**OBJECT RECOGNITION FOR AUTONOMOUS DRIVING SYSTEM**

**ABSTRACT:-**

Computer vision is an essential component for autonomous scars. Accurate detection of vehicles, street buildings, pedestrians and road signs could assist self-driving cars the drive as safely as humans. However, object detection has been a challenging task for years since images of objects in the real-world environment are affected by illumination, rotation, scale, and occlusion. In recent years, many Convolutional Neural Network (CNN) based classification-after-localization methods have improved detection results in various conditions. However, the slow recognition speed of these two-stage methods limits their usage in real-time situations. using sliding windows is very costly since a classifier has to be used many times for each image, which makes real-time detections very difficult with this approach. Recently, a unified object detection model, You Only Look Once (YOLO) , was proposed, which could directly regress from input image to object class scores and positions. YOLO has been improved upon it first original introduction in a second version, YOLOv2. YOLOv2 is similar to the original version, but some adjustments in the model makes it both more accurate and faster. However, when applied to auto-driving object detection tasks, this model still has limitations. It processes images individually despite the fact that an object's position changes continuously in the driving scene. In this research, we applied YOLO to two different datasets to test its general applicability. We consider one datasets to perform an experiment, KITTI We fully analyzed its performance from various aspects on KITTI data set which is specialized for autonomous driving. We proposed a novel technique called memory map, which considers inter-frame information, to strengthen YOLO's detection ability in driving scene. We broadened the model's applicability scope by applying it to a new orientation estimation task. KITTI is our main dataset. The main objective is to provide a information of poor quality and environmental model on traffic activity and to signal potentially anomalous situation.

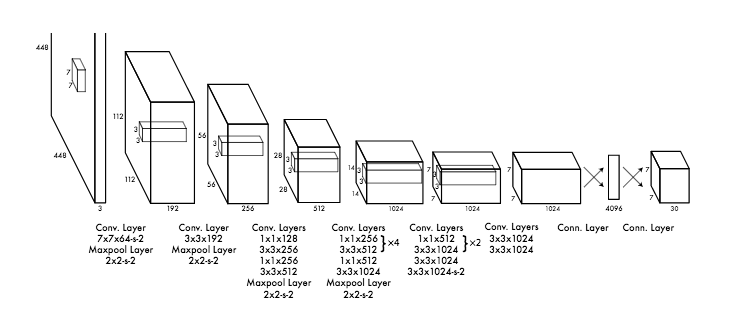
**INTRODUCTION:-**

A safe and robust autonomous driving system relies on accurate perception of the environment. To be more specific, an autonomous vehicle needs to accurately detect cars, pedestrians, cyclists, road signs, and other objects in real time in order to make the right control decisions that ensure safety.

Applications such as autonomous vehicles or surveillance systems would play a big role in our life in the very near future. The decreasing amount of car accidents and the identication of abnormal events are considered an important problem that has to be solved. In order to attain these goals, smart systems need to be developed for monitoring and understanding our surroundings using images, videos, sensors, or depth cameras Moreover, to be economical and widely deployable, this object detector must operate on embedded processors that dissipate far less power than powerful GPUs (Graphics Processing Unit) used for benchmarking in typical computer vision experiments.Object detection is a crucial task for autonomous driving. Different autonomous vehicle solutions may have different combinations of perception sensors, but image based object detection is almost irreplaceable. Recent progress in deep learning shows a promising trend that with more and more data that cover all kinds of long-tail scenarios, we can always design more powerful neural networks with more parameters to digest the data and become more accurate and robust.

While recent research has been primarily focused on improving accuracy, for actual deployment in an autonomous vehicle, there are other issues of image object detection that are equally critical. For autonomous driving some basic requirements for image object detectors include the following: a) Accuracy. More specifically, the detector ideally should achieve 100% recall with high precision on objects of interest. b) Speed. The detector should have real-time or faster inference speed to reduce the latency of the vehicle control loop. c) Small model size .

