**CHAPTER 3**

**BACKGROUND THEORY**

This chapter describes about background theories that are used in Driver Drowsiness Detection and Alert System such as Ensemble of Regression Tree, Cascade Classifiers, Eye Aspect Ratio and Machine Learning, etc.

**3.1. Drowsiness**

Drowsiness is a physiological state with a tendency to fall asleep. Technically, drowsiness is different from the fatigue that is the lack of willingness to continue performing the same activity. Traffic accidents due to human errors cause many deaths and injuries around the world. Drowsiness and sleeping while driving is now identified as one of the reasons behind fatal crashes and highway accidents caused by drivers. Fatigue occurs by performing tasks that are always performed in the same way using the same muscle groups, their repetition rate is high and are usually performed by adopting forced postures such as monitoring a screen. A person may be fatigued without being drowsy, but conditions that produce fatigue such as driving cars over great distances unmask the presence of physiological drowsiness, but do not cause fatigue.

According to many researches, drowsiness is related to thousands of traffic accidents each year, the accidents produces approximately 50% of death or serious injuries, as they tend to be impacts at high speed because the driver who has fallen asleep cannot brake or deviate to avoid or reduce impact. To mitigate these accidents, manufacturers have developed drowsiness detection systems that recognize signs of possible drowsiness, alerting the driver to their condition.

Among the effects of being sleepy we have a lowered wakefulness, reaction time, psychomotor coordination and decreased information processing. For the driver the main effect is the progressive withdrawal of attention in demands of road, traffic and signaling, which causes a low driving performance producing accidents. People who are drowsy have signs like frequent blinking, rubbing eyes, repeated yawning, head tilt, and distractions are the most important among which it can mention.

**3.2. Machine Learning**

Machine learning is the scientific study of algorithms and statistical models that computer systems use to effectively perform a specific task without using explicit instructions, relying on models and inference instead. It is seen as a subset of artificial intelligence. Machine learning algorithms build a mathematical model of sample data, known as training data, in order to make predictions or decisions without being explicitly programmed to perform the task. Machine learning algorithms are used in the applications of email filtering, detection of network intruders, and computer vision, where it is infeasible to develop an algorithm of specific instructions for performing the task.

Machine learning is closely related to computational statistics, which focuses on making predictions using computers. The study of mathematical optimization delivers methods, theory and application domains to the field of machine learning. Data mining is a field of machine learning. And focuses on exploratory data analysis through unsupervised learning. In its application across business problems, machine learning is also referred to as predictive analytics.

The name machine learning was coined in 1959 by Arthur Samuel. Tom M Mitchell provided a widely quoted, more formal definition of the algorithms studied in the machine learning field A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E. This definition of the tasks in which machine learning is concerned offers a fundamentally operational definition rather than defining the field in cognitive terms. In Alan Turing’s proposal paper Computing Machinery and intelligence, the question can machine think? is replaced with the question can machines do what we can do?

Arthur Samuel, an American pioneer in the field of computer gaming and artificial intelligence, coined the term Machine Learning in 1959 while at IBM.As a scientific endeavor, machine learning grew out of the quest for artificial intelligence. Already in the early days of AI as an academic discipline, some researchers were interested in having machines learn from data. They attempted to approach the problem with various symbolic methods, as well as what were then termed neural networks; these were mostly perceptron and other models that were later found to be reinventions of the generalized linear models of statistics. Probabilistic reasoning was also employed, especially in automated medical diagnosis. Machine learning and data mining often employ the same methods and overlap significantly, but while machine learning focuses on prediction, based on known properties learned from the training data, data mining focuses on the discovery of unknown properties in the data.

