template is passed to a matcher for making comparison between the obtained template and other existing templates. A number of algorithms can be used for estimating the distance between them [4]. Finally the matcher decision is sent as output, which can be used for recognition purpose.

2.4 Functions

A biometric system has two important functions such as verification and recognition/identification.

2.4.1 Person Verification

Verification is the process that aims to find the answer to the question "Is he/she the person who he/she claims to be?". In this process the subject claims to be a person whose biometric information are already existent or known. This information may be stored in a database or on a card. During this process new biometric information is taken from the subject and a comparison is done between the new biometric information and the stored data. If the new information is matched with a stored template, the verification process will finish successfully. So, this type of check is called 1:1 match verification [4].

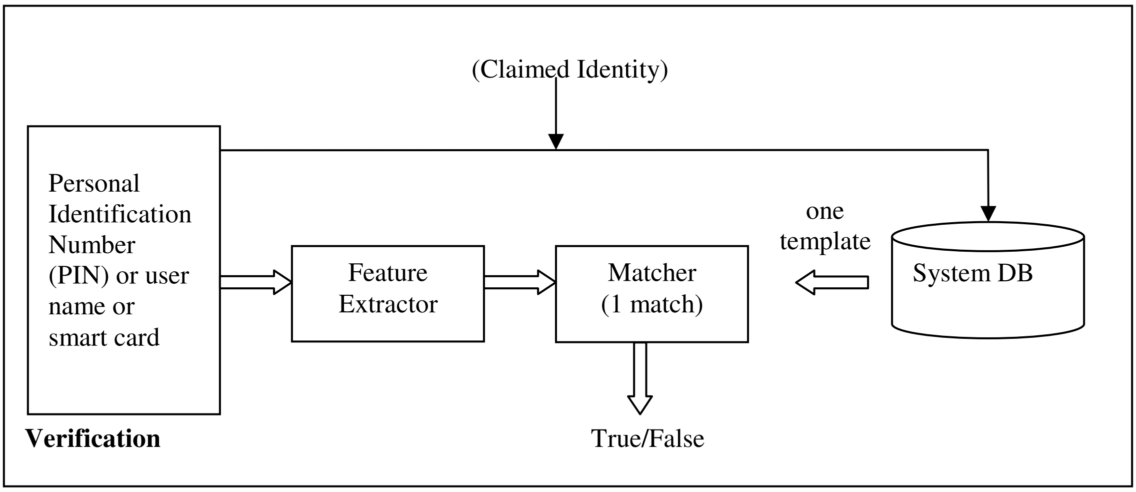

Figure 2.3 – Block diagram of verification task by using the four main modules of a biometric system such as sensor, feature extraction, matcher, and system database [2]

2.4.2 Person Recognition / Identification

Identification is the process that aims to find the answer to the question "Who is he/she?". In this process biometric information of the subject is taken then this information is compared with the other data which are stored in the database. This process is more difficult than the verification process. Since, the subjects' information is compared with all the other subjects in the database [4].

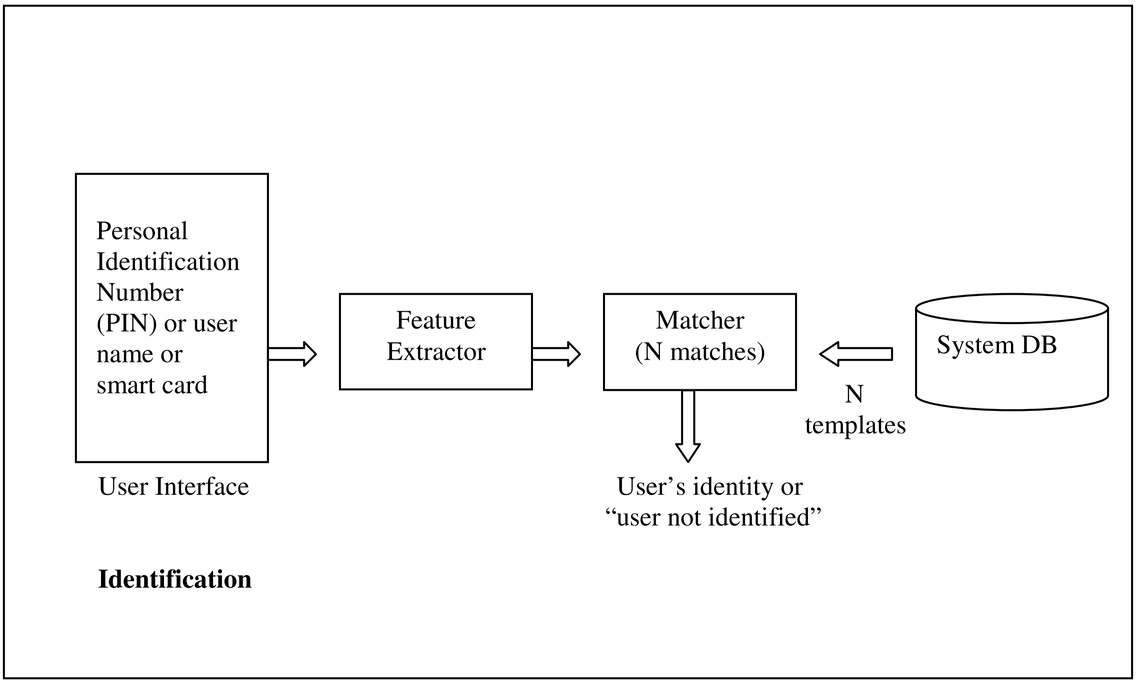

Figure 2.4 – Block diagram of identification task by using the four main modules of a biometric system such as sensor, feature extraction, matcher, and system database [2]

2.5 Performance Measurement

The performance of verification systems can be measured in the following dimensions:

- **False Accept Rate (FAR) or False Match Rate (FMR)**

False accept rate is the probability that the system incorrectly declares a successful match between the input pattern and a non-matching pattern in the database. It gives the percent of invalid matches as an output. This type of error is very critical for security issues because these invalid users are accepted by the system. In secure systems these kinds of entries are forbidden to the non-allowed people.

- **False Reject Rate (FRR) or False Non-Match Rate (FNMR)**

False reject rate is the probability that the system incorrectly declares a failure of match between the input pattern and a non-matching pattern in the database. It gives the percent of valid users who are rejected as an output.

- **Receiver (or relative) Operating Characteristic (ROC)**

Generally, the matching algorithm makes a decision using some parameters such as a threshold. If we change these parameters, the false accept rate and false reject rate can typically be traded off against each other. ROC plot is obtained by graphing the values of false accept rate and false reject rate, also changing the variables implicitly. Detection error- trade off (DET) is a common variation and obtained using logarithmic scales on both axes. This more linear graph illuminates the differences for higher performances.

- **Equal Error Rate (EER)**

The equal rate is the rate where both accept and reject rates are equal. ROC or DET plot is used for showing the performance of a biometric system. These plots show how FAR and FRR can be changed. EER is used for making quick comparison between two systems. Also, EER can be obtained from the ROC plot by taking the point where FAR and FFR have the same value. If the EER is lower, the system considered to be more accurate.

- **Failure to Enroll Rate (FTE or FER)**

Failure to enroll rate shows the percentage of people who fail to enroll in the system. If the data which was obtained by the sensor is invalid, failure to enroll will happen.

- **Failure to Capture Rate (FTC)**

Failure to capture rate is the probability that the system fails to detect a biometric characteristic if it is presented to it correctly.

- **Template Capacity**

Template capacity is the maximum number of people that is possible to discriminate. For example if we use a template of *n* bits and choose the features so that each individual generates a different template, then theoretically 2^n individuals are discriminated. But such ideal features can not be found and the noise and a certain range of uncertainty have to be considered, so the capacity must be much smaller than 2^n [4].

Chapter 3
Data Acquisition

The State Planning Organization (DPT) supported Drive-Safe project which aims to increase driving safety and decrease the loss of manpower, wealth and death. A consortium of three universities, three automotive companies and a research center realize the Drive-Safe project. The objective of the study is to collect data from driver, passenger, car and road condition by using cameras, microphones and the other sensors. The next stages of the project aim to analyze these data and automatically determine dangerous situations such as driver fatigue. The ultimate target is to take precautions and make necessary warnings to improve the driving safety.

The first stage of the project aims to realize an infrastructure to collect data, prepare data collection scenarios, design and implementation of database for using the data robustly and finally collect the data from the vehicle and simulator environment.

3.1 Vehicle for Data Collection

Ford and Renault Companies donated a car each for collecting data. In first step, Renault Company's Megan car is used for the data collection, because Megan car has a CanBus.

The first vehicle's all sensors are completed and put into the car by the Renault Team, and this vehicle is named as "Uyanık". Figure 3.1 shows the sensors of the vehicle and how they were mounted in the vehicle. In Table 3.1 detailed list and properties of the sensors are given. Figure 3.2 shows some of the sensors in the vehicle.

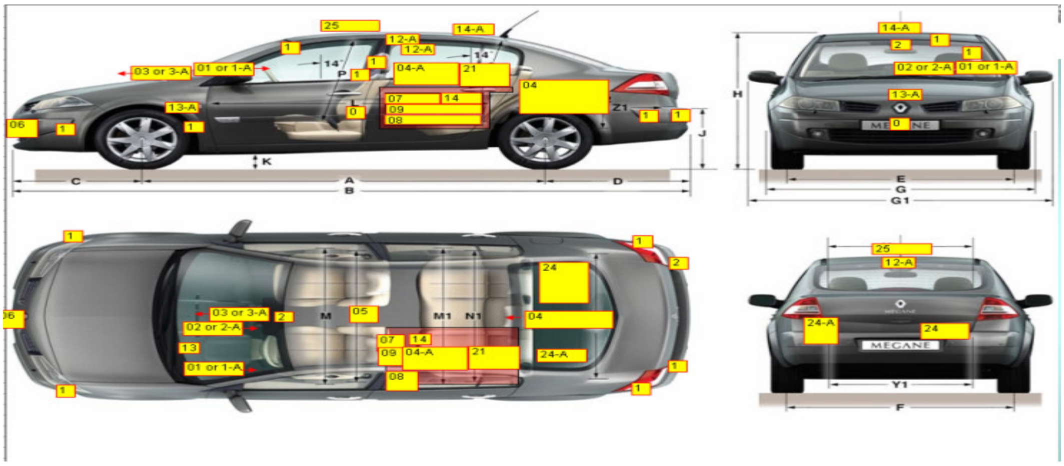

Figure 3.1 - The sensors of the vehicle and their settlement into the vehicle.

Laser Scanner | Sonar Sensors

GPS | Cameras

EEG and Mouth Microphone | Data Collection Center

Brake Pressure Sensor | 3 Axis Angular Speed-Acceleration Measure

Figure 3.2 - Some of the sensors in the vehicle 'Uyanık'

Table 3.1- Detailed list and properties of the sensors

Product #	Sensor	Product Code / Producer	Supply Socket	Power Usage (Watts) Total	Interface	Dimensions (HxBxT)
1	High resolution camera	Basler A601fc-2	Firewire	0	Firewire	0
1-A	Night vision camera	Basler A601fc-HDR	Firewire	0	Firewire	0
2	High resolution camera	Basler A601fc-2	Firewire	0	Firewire	0
2-A	Night vision camera	Basler A601fc-HDR	Firewire	0	Firewire	0
3	High resolution camera	Basler A601fc-2	Firewire	0	Firewire	0
3-A	Night vision camera	Basler A601fc-HDR	Firewire	0	Firewire	0
4	Video Image Acquisition system	NORPIX-StreamPix, digital video recording software version 3.20.1, USB product key, Pentium 4 computer	220/110V AC	250	Inputs = 1 parallel, 1 optic, 1 eSATA, 6 USB, 5 firewire, 1 ethernet, 4 line out sound, 2 line in sound, 1 VGA, 1 Gamepad	
4-A	Monitor + mouse + keyboard					
5	XYZ accelerometer	IMU 400 Crossbow	9-25 V DC	3	15 PinD Output Connector Male / 6 Digital Input or DB-9 Standard Com Port	
6	2D laser scanner	LMS221-30206	24 V DC	20	DB-9 Std. COM Port Output / Input	
7	EEG	GRASS TELEFACTOR AURA24	Self Powered	0	RS232	
8	24 channel Data Acquisition Device	Alesis ADAT HD24	110 / 220 VAC	60	Audio inputs = 24 x 1/4" TRS jacks, Audio outputs= 24 x 1/4 TRS jacks	133mm x 483mm x 342 mm
9	8-channel Input Signal Mixer Amplifiers to convert 8 audio channels in Alesis	Behringer/UltraGain Pro8 Digital ADA8000 ADC/DAC	110V AC	25	MIC IN->XLR Female LINE IN->1/4" TRS LINE OUT-> XLR Male DIGI IN-> Toslink-Optical DIGI OUT-> Toslink-Optical	44.5mm x 482.6mm x 217 mm
10	Driver headphone and mouth microphone	Beyer Dynamic			Stereo Audio IN/OUT	
11	Visor Microphone	Sony ECM-C115	lithium batery CR2025	0	L shape Male Stereo Jack	
12	Collar Microphone	Sony ECM-C115	lithium batery CR2025	0	L shape Male Stereo Jack	
12-A	Room Microphone	Sony ECM-C115	lithium batery CR2025	0	L shape Male Stereo Jack	
13	Break Pedal Pressure Sensor	Custom beak sensor made in Japan		0	Special Connector	
13-A	Break Pedal Pressure Sensor Amplifier	Custom amplifier made in Japan	12 DC	0	Special Connector	
14	GPS receiver	Trimble Pathfinder Pro XRS + Omnistar VBR 1 year subscription	10-32 V DC	7	DB-9 Std. COM Port Output / Input	
14-A	GPS receiver					
15-20	Megane Layout and Sensors for distance measurement	SensComp 600 Pakage (2x4 pcs)	4,5-6,8 V DC	53.2	JST Polaroid Coonector Output / 4 Digital Input	
21	Laptop	Toshiba Tecra S3 Sonoma 2Ghz/Alternative	AC-DC			
22	PCMCI serial transformer for Laptop	Quatech QSP-100, 4 port RS-232 serial PCMCIA Card			4 port RS-232	
23	Cellular Phone	Sony Ericson W800i Walkman-Phone	Self Powered	0	Handset microphone and headphone	100mm x 46mm x 20.5mm
24	12V DC - 220 AC Power Inverter	Pure Sine Wave 1500 W S1500-212E3	12 DC	1500 W	Special Cable to Megane Battery; 220 ground two exit socket	390mm x 275mm x 105mm
24-A	Power distributor panel					
25	Panoramic and remote camera	RPU C2512 FG Panoramic Camera + RPU SP Remote Control Camera + RPU K 3m Cable				

3.2 Data Collection Studies with CAN Data Bus

The data on the CAN data bus has an important role for modeling driver behavior, for security related recognition/verification and fatigue detection tasks. The following data were obtained from the CanBUS of the vehicle:

- Steering Wheel Angle
- Steering Wheel Angular Speed
- Engine Period
- Vehicle Speed
- Every Wheel’s Angular Speed