Often there are multiple targets present in the same image for the object detection system to detect. YOLO (You Only Look Once) is a fast and accurate object detection system consisting of one single convolutional network. This means YOLO proposes bounding boxes and their corresponding class scores simultaneously. In this research first we analyse whether the YOLO and Yolo version is an appropriate model for general object detection, for that we trained the model on data sets such as KITTI . We perform a comprehensive analysis of YOLO over Road - object detection and orientation estimation benchmark. According to the comprehensive analysis results the YOLO have a limitations in on realtime autonomous driving system. To implement this process for realtime autonomous driving system we design a novel YOLO method by combining the memory map and on-max suppression (NMS)  modelto make YOLO suitable for real-time object detection in video streaming and improve the YOLO potentiality by applying YOLO on orientation estimation, the YOLO can estimate the orientations of the detected objects.



The KITTI Vision Benchmark Suite is specialized for autonomous driving.. KITTI also extract benchmarks for 2D/3D object detection, object tracking and pose estimation. In this thesis, we used the object detection and orientation estimation benchmark. This benchmark includes 7481 labelled images from 80 classes. Single image size is 800kb to 900kb and total size of 7481 images is 6.2GB. Among these classes, only the frequent objects (car, truck, pedestrian, cyclist, tram, and sitting people) are labelled independently, all the other classes are labelled as 'M isc' or 'DontCare' class. The 7481 images were divided into 70% (5237 images) for training and 30% (2244 images) for testing.

. In this research work we consider self-driving application, where we need to improve object detections which could assist self-driving cars to drive as safely as humans. In this application we extend the YOLO model with memory map component to the end of neural network. When we validate the effect of gridding size, the size of last layer will be changed to multiple sizes. In orientation estimation experiment, we will revise the cost function, so that the network could learn to distinguish objects' orientations. Late we will implement a memory map for increasing detection accuracy in video for different gridding size's effect by applying YOLO on orientation estimation task.

**Related Work**

The initial vehicle detection stage is based on the adaptive smoothness method to build a background model. This model is updated continuously, at every time instant, based on those pixels that were not detected as moving regions. This approach assumes that the camera is static, whereas in outdoor scenarios, cameras are frequently subject to small motion disturbances (e.g., due to wind). In this case, more sophisticated background modeling techniques can be used, e.g., employing multiple background models instead of only one. Lane detection and classification is an important first step in building a semantic scene description of the highway. This can lead to generation of statistical data on traffic activity, such as vehicle density, lane changes, and detection of anomalous situations (e.g., accidents, congestion, and dangerous driving).

In recent years many scholars both at home and abroad have carried on researches about vehicle detection and classification based on magnetic sensor. Currently, research on vehicle classification using magnetic sensor has been developing a little slowly because of low classification accuracy and immature application on vehicle identification. Sing Y. C. et al proposed a vehicle classification method simply based on magnetic sensors to classify vehicle into several types [4]. Based on the previous work, Saowaluck K. et al[54] built up an automatic vehicle classification system using magnetic sensors. In [6], a vehicle detection and classification approach by the magnetic signal measured by a MEMS magnetic sensor. However, the accuracies of classification in these researches are not very high which are mostly between 70%-90%. What’s more, most methods lack reasonable feature extraction and selection process, which are of crucial importance to efficiency and result of classification.

The problem of vehicle counting is most commonly solved by deploying inductive loops. Those loops provide high precision, but are very intrusive to the road pavement and therefore come with a high maintenance cost. Most video analytics systems on highways focus on counting and possibly classification to allow for more detailed statistics. Some systems have also been adapted for urban environments, with cameras mounted on high poles. This provides a higher viewing angle, which limits the occlusion between densely spaced vehicles, which results in similar conditions to highways. However, those highly mounted cameras are specifically for video analytics, because standard CCTV cameras for human operators are mounted lower.

A typical video analytics application uses a pipeline of foreground estimation classification and tracking. A statistical model typically estimates foreground pixels, which are then grouped with a basic model (e.g. connected regions) and propagated through the system until the classification stage Classification then uses prior information (previously learned or pre-programmed) about the object classes to assign a class label. For the remainder of the review, this class of algorithms will be referred to as 'top-down' or 'object-based', because pixels are grouped into objects early during the processing.