Data mining uses many machine learning methods, but with different goals; on the other hand, machine learning also employs data mining methods as unsupervised learning or as a preprocessing step to improve learner accuracy. Machine learning also has intimate ties to optimization: many learning problems are formulated as minimization of some loss function on a training set of examples. Loss functions express the discrepancy between the predictions of the model being trained and the actual problem instances. The difference between the two fields arises from the goal of generalization: while optimization algorithms can minimize the loss on a training set, machine learning is concerned with minimizing the loss on unseen samples.

3.2.1. Machine Learning Algorithms

Machine learning algorithms are programs that can learn from data and improve from experience, without human intervention. Machine learning algorithms are described as learning a target function (f) that best maps input variables (x) to an output variable (y): y = f(x). The most common type of machine learning is to learn the mapping y = f(x) to make predictions of y for new x. This is called predictive modeling or predictive analytics and the goal is to make the most accurate predictions possible. Top 10 machine learning algorithms used by data scientists are:

* Linear Regression: Linear Regression is a basic and commonly used type of predictive analysis. The overall idea of regression is to examine two things: does a set of predictor variables do a good job in predicting an outcome variable and which variables in particular are significant predictors of the outcome variables.
* Logistic Regression: Logistic Regression is actually a binomial classifier that acts like a light switch. A light switch essentially has two states, on and off. Logistic regression takes input data and classifies it as category or not category, on or off expressed as 1 or 0, based on the strength of the input’s signal.
* Decision Tree: A decision tree is a series of nodes, a directional graph that starts at the base with a single node and extends to the many leaf nodes that represent the categories that the tree can classify. Another way to think of a decision tree is as a flow chart, where the flow starts at the root node and ends with a decision made at the leaves. It is a decision-support tool. It uses a tree-like graph to show the predictions that result from a series of feature-based splits.
* Random Forest: Random forests are made of many decision trees. They are ensembles of decision trees, each decision tree created by using a subset of the attributes used to classify a given population. Those decision trees vote on how to classify a given instance of input data, and the random forest bootstraps those votes to choose the best prediction. This is done to prevent over fitting, a common flaw of decision trees.
* Linear Discriminant Analysis: Linear discriminant analysis (LDA), Normal Discriminant Analysis (NDA), or discriminant function analysis is a generalization of Fisher’s linear discriminant, a method used in statistic, pattern recognition and machine learning to find a linear combination of features that characterizes or separates two or more classes of objects or events. The resulting combination may be used as a linear classifier or dimensionality reduction.
* Naïve Bayes: Naïve Bayes is a very popular classification algorithm that is mostly used to get the base accuracy of the dataset. Naive Bayes classifiers are a family of simple “probabilistic classifiers” based on applying Bayes’ theorem with strong independence assumptions between the features. It has been successfully used for many purposes such as medical diagnosis, spam filters.
* K-Nearest Neighbors: In pattern recognition, the k-nearest neighbors algorithm (K-NN) is a non-parametric method used for classification and regression. In both cases, the input consists of the k closest training examples in the feature space. The similarity between instances is calculated using measures such as Euclidean distance and Hamming distance. The output depends on whether K-NN is used for classification or regression.
* Learning Vector Quantization: In computer science, Learning Vector Quantization (LVQ), is a prototype-based supervised classification algorithm. LVQ is the supervised counterpart of vector quantization systems.
* Support Vector Machine: Support Vector Machines (SVM) are supervised learning models with associated learning algorithm models with associated learning algorithms that analyze data used for classification and regression analysis. SVM finds a hyper plane in an N-dimensional space (N is the number of features) that distinctly classifies the data points.
* Bagging: Bootstrap Aggregation (or Bagging for short), is a simple and very powerful ensemble method. Bagging is the application of the Bootstrap procedure to a high-variance machine learning algorithm, typically decision trees.
* Boosting and AdaBoost: Boosting is a general ensemble method that creates a strong classifier from a number of weak classifiers. AdaBoost was the first really successful boosting algorithm developed for binary classification. It is the best starting point for understanding boosting.