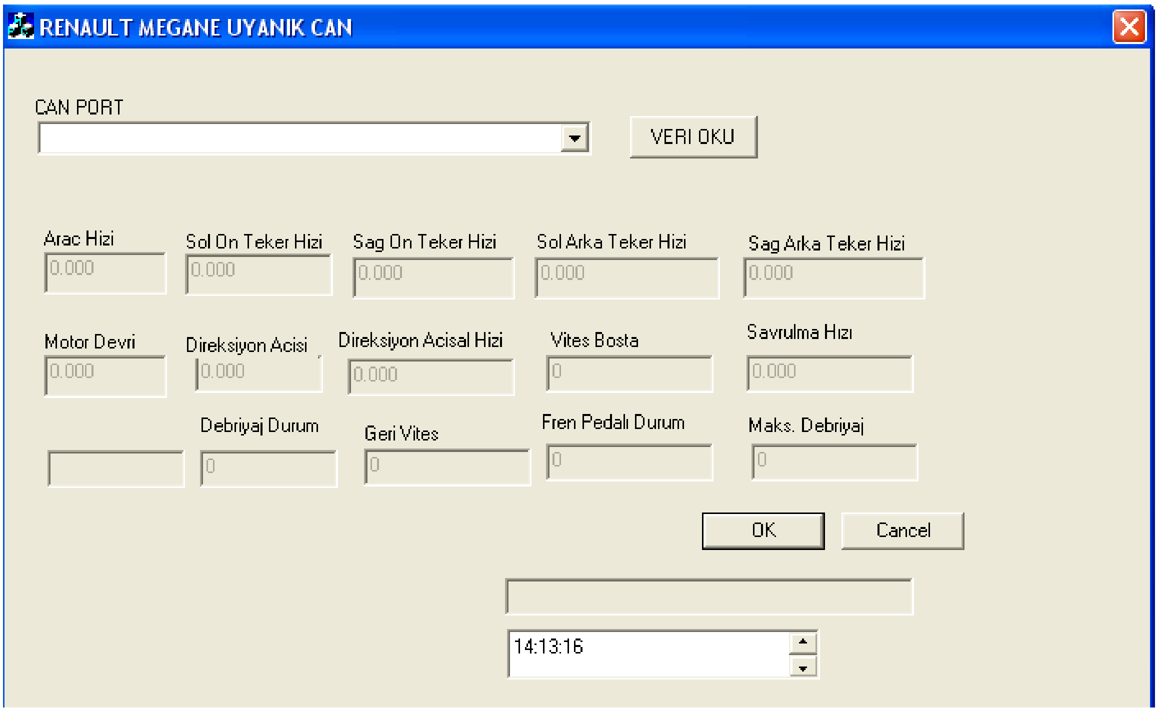

Figure 3.3 – Interface for CAN Data Bus

A card Canalyzer is attached to the data collection computer’s PCMCI socket. Data can be observed and recorded at required intervals, with the help of this card and the program which is developed in Visual Studio platform and C++ language.

3.3 Synchronization of Sensors

For the activity of data analysis, it is important to gather the data from vehicle's instruments in a synchronous way. All sensor systems have different interfaces. For running these systems synchronously, it is first planned to send a trigger signal to the systems, as a result of this, trigger signal and different modules record starting time. However only marking the starting times during the records can cause some problems. Therefore, it is planned to send the trigger signals with certain time intervals at the record time. The systems, that are used to trigger signal, are different from each other and additional electronic circuit is needed for some of the systems. In this system, times of all systems in the vehicle are arranged according to a reference point. As a result of this, during the time marked voice, image, and other signals synchronization of each other is satisfied for required any time intervals. Graphical representation of the system is depicted in Figure 3.4.

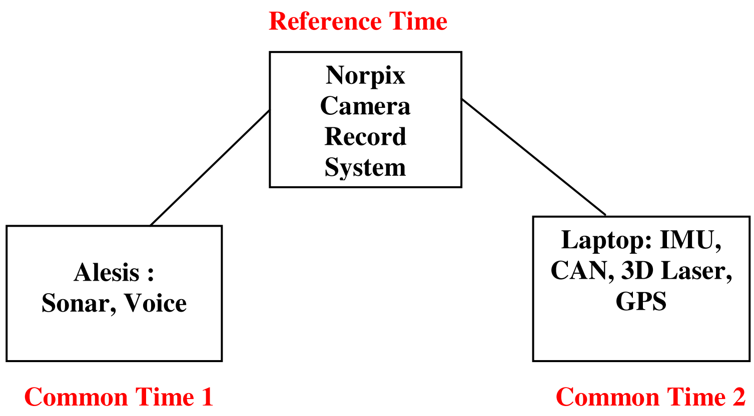

Figure 3.4 - Synchronization structure

3.4 Determination of Data Collection Track

Most of the fatigue-related experiments will be done in the simulator environment. For determining simulator experiments realities, a track is determined, 3 dimensional model of this track will be rendered to the simulator. According to the objective the following track is determined. The track is started from OTAM, which is a research center in ITU Ayazağa Campus, certain proportion main road, certain proportion city traffic. Also, this study is determined in coordination between Japanese partners as a scope of the NEDO project. Figure 3.5 shows the Maslak track of the experiment. This track is planned to take about 40 minutes.

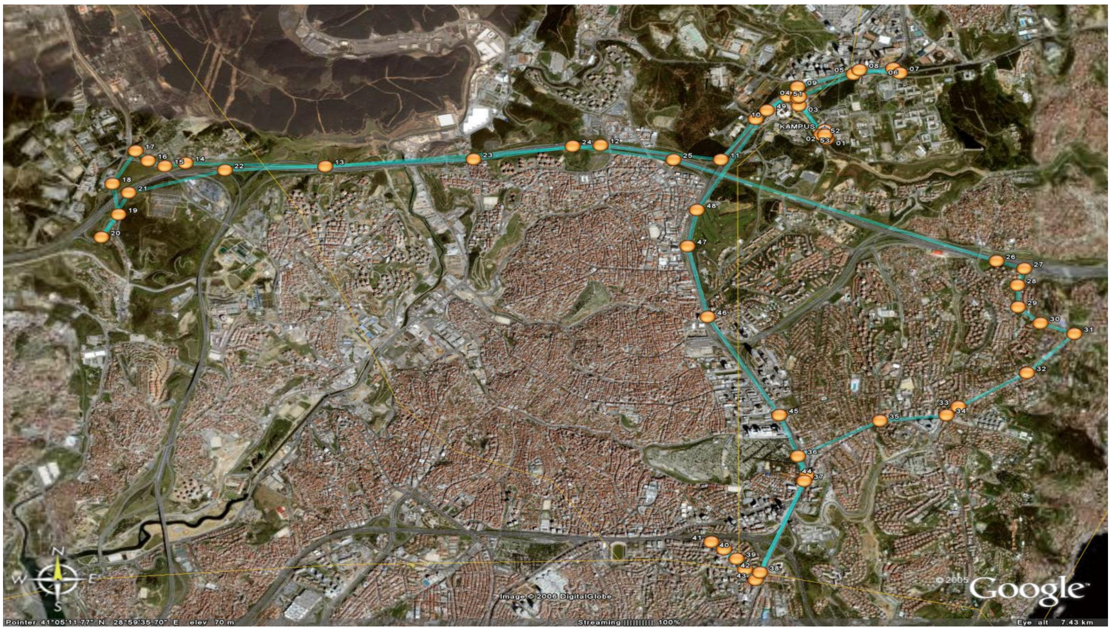

Figure 3.5 - Track of experiment vehicle

Chapter 4
Classifier Theory

Classifier is an algorithm which takes features as input and interprets what it means according to the information. The information is encoded into the algorithm. At the end it may give a single output like a label or a range of confidence values.

Knowledge of a classification task combines into a classifier by selecting a suitable classifier type such as neural network, a distance transform or Bayesian classifier. Also, knowledge is required for deciding an appropriate inner structure for the classifier such as the number of neurons and layers in a neural network classifier. In Bayesian classifier the probability density models or functions are selected. These choices are important for the determination of the classifier complexity.

Classifier complexity is a trade off between representational power and generality for a task. A simple classifier may not be talented for learning or representing classes well which bring in poor accuracy. For example an over-fitted classifier may classify the training data 100 percent correct. However when a different data set for the same task is offered, the accuracy may be poor. Because of this the training data is usually divided into two disjoint set such as an actual training set and a test set for estimating the classifier performance objectively.

A classifier may have numerous parameters that have to be adapted according to the task. This process is called *training* or *learning*. In the case of supervised learning the training samples are labeled and the aim is to minimize the classification error of the training set using training algorithm. In the case of unsupervised learning or clustering the training samples are not labeled however training algorithm aims to find clusters and form classes. Also, in the case of reinforcement learning the training labels are not labeled however the training algorithm uses feedback such as it tells that it classifies a sample correctly or not [5].

4.1 Bayesian Classification

A fundamental of the Bayesian classification and decision making is the probability theory. It aims to choose the most probable or the lowest risk option. Suppose that there is a classification task for classifying feature vectors to K different classes. A feature vector is represented as $x = [x_1, x_2, ..., x_D]^T$ where D represents the dimension of a vector. The probability that a feature vector x belongs to class ω_k represented as $P(\omega_k \mid x)$, and it is mentioned as a posteriori probability. According to the posterior probabilities the classification vector is done or decision risks are calculated.

The posterior probabilities computed using Bayes rule

$$P(\omega_k \mid x) = \frac{p(x \mid \omega_k) P(\omega_k)}{p(x)} \tag{4.1}$$

where $p(x \mid \omega_k)$ represents the probability density function of class ω_k in the feature space and $P(\omega_k)$ represents the a priori probability, that denotes the probability of the class before measuring any features. In the case of the unknown prior probabilities, they can be estimated by the class proportions in the training set. The divisor

$$p(x) = \sum_{i=1}^{K} p(x \mid \omega_i) P(\omega_i) \tag{4.2}$$

is only a scaling factor that guarantees that posterior probabilities are really probabilities such as their sum is equal to one.

The class-conditional probability density function $p(x \mid \omega_k)$ is the major problem for Bayesian classifier. The function describes the distribution of feature vectors in the feature space inside a particular class. Generally it is always known except in some artificial classification tasks. The distribution evaluated from the training set using some methods [5].

4.2 Gaussian Mixture Probability Density Function

In one dimension the Gaussian probability density function is a bell shaped curve described by two parameters such as mean μ and variance σ^2. In D-dimensional space it is described in a matrix form as

$$N(x;\mu,\Sigma) = \frac{1}{\sqrt{(2\pi)^{D/2}\,|\Sigma|}} \exp\left\{-\frac{1}{2}(x-\mu)^T \sum{}^{-1}(x-\mu)\right\} \tag{4.3}$$

where μ represents the mean vector, Σ represents the covariance matrix.

The Gaussian distribution is usually a good approximation for a class model shape in an appropriate selected feature space. It is a mathematically sound function and extends to multiple dimensions. In the Gaussian distribution there is an assumption that the class model is a complete model of one basic class. It fails when the actual model, the actual probability density function, is multimodal. For example, assume that we are searching for different face parts from a picture and there are many basic types of eyes, because the people are from different races. Therefore single Gaussian approximation defines a wide mixture of all eye type also including the patterns which do not look like an eye.

Gaussian Mixture Model (GMM) is a mixture of several Gaussian distributions. Therefore it can represent different subclasses inside a one class. The probability density function is represented as a weighted sum of Gaussians

$$p(x;\theta) = \sum_{c=1}^{C} \alpha_c N(x;\mu_c,\Sigma_c) \tag{4.4}$$

where α_c represents the weight of the component *c*, $0 \prec \alpha_c \prec 1$ for all components, and $\sum_{c=1}^{C} \alpha_c = 1$. The parameter list

$$\theta = \{\alpha_1, \mu_1, \Sigma_1, ..., \alpha_C, \mu_C, \Sigma_C\} \tag{4.5}$$

describes a particular Gaussian mixture probability density function.

Estimation of the Gaussian mixture parameters for one class is referred to as *unsupervised learning*. This is the case where the samples are generated by individual components of the mixture distribution and there is no knowledge about which sample was generated by which component. Clustering attempts to identify the exact components, but Gaussian mixtures can also used as approximation of an arbitrary distribution [5].

4.3 Maximum Likelihood Estimation

The class-conditional probability density functions are determined for the construction of the Bayesian classifier. The initial model selection can be done by visualizing the training data. But the model parameters' setting requires measure of goodness, like how well the distribution fits the observed data. So, data likelihood is an example of goodness value.

Let us assume that $X = \{x_1, ..., x_N\}$ is a set of independent samples and drawn from a single distribution which is described by a probability density function $p = (x; \theta)$ where θ represents the PDF parameter list. The likelihood function

$$\mathcal{L}(X; \theta) = \prod_{n=1}^{N} p(x_n; \theta) \tag{4.6}$$

shows the likelihood of the data *X* given the distribution or given the distribution parameters θ. The aim is to find $\hat{\theta}$ which maximizes the likelihood.

$$\hat{\theta} = \arg\max_{\theta} \mathcal{L}(X;\theta) \tag{4.7}$$

This function is not maximized directly but the logarithm

$$L(X;\theta) = \ln \mathcal{L}(X;\theta) = \sum_{n=1}^{N} \ln p(x_n;\theta) \tag{4.8}$$

called the log-likelihood function. Also, it is easy to handle. It is the same using $\mathcal{L}$ $(X;\theta)$ or $L(X;\theta)$ because the logarithm functions is monotone.

According to $p = (x;\theta)$ it is possible to find the maximum analytically by setting the derivatives of the log-likelihood function to zero then solving for θ. This can be done for a Gaussian PDF, which leads to the estimates for a mean and variance. But the analytical approach is intractable. In practice an iterative method such as expectation maximization (EM) algorithm is used. In some cases maximizing the likelihood leads to singular estimates, this is the fundamental problem of maximum likelihood methods with Gaussian Mixture Models.

Consider the task of classifying vectors into *K* classes. If the different classes can be seen as independent, the estimation problem of *K* class-conditional PDFs can be divided into *K* distinct estimation problems [5].

4.4 Basic EM Estimation

The expectation maximization (EM) algorithm is an iterative method which is used for calculating maximum likelihood distribution parameter estimates from incomplete data. Also, it is used for handling the cases where an analytical approach for maximum likelihood estimation is feasible like Gaussian mixtures with unknown and unrestricted covariance matrices and means.

Suppose that each training sample contains both known and missing features. Mark all good features of all samples with X and all missing features of all samples with Y. The expectation step (E-step) for the EM algorithm has the form

$$Q(\theta;\theta^i) \equiv E_Y\left[\ln \mathcal{L}\ (\mathrm{X},\mathrm{Y};\theta) \mid \mathrm{X};\theta^i\right] \tag{4.9}$$

where θ^i represents the presents the previous estimate for the distribution parameters and θ represents the variable for a new estimate describing the distribution. $\mathcal{L}$ represents the likelihood function which is described in equation (4.6). This function calculates the likelihood of the data, including the missing feature Y marginalized with respect to the current estimate of the distribution described by θ^i. The maximization step (M-step) maximizes $Q(\theta;\theta^i)$ with respect to θ and the set

$$\theta^{i+1} \leftarrow \arg\max_{\theta} Q(\theta;\theta^i) \tag{4.10}$$

Repeat the steps until a convergence criterion is met.

It is suggested for the convergence criterion that

$$Q\left(\theta^{i+1};\theta^i\right) - Q\left(\theta^i;\theta^{i-1}\right) \le T \tag{4.11}$$

where suitably selected T and

$$\left\|\theta^{i+1} - \theta^i\right\| \le \varepsilon \tag{4.12}$$

appropriately chosen vector norm and ε. It is common for both of the criteria that iterations are stopped when the change in the values falls below a threshold. More complicated criterion can be derived using equation (4.11) by using relative rate of change instead of absolute rate of change.

EM algorithm starts with an initial guess θ^0 for the distribution parameters. The log-likelihood increase on each iteration until it converges. The convergence takes to a local or global maximum. But also it takes to singular estimates which are true especially for Gaussian mixture distributions with arbitrary covariance matrices.