In contrast, a 'bottom-up' approach is defined as one which detects and classifies parts of an object first. This initial classification of the parts uses learned prior information about the final object classes, e.g. an image area is classified to be a car wheel or a pedestrian head based on previously learned appearances of wheels and heads. The combination of those parts into valid objects and trajectories is the final step of the algorithm. This type of approach is typically used in generic object recognition.

Foreground estimation and segmentation is the first stage of many visual surveillance systems. The foreground regions are marked for processing in the subsequent steps. The foreground is defined as every object, which is not fixed furniture of a scene where fixed could normally mean months or years. This definition conforms to human understanding, but it is difficult to implement algorithmically. There are two main different approaches to estimate the foreground, which both use strong assumptions to comply with the above definition. Firstly, a background model of some kind can be used to accumulate information about the scene background of a video sequence. The model is then compared to the current frame to identify differences (or ‗motion‘), provided that the camera is stationary. This concept lends itself well for computer implementation, but leads to problems with slow moving traffic. Any car should be considered foreground, but stationary objects are missed due to the lack of motion.

This group of background models estimates a background image which is subtracted from the current video frame. A threshold is applied to the resulting difference image to give the foreground mask. The threshold can be constant or dynamic as used in [7]. The methods described below differ in the way the background picture is obtained.

To improve robustness, a single Gaussian model can be used for the background. Instead of only the mean value as for averaging, the variance of the background pixels is calculated additionally. This results in a mean image and variance image for the background model. A new pixel is classified depending on the position in the Gaussian distribution, which is the statistical equivalent to a dynamic threshold. [8]use a single Gaussian background model.

A Kalman filter can be used to estimate the background image, where the colour of each pixel is modelled by one filter. The foreground can be interpreted as noise for the filter state. However, illumination changes are non Gaussian noise and violate basic assumptions for the use of Kalman filters. [9] proposes a Kalman filter approach which can deal with illumination changes. The illumination distribution over the image is estimated and used to adjust the individual Kalman filter states. The foreground estimation was tested in [9] indicating superior performance compared to the Kalman filter based algorithm proposed

Decision tree is very useful classification and regression technique. Decision Trees are extremely adaptable, straightforward, and simple to investigate. They will work with characterization issues and relapse issues. So on the off chance that you are attempting to anticipate a straight out worth like (red, green, up, down) or in the event that you are attempting to foresee a persistent quality trees, decision trees will handle both issues. a decent aspect regarding Decision Trees is they just need a table of information and they will assemble a classifier specifically from that information without requiring any in advance configuration work to occur

Random Decision Tree(RDT) algorithm is more exact than other choice tree calculation. One vital part of RDTs is that the structure of an arbitrary tree is built totally autonomous of the preparation data[1]. The RDT calculation can be broken into two stages, preparing and arrangement. The preparation stage comprises of building the trees) and populating the hubs with preparing example information

Intelligent transportation system has a wide range of traffic data sources, including the dynamic traffic flow data and intelligent transportation subsystem management control data, as well as static road environment data. Intelligent transportation system management and control of the object is the traffic flow, traffic flow data is sampled by the time sequence of a series of numerical data sequence, is the most important data in the transport system[4]. Intelligent transportation system (ITS) recorded a lot of traffic information, such as electronic police system the vehicle traffic violations of images and data are recorded, providing information for traffic violations, including vehicle illegal sites, illegal date and time of illegal, illegal, illegal parameter, illegal vehicle panoramic image sequence, illegal vehicle license plate image; traffic accident alarm system provides the alarm time and alarm location, alarm phone number and related traffic accident information; traffic signal control system with intersection of running state, color step progressive information[7].