There are many algorithms to implement this approach, in other side that algorithms depend on many factor, including (1) the size, quality and nature of data, (2) available computational time, (3) the urgency of the task, (4) what you want to do with the data.

**3.3. Cascading classifiers**

Cascading is a particular case of ensemble learning based on the concatenation of several classifiers, using all information collected from the output from a given classifier as additional information for the next classifier in the cascade. Unlike voting or stacking ensembles, which are multi expert systems, cascading is a multistage one.

Cascading classifiers are trained with several hundred "positive" sample views of a particular object and arbitrary "negative" images of the same size. After the classifier is trained it can be applied to a region of an image and detect the object in question. To search for the object in the entire frame, the search window can be moved across the image and check every location for the classifier. This process is most commonly used in image processing for object detection and tracking, primarily facial detection and recognition.

3.3.1. Cascade training

Cascades are usually done through cost-aware ADAboost. The sensitivity threshold (0.8 in our example) can be adjusted so that there is close to 100% true positives and some false positives. The procedure can then be started again for stage 2, until the desired accuracy/computation time is reached.

After the initial algorithm, it was understood that training the cascade as a whole can be optimized, to achieve a desired true detection rate with minimal complexity. Examples of such algorithms are RCBoost, ECBoost or RCECBoost. In their most basic versions, they can be understood as choosing, at each step, between adding a stage or adding a weak learner to a previous stage, whichever is less costly, until the desired accuracy has been reached. Every stage of the classifier cannot have a detection rate (sensitivity) below the desired rate, so this is a constrained optimization problem. To be precise, the total sensitivity will be the product of stage sensitivities.

Cascade classifiers are available in OpenCV, with pre-trained cascades for frontal faces and upper body. Training a new cascade in OpenCV is also possible with either haar-training or train cascades methods. This can be used for rapid object detection of more specific targets, including non-human objects with Haar-like features. The process requires two sets of samples: negative and positive, where the negative samples correspond to arbitrary non-object images. The time constraint in training a cascade classifier can be circumvented using cloud-computing methods.

**3.4. Regression Tree**

A regression tree is built through a process known as binary recursive partitioning, which is an iterative process that splits the data into partitions or branches, and then continues splitting each partition into smaller groups as the method moves up each branch.

Initially, all records in the Training Set (pre-classified records that are used to determine the structure of the tree) are grouped into the same partition. The algorithm then begins allocating the data into the first two partitions or branches, using every possible binary split on every field. The algorithm selects the split that minimizes the sum of the squared deviations from the mean in the two separate partitions. This splitting rule is then applied to each of the new branches. This process continues until each node reaches a user-specified minimum node size and becomes a terminal node. (If the sum of squared deviations from the mean in a node is zero, then that node is considered a terminal node even if it has not reached the minimum size.)

All regression techniques contain a single output (response) variable and one or more input (predictor) variables. The output variable is numerical. The general regression tree building methodology allows input variables to be a mixture of continuous and categorical variables. A decision tree is generated when each decision node in the tree contains a test on some input variable's value. The terminal nodes of the tree contain the predicted output variable values. A Regression tree may be considered as a variant of decision trees, designed to approximate real-valued functions, instead of being used for classification methods.

**3.5. Ensemble Methods**

XLMiner V2015 offers three powerful ensemble methods for use with Regression trees: bagging (bootstrap aggregating), boosting, and random trees. The Regression Tree Algorithm can be used to find one model that results in good predictions for the new data. We can view the statistics and confusion matrices of the current predictor to see if our model is a good fit to the data; but how would we know if there is a better predictor just waiting to be found? The answer is that we do not know if a better predictor exists.