One of the problems of EM algorithm is the initialization. The selection of θ^0 decides where the algorithm converges the boundary of the parameter space producing singular results. Some solutions for initialization are multiple random starts or clustering algorithms.

The application of the EM algorithm to Gaussian mixtures goes as follows. The known data X is interpreted as incomplete data. The missing part Y has the knowledge of which component produced each sample x_n. For each x_n there is a binary vector $y_n = \{y_{n,1},..., y_{n,C}\}$, where $y_{n,C} = 1$, if the sample produced by the component c, otherwise it is zero. The complete data log-likelihood is

$$\ln \mathcal{L}(X, Y; \theta) = \sum_{n=1}^{N} \sum_{c=1}^{C} y_{n,c} \ln(\alpha_c p(x_n \mid c; \theta)) \tag{4.13}$$

In the E-step the conditional expectation of the complete data log-likelihood is computed with the Q-function, given X and the current estimate θ^i of the parameters. The complete data log-likelihood $\ln\mathcal{L}(X,Y;\theta)$ is linear with respect to the missing Y, the conditional expectation $W \equiv E[Y \mid X, \theta]$ is computed and put it into $\mathcal{L}(X,Y;\theta)$. For this reason

$$Q(\theta, \theta^i) \equiv E[\ln \mathcal{L}(\mathrm{X}, \mathrm{Y}; \theta) \mid \mathrm{X}, \theta^{\mathrm{i}}] = \ln \mathcal{L}(\mathrm{X}, \mathrm{W}; \theta) \tag{4.14}$$

where the elements of W can be shown as

$$w_{n,c} \equiv E[y_{n,c} \mid X, \theta^i] = \Pr[y_{n,c} = 1 \mid x_n, \theta^i] \tag{4.15}$$

The probability calculated using Bayes Rule

$$w_{n,c} = \frac{\alpha_c^i p(x_n \mid c; \theta^i)}{\sum_{j=1}^{C} \alpha_j^i p(x_n \mid j; \theta^i)} \tag{4.16}$$

where α_c^i presents the a priori probability (of estimate θ^i) and $w_{n,c}$ represents the a posteriori probability that $y_{n,c} = 1$ after observing x_n. Also, $w_{n,c}$ represents the probability that x_n was produced by component c.

Apply the M-step to the problem of estimating the distribution parameters for C-component Gaussian mixture with arbitrary covariance matrices; the following iteration formula is obtained.

$$\alpha_c^{i+1} = \frac{1}{N}\sum_{n=1}^{N} w_{n,c} \tag{4.17}$$

$$\mu_c^{i+1} = \frac{\sum_{n=1}^{N} x_n w_{n,c}}{\sum_{n=1}^{N} w_{n,c}} \tag{4.18}$$

$$\Sigma_c^{i+1} = \frac{\sum_{n=1}^{N} w_{n,c}(x_n - \mu_c^{i+1})(x_n - \mu_c^{i+1})^T}{\sum_{n=1}^{N} w_{n,c}} \tag{4.19}$$

The new estimates are collected to θ^{i+1} in equation (4.5). If the convergence criterion in equations (4.11) and (4.12) is not satisfied, $i \leftarrow i+1$ and the equations between (4.16) to (4.19) are evaluated with new estimates.

The explanation of the equations (4.17) to (4.19) is intuitive. The weight α_c of a component represents the portion of samples belonging to that component. The computation is done by approximating the component-conditional PDF with the previous parameter estimates and taking the posterior probability of each sample point belonging to the component c in equation (4.16).

Also, the component mean μ_c and the covariance matrix Σ_c are estimated in the same way. First the samples are weighted with their probabilities of belonging to the component, and then the computations of sample mean and covariance matrix are done.

Note that the number of components C assumed to be known. Clustering techniques aim to find true clusters and components from a training set. But training a classifier needs a good approximation of the distribution of each class. So, C does not need to be guessed definitely. It is a parameter and represents the complexity of the approximation distribution. If C is too small, it prevents the classifier from learning the sample distributions well enough. If C is too large, it leads to an over fitted classifier. Moreover large values of C will lead to singularities in case the training data becomes insufficient [5].

Chapter 5
Data Fusion Techniques

Fusion has the meaning of combining information from different sources for improving the performance of a system. The most known example of fusion is the use of different sensors for detecting a target. In general, different inputs may be found from a single sensor at different times. Also, many experts make different processing from a single sensor at a given time. In this case, many experts may be "consulted" to come up with the decision with highest confidence.

Fusion can be used for many purposes like detection, recognition, identification, tracking, change detection, decision making, etc. Fusion has application areas of Defense, Robotics, Medicine, Space, etc.

The advantages of using an efficient fusion scheme are as follows [6]:

- Improvement of the confidence of decision
- Improvement of countermeasures performance (camouflage an object in all possible wave-bands)
- Improvement of performance in existence of unsuitable environmental conditions. For example some weather conditions like fog and smoke cause bad visible contrast also some weather conditions like rain cause low thermal contrast, so by combining both types of sensors better overall performance is obtained.

5.1 Types of Fusion Processes

Fusion processes can be categorized into the three such as low, intermediate and high level fusion according to the processing stage where the fusion takes place.

5.1.1 Low Level Fusion

Low level fusion is also called as *data fusion*. It combines raw data from the several sources for producing new raw data which is expected to be more informative than the inputs. In image processing, images present several spectral bands of the same scene. So, images are fused for producing a new image. The new image contains in a single channel has most of the information that is available in the different spectral bands. This single image can be used by an operator or an image processing algorithm instead of the original images. If the number of available spectral bands becomes very large, looking at the images separately becomes impossible. A precise (pixel-level) registration of the available images is required in this kind of fusion. If the various bands come from the same sensor, the registration is intrinsic. But it is more complicated than the bands which use several different sensors like SAR, IR, scanner, camera, etc. The aim of this kind of fusion is fusing relevant information from the different images. The problem is defining that what the relevant information is. Human beings do not make the classification on the basis of local information. They rather use secondary information and trust their global interpretation capabilities. Relevant information is defined as local variations by the most of the fusion algorithms. This becomes a poor modeling of relevant information if the noisy spectral bands (SAR) are used [6].

5.1.2 Intermediate Level Fusion

Intermediate level fusion is also called as *feature level fusion*. It combines several features like edges, corners, lines, texture parameters, etc into a feature map. Those features may come from several raw data sources like several sensors, different moments, etc. or come from the same raw data. Also, the objective is to find relevant features from between the available features. So, these features come from the several feature extraction methods. Moreover, limited number of relevant features must be obtained. The feature fusion methods are Principal Component Analysis (PCA), Diablo shaped Multi-layer Perceptions (MLP) for the non-linear counterpart, etc. In image processing, for segmentation or detection feature maps are computed as pre-processing. Several features like edges, corners, lines, texture parameters, etc are computed and then combined into a fused feature map. So, it can be used for segmentation or detection [6].

5.1.3 High Level Fusion

High level fusion is also called *decision fusion*. It combines decisions which come from several experts. In other words, if the experts return a confidence (score) instead of a decision, it is a decision fusion problem. For differentiating two cases, one of them is hard fusion and the other is soft fusion. Decision fusion methods are voting methods, statistical methods, fuzzy logic based methods, etc.

The above categorization does not include all possible fusion paradigms. For example, input and output of the fusion process may denote different levels of processing.

The fusion procedures are categorized into five categories according to their input and output characteristics like Data in-Data out, Data in-Feature out, Feature in-Feature out, Feature in-Decision out, Decision in-Decision out. In majority, Data in-Decision out may be considered as a sixth category. In spite of it is not the most promising approach but it is good practice for extracting relevant features from raw data before attacking a classifier.

Also, there is a terminology like temporal fusion, spatial fusion or spectral fusion which can be found in the literature. But these are not considered as new fusion types. The last two ones are the examples of low or intermediate level fusions. The temporal fusion characterizes the type of the input used. Moreover, it may occur at any level. In the above part mentioned that, the input may come from the one or several sensors. In the temporal fusion, the inputs, which taken from the one sensor at different moments, are combined. If some kind of noise reduction is introduced, this type of fusion may improve the performance. In motion fusion case, the target may indicate many aspects or occlusion profile [6].

5.2 Fixed Rules

Consider a pattern recognition problem where Z is a pattern and classes are (ω_1,..., ω_m). The goal is to assign Z to the one of the m possible classes. Assume that there are R classifiers each of which represents the given pattern by a distinct measurement vector, where x_i is the measurement vector that is used by the *i*th classifier. Each class ω_k is modeled by the probability density function $p(x_i \mid \omega_k)$ and the priori probability of occurrence is determined by $P(\omega_k)$. Consider that the models are mutually exclusive. It means that only one model is associated with each pattern.

The given measurements are as follows:

- x_i, is the *i*th classifier, where i = 1,..., R
- Z, is the pattern
- ω_j, is the class

the pattern Z should be assigned to class ω_j obtain the posteriori probability of that interpretation is maximum, i.e.

$$\begin{aligned} &assign \quad Z \to \omega_j \quad if \\ &P(\omega_j \mid x_1,..., x_R) = \max_k P(\omega_k \mid x_1,..., x_R) \end{aligned} \tag{5.1}$$

In the above equation the Bayesian decision rule shows that in order to use all available information correctly to obtain a decision, it is necessary to compute the probabilities of the different hypotheses by considering all measurements simultaneously. Posteriori probability functions computation depend on the knowledge of high-order measurement statistics which is described in terms of joint probability density functions $p(x_1,..., x_R \mid \omega_k)$. In this case it is difficult to draw a conclusion. So, the above rule must be simplified. This rule will be determined in terms of decision support computations which are done by the individual classifiers and the information is expressed by the vector x_i. This procedure will make the above rule computationally useful and it will manage the combination rules. The computation rules are generally used in practice. Furthermore, this procedure will obtain an opportunity for the development of a range of efficient classifier combination strategies.

According to the above rule, by using Bayes theorem, the posteriori probability $P(\omega_k \mid x_1,..., x_R)$ can be rewritten in the following form

$$P(\omega_k \mid x_1,...,x_R) = \frac{p(x_1,...,x_R \mid \omega_k)P(\omega_k)}{p(x_1,...,x_R)} \tag{5.2}$$

In the above equation $p(\mathbf{x}_1, \ldots, \mathbf{x}_R)$ is the unconditional measurement joint probability density. Also this expression can be written in terms of the conditional measurement distributions like

$$p(x_1,...,x_R) = \sum_{j=1}^{m} p(x_1,...,x_R \mid \omega_j)P(\omega_j) \tag{5.3}$$

Because of this, in the next parts, we will deal with the numerator terms of the equation (5.2) [7].

5.2.1 Product Rule

As already described in the above parts, $p(x_{1, \ldots}\ x_R \mid \omega_{k,})$ shows the joint probability distribution of the measurements obtained by the classifiers. Assume that the representations used are conditionally statistically independent. As a result of this assumption the following expression can be written

$$p(x_1,...,x_R \mid \omega_k) = \prod_{i=1}^{R} p(x_i \mid \omega_k) \tag{5.4}$$

where $p(\mathbf{x}_i|\omega_k)$ is the measurement process model of the ith representation. By substituting equations (5.3) and (5.4) into the equation (5.2) the following equation is obtained

$$P(\omega_k \mid x_1,...,x_R) = \frac{P(\omega_k)\prod_{i=1}^{R} p(x_i \mid \omega_k)}{\sum_j^m P(\omega_j)\prod_{i=1}^{R} p(x_i \mid \omega_j)} \tag{5.5}$$

Also, by using equation (5.5) in equation (5.1), the following decision rule is obtained

$$assign \qquad Z \to \omega_j \qquad if$$
$$P(\omega_j)\prod_{i=1}^{R} p(x_i \mid \omega_j) = \max_{k=1}^{m} P(\omega_k)\prod_{i=1}^{R} p(x_i \mid \omega_k) \tag{5.6}$$

or in terms of the a posteriori probabilities brought by the respective classifiers can written in the following form

$$assign \qquad Z \to \omega_j \qquad if$$
$$P^{-(R-1)}(\omega_j)\prod_{i=1}^{R} P(\omega_j \mid x_i) = \max_{k=1}^{m} P^{-(R-1)}(\omega_k)\prod_{i=1}^{R} P(\omega_k \mid x_i) \tag{5.7}$$

The decision rule, which is shown in equation (5.7), considers the likelihood of a hypothesis by combining the a posteriori probabilities. The a posteriori probabilities are formed by the individual classifiers. So, it is called as product rule [7].

5.2.2 Sum Rule

In some cases the decision rule, which is shown in equation (5.7), may be appropriate under the assumption that the a posteriori probabilities computed by the respective classifiers will not diverge dramatically from the prior probabilities. It is a strong assumption. When the available observational discriminatory information is undetermined due to high levels of noise, it can be satisfied.

In this situation, assume that the a posteriori probabilities can be written as the following form.

$$P(\omega_k \mid x_i) = P(\omega_k)(1 + \delta_{ki}) \tag{5.8}$$

where δ_{ki} satisfies $\delta_{ki} << 1$. If the equation (5.8) for the a posteriori probabilities is substituted into the equation (5.7), the following result is obtained.

$$P^{-(R-1)}(\omega_k)\prod_{i=1}^{R} P(\omega_k \mid x_i) = P(\omega_k)\prod_{i=1}^{R}(1 + \delta_{ki}) \tag{5.9}$$

For approximating the equation (5.9) in the right-hand side, expand the product and neglect the second and higher order terms.

$$P(\omega_k)\prod_{i=1}^{R}(1+\delta_{ki}) = P(\omega_k) + P(\omega_k)\sum_{i=1}^{R}\delta_{ki} \tag{5.10}$$

By substituting equations (5.10) and (5.8) into the equation (5.7), sum decision rule is obtained [7].

$$\begin{aligned}&assign \qquad Z \to \omega_j \qquad\qquad if \\ &(1-R)P(\omega_j) + \sum_{i=1}^{R} P(\omega_j \mid x_i) = \max_{k=1}^{m}\left[(1-R)P(\omega_k) + \sum_{i=1}^{R} P(\omega_k \mid x_i)\right]\end{aligned} \tag{5.11}$$

The basic schemes for classifier combination are formed by the decision rules equations (5.7) and (5.11). Also, these rules are used for developing many commonly classifier combination strategies. Note that

$$\prod_{i=1}^{R} P(\omega_k \mid x_i) \le \min_{i=1}^{R} P(\omega_k \mid x_i) \le \frac{1}{R}\sum_{i=1}^{R} P(\omega_k \mid x_i) \le \max_{i=1}^{R} P(\omega_k \mid x_i) \tag{5.12}$$

The above expression determines that the product and sum combination rules can be approximated by the above upper or lower bounds. Moreover, producing binary valued functions Δ_{ki} from the a posteriori probability $P(\omega_k \mid x_i)$ is the following

$$\Delta_{ki} = \begin{cases} 1 & if \ P(\omega_k \mid x_i) = \max_{j=1}^{m} P(\omega_j \mid x_i) \\ 0 & otherwise \end{cases} \tag{5.13}$$

Binary valued functions are resulted in combining decision outcomes rather than combining a posteriori probabilities. Using these approximations we obtain the max, min and median rules [7].