Short Text Classification[2], The intrinsic idea of this part is that we observed that a word may have different meanings in different domains. So the unigram Naïve Bayes classifier method is used together with the pre-labeled training data to build the multi-classifier. We use distinct categories of training data. However, since for unigram features, there are usually many different features, and as such it is helpful if we discard some useless features. In order to solving this problem, we try two different feature selection algorithms. The first is Mutual Information The idea of mutual information is that, for each class *C* and each feature *F*, there is a score to measure how much *F* can contribute to making a correct decision on class *C.*

The Naive Bayes classifier is a straightforward probabilistic classifier which depends on Bayes hypothesis with solid and innocent self-government suspicions. It is a standout amongst the most essential content arrangement system with different applications in email spam presentation, privatemail sorting, record classification, , dialect disclosure and estimation revelation. Innocent Bayes executes well in numerous troublesome certifiable inconveniences. Despite the fact that it is much of the time beat by different strategies, for example, helped trees, Max Entropy, Support Vector Machines and so on, Naive Bayes classifier is amazingly productive since it is less computationally and it requires a little measure of readiness data. One all around enjoyed approach to execute multi-mark classifier is to change over the multi-name association issue into numerous single-name classification issues [7]

Binary classification presents two different kind of classes for representing binary classification and multiclass classification. Both binary classification and multiclass classification are single-label classification systems. Singlelabel classification means each data point can only fall into one class where all classes are mutually exclusive[3]. Most existing studies found on traffic flow arrangement are either paired order on important and unimportant substance, or multi-class grouping on non specific classes, for example, news, occasions, sentiments, arrangements, and private messages. Notion investigation is another exceptionally prevalent three-class arrangement on positive, negative, or nonpartisan feelings/suppositions. Slant examination is extremely helpful for mining client assessments on items or organizations through their surveys or online posts. It finds wide selection in showcasing and client relationship administration. Numerous systems have been created to mine feeling from texts.However, does not give much noteworthy learning on important mediations and administrations for understudies.

Image mining is a technique which handles the mining of information, image data association, or additional patterns not unambiguously stored in the images. It utilizes methods from computer vision, image processing, image retrieval, data mining, machine learning, database, and artificial intelligence [18]. Rule mining has been implemented to huge image databases in 1999[19]. There are two most significant techniques. The first technique is to mine from huge amount of images alone and the second technique is to mine from the integrated collections of images and related alphanumeric data. Rule mining technique is exploited to determine relations between structures and functions of human brain. An image mining algorithm using blob required to be carry out the mining of relations within the context of images is provided by Zaiane& Han [20]. The main intention of image mining is to produce all considerable patterns without any information of the image content, the patterns types are different. They could be classification patterns, description patterns, correlation patterns, temporal patterns and spatial patterns. Image mining handles with all features of huge image databases which comprises of indexing methods, image storages, and image retrieval, all regarding in an image mining system[21]. The establishment of an image mining system is frequently an intricate process because it implies joining diverse techniques ranging from image retrieval and indexing schemes up to data mining and pattern recognition. Further, it is anticipated that a good quality image mining system provides users with a useful access into the image storage area at the same time it recognizes data patterns and generates knowledge beneath image representation. Such system basically be supposed to bring together the following functions: image storage, image processing, feature extraction, image indexing and retrieval and, pattern and knowledge discovery.

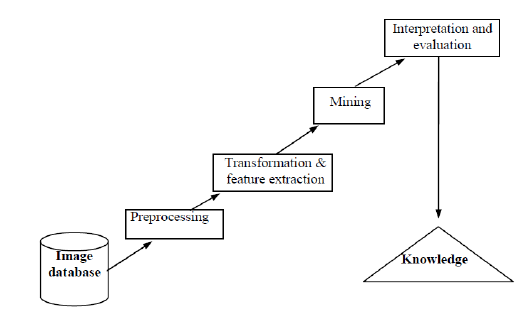


Fig 1- Image Mining process [16]

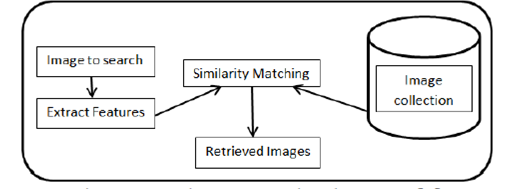


Fig -2 Image Retrival process [17]