However, ensemble methods allow us to combine multiple weak regression tree models, which when taken together form a new, accurate, strong regression tree model. These methods work by creating multiple diverse regression models, by taking different samples of the original data set, and then combining their outputs. (Outputs may be combined by several techniques for example, majority vote for classification and averaging for regression) This combination of models effectively reduces the variance in the strong model. Three types of ensemble methods are:

* Bagging (bootstrap aggregating) was one of the first ensemble algorithms ever to be written. It is a simple algorithm, yet very effective. Bagging generates several Training Sets by using random sampling with replacement (bootstrap sampling), applies the regression tree algorithm to each data set, then takes the average amongst the models to calculate the predictions for the new data. The biggest advantage of bagging is the relative ease that the algorithm can be parallelized, which makes it a better selection for very large data sets.
* Boosting builds a strong model by successively training models to concentrate on records receiving inaccurate predicted values in previous models. Once completed, all predictors are combined by a weighted majority vote. XLMiner offers three variations of boosting as implemented by the AdaBoost algorithm (one of the most popular ensemble algorithms in use today): M1 (Freund), M1 (Breiman), and SAMME (Stagewise Additive Modeling using a Multi-class Exponential).
* The random trees method (random forests) is a variation of bagging. This method works by training multiple weak regression trees using a fixed number of randomly selected features (sqrt[number of features] for classification and number of features/3 for prediction), then takes the average value for the weak learners and assigns that value to the strong predictor. Typically, the number of weak trees generated could range from several hundred to several thousand depending on the size and difficulty of the training set. Random trees are parallelizable, since they are a variant of bagging. However, since random trees selects a limited amount of features in each iteration, the performance of random trees is faster than bagging.

**3.6. Face Detection**

Face detection is a computer technology that determines the location and size of a human

face in the digital image. The facial features are detected and any other objects like trees,

buildings and bodies are ignored from the digital image. It can be regarded as a speciﬁc case

of object-class detection, where the task is ﬁnding the location and sizes of all objects in

an image that belongs to a given class. Face detection, can be seen as a more general case

of face localization. In face localization, the task is to identify the locations and sizes of a

known number of faces (usually one). Basically, there are two types of approaches to detect

facial part in the given digital image i.e. feature based and image based approach. Feature

based approach tries to extract features of the image and match it against the knowledge of

the facial features. While image based approach tries to get the best match between training

and testing images.

Face detection is a computer technology that determines the location and size of a human

face in the digital image. The facial features are detected and any other objects like trees,

buildings and bodies are ignored from the digital image. It can be regarded as a speciﬁc case

of object-class detection, where the task is ﬁnding the location and sizes of all objects in

an image that belongs to a given class. Face detection, can be seen as a more general case

of face localization. In face localization, the task is to identify the locations and sizes of a

known number of faces (usually one). Basically, there are two types of approaches to detect

facial part in the given digital image i.e. feature based and image based approach. Feature

based approach tries to extract features of the image and match it against the knowledge of

the facial features. While image based approach tries to get the best match between training

and testing images.

Face detection is a computer technology that determines the location and size of a human

face in the digital image. The facial features are detected and any other objects like trees,

buildings and bodies are ignored from the digital image. It can be regarded as a speciﬁc case

of object-class detection, where the task is ﬁnding the location and sizes of all objects in

an image that belongs to a given class. Face detection, can be seen as a more general case

of face localization. In face localization, the task is to identify the locations and sizes of a

known number of faces (usually one). Basically, there are two types of approaches to detect

facial part in the given digital image i.e. feature based and image based approach. Feature

based approach tries to extract features of the image and match it against the knowledge of

the facial features. While image based approach tries to get the best match between training

and testing images.

Face detection is a computer technology that determines the location and size of a human face in the digital image. The facial features are detected and any other objects like trees, buildings and bodies are ignored from the digital image. It can be regarded as a speciﬁc case of object-class detection, where the task is ﬁnding the location and sizes of all objects in an image that belongs to a given class. Face detection, can be seen as a more general case of face localization. In face localization, the task is to identify the locations and sizes of known number of faces (usually one). Basically, there are two types of approaches to detect facial part in the given digital image i.e. feature based and image-based approach. Feature based approach tries to extract features of the image and match it against the knowledge of the facial features. While image-based approach tries to get the best match between training and testing images. The actual advantages of face-based identification over other biometrics are uniqueness and acceptance. As human face is a dynamic object having high degree of variability in its appearance, that makes face detection a difficult problem in computer vision.