5.2.3 Max Rule

Equation (5.14) is obtained from the equation (5.11). Equation (5.14) is approximating the sum by the maximum of the posterior probabilities.

$$\begin{aligned}&assign \qquad Z \rightarrow \omega_j \qquad\qquad if\\&(1-R)P(\omega_j) + R\max_{i=1}^{R} P(\omega_j \mid x_i) = \max_{k=1}^{m}\left[(1-R)P(\omega_k) + R\max_{i=1}^{R} P(\omega_k \mid x_i)\right]\end{aligned} \tag{5.14}$$

under the assumption of equal priors simplifies to [7]

$$\begin{aligned}&assign \qquad Z \rightarrow \omega_j \qquad\qquad if\\&\max_{i=1}^{R} P(\omega_j \mid x_i) = \max_{k=1}^{m}\max_{i=1}^{R} P(\omega_k \mid x_i)\end{aligned} \tag{5.15}$$

5.2.4 Min Rule

For obtaining the equation (5.16), start from equation (5.7) and bound the product of posterior probabilities from above equation.

$$\begin{aligned}&assign \qquad Z \rightarrow \omega_j \qquad\qquad if\\&P^{-(R-1)}(\omega_j)\min_{i=1}^{R} P(\omega_j \mid x_i) = \max_{k=1}^{m} P^{-(R-1)}(\omega_k)\min_{i=1}^{R} P(\omega_k \mid x_i)\end{aligned} \tag{5.16}$$

It simplifies under the assumption of equal priors into the following form [7]

$$\begin{aligned}&assign \qquad Z \rightarrow \omega_j \qquad\qquad if\\&\min_{i=1}^{R} P(\omega_j \mid x_i) = \max_{k=1}^{m}\min_{i=1}^{R} P(\omega_k \mid x_i)\end{aligned} \tag{5.17}$$

5.2.5 Median Rule

The sum rule in equation (5.11) can be interpreted as computing the average a posteriori probability for each class over all the classifier outputs, under the equal prior assumption. As an example,

$$assign \quad Z \to \omega_j \quad if$$
$$\frac{1}{R}\sum_{i=1}^{R} P(\omega_j \mid x_i) = \max_{k=1}^{m} \frac{1}{R}\sum_{i=1}^{R} P(\omega_k \mid x_i) \tag{5.18}$$

So, the rule assigns a pattern to that class and the a posteriori probability of the class is maximum. If some of the classifiers output an a posteriori probability for some outlier class, it will influence the average. This results in an incorrect decision. It is known that a reliable estimate of the mean is the median. As a result of this, the combined decision is based on the median of the posteriori probabilities. Then it indicates the following equation [7]

$$assign \quad Z \to \omega_j \quad if$$
$$\operatorname*{med}_{i=1}^{R} P(\omega_j \mid x_i) = \max_{k=1}^{m} \operatorname*{med}_{i=1}^{R} P(\omega_k \mid x_i) \tag{5.19}$$

5.3 Trainable Combiners

5.3.1 Linear Discriminant Classifier

A discriminant is a function which takes the vector x as an input and assigns it to the one of the K classes which is shown by C_k. The basic representation of the linear discriminant function is shown as

$$y(x) = w^T x + w_0 \tag{5.20}$$

where w is represent weight vector and w_0 is represent bias. Sometimes the negative of bias is called threshold.

If $y(x) \geq 0$ then the input vector is assigned to class C_1 otherwise it is assigned to class C_2.

Therefore the decision boundary is defined by the relation $y(x) = 0$, which corresponds to a (D-1) - dimensional hyper plane within D-dimensional input space.

Take two points x_A and x_B which lie on the decision surface. Because $y(x_A) = y(x_B) = 0$, the equation becomes $w^T (x_A - x_B) = 0$. So, vector w is orthogonal to every vector which lies within the decision surface. Also, if point x is on the decision surface, then the equation becomes $y(x) = 0$. So, the normal distance from the origin to the decision surface is computed as

$$\frac{w^T x}{\| w \|} = -\frac{w_0}{\| w \|} \tag{5.21}$$

where w_0, bias parameter, determines the location of the decision surface.

Note that $y(x)$ gives a signed measure of the perpendicular distance r of the point x from the decision surface. As an example, choose an arbitrary point x and $x_\perp$ be its orthogonal projection onto the decision surface,

$$x = x_\perp + r\frac{w}{\| w \|} \tag{5.22}$$

multiply both sides of the above equation with w^T and add w_0, also use the following equations $y(x) = w^T x + w_0$ and $y(x_\perp) = w^T x_\perp + w_0 = 0$, so equation is considered [8]

$$r = \frac{y(x)}{\| w \|} \tag{5.23}$$

5.3.2 Fisher's Discriminant (Minimum Least Square Linear Classifier)

Fisher criterion can be obtained as a special case of Least Squares. Particularly, the targets for class C_1 must be taken as N / N_1, where N_1 represents the number of patterns in class C_1, and N represents the total number of patterns. The target value approximates to the reciprocal of the prior probability for class C_1. In the case of class C_2, the targets must be taken as $- N / N_2$, where N_2 represents the number of patterns in class C_2. So, the sum-of-squares error function can be defined as

$$E = \frac{1}{2}\sum_{n=1}^{N}(w^T x_n + w_0 - t_n)^2 \tag{5.24}$$

The following equations can be obtained by setting the derivatives of E with respect to w_0 and w to zero.

$$\sum_{n=1}^{N}(w^T x_n + w_0 - t_n) = 0 \tag{5.25}$$

$$\sum_{n=1}^{N}(w^T x_n + w_0 - t_n)x_n = 0 \tag{5.26}$$

go back to the equation (5.25) and make use of the choice of target coding scheme for the t_n, the following equation can be obtained

$$w_0 = -w^T m \tag{5.27}$$

where the below equation is used

$$\sum_{n=1}^{N} t_n = N_1 \frac{N}{N_1} - N_2 \frac{N}{N_2} \tag{5.28}$$

where m represent the mean of the total data set and it is shown as

$$m = \frac{1}{N}\sum_{n=1}^{N} x_n = \frac{1}{N}(N_1 m_1 + N_2 m_2) \tag{5.29}$$

After doing some algebraic operations and using the choice of t_n, and the equation three the following equation is obtained,

$$(S_w + \frac{N_1 N_2}{N} S_B)w = N(m_1 - m_2) \tag{5.30}$$

where S_w represents total within- class covariance matrix

$$S_w = \sum_{n \in C_1} (x_n - m_1)(x_n - m_1)^T + \sum_{n \in C_2} (x_n - m_2)(x_n - m_2)^T \qquad (5.31)$$

and S_B represents between-class covariance matrix

$$S_B = (m_2 - m_1)(m_2 - m_1)^T \qquad (5.32)$$

by substituting the equations (5.27),(5.31) and (5.32) and ignoring irrelevant scale factors the following equation is obtained,

$$w \alpha \quad S_w^{-1}(m_2 - m_1) \qquad (5.33)$$

Note that $S_B w$ is always in the direction of (m_2-m_1). The weight vector matches with that found from Fisher Criterion.

Also an expression found for bias value w_0

$$w_0 = -w^T m \qquad (5.34)$$

which means that a new vector x should be classified as belonging to class C_1 if

$$y(x) = w^T (x - m) > 0 \qquad (5.35)$$

otherwise it belongs to class C_2 [8].

5.3.3 Naive Bayes Classifier

Naïve Bayse Classifier is a probabilistic classifier and it is based on Bayes' Theorem with independence assumptions. Bayes' Theorem is related to the Conditional and Marginal Probability distributions of random variables. The derivation of Bayes Theorem is the following:

The Conditional Probability of event A given event B is

$$\Pr(A \mid B) = \frac{\Pr(A \cap B)}{\Pr(B)} \tag{5.36}$$

Also, the probability of event B given event A is

$$\Pr(B \mid A) = \frac{\Pr(A \cap B)}{\Pr(A)} \tag{5.37}$$

By combining these two equations we can get the following rule

$$\Pr(A \mid B)\Pr(B) = \Pr(A \cap B) = \Pr(B \mid A)\Pr(A) \tag{5.38}$$

This lemma is called the product rule for probabilities. By dividing both sides with Pr(B), we can get Bayes' Theorem

$$\Pr(A \mid B) = \frac{\Pr(B \mid A)\Pr(A)}{\Pr(B)} \tag{5.39}$$

5.3.3.1 Naive Bayes Probabilistic Model

The probability model for a classifier has a conditional model of the form

$$p(C \mid F_1, \ldots, F_n) \tag{5.40}$$

where C represents a dependent class variable and F_1 through F_n represent feature variables.

If the feature take large number of values or the number of features n is large we can have problem. Then by using Bayes' theorem we reformulate the model

$$p(C \mid F_1, \ldots, F_n) = \frac{p(C)\, p(F_1, \ldots, F_n \mid C)}{p(F_1, \ldots, F_n)} \tag{5.41}$$

The important part of the formula is the numerator part of the fraction, because the denominator part of the fraction is independent from C. Also, the values of features F_i are given, so the denominator part becomes constant. And the numerator part becomes Joint Probability model

$$p(C, F_1, ..., F_n) \tag{5.42}$$

Also, formula can be rewritten by using the definition of Conditional Probability

$$\begin{aligned} p(C,F_1,...,F_n) &= p(C)p(F_1,...,F_n \mid C) \\ &= p(C)p(F_1 \mid C)p(F_2,...,F_n \mid C,F_1) \\ &= p(C)p(F_1 \mid C)p(F_2 \mid C,F_1)p(F_3,...,F_n \mid C,F_1,F_2) \\ &= p(C)p(F_1 \mid C)p(F_2 \mid C,F_1)p(F_3 \mid C,F_1,F_2)p(F_4,...,F_n \mid C,F_1,F_2,F_3) \end{aligned} \tag{5.43}$$

Assume that every feature F_i is conditionally independent from other feature F_j for $j \neq i$ so,

$$p(F_i \mid C, F_j) = p(F_i \mid C) \tag{5.44}$$

and joint model can be shown as

$$\begin{aligned} p(C,F_1,...,F_n) &= p(C)p(F_1 \mid C)p(F_2 \mid C)p(F_3 \mid C)... \\ &= p(C)\prod_{i=1}^{n} p(F_i \mid C) \end{aligned} \tag{5.45}$$

The conditional distribution over the class C can be shown as the following form:

$$p(C \mid F_1,...,F_n) = \frac{1}{Z} p(C)\prod_{i=1}^{n} p(F_i \mid C)$$

Here the Z value is a scaling factor that depends on feature variables $F_1,...,F_n$. If the feature variables values are known, Z becomes constant [9].

5.3.4 Kernel Density Estimator

Let us assume that observations are done from p(x) which is unknown probability density in some D-dimensional Euclidean space and the aim is to estimate the value of p(x). Let us consider a small region R which contains x. So, the probability associated with this region becomes

$$P = \int_R p(x)dx \tag{5.46}$$

where P represents probability, R represents region and p(x) represents probability density.

Suppose that there is a data set which includes N observations from p(x). Every data point has a probability P of falling within the region R. According to the binomial distribution, total numbers of K points, which lie inside the R, are distributed.

$$Bin(K \mid N, P) = \frac{N!}{K!(N-K)!} P^K (1-P)^{1-K} \tag{5.47}$$

where K represents the total number of points, N represents the number of observations and P represents the probability.

According to the below equation,

$$E[m] = \sum_{m=0}^{N} mBin(m \mid N, \mu) = N\mu \tag{5.48}$$

the mean fraction of points falling inside the region is shown by

$$E[K / N] = P \tag{5.49}$$

Also by using the below equation,

$$\operatorname{var}[m] = \sum_{m=0}^{N} (m - E[m])^2 \, Bin(m \mid N, \mu) = N\mu(1-\mu) \tag{5.50}$$

the variance around the mean becomes,

$$\operatorname{var}[K / N] = P(1-P) / N \tag{5.51}$$

For large values of N the distribution has the form

$$K \cong NP \tag{5.52}$$

Assume that the region R is sufficiently small and the probability density is approximately constant over the region, then the equation becomes

$$P \cong p(x)V \tag{5.53}$$

where p(x) represents probability density and V represents volume over R.

By combining the equations (5.52) and (5.53), the density estimation is obtained in the form of

$$p(x) = \frac{K}{NV} \tag{5.54}$$

For the validity of the above equation, assume that R region is sufficiently small and because of this the density is approximately constant over the region. Also, note that R region is sufficiently large that K points are falling inside the region is sufficient for the binomial distribution [8].

5.3.4.1 Parzen Classifier

The region R is taken as a small hypercube which is centered on the point x. And the aim is to determine the probability density function. Instead of counting the K number of points that are falling within the region, the following function is defined as

$$k(u) = \begin{cases} 1, & |u_i| \leq 1/2, \quad i = 1, \ldots, D, \\ 0, & otherwise \end{cases} \tag{5.55}$$

where the function k(u) represents a unit cube centered on the origin. And, the function k(u) is an example of a kernel function, also it is called a *Parzen Window*.

According to the equation (5.55), if the data point x_n lies inside a cube of side h centered on x, the quantity k $((x-x_n)/h)$ will be 1 and 0 otherwise.

For this reason the total number of data points lying inside the cube will be shown as

$$K = \sum_{n=1}^{N} k(\frac{x - x_n}{h}) \tag{5.56}$$

Also, substituting the equation (5.56) into the equation (5.54), the following result is obtained for the estimated density at x,

$$p(x) = \frac{1}{N} \sum_{n=1}^{N} \frac{1}{h^D} k(\frac{x - x_n}{h}) \tag{5.57}$$

where h^D represents the volume of a hypercube of side h in D dimensions. Also, instead of computing the probability density of a single cube centered on x, compute the sum over N cubes centered on N data points x_n. So, by using the symmetry of the function k(u), the equation (5.57) is re-interpreted.

The kernel density estimator in equation 12 has a problem, namely the presence of artificial discontinuities in this case at the boundaries of the cube. A common choice is Gaussian, which gives us the following kernel density model,

$$p(x) = \frac{1}{N} \sum_{n=1}^{N} \frac{1}{(2\pi h^2)^{1/2}} \exp\left\{ -\frac{\| x - x_n \|^2}{2h^2} \right\} \tag{5.58}$$

where h is the standard deviation of the Gaussian components. In this model, a Gaussian over each data point is placed, then the contributions over the whole data set is adding up and finally the density is normalized by dividing N.

Choose another kernel function k(u) in equation (5.57) with the following conditions,

$$k(u) \geq 0, \tag{5.59}$$

$$\int k(u)du = 1 \tag{5.60}$$

which make definite that the resulting probability distribution is non negative everywhere and integrates to 1.

The class of density model which is given in equation (5.57) is called a kernel density estimator or *Parzen* estimator. It has an advantage that no computation is involved in 'training' phase because this requires storage of the training set. But at the same time this is a disadvantage because the computational cost of evaluating density is linearly growing according to the size of the data set [8].