1. **Data preprocess**

Image mining deals with large collection of image datasets that are high-dimensional and have multiple features, so time and space cost are relative high when analysing them. There exist a lot of dirty and noisy data in large image databases, for instance, images that are extremely unclear and images that are already breached. Those data often cause chaos in mining process and give birth to bad mining results, so it is necessary to pre-process data, clean up the noisy, broken, dirty data. In order to improve quality and efficiency of the following mining steps, it is vital to discover suitable pre-processing technologies to clean up the un-related data and make useful hidden information more obvious. Traditional image processing technologies are applied to the image data ready to be mined. Some image pre-process has been introduced, for instance, Han propose a palm-print-based identification system in [22], the pre-processing steps including image thresholding, border tracing and wavelet-based segmentation, the pre-processing method is proved to be effective and can be simulated in other scenarios as well.

1. **Extracting multi-dimensional feature vectors**

One of the key problems is how to express image data, which can usually be represented by features such as texture, color, edge, shape. According to the mining object, extract the basic elements that can present the images, omit features in essential to mining result. In some cases, to get better mining result, it is necessary to converge many features to form multidimensional feature vectors. Color, edge, texture are very important features in image mining and are widely used. [23] Presents a feature extraction method that uses a combination of features: color, edge, texture, the method achieves high recall and precision. Using image processing technologies such as image segmentation, picking up the edge to extract task related feature vectors, form multi-dimensional feature vectors.

1. **Mining on vectors and acquire high-level knowledge**

Most commonly used image mining technologies are image classification, image clustering, mining association rules and neural network. Various methods such as object recognition, image indexing and retrieval, image classification and clustering, neural network are used on feature vectors for mining and acquiring hidden and valuable high-level knowledge, then evaluate and explain that knowledge.

1) Image classification - Image classification is to do quantity analysis on image, and, it is a supervised learning method, a set of pre-labeled images are provided, then based on prior knowledge, tag the new images with suitable labels, often there are three processes involved in image classification:(1): Feature extraction, first build up a image representation model, extract features from sample images that are already labeled and establish feature description for each image; (2): Train the samples of each class and establish model description for each class; (3): Use the model to classify and index images that are not labeled. The most commonly used image classification technologies are as follows: Bayes, neural network, decision tree, support vector machine, K-nearest–neighbor-classifier, genetic algorithm etc. Performance of a classifier is normally measured by accuracy of prediction, speed, robustness, extensibility.

**TASK DESCRIPTION:-**

The object detection system You Only Look Once (YOLO) on the other hand, proved it is possible to unify the region proposal and class score prediction in one single convolutional neural network. YOLO has been improved upon it first original introduction in a second version, YOLOv2. YOLOv2 is similar to the original version, but some adjustments in the model makes it both more accurate and faster.

The KITTI Vision Benchmark Suite is specialized for autonomous driving. . KITTI also extract benchmarks for 2D/3D object detection, object tracking and pose estimation. In this thesis, we used the object detection and orientation estimation benchmark. This benchmark includes 7481 labelled images from 80 classes. Single image size is 800kb to 900kb and total size of 7481 images is 6.2GB. Among these classes, only the frequent objects (car, truck, pedestrian, cyclist, tram, and sitting people) are labelled independently, all the other classes are labelled as 'Misc' or 'don’t Care' class. The 7481 images were divided into 70% (5237 images) for training and 30% (2244 images) for testing.

Here we first, we analyse whether the YOLO and Yolo version is an appropriate model for general object detection, for that we trained the model on dataset and KITTI . We perform a comprehensive analysis of YOLO over Road - object detection and orientation estimation benchmark. According to the comprehensive analysis results the YOLO have a limitations in on real time autonomous driving system. To implement this process for real time autonomous driving system we design a novel YOLO method by combining the memory map and on-max suppression (NMS)  model to make YOLO suitable for real-time object detection in video streaming and improve the YOLO potentiality by applying YOLO on orientation estimation, the YOLO can estimate the orientations of the detected objects .