Human Face Detection is an active area of research include several disciplines such as Digital Image Processing, Image Enhancement, Edge detection and Computer Vision. Face detection is the first stage in various applications ranges from person’s identification, video supervision, lips tracing, obtaining face expressions, gender categorization and modern human and machine interactions. For detection of the face, different face areas such as eyes, nose, mouth and chicks are detected as a feature component by using template matching technique and then combined them to identify face portion in the given input image. Face detection falls into a study of Object-Class detection in which the goal is to locate and identify all the objects in the given image. Face-detection technique focus on the frontal human faces. It is similar to image recognition system in which the image of an objector person is matched with reference image (in database) bit by bit. This matching of the image is done with reference images that stores in the image/object database. Any feature data change in the database will fails the matching process.

**3.7. Ensemble of Regression Trees (ERT)**

An ensemble of regression trees can be used to estimate the face’s landmark positions directly from a sparse subset of pixel intensities, achieving super-real-time performance with high quality predictions. It’s to precisely estimate the position of facial landmarks in a computationally efﬁcient way. An ensemble of regression trees is a predictive method composed of a weighted combination of multiple regression trees.

In general, combining multiple regression trees increases predictive performance. It is based on gradient boosting for learning an ensemble of regression trees that optimizes the sum of square error loss and naturally handles missing or partially labelled data. An ensemble of regression trees works with six steps such as step (1) The cascade of regressors, step (2) Learning each regressor in the cascade, step (3) Shape invariant split tests, step (4) Choosing the node splits, step (5) Feature selection and step (6) Handling missing labels.

3.7.1. The cascade of regressors

Let xi ϵ R2 be the x, y-coordinates of the ith facial landmark in an image I. Then the vector S =(x1T , x2T ,..., xpT ) ϵ R2p denotes the coordinates of all the p facial landmarks in I. Each regressor in cascade predicts an update vector from the image and Š(t) that is added to the current shape estimate Š(t) to improve the estimate:

Where; Š(t) = current estimate of S, rt(.,.) = regressors, I = image

The critical point of the cascade is that the regressor makes its predictions based on features, such as pixel intensity values, computed from I and indexed relative to the current shape estimate. This introduces some form of geometric invariance into the process and as the cascade proceeds one can be more certain that a precise semantic location on the face is being indexed. Later we describe how this indexing is performed.

Note that the range of outputs expanded by the ensemble is ensured to lie in a linear subspace of training data if the initial estimate Š(0) belongs to this space.

3.7.2. Learning each regressor in the cascade

Assume the training data (I1,S1),...,(In,Sn) where each In is a face image and Sn its shape vector. To learn the ﬁrst regression function r0 in the cascade the training data triplets of a face image, an initial shape estimate and the target update step, that is, where

for i =1,...,N. We set the total number of the set rip lets to N = nR where R is the number of initializations used per image Ii. Each initial shape estimate for an image is sampled uniformly from without replacement.

The regression function r0, using gradient tree boosting with a sum of square error loss. The set of training triplets is then updated to provide the training data, for the next regressor in the cascade by setting (with t =0 ).

This process is iterated until a cascade of T regressors r0, r1,...,rT-1 are learnt which when combined give a sufﬁcient level of accuracy.

As stated each regressor is learned using the gradient boosting tree algorithm. It should be remembered that a square error loss is used and the residuals computed in the innermost loop correspond to the gradient of this loss function evaluated at each training sample. Included in the statement of the algorithm is a learning rate parameter 0 < v ≤ 1 also known as the shrinkage factor. Setting v < 1 helps combat over-ﬁtting and usually results in regressors which generalize much better than those learnt with v =1.