5.3.5 K-Nearest Neighbor Method

In previous sections, the general result for local density estimation is found as

$$p(x) = \frac{K}{NV} \tag{5.54}$$

Instead of fixing V and determining the value of K from the data, we do something different and this time fix the value of K and determine an appropriate value for V by using the data. For doing this, consider a small sphere centered on the point x at which the density *p(x)* will be estimate. Also, the radius of the sphere will grow until it contains exactly K data points. The estimate of the density *p(x)*, which is given by in equation (5.54), with V is set to the volume of the resulting sphere. This technique is known as *K nearest neighbors.*

For extending K-nearest-neighbor technique to the problem of classification, first apply K-nearest-neighbor density estimation technique to each class separately and then use Bayes' theorem. Assume that there is a data set contains N_k points in class C_k with N points in total, so $\sum_k N_k = N$. If a new point x is to be classified, then draw a sphere centered on x containing exactly K points irrespective of their class. Assume that the sphere has volume V and contains K_k points from class C_k. Then the equation (5.54) obtains an estimate of the density of each class

$$p(x \mid C_k) = \frac{K_k}{N_k V} \tag{5.61}$$

Also, the unconditional density is the following

$$p(x) = \frac{K}{NV} \tag{5.62}$$

and the class priors are the following

$$p(C_k) = \frac{N_k}{N} \tag{5.63}$$

So, for obtaining the posterior probability of class membership, combine the equations (5.61), (5.62) and (5.63) and use Bayes' theorem [8].

$$p(C_k \mid x) = \frac{p(x \mid C_k)\, p(C_k)}{p(x)} = \frac{K_k}{K} \tag{5.64}$$

5.3.6 Nearest Mean Linear Classifier

In a two-class problem, the normal density based linear classifier (NLC) built on the set R is shown as

$$f(D(x,R)) = \left[D(x,R) - \frac{1}{2}(m_1 + m_2) \right]^T C^{-1}(m_1 - m_2) + \log \frac{p(\omega_1)}{p(\omega_2)} \tag{5.65}$$

and the normal density based quadratic classifier (NQC) is shown as

$$f(D(x,R)) = \sum_{i=1}^{2} (-1)^i (D(x,R) - m_i)^T C_i^{-1} (D(x,R) - m_i) + \log \frac{p(\omega_1)}{p(\omega_2)} + \frac{1}{2} \log \frac{\det(C_1)}{\det(C_2)} \tag{5.66}$$

where C_1 and C_2 represent the estimated class covariance matrices and $C = \frac{1}{2}(C_1 + C_2)$ represents the sample covariance matrix which determined in a dissimilarity space. The square Mahalanobis distance between $D(x,R)$ and the class mean m_i represented as

$$(D(x,R) - m_i)^T C_i^{-1} (D(x,R) - m_i)$$

If the covariance matrix C or C_1 or C_2 is singular, then its inverse will not be computed. A solution is using the regularized version which is defined as $C_{reg} = (1-\lambda)C + \lambda I$ where I represents the identity matrix. The following regularization is used for choosing a proper λ.

$C_{reg}^{\lambda} = (1-2\lambda)C + \lambda diag(C) + \frac{\lambda}{n} tr(C) I$, $n = ||R||$. So, the regulation term is expressed relatively to the variance.

If the covariance matrix C is the identity matrix, NLC reduces to nearest mean classifier (NMC), assigning an object to the class of its nearest mean vector in the Euclidean sense. If C is a diagonal matrix, then the resulting decision rule is the weighted nearest mean classifier (WNMC). So, these are multi-class classifiers [10].

The nearest mean classifier stores only the mean of each class such as one prototype per class. It classifies the objects with the label of the nearest class prototype. The nearest mean classifier is very robust. It generally has a high error on the training data and on the test data, but the error on the training data is a good prediction of the error on the test data [11].

5.3.7 Perl Classifier - Linear Classifier by Linear Perceptron

The perceptron is a kind of binary classifier. In binary classification, the set of objects are classified into the two groups like the first group has some property and the second group does not have that property. So, the perceptron maps its input binary vector *x* to an output value *f(x)* according to the following expression

$$f(x) = \begin{cases} 1 & if \;\; w \cdot x + b > 0 \\ 0 & else \end{cases} \tag{5.67}$$

Where w represents a vector of real-valued weights and $w \cdot x$ represents the dot product. Here the dot product computes the weighted sum. Also, the term b represents the bias and it is constant. So, it does not depend on any input value.

The output value of *f(x)* is either 0 or 1. Because of binary classification, these values are used for classifying the input x as either positive or negative instance. Here, the bias b can be used for offsetting the activation function or giving a base level activity to the output. So, if the value of b is negative, then the value of the dot product, which is the combination of weighted inputs, must be greater than the value of $-b$ for producing a positive value. Also, the bias changes the position of the decision boundary.

Since, the output is directly related with the inputs, the perceptron can be assumed as a simple form of feed-forward neural network [12].

Chapter 6
Driver Recognition and Verification/Approval

This thesis aims to recognize/approve the driver by using driving signals. The results of driver recognition and approval can be used for the security purposes and also, vehicle's some settings can be personalized automatically. In this study, both Drive-Safe and Nagoya University's driver data are used. The Center for Integrated Acoustic Information Research (CIAIR) at Nagoya University is recorded a multi-model corpus inside a vehicle. In these experiments different driving behavior signals were collected from 5 analog channels, each sample except steering wheel angle was converted into 1 kHz with an unsigned 16-bit format.

1- Break pedal pressure (kgf/cm^2): 0-50 kgf/cm^2 was mapped to 0-5.0 V and linearly digitized in the range 0 to 32767.
2- Accelerator pedal pressure (kgf/cm^2): 0-50 kgf/cm^2 was mapped to 0-5.0 V and linearly digitized in the range 0 to 32767.
3- Engine speed (period/minute): 0-8000 period/minute was mapped to 0-5.0 V and linearly digitized in the range 0 to 32767.
4- Vehicle speed (km/h): 0-120 km/h was mapped to 0-5.0 V and linearly digitized in the range 0 to 32767.
5- Steering wheel angle (degree): -1800 degrees to +1800 degrees and linearly digitized in the range -32769 to 32767.

We will be using these five channels of driving data in our study.

Driver recognition is the process of determining whether the person in question belongs to the driver subjects' database. In the driver verification/approval process, the goal is to determine whether the identity of the driver is the same which he/she claims. For example, if the driver sits in the driver seat and the vehicle determines the driver directly, this process is not a driver verification process. If the driver claims that he/she is a certain driver and the vehicle determines whether the driver is certain driver or not, then this process is a driver verification process. In this project, according to the scenario driver recognition or a driver verification/approval will be done.

Other biometric features, which are generally used for recognition and verification tasks are the face, sound and driving behavior. In daily life, people use face and sound methods for recognize each other. So, these techniques are very natural and do not disturb the driver. Driving behavior is an important feature that the vehicle can recognize the driver easily during the driving. Driver recognition by using driving behavior is a very original research area. We will focus on driving behavior and not face or sound recognition in this thesis.

6.1 Driver Recognition Using Driving Signals

At the preprocessing stage, time domain signals are smoothed to remove noise and resample. First order differences are determined in the same time domain to extract new features from the data. No evidence is reported which show the existence of periodicity in driving signals. Consequently, in this study, the signals are used in the time domain directly.

The signals and first order difference vectors are modeled using statistical methods. Most time series signals change slowly, so in the modeling stage quasi-stationary assumption can be done. As a result, time series data can be modeled using dynamic models like Hidden Markov Model. By using time series data in biometric identification studies, signal's underlying state topology is usually unclear (except text dependent speaker recognition studies) and single-state probabilistic models have good performance. This is valid for using functions which have more than one distribution peak and parametric continuous distribution functions which comprise the changes of the time area. Gaussian Mixture Models can approximate any continuous differentiable curves even if they have multiple distribution peaks. Good statistical models can be obtained for time series data if GMM has enough number of distributions. GMM is used for the first time for driving signals by Igarashi [3] and in this thesis also same methods are used for modeling the driver behavior.

6.2 Classifier Fusion for Driver Recognition and Verification

In recent years, studies in multi-biometric systems that depend on more than one biometric property have become more popular. The reason for this motivation to the multi-biometric system is, some restrictions occur in systems which use single-biometric quality. By using multi-biometric system, these restrictions can be removed. At the end of this study, using driving behavior signals, a multi-biometric person recognition/approval system will be done.

Combination methods can be divided into the two main categories as Fixed Rules and Trainable Combiners.

Fixed Rules use simple and unchangeable rules for combining different classifiers data. Trainable combiner rules have free parameters which can be trained on a separate part of the study data. At the same time trainable combiners are typically classifiers. These combiners' classification is done in the score space instead of feature space.

A test data x, let us assume that $S(i,j)$ be the score of person i in the modality j. We drop x from the notation for a basic representation. The goal is to get a score value $S(i)$ for person i by using combination methods. As a summary of a previous chapter, the different classifier combination methods are defined in the below part.

Fixed Rules:

1. Maximum Rule : $S(i) = \max_j S(i,j)$
2. Minimum Rule : $S(i) = \min_j S(i,j)$
3. Mean (Sum) Rule : $S(i) = \text{sum}_j S(i,j)$
4. Product Rule : $S(i) = \text{prod}_j S(i,j)$
5. Median Rule : $S(i) = \text{median}_j S(i,j)$

Trainable Combiners:

1. Nearest Mean Combiner (NMC): Simple linear combiner that accepts nearest class mean as a classifier output.
2. Fisher Classifier (Fisher): Linear classifier that uses the least squares method for matching features and class labels.
3. Linear Discriminant Combiner (LDC): Linear classifier that is modeled every class as a Gauss distribution which owner of the same covariance matrix.
4. Naïve Bayes Combiner (NB): A combiner which assumes that the feature vector's conditional class probabilities, is statistically independent. Every vector is modeled as a distribution model which occurs from 10 non-parametric bins.
5. Parzen Combiner (Parzen): It uses Parzen density distribution function.
6. K-Nearest Neighbour Classifier (KNN): A method for classifying objects based on closest training examples in the feature space.
7. Perl Classifier (Perl): Linear classifier by linear perceptron. The perceptron is a kind of binary classifier.

Chapter 7
Experiments and Results

7.1 Experiments

In this study we carried out three experiments; driver recognition, driver verification and driver fatigue detection. Driver recognition and verification experiments were done with a 100 person subset of the Nagoya University CIAIR database whereas driver fatigue experiment was done with data Drive-Safe. This database consists of 50 female drivers and 50 male drivers. During the experiments five different “driving behavior signal” were used. These signals were:

1. Break pedal pressure
2. Accelerator pedal pressure
3. Engine speed
4. Vehicle speed
5. Steering wheel angle

The driving behavior data was collected by 5 analog channels, each sample was converted into 1 kHz with an unsigned 16-bit format. These signals were used for identifying and verifying driver identities.

In the first part of the experiment a noise removal operation was applied to the signals by using the filter function of the Matlab. The filter function filters a data sequence using a digital filter and it works for both real and complex inputs [13]. Figure 7.1 shows the implementation of filter function.

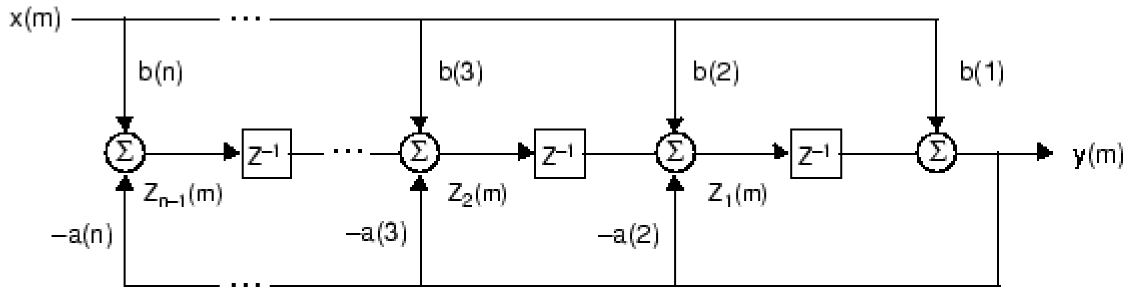

Figure 7.1 – Implementation of filter function [13]

This step was followed by decimation procedure which was done with the decimate function of the Matlab. Decimation reduces the original sampling rate for a sequence to a lower rate, the opposite of interpolation. The decimation process filters the input data with a lowpass filter and then resamples the resulting smoothed signal at a lower rate [14]. Figure 7.2 shows graphical representation of decimation process.

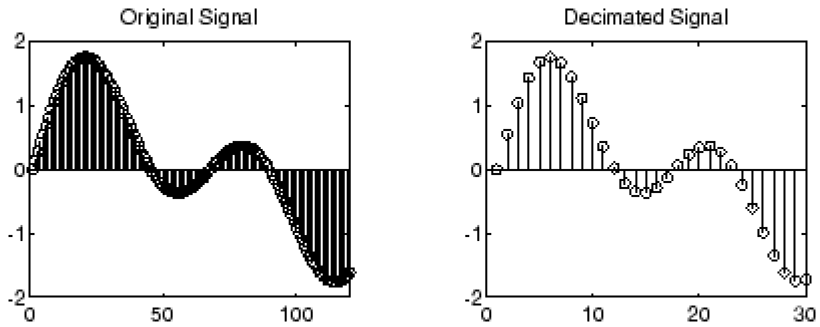

Figure 7.2 – Graphical representation of decimation process [14]

After these operations the driving features were obtained. In the second part driving features of each driver were divided into the 20 equal length parts. First 17 parts were used for training, following 2 parts were used for held-out and the final part was used for testing. So, 20 sets of observations were available for each person. The held-out data is a part of available training data that is not used during training or testing, but it is used to regulate certain parameters of the recognition system. Held-out data is also called *validation data* [1].

In the literature Gaussian Mixture Model is frequently used in text-independent speaker recognition. GMMs were first used by Igarashi for modeling the driving signals [15]. In his study, it was used for modeling the driver behavior. During the experiments, eight mixture components of GMM were used for modeling the driving signals of every people. Also, background GMM models were trained for each modality. In background model, sixteen mixture (twice the number of mixtures) components of GMM were used. Background GMM was used for normalization in likelihood ratio testing for biometric recognition. Igarashi preferred to use well-known Expectation-Maximization algorithm for training GMMs. Block diagram of the training procedure is shown in the below figure.