In this research work, we need a Image vision simulators such as OPENCV, MATLAB and Python. In this research we are using large dataset models to analyse the proposed model performance by varying different datasets. In order to perform our experiment we considered MATLAB simulator. To simulate the real time video streaming data we need minimum 4 GB RAM and Min 120GB hard disk.

**MAJOR CHALLENGES AND SOLUTIONS**

One of the major open challenges in self-driving cars is the ability to detect objects and pedestrians to safely navigate in the world.

For a safety critical application such as autonomous driving, the error rates of the current state of-the-art are still too high to enable safe operation.

The main objective is to provide a information of poor quality and environmental model on traffic activity and to signal potentially anomalous situations, e.g., accident detection or dangerous driving

In this We perform a comprehensive analysis of YOLO over Road - object detection and orientation estimation benchmark. According to the comprehensive analysis results the YOLO have a limitations in on realtime autonomous driving system. In this we use an important dataset called KITTI. which is specialized for autonomous driving. By driving the car equipped with multiple sensors around in mid-size city, rural areas and on highways, they collected rich data, including images and optical flow from camera, points cloud from laser scanner and odometry information from a GPS.

**Flow Diagram**

**Flowchart**

Load ‘N’ training image into database database

.

I=0

Load test images

I=I+1

Is I<=N+1

NO

YES

NO

Apply image preprocessing

Extract features

Is HD<0.3

BPN

Find Hamming distance

Compute feature vectors

YES

Display “ vecihle found”

**EXPERIMENTS:-**

In this chapter we discuss about image preprocessing and feature extraction process to perform following functions thresholding, image segmentation and image coding. Based on these functions the back propagation algorithm is employed to determine the results

4.1 Thresholding

The main idea of thresholding is to organize image segmentation, the thresholding function extraction the different image characters from image background based on the threshold value, and assign 0 to the pixels for smaller values and 1’s for higher values of threshold value (i.e if pixl value grater than Threshold T value the value assignment become as a 1 or else it will become as 0). This process iterate to different back ground color formats, and here we obtain this thresholding process for dark characters on a light background. The following function derives to the image as follows

Thresholding is the simplest method of image segmentation. From a grayscale image, thresholding can be used to create binary images. The simplest thresholding methods replace each pixel in an image with a black pixel if the image intensity I_{i,j} is less than some fixed constant T (that is, I_{i,j}<T), or a white pixel if the image intensity is greater than that constant.

**4.2 Image segmentation**

**To recognize the particular character of a image and image characteristic’s, the image segmentation process divides the image into the different segments based on thresholding process,** The thresholded image has only the featureset which are identified using back propogation neural network model. To recognize the particular vehicle and vehicle type, it is important to segment all features in a extracted image.

**Algorithm**

1. Compute histogram and probabilities of each intensity level.
2. Set up initial \omega_i(0) and\mu_i(0).
3. Step through all possible thresholds t = 1 \ldots  maximum intensity.

* Update \omega_i and \mu_i
* Compute \sigma^2_b(t)

1. Desired threshold corresponds to the maximum\sigma^2_b(t).
2. You can compute two maxima (and two corresponding thresholds). \sigma^2_{b1}(t) Is the greater max and \sigma^2_{b2}(t) is the greater or equal maximum.
3. Desired threshold = \frac{\text{threshold}_1 + \text{threshold}_2 }{2}

**Image coding**

**Image coding process converts the image into the binaries, in this process the extracted image convert the image pixels in to the binaries by using** Fourier Descriptors feature extraction process. Each character of image is coded 11x19 pixels matrix.

These inut bits are given as input vector set of BPN approach, these binary input code is applied as a input of BPN input

****

**Fig: Binary coded format**

**Edge Detection**

Edges occur in images due to sharp discontinuities which bring about pixel intensities which either occlusions or separation between different regions. Edge detection involves identifying the pixels that fall along the edges.

Edge detection is gradient based and considering a pixel at position 𝑥,𝑦 where x represents horizontal pixel position from left to right and y represents the vertical position from top to bottom. Pixel intensity is signified by 𝑓(𝑥,𝑦).