3.7.3. Shape invariant split tests

At each split node in the regression tree that make a decision based on thresholding the difference between the intensities of two pixels. The pixels used in the test are at positions u and v when deﬁned in the coordinate system of the mean shape. For a face image with an arbitrary shape, index of the points that have the same position relative to its shape as u and v have to the mean shape. To achieve this, the image can be warped to the mean shape based on the current shape estimate before extracting the features. Since only use a very sparse representation of the image, it is much more efﬁcient to warp the location of points as opposed to the whole image. Furthermore, a crude approximation of warping can be done using only a global similarity transform in addition to local translations.

The precise details are as follows. Let ku be the index of the facial landmark in the mean shape that is closest to u and deﬁne its offset from u as

Then for a shape Si deﬁned in image Ii, the position in I­i that is qualitatively similar to u in the mean shape image is given by

Where si and Ri are the scale and rotation matrix of the similarity transform which transforms Si to S̄, the mean shape. The scale and rotation are found to minimize

the sum of squares between the mean shape’s facial landmark points, , and those of the warped shape. vˊ is similarly deﬁned. Formally each split is a decision involving 3 parameters and is applied to each training and test example as

where u0 and v0 are deﬁned using the scale and rotation matrix which best warp to S̄.

In practice the assignments and local translations are determined during the training phase. Calculating the similarity transform, at test time the most computationally expensive part of this process, is only done once at each level of the cascade.

3.7.4. Choosing the node splits

For each regression tree, we approximate the underlying function with a piecewise constant function where a constant vector is ﬁt to each leaf node. To train the regression tree randomly generate a set of candidate splits, that is ’s, at each node. If Q is the set of the indices of the training examples at a node, this corresponds to minimizing.

where is the indices of the examples that are sent to the left node due to the decision induced by , ri is the vector of all the residuals computed for image i,is the average of the targets at the parent node.

3.7.5. Feature selection

The decision at each node is based on thresholding the difference of intensity values at a pair of pixels. This is a rather simple test, but it is much more powerful than single intensity thresholding because of its relative insensitivity to changes in global lighting. Unfortunately, the drawback of using pixel differences is the number of potential split(feature) candidates is quadratic in the number of pixels in the mean image. This makes it difﬁcult to ﬁnd good ’s without searching over a very large number of them. However, this limiting factor can be eased, to some extent, by taking the structure of image data into account.

where is the effective maximum distance between the two pixels in features

3.7.6. Handling missing labels

Some of the landmarks are not labeled in some of the training images, for each training image i and each landmark j. Setting to 0 indicates that the landmark j is not labeled in the i­th image while setting it to1indicates that it is.

where Wi is a diagonal matrix with the vector

**3.8. Eye Aspect Ratio**

For every video frame, the eye landmarks are detected. The eye aspect ratio (EAR) between height and width of the eye is computed.

where ,...,are the 2D landmark locations, depicted in Fig. 3.1.

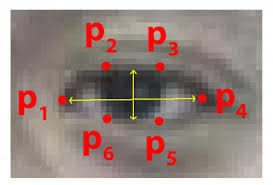


Figure 3.1 Eye’s Landmark Location

Figure

The EAR is mostly constant when an eye is open and is getting close to zero while closing an eye. It is partially person and head pose insensitive. Aspect ratio of the open eye has a small variance among individual sand it is fully invariant to a uniform scaling of the image and in-plane rotation of the face. Since eye blinking is performed by both eyes synchronously, the EAR of both eyes is averaged A similar feature to measure the eye opening, but it was derived from the eye segmentation in a binary image.

**3.9. Summary**

In this chapter, background theories of Driver Drowsiness Detection and Alert System Using Ensemble of Regression Trees such as Ensemble of Regression Tree, Cascade Classifiers, Eye Aspect Ratio and Machine Learning have been discussed.