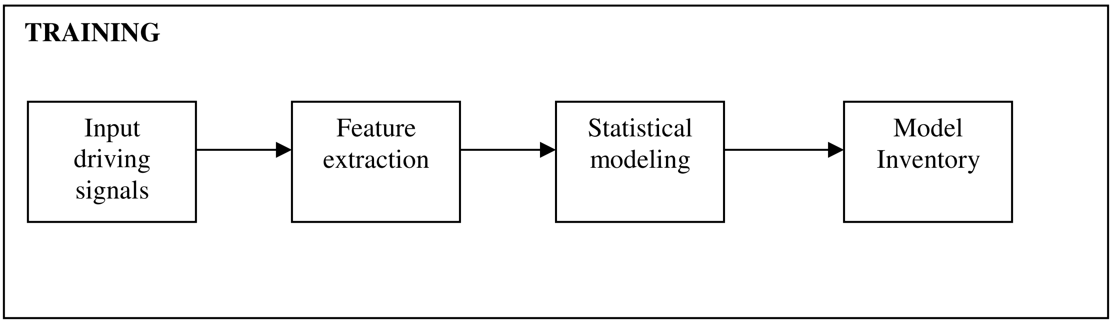

Figure 7.3 – System block diagram for training the multimodal driver recognition system [1]

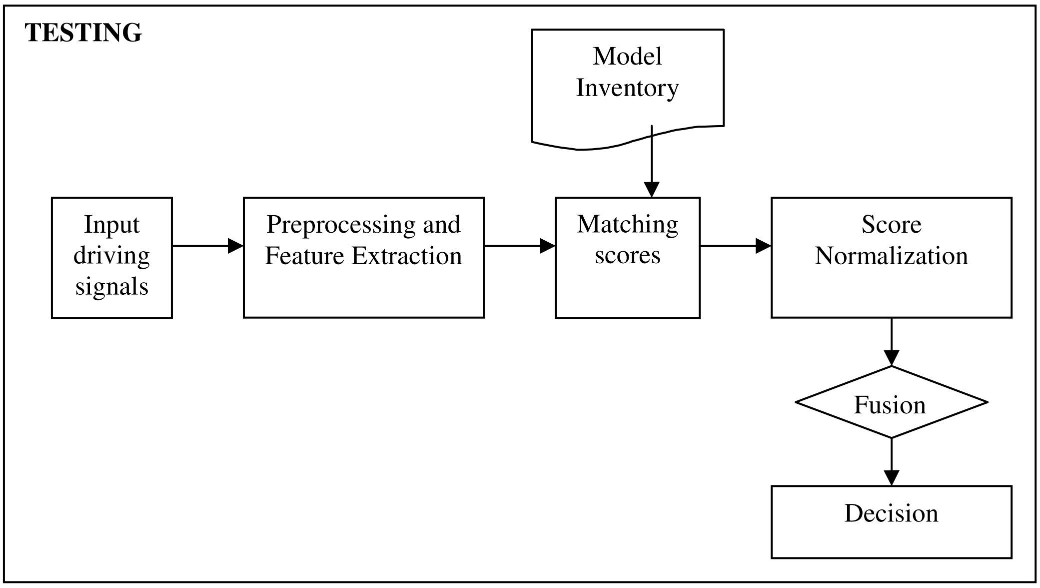

Figure 7.4 – System block diagram for testing the multimodal driver recognition system [1]

Multi-modal driver recognition system is illustrated in the above figure. In the recognition study, the posterior probabilities of the identities were obtained by the given test data. The largest one was chosen as identity of the test segment. These probabilities are called scores. The similarity between biometrics data is shown with these scores [15]. The important part of the classifier combination at the score level is to normalize the scores from each modality before the combinations. Typical likelihood ranges for genuine and impostors can be differing among modalities. So, loglikelihood-ratio scores from different modalities cannot be directly added. Therefore, for making the scores compatible, it is needed to normalize the scores. The way that was used to normalize the scores is, to use the mean and standard deviation of likelihood scores which were obtained from held-out validation data. Normalization can be done using a sigmoid function which can map the scores to the (0,1) range [1].

$$S_k^{'} = \frac{1}{1 + \exp(-(S_k - \mu)/\sigma)} \tag{7.1}$$

In the equation S_k represents the old log-likelihood-ratio score for the k^{th} modality and S_k' represents the new score. Also, μ and σ represent mean and standard deviation of old scores obtained on the validation set. In this study, we used top $3N_t$ scores for N_t validation instances for computing the mean and standard deviation of scores, otherwise the mismatch scores may have effected the statistics. After normalization these scores were combined by using fusion methods [1].

During the combination process two main types of strategies were used such as fixed and trainable. Fixed rules are simple and fixed. The rules that were used in the experiments are Max Rule, Min Rule, Mean Rule, Product Rule and Median Rule. Trainable combiners are also classifiers. Both fixed and trainable rules classify in the score space instead of the original feature space. The trainable combiners that were used in the experiments are Linear Discriminant Classifier, Fisher's Least Square Linear Classifier, Naive Bayes Classifier, Parzen Classifier, K-Nearest Neighbour Classifier, Nearest Mean Classifier and Perl Classifier - Linear classifier by linear perceptron. During these experiments PRTools software library was used for evaluating the results and combining classifiers.

The verification study is formed as follows:

Given an input feature vector X_Q (extracted from the biometric data) and a claimed identity I, determine if (I,X_Q) belongs to class w_1 or w_2, where w_1 represents that the claim is true (a genuine user) and w_2 represents that the claim is false (an impostor). Typically, X_Q is matched against X_I, the biometric template corresponding to user I, to determine its category [2]. So,

$$(I, X_Q) \in \begin{cases} w_1 & \text{if } S(X_Q, X_I) \geq t, \\ w_2 & \text{otherwise}, \end{cases} \tag{7.2}$$

where S represents the function that measures the similarity between feature vectors X_Q and X_I, and t represents a predefined *threshold*. The value $S(X_Q, X_I)$ is termed as a similarity or matching score between the biometric measurements of the user and the claimed identity. Therefore, every claimed identity can be classified into two classes like w_1 or w_2 based on the variables X_Q, I, X_I and t, and the function S. Also, biometric measurements of the same individual taken at different times are almost never identical. Because of this the threshold t is introduced [2]. The below figure shows the multi-biometric person verification system.

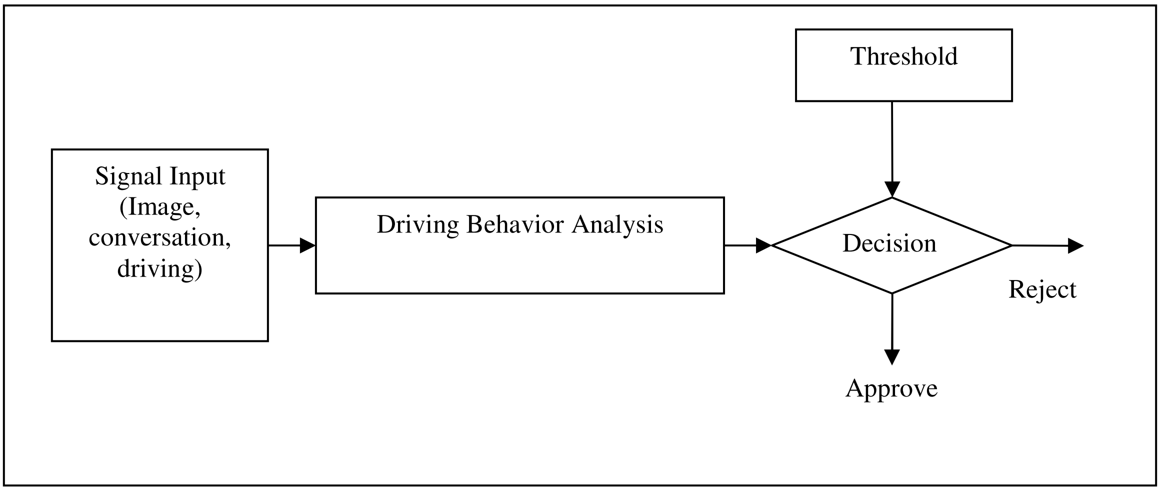

Figure 7.5– Block diagram of multi-biometric person verification system

In the verification experiment, false accept and false reject rates were used as the performance measurement criteria. We defined a threshold value and according to this value and calculated the false accept and false reject rates. False accept rate shows the probability that the system incorrectly tells a successful match between the input pattern and non- matching pattern. This measurement gives the result of invalid matches. False reject rate shows the probability that the system incorrectly tells a failure of match between the input pattern and matching pattern. Therefore, this measurement gives the result of valid users who are rejected [4].

The results presented in the next sections show the error rates obtained in the experiments.

7.2 Results

7.2.1 Driver Recognition Results

The tables in the below part show the driving recognition experiments results using the fixed rules and trainable combiners. Recognition is the process that aims to find the answer to the question "Who is he/she?". In this process biometric information of the subject is taken then this information is compared with the other data which are stored in the database.

Driver recognition experiments were done with 100 subjects that are randomly chosen from the CIAIR database, 50 male and 50 female. In these experiments five different channel were used. These channels are break pedal pressure, accelerator pedal pressure, engine speed, vehicle speed and steering wheel angle. During these experiments two different combination methods were used. The first method is the fixed methods which have simple fixed rules to combine information from a set of classifiers. We used five fixed rules such as maximum rule, minimum rule, median rule, mean rule and product rule. The second method is the trainable combination methods. These methods have some free parameters that can be trained on a separate part of the training data. We used seven trainable combiners such as fisher, linear discriminant, nearest mean, naïve bayes, perl, parzen and k-nearest neighbor. Tables 2 to 12 below present the results of driver recognition experiments.

7.2.1.1 Fixed Rules

The following tables show the decision fusion results using the fixed rules. In the combine list, comma (,) shows the decision fusion where the posterior probabilities of classifier are combined. In the tables below Driving 1 through 5 are used as abbreviations for different channels.

- Driving 1 - Break pedal pressure
- Driving 2 - Accelerator pedal pressure
- Driving 3 - Engine speed
- Driving 4 - Vehicle speed
- Driving 5 - Steering wheel angle

The following table presents the individual performance results of break pedal pressure, accelerator pedal pressure, engine speed, vehicle speed and steering wheel angle.

Table 7.1 – Individual performance results for different modalities

Modality	**Percent Error (%)**
Driving 1	90.35
Driving 2	86.15
Driving 3	98.05
Driving 4	97.35
Driving 5	97.90

The above table shows the individual performance results for each modality. The best experiment result was obtained with the accelerator pedal pressure. The highest error rate was obtained with engine speed. The results showed that individually, the driving signals are not appropriate for biometric identification.

The following tables present the results of the fixed rules.

Table 7.2 – Error rates using fixed combination rules for combining the break pedal pressure and accelerator pedal pressure

	Combine List = Driving1,Driving 2		
	COMBINE	**METHOD**	**ERROR (%)**
1	1	maxc	85.95
2	1	minc	81.40
3	1	medianc	78.95
4	1	meanc	78.95
5	1	prodc	77.95

The above table shows the error rates of the first experiment. In the first experiment, we combined the break pedal pressure and accelerator pedal pressure channels. Then we run five different fixed rules for getting the decision fusion. According to the results, the product rule showed the best performance. Median and Mean rules had the same performance. And the maximum rule had the worst performance.

Table 7.3 – Error rates using fixed combination rules for combining the break pedal pressure, accelerator pedal pressure, engine speed, vehicle speed and steering wheel angle

	Combine List = Driving 1,Driving 2,Driving 3, Driving 4,Driving 5		
	COMBINE	**METHOD**	**ERROR (%)**
1	2	maxc	91.60
2	2	minc	90.55
3	2	medianc	87.70
4	2	meanc	85.95
5	2	prodc	84.85

The above table showed the error rates of the second experiment. In the second experiment we combined five channels such as break pedal pressure, accelerator pedal pressure, engine speed, vehicle speed and steering wheel angle. Then we run five different fixed rules for getting the decision fusion. According to the results, the product rule showed the best performance and the maximum rule had the worst performance. In this experiment we increased the dimension for getting better and more accurate results but we faced with the problems of high dimensional data, which called the *curse of dimensionality*.

Table 7.4 – Error rates using fixed combination rules for combining the break pedal pressure, accelerator pedal pressure and steering wheel angle

	Combine List = Driving 1,Driving 2,Driving 5		
	COMBINE	**METHOD**	**ERROR (%)**
1	3	maxc	89.30
2	3	minc	86.50
3	3	medianc	86.45
4	3	meanc	82.30
5	3	prodc	81.20

The above table shows the error rates of the third experiment. In the third experiment we combined three channels such as break pedal pressure, accelerator pedal pressure and steering wheel angle. Then we run five different fixed rules for getting the decision fusion. According to the results, the product rule showed the best performance. Also, mean rule had good performance and the results are very similar with product rule. Median and mean rules had nearly same performances. Finally, maximum rule had the worst performance. If we make a comparison between the results of Table 7.2 and Table 7.4, the results of Table 7.2 were better than the results of Table 7.4.

Table 7.5 – Error rates using fixed combination rules for combining the break pedal pressure, accelerator pedal pressure, engine speed and vehicle speed

	Combine List=Driving 1,Driving 2,Driving 3, Driving 4		
	COMBINE	**METHOD**	**ERROR (%)**
1	4	maxc	90.75
2	4	minc	90.50
3	4	medianc	86.10
4	4	meanc	85.35
5	4	prodc	85.05

The above table showed the error rates of the fourth experiment. In the fourth experiment we combined four channels such as the break pedal pressure, accelerator pedal pressure, engine speed and vehicle speed. Then we run five different fixed rules for getting the decision fusion. According to the results, product and mean rules had similar results and showed the best performances. Also, median rule had good performance. Finally, maximum and minimum rules had the worst performances.

Table 7.6 – Driver recognition using fixed rules

Methods	Experiment 1	Experiment 2	Experiment 3	Experiment 4
Maxc	**85.95**	**91.60**	**89.30**	**90.75**
Minc	81.40	90.55	86.50	90.50
Medianc	78.95	87.70	86.45	86.10
Meanc	78.95	85.95	82.30	85.35
Prodc	**77.95**	**84.85**	**81.20**	**85.05**

As a summary of the four experiments results, according to the above table, fixed rules do not show good performances. Their error rates are very high. The lowest error rate was taken from the first experiment which was done with the break pedal pressure and accelerator pedal pressure channels. In this experiment product rule showed the best result and the error rate was 77.95 percent. Also, this error rate is the best error rate in all of the experiments results. Moreover, product rule shows the best performance in all experiments and has the lowest error rates. Mean rule follows the product rule with the secondary lowest error rates. We expected that the second experiment will show the best results. Because this experiment was done with the all channels such as break pedal pressure, accelerator pedal pressure, engine speed, vehicle speed and steering wheel angle. So, the combination of these channels can have lower error rates than the others and be the most reliable combination. But the results of this experiment were the worst ones because of the curse of dimensionality. In the third experiment three channels were combined. These channels are break pedal pressure, accelerator pedal pressure and steering wheel angle. The results of the minimum and median rule were approximately the same. In the last experiment we combine the four channels such as break pedal pressure, accelerator pedal pressure, engine speed and vehicle speed. The error rates of this experiment were also high. The highest error rate was taken from the maximum rule. Moreover, maximum rule showed the worst performance in all experiments and had the highest error rates. In the next part we will present the experiment results of the trainable combiners.

7.2.1.2 Trainable Combiners

The following tables show the decision fusion when using the trainable combiners. In the combine list, comma (,) shows the decision fusion where the posterior probabilities of classifier are combined. In the tables below Driving 1 through 5 are used as abbreviations for different channels.

- Driving 1 - Break pedal pressure
- Driving 2 - Accelerator pedal pressure
- Driving 3 - Engine speed
- Driving 4 - Vehicle speed
- Driving 5 - Steering wheel angle

The following tables present the results of the trainable combiners.

Table 7.7 – Error rates using Fisher trainable combination method for different modalities

	COMBINE	COMBINE LIST	ERROR (%)	METHOD
1	1	Driving 1,Driving 2	43.45	Fisherc
2	2	Driving 1, Driving 2, Driving 3, Driving 4, Driving 5	16.10	Fisherc
3	3	Driving 1, Driving 2, Driving 5	29.50	Fisherc
4	4	Driving 1, Driving 2, Driving 3, Driving 4	24.55	Fisherc

The above table shows the error rates of Fisher trainable combination method. We constituted four different combination lists and run Fisher trainable combiner. The results showed that for all combination lists Fisher method reached lower error rates than all fixed rules. The best result of Fisher method was obtained in the second combination list where we combined all channels. Also, for this combination list we expected same good result from fixed rules but we did not reach. Fisher method had the worst result in the first experiment which was the 43.45 percent. This result was still better than the results of fixed rules.

Table 7.8 – Error rates using LDC trainable combination method for different modalities

	COMBINE	COMBINE LIST	ERROR (%)	METHOD
1	1	Driving 1, Driving 2	35.50	LDC
2	2	Driving 1, Driving 2, Driving 3, Driving 4, Driving 5	85.55	LDC
3	3	Driving 1, Driving 2, Driving 5	48.00	LDC
4	4	Driving 1, Driving 2, Driving 3, Driving 4	78.70	LDC

The above table shows the error rates of LDC trainable combination method. In the first combination list, LDC method reached better error rates than all fixed rules and Fisher method. In the second combination list, LDC method obtained same error rate with mean rule and the result of product rule was better than LDC method but the best result was obtained by Fisher method. In the third combination list, LDC method had better error rates than all fixed rules but again the best result was obtained by Fisher method. In the last combination list, LDC method showed better error rates than all fixed rules however Fisher method reached the best result.

As a summary LDC method had low error rates in the first and third combination list. But in the second combination list it showed worst performance than the product rule and Fisher Method. The best result of the LDC method was obtained in the first combination list where we combined two channels. LDC method had the worst result in the second combination list which was the 85.55 percent.

Table 7.9 – Error rates using NMC trainable combination method for different modalities

	COMBINE	COMBINE LIST	ERROR (%)	METHOD
1	1	Driving 1, Driving 2	66.65	NMC
2	2	Driving 1, Driving 2, Driving 3, Driving 4, Driving 5	54.65	NMC
3	3	Driving 1, Driving 2, Driving 5	61.40	NMC
4	4	Driving 1, Driving 2, Driving 3, Driving 4	60.40	NMC

The above table shows the error rates of NMC trainable combination method. In the first combination list, NMC method obtained better error rates than all fixed rules. However if we make a comparison between Fisher, LDC and NMC methods, NMC obtained the worst result. In the second combination list, NMC method showed better error rates than all fixed rules but Fisher method reached the lowest error rate. In the third combination list, NMC method had better error rates than all fixed rules but Fisher and LDC methods obtained lower error rates than NMC. In the last combination list, NMC method got better error rates than all fixed rules and LDC method however Fisher method obtained lower error rate than NMC.

As a summary the best result of the NMC method was obtained in the second combination list where we combined all the channels. NMC method had the worst result in the first combination list which was 66.65 percent. Also this result was better than the results of the fixed rules.

Table 7.10 – Error rates using Naïve Bayes trainable combination method for different modalities

	COMBINE	COMBINE LIST	ERROR (%)	METHOD
1	1	Driving 1, Driving 2	43.70	Naivebc
2	2	Driving 1, Driving 2, Driving 3, Driving 4, Driving 5	99.00	Naivebc
3	3	Driving 1, Driving 2, Driving 5	33.85	Naivebc
4	4	Driving 1,Driving 2,Driving 3, Driving 4	91.80	Naivebc

The above table shows the error rates of Naïve Bayes trainable combination method. In the first combination list, Naïve Bayes method had better error rates than all fixed rules. However LDC method obtained better error rate than the Naïve Bayes method. In the second combination list, Naïve Bayes method showed worse error rates than all fixed rules. Moreover Fisher, LDC and NMC methods reached better error rates than Naïve Bayes. In the third combination list, Naïve Bayes method obtained better error rates than the all fixed rules. Also, it is better than LDC and NMC methods. However for this combination list the best result was obtained from Fisher method. In the last combination list, Naïve Bayes method showed worse error rates than all fixed rules and trainable combiners.

As a summary Naïve Bayes rule obtained very high error rates in second and fourth combination lists. In these lists fixed rules error rates became better. Especially in second experiment Naïve Bayes method showed the worst error rate of all the other combination lists with both fixed rules and trainable combiners. The best result of the Naïve Bayes method was obtained in the third combination list where we combined three channels.

Table 7.11 – Error rates using Perl trainable combination method for different modalities

	COMBINE	COMBINE LIST	ERROR (%)	METHOD
1	1	Driving 1, Driving 2	66.00	Perlc
2	2	Driving 1, Driving 2, Driving 3, Driving 4, Driving 5	38.70	Perlc
3	3	Driving 1, Driving 2, Driving 5	55.05	Perlc
4	4	Driving 1, Driving 2, Driving 3, Driving 4	44.25	Perlc

The above table shows the error rates of Perl trainable combination method. In the first combination list, Perl method showed better error rates than all fixed rules. However Fisher, LDC and Naïve Bayes methods reached better error rates than Perl method. In the second combination list, Perl method obtained better error rates than all fixed rules but in this experiment the best result was obtained from Fisher method. In the third combination list, Perl method had better error rates than all fixed rules. However for trainable combiners Fisher, LDC and Naïve Bayes methods obtained better results than Perl method. In the last combination list, Perl method got better error rates than all fixed rules. Also this result was better than the results of LDC, NMC and Naïve Bayes methods but for this list the best result was obtained from Fisher method.

As a summary Perl rule had low error rates. The best result of the Perl method was obtained in the second combination list where we combined all the channels. Perl method showed the worst result in the first combination list which was the 66 percent. Also this result was better than the results of fixed rules.

Table 7.12 – Error rates using Parzen trainable combination method for different modalities

	COMBINE	COMBINE LIST	ERROR (%)	METHOD
1	1	Driving 1, Driving 2	0.55	Parzenc
2	2	Driving 1, Driving 2, Driving 3, Driving 4, Driving 5	0.35	Parzenc
3	3	Driving 1, Driving 2, Driving 5	0.65	Parzenc
4	4	Driving 1, Driving 2, Driving 3, Driving 4	0.80	Parzenc

The above table shows the error rates of Parzen trainable combination method. We constituted four different combination lists and run the Parzen trainable combiner. The results showed that for all combination lists Parzen method reached lower error rates than fixed rules and trainable combiners. Parzen rule showed very low error rates. The best result of the Parzen method was obtained in the second experiment where we combined all the channels. Parzen method had the worst result in the fourth experiment which was the 0.8 percent.

Table 7.13 – Error rates using K-NN trainable combination method for different modalities

	COMBINE	COMBINE LIST	ERROR (%)	METHOD
1	1	Driving 1, Driving 2	0.55	KNNc
2	2	Driving 1, Driving 2, Driving 3, Driving 4, Driving 5	0.35	KNNc
3	3	Driving 1, Driving 2, Driving 5	0.65	KNNc
4	4	Driving 1, Driving 2, Driving 3, Driving 4	0.75	KNNc

The above table shows the error rates of K-NN trainable combination method. We constituted four different combination lists and run K-NN trainable combiner. The results showed that for all combination lists K-NN method reached lower error rates than fixed rules and trainable combiners except Parzen method. For first three combination list K-NN and Parzen reached same error rates. However in the last combination list K-NN rule showed the best performance and obtained better result than Parzen method.

As a summary K-NN method reached very low error rates. The best result of the K-NN method was obtained in the second combination list where we combined all the channels. K-NN method showed the worst result in the last combination list which was the 0.75 percent. Also this result was better than the results of all fixed rules and trainable combiners.

Table 7.14 – Driver Recognition results using fixed rules and trainable combiner

Combination Methods	Combination List 1	Combination List 2	Combination List 3	Combination List 4
Maxc	**85.95**	91.60	**89.30**	90.75
Minc	81.40	90.55	86.50	90.50
Medianc	78.95	87.70	86.45	86.10
Meanc	78.95	85.95	82.30	85.35
Prodc	77.95	84.85	81.20	85.05
Fisher	43.45	16.10	29.50	24.55
LDC	35.50	85.55	48.00	78.70
NMC	66.65	54.65	61.40	60.40
Naive Bayes	43.70	**99.00**	33.85	**91.80**
Perl	66.00	38.70	55.05	44.25
Parzen	**0.55**	**0.35**	**0.65**	0.80
KNN	**0.55**	**0.35**	**0.65**	**0.75**

As a general summary, according to the above table, most cases trainable combiners showed better results than the fixed rules. In the first combination list we combined two channels, break pedal pressure and accelerator pedal pressure then we run seven different trainable combiners. The above results show that we obtained the best results from the Parzen and K-NN methods with the error rate of 0.55 percent. The worst results were obtained from the NMC and Perl methods. But these results were better than the results of fixed rules. In the second combination list we combined all the channels, break pedal pressure, accelerator pedal pressure, engine speed, vehicle speed and steering wheel angle. The above results show that we reached the best results from the Parzen and K-NN methods with the error rate of 0.35 percent. Also, Fisher method showed the secondary good result. The worst results were obtained from the LDC and Naïve Bayes methods. In this experiment both of the LDC and Naïve Bayes showed their worst performances in all the other experiments. In the third combination list we combined three channels, break pedal pressure, accelerator pedal pressure and steering wheel angle then we run all trainable combiners. The above results show that we reached the best results from the Parzen and K-NN methods with the error rate of 0.65 percent. Also, Fisher method obtained the secondary good result. The worst results were obtained from the NMC method. In the last combination list we combine four channels, break pedal pressure, accelerator pedal pressure, engine speed, and vehicle speed then we run all trainable combiners. The above results show that we obtained the best results from the Parzen and K-NN methods with the error rates of 0.80 and 0.75 percent respectively. Also, Fisher method showed the secondary good result. The worst results were obtained from the Naïve Bayes method.

The best experiment results were obtained from the Parzen and K-NN methods. These methods have very low error rates and best performances. The lowest error rate was obtained from the second combination list which was done with the all channels. In this experiment Parzen and K-NN methods showed the same best result and the error rate was the 0.35 percent. Also, this error rate was the best error rate in all of the experiments results. We expected that the second experiment will show the best results. Because this experiment was done with all channels and the combination of these channels can have lower error rates than the others also can be the most reliable combination. So, our expectation became fact for trainable combiners except LDC and Naïve Bayes. Moreover, Naïve Bayes rule showed very high error rates in second and fourth combination lists. In these combination lists the fixed rules error rates became better. Especially in second combination list Naïve Bayes method showed the worst error rate of the all the experiments both fixed rules and trainable combiners.

In the next part we will present the driver verification experiment results.

7.2.2 Driver Verification Results

The tables in the below part show the driving verification experiments results using the fixed rules and trainable combiners. Verification is the process that aims to find the answer to the question “Is he/she the person who he/she claims to be?”. In this process the subject claims to be a person whose biometric information are already existent or known. During this process new biometric information is taken from the subject and a comparison is done between the new biometric information and the stored data. If the new information is matched with a stored template, the verification process will finish successfully.

The performance measurement criteria of these experiments are the false accept and false reject rates. The false accept rate is the probability that the system incorrectly declares a successful match between the input pattern and a non-matching pattern in the database. It gives the percent of invalid matches as an output. This type of error is very critical for security issues. Because these invalid users are accepted by the system. The false reject rate is the probability that the system incorrectly declares a failure of match between the input pattern and a non-matching pattern in the database. It gives the percent of valid users who are rejected as an output. During these experiments we determined a threshold value and according to this value we obtained the results of false acceptance and rejection.

7.2.2.1 Fixed Rules

The tables in the below part show the driving verification experiments results using the fixed rules.

Table 7.15 – False Accept and False Reject Rates using fixed rules

Method	False Accept (%)	False Reject (%)
Maxc	90.75	0
Minc	90.50	2.30
Medianc	86.10	0
Meanc	85.35	0
Prodc	85.05	5.65

Before the experiment we determined a threshold value which is valid for all combiners, but all the methods did not obtain successful results with this threshold. If we applied different suitable threshold values for every combiner then we obtained results different than our results. So, these results were depended on the threshold.

If we look at the fixed rules Product rule shows better performance than the others in the false acceptance. In false reject rate Maximum, Median and Mean rules show the best performance. So, they never reject a valid user.

Maximum fixed rule got 90.75 percent false accept rate which is very high rate and it did not have false reject. Minimum fixed rule got 90.50 percent false accept rate which is very high rate and it got 2.30 percent false reject rate. Median fixed rule got 86.10 percent false accept rate and it did not any have false reject. Mean fixed rule got 85.35 percent false reject rate which is very high rate and it did not have any false reject. Product fixed rule got 85.05 percent false accept rate and 5.65 percent false reject rate.

7.2.2.2 Trainable Combiners

The tables in the below part show the driving verification experiments results using the trainable combiners.

Table 7.16 – False Accept and False Reject Rates using trainable combination methods

Method	False Accept (%)	False Reject (%)
Fisherc	24.55	0
LDC	0	21.35
NMC	60.40	0
Naivebc	0	5.45
Perlc	44.20	0.75
Parzenc	0.4	6.25
knnc	0.75	0

Before the experiment we determined a threshold value which is valid for all combiners, but all the methods did not obtain successful results with this threshold. If we applied different suitable threshold values for every combiner then we obtained results different than our results. So, these results were depended on the threshold.

According to our results, for the same threshold value K-NN trainable combiner shows the best performance. It has only 0.75 percent false accept which is very low rate because we were done the experiment with 100 subjects and it has not got false reject.

Fisher trainable combiner got 24.55 percent false accept rate which is low rate and it did not have false reject. LDC trainable combiner got 21.35 percent false reject rate which is low rate and it did not have false accept. NMC trainable combiner got 60.40 percent false accept rate and it did not have false reject. Naïve Bayes trainable combiner got 5.45 percent false reject rate which is low rate and it did not have false accept. Perl trainable combiner got 44.20 percent false accept rate and 0.75 percent false reject rate which is very low rate. Parzen trainable combiner got 0.4 percent false accept rate and 6.25 percent false reject rate which are very low error rates.

In the trainable combiners, LDC and Naïve Bayes show the best performance in false acceptance. So, they never accept an invalid user to the system. Also, Parzen and K-NN show secondary better performances. In false reject rate Fisher, NMC and K-NN rules show the best performances. So, they never reject a valid user to the system.

In verification study a comparison done between the possibility ratios to a threshold value. For different thresholds, false-accept rate versus false-reject rate was plotted using the receiver operating characteristics curve. In figure 7.6 there are ROC curves for different combiners.

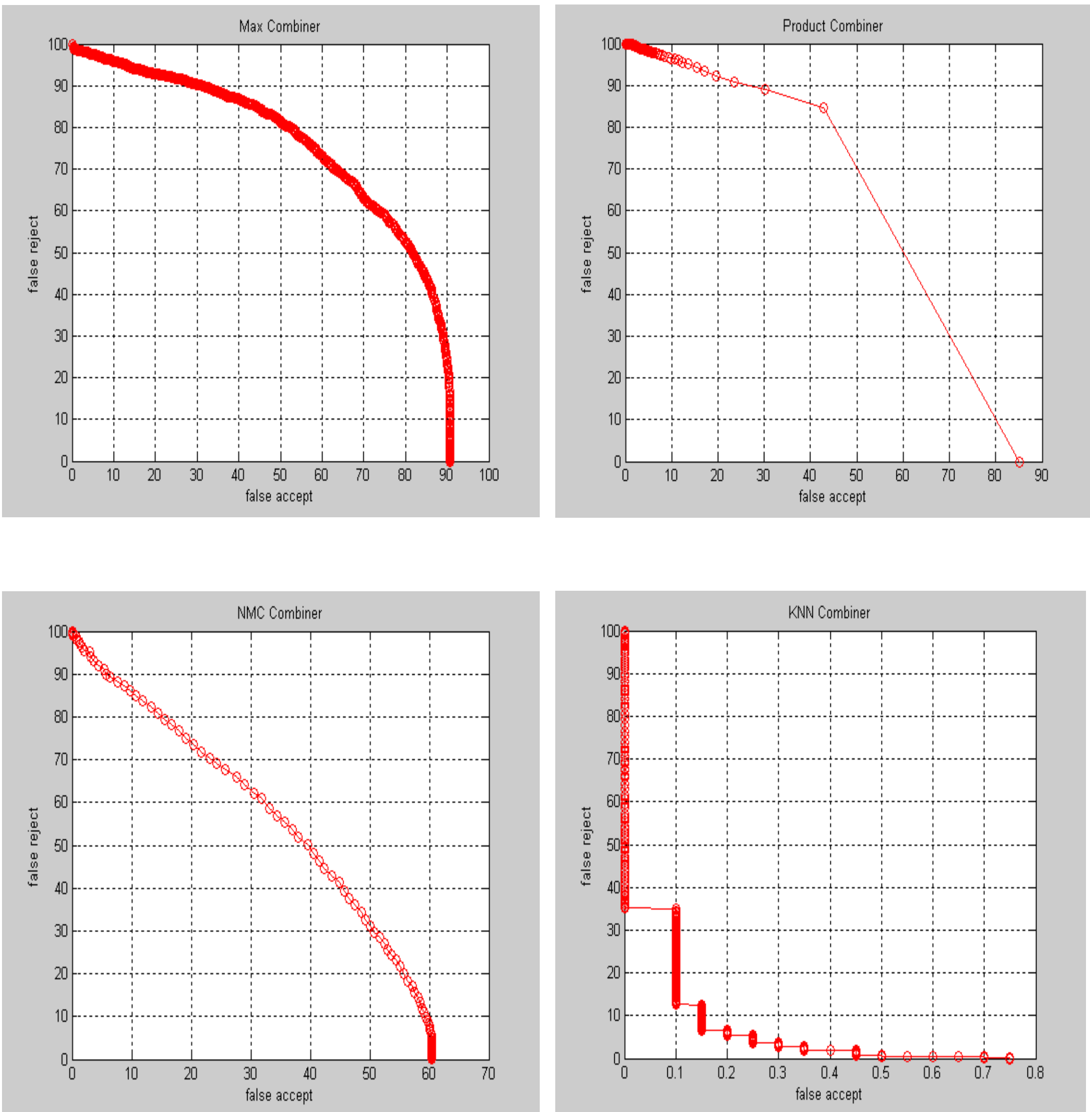

Figure 7.6– ROC curves of Max Rule, Product Rule, NMC and KNN

Chapter 8
Conclusion and Future Work

The main goal of our research is to facilitate vehicle-person interaction. In this study using driver behavior signals we accomplished driver recognition and driver verification. During this study, we followed the basic steps of a multi-modal biometric system as explained in chapter 2.

The first step of recognition/verificaion system is the data acquisition. In our study instead of collecting the driver data we used the CIAIR database of the Nagoya University for driver recognition and verification. So, our study did not involve the data acquisition step. We take five different driving behavior signals from the CIAIR database. These signals were:

1. Break pedal pressure
2. Accelerator pedal pressure
3. Engine speed
4. Vehicle speed
5. Steering wheel angle

Driver recognition and verification experiments were done with the 100 person subset of the CIAIR database. This database consists of 50 female drivers and 50 male drivers.

The second and the third parts of a biometric system are data compression and decompression. As our study did not involve data storage or transfer, we omitted these stages. The fourth step is the feature extraction algorithm which is used for producing a feature vector.

The components of the feature vector are numerical characterization of the biometrics. Gaussian Mixture Model is used for modeling the driver behavior.

The fifth part of the system is the matcher which compares the feature vectors for obtaining a similarity score. In this step we used well-known Expectation Maximization algorithm for training the GMMs.

The final step of the system is the decision maker. In our study we used decision fusion which combines decisions that come from several experts. In other words, if the experts return a confidence (score) instead of a decision, and we deal with a decision fusion problem. In this study we used two different combination methods, namely fixed methods and trainable combiners.

Fixed methods have simple fixed rules to combine information from a set of classifiers. In this study five fixed rules were used. These are maximum, minimum, median, mean and product rules. Trainable methods have some free parameters that can be trained on a separate part of the training data. In this study seven trainable combiners were used. These combiners are fisher, linear discriminant, nearest mean, naïve bayes, perl, parzen and k-nearest neighbor.

In this study we use biometric system for person recognition and verification. The recognition process aims to find the answer of the question “Who is he/she?” In this process there exists biometric information of the subject in the database or stored somewhere. At this time new information is taken from the subject and a comparison is done between this new biometric information and all the other stored biometric information.

The verification process aims to find the answer of the question “Is he/she the person he/she claims to be”. In this process the subject claims that he/she is a person whose biometric information is already exists or stored in the database. In this case again new biometric information is taken from the subject and a comparison is done between this new biometric information and claimed biometric information. If the new biometric information is matched with the stored biometric information, the verification process will finish successfully and the subject is accepted otherwise he/she is rejected.

The first study is the driver recognition using the individual modalities. The individual performance results showed that, a single driving signal is not appropriate for biometric identification. GMMs trained independently on single modalities had very high error rates.

The second study is the driver recognition using the fixed rules and trainable combiners. Four different driver recognition experiments were done using five different fixed rules. But fixed rules do not show good performances. Their error rates are very high. The lowest error rate was taken from the first experiment which was done with the break pedal pressure and accelerator pedal pressure channels. In this experiment product rule showed the best result and the error rate was 77.95 percent. Also, this error rate is the best error rate in all of the fixed rules' experiments results. Moreover, product rule shows the best performance in all the experiments and has the lowest error rates. Mean rule follows the product rule with the secondary lowest error rates.

We expected that the second experiment will show the best results. Because this experiment was done with the all channels such as break pedal pressure, accelerator pedal pressure, engine speed, vehicle speed and steering wheel angle. So, the combination of these channels can have lower error rates than the others and be the most reliable combination. But the results of this experiment were the worst ones. In this experiment we increased the dimension for getting better and more accurate results but we faced with the problems of high dimensional data, which called the *curse of dimensionality*.

In the third experiment three channels were combined. These channels are break pedal pressure, accelerator pedal pressure and steering wheel angle. The results of the minimum and median rule were approximately the same.

In the last experiment we combine the four channels which are break pedal pressure, accelerator pedal pressure, engine speed and vehicle speed. The error rates of this

experiment were also high. The highest error rate was taken from the maximum rule. Moreover, maximum rule shows the worst performance in all experiments and has the highest error rates.
In driver recognition with trainable combiners we constituted four different combination lists and run seven algorithms. In most cases trainable combiners showed better results than the fixed rules. In the first combination list we combined two channels which are break pedal pressure and accelerator pedal pressure and then ran seven different trainable combiners. We obtained the best results from the Parzen and K-NN methods with the error rate of 0.55 percent. The worst results were obtained from the NMC and Perl methods. But these results were better than the fixed rules results.

In the second combination list we combined all the channels break pedal pressure, accelerator pedal pressure, engine speed, vehicle speed and steering wheel angle. We achieved the best results from the Parzen and K-NN methods with the error rate of 0.35 percent. Also, Fisher method had the secondary good result. The worst results were obtained from the LDC and Naïve Bayes methods. In this list both of the LDC and Naïve Bayes showed their worst performances in all experiments.

In the third experiment we combined three channels break pedal pressure, accelerator pedal pressure and steering wheel angle then we run all trainable combiners. We obtained the best results from the Parzen and K-NN methods with the error rate of 0.65 percent. Also, Fisher method had the secondary good result. The worst results were obtained from the NMC method.

In the last experiment we combined four channels break pedal pressure, accelerator pedal pressure, engine speed, and vehicle speed then we run all trainable combiners. We obtained the best results from the Parzen and K-NN methods with the error rate of 0.80 percent. Also, Fisher method had the secondary good result. The worst results were obtained from the Naïve Bayes method.

Overall, the best experiment results were obtained from the Parzen and K-NN methods. These methods reached very low error rates and best performances. The lowest error rate was obtained from the second combination list which was done with the all channels break pedal pressure, accelerator pedal pressure, engine speed, vehicle speed and steering wheel angle. In this experiment Parzen and K-NN methods had the same best result and the error rate was the 0.35 percent. Also, this error rate was the best error rate in all of the experiments results. We expected that the second experiment will show the best results. Because this experiment was done with the all channels and the combination of these channels can have lower error rates than the others also can be the most reliable combination. So, our expectation became fact for trainable combiners except LDC and Naïve Bayes. Moreover, Naïve Bayes rule had very high error rates in second and fourth combination lists. In these lists the fixed rules error rates became better. Especially in second combination list Naïve Bayes method showed the worst error rate of the all the experiments both fixed rules and trainable combiners.

In the verification experiment we first determined a threshold value, but all the methods did not obtain successful results with this threshold. If we applied different suitable threshold values for every combiner then we obtained results different than our results. So, these results were depended on the threshold.

According to our results, for the same threshold value K-NN trainable combiner shows the best performance. It has only 0.75 percent false acceptance which is very low rate because we were done the experiment with 100 subjects and it has not got false reject.

In the trainable combiners, LDC and Naïve Bayes showed the best performance in false acceptance. So, they never accepted an invalid user to the system. Also, Parzen and K-NN obtained better performance. In false reject rate Fisher, NMC and K-NN rules showed the best performance. So, they never rejected a valid user to the system.

If we look at the fixed rules Product rule obtained better performance than the others in the false acceptance. In false reject rate Maximum, Median and Mean rules showed the best performance. So, they never rejected a valid user to the system.

In conclusion, in this study we accomplished driver recognition and driver verification. In driver recognition most cases trainable combiners showed better results than the fixed rules. In the results of the fixed rules, the product rule showed the best performance. Also, the secondary good results were obtained from the mean rule. The most reliable combine list was the first one which was done with the break pedal pressure and accelerator pedal pressure. The combination of the all driving signals had the worst performance. In the results of the trainable combiners, the Parzen and K-NN showed the best performance. In Parzen and K-NN, the combination of the all driving signals had the best result. Also, in driver verification study trainable combiners showed better results than the fixed rules. K-NN trainable combiner showed the best performance. So, note that driver verification results were depended on the threshold.

Finally after this thesis we planned to realize two more studies as future work. In the first study we will do driver recognition and driver verification with the all CIAIR database subjects (343). In this thesis we did not use all the database because of the insufficient memory of our computers. The future work will be on the implementation of these studies.

The second study will be the detection of the driver fatigue. Traffic safety is very important issue and one of the important reasons of the accidents is the driver fatigue. If the fatigue of the driver is detected early, the drivers and passengers will be prevented from the accidents. Also, we started to work on this project and we obtained the primary results.

In driver fatigue detection study we used the fatigue data of OTAM database. The experiment was done with 33 people, male and female. The data collected according to a track which was started from OTAM, which is a research center in ITU Ayazağa Campus, and then continues with the certain proportion of main road and certain proportion of city traffic (Figure 3.5).

In the driver fatigue experiment, a cleaning or eliminating procedure was applied to the OTAM data. First two minutes and last two minutes of the data were eliminated. This elimination procedure was applied to the entirety of the data. The obtained results of these intervals can not give a correct or valid result to us. Because, these intervals were track dependent such as in the last two minutes of the driving the driver can park his car and the driving signals can mislead us. So, after the elimination step the data were divided into the two parts. First minutes of the driving data grouped as not fatigued data and the last minutes of the driving data grouped as fatigued data. After this grouping some classification algorithms were run with these data. The classification algorithms that were used in the experiments are Ada_Boost, C 4_5, CART, EM, LMS, Nearest Neighbor and Parzen. This experiment was done with the 33 subject both male and female and Classification Toolbox was used [17]. The following table shows the result of one subject.

Table 8.1 – Driving Fatigue detection of Subject 03-04-2007 IM1009 with different classification algorithms

Ada_Boost	**Class 1**	**Class 2**
Test Set Errors (%)	1.9	18
Train Set Errors (%)	1.9	20

C 4_5	**Class 1**	**Class 2**
Test Set Errors (%)	5.2	17
Train Set Errors (%)	4.1	13

CART	**Class 1**	**Class 2**
Test Set Errors (%)	29	20
Train Set Errors (%)	25	17

EM	**Class 1**	**Class 2**
Test Set Errors (%)	30	16
Train Set Errors (%)	28	14

LMS	**Class 1**	**Class 2**
Test Set Errors (%)	0	100
Train Set Errors (%)	0	100

Nearest Neighbour	**Class 1**	**Class 2**
Test Set Errors (%)	6.3	17
Train Set Errors (%)	4	11

Parzen	**Class 1**	**Class 2**
Test Set Errors (%)	0	100
Train Set Errors (%)	0	99

After the experiment we obtained good results but observed that these data may be track dependent. If they are track dependent, our results will become incorrect. Because according to the results we said that this driver is tired or not and while we are telling this we examine the driving behavior signals of the driver like steering wheel angle. For example if the driver has a large steering wheel angle value we can said that this driver is tired and he/she did not see the barrier early when driving the car then he/she suddenly see the barrier and make unexpected steering wheel movement for escape. It is a one option and it is very sensible. But there is another option such that the roads condition may be caused this type of movement. In this case the driver is not tired the movement is caused by the roads condition. This second option is also sensible. So, there is a trade off situation between two options. Therefore, we want to plan to study on this issue later on as a future work.

References

[1] H. Erdogan, A. Ercil, H.K. Ekenel, S.Y. Bilgin, I. Eden, M. Kirisci, H. Abut, "Multi-modal person recognition for vehicular applications," *N.C. Oza et al. (Eds.): MCS 2005, LNCS 3541*, pp. 366 – 375, Monterey CA, Jun. 2005.

[2] A.K. Jain*,* A. Ross*,* and S. Prabhakar, "An Introduction to Biometric Recognition," *IEEE Transactions on Circuits and Systems for Video Technology, Special Issue on Image- and Video-BasedBiometrics, Vol. 14, No. 1,* January 2004.

[3] W. Shen and T. Tan, "Automated biometrics-based personal identification," *Proc. Natl. Acad. Sci. USA* Vol. 96, pp. 11065–11066, From the Academy September 1999

[4] Wikipedia, the free encyclopedia : *Biometrics*
http://en.wikipedia.org/wiki/Biometrics

[5] P. Paalanen, "Bayesian Classification using Gaussian Mixture Model and EM Estimation : Implementations and Comparisons"
http://www.it.lut.fi/project/gmmbayes/downloads/doc/report04.pdf

[6] "Introduction to Data Fusion"
http://www.sic.rma.ac.be/Research/Fusion/Intro/content.html

[7] J. Kittler, M. Hatef, R.P.W. Duin, and J. Matas
"On Combining Classifiers, " IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol: 20, No.3, MARCH 1998

[8] C.M. BISHOP, *Pattern Recognition and Machine Learning*

[9] Wikipedia, the free encyclopedia : *Naïve Bayes* http://en.wikipedia.org/wiki/Naive_Bayes

[10] R.P.W. Duin, E. Pękalska, " The Dissimilarity Representation for Pattern Recognition: Foundations And Application"

[11] C.J. Veenman and M.J.T. Reinders, "The Nearest Sub-class Classi_er: a Compromise between the Nearest Mean and Nearest Neighbor Classifer," *IEEE Transactions on PAMI, Vol. 27, No. 9*, pp. 1417-1429, September 2005

[12] Wikipedia, the free encyclopedia : *Perceptron* http://en.wikipedia.org/wiki/Perceptron

[13] The MathWorks, Accelerating the pace of engineering and science : filter http://www.mathworks.com/access/helpdesk/help/techdoc/index.html?/access/helpdesk/help/techdoc/ref/filter.html&http://www.google.co.jp/search?hl=en&q=filter+function+in+matlab

[14] The MathWorks, Accelerating the pace of engineering and science : decimate http://www.mathworks.com/access/helpdesk/help/toolbox/signal/index.html?/access/helpdesk/help/toolbox/signal/decimate.html&http://www.google.com.tr/search?hl=tr&q=decimate+in+matlab&meta=

[15] K. Igarashi, C. Miyajima, K. Itou, K. Takeda, H. Abut and F. Itakura, "Biometric Identification Using Driving Behavior," *Proceedings IEEE ICME 2004*, Taipei, Taiwan, June 27-30, 2004

[16] General Directorate of Security Head of Traffic Services Traffic Research Center http://www.trafik.gov.tr/english/traffic_safety/traffic_safety_sleepless_and_tired_driving.asp

[17] R.O. Duda, P.E. Hart, D.G. Stork , *Pattern Classification*

Curriculum Vitae