We can then define the local extreme gradient as:

We can then apply a threshold to the magnitude as:

Edge is defined by: ǀ∇𝑓ǀ ≥ 𝑇

No edge is defined by: ǀ∇𝑓ǀ < 𝑇

If T is too large, we‟ll have missed detections and if it‟s too small, we‟ll have effect of producing false alarms.

**Euclidean distance**

It is also called the 𝐿2 distance. With the same points described above, we can calculate the Euclidean Distance between 𝑢 and 𝑣 as:

𝐸𝑈 (𝑢,𝑣) = ( (𝑥1 − 𝑥2)2 + (𝑦1 − 𝑦2)2)(1/2)

Thus for n-dimensions:

Where, j=i+1

**Conventional neural network**

**In this network we organize this process by following sets**

* TrainingSet  
  A collection of input-output patterns that are used to train the network
* TestingSet  
  A collection of input-output patterns that are used to assess network performance
* LearningRate-*η*  
  A scalar parameter, analogous to step size in numerical integration, used to set the rate of adjustments

**Feedforward network**

* **Image binary Inputs** **xi** attain through pre-synaptic connections
* Synaptic efficacy is shown using real **weights wi**
* The result of the neuron is a **nonlinear function** ***f*** of its inputs **wi**

Apply each binary input to each neuron as input node, then computes the node idenfication feedforward network model

**Calculate Outputs For Each Neuron Based On The Pattern**

* The output from neuron j for pattern p is Opj where\



and

k ranges over the input indices and Wjk is the weight on the connection from input k to neuron j



* The output neuron error signal *dpj* is given by *dpj=(Tpj-Opj) Opj (1-Opj)*
* *Tpj* is the target value of output neuron j for pattern p
* *Opj* is the actual output value of output neuron j for pattern p
* The hidden neuron error signal *dpj* is given by

where *dpk* is the error signal of a post-synaptic neuron k and *Wkj* is the weight of the connection from hidden neuron j to the post-synaptic neuron k

* Compute weight adjustments *DWji* at time t by  
    
  *DWji(t)= η dpj Opi*
* Apply weight adjustments according to  
    
  *Wji(t+1) = Wji(t) + DWji(t)*
* *Some add a momentum term* ***a****\*DWji(t-1)*

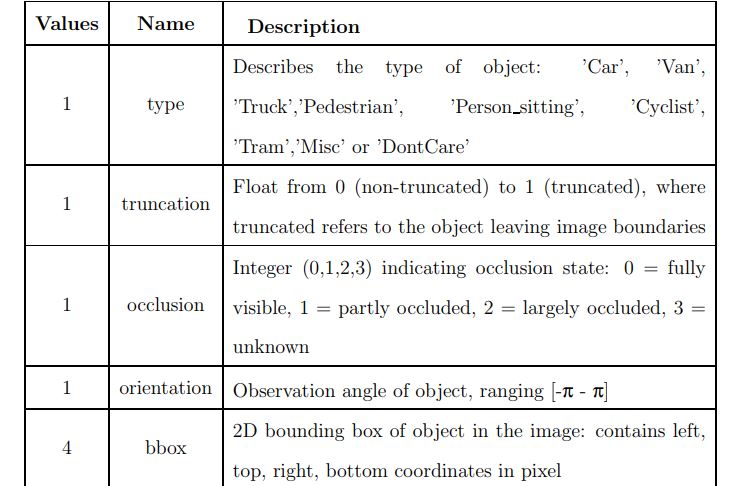
**DATASET:-**

A **data set** is a collection of related, discrete items of related **data** that may be accessed individually or in combination or managed as a whole entity. A **data set** is organized into some type of **data** structure

Here we are using one data set called as KITTI.

**KITTI :-**

The KITTI Vision Benchmark Suite is specialized for autonomous driving. By driving the car equipped with multiple sensors around in mid-size city, rural areas and on highways, they collected rich data, including images and optical flow from camera, points cloud from laser scanner and odometry information from a GPS. KITTI also 31 extract benchmarks for 2D/3D object detection, object tracking and pose estimation. In this thesis, we used the object detection and orientation estimation benchmark. This benchmark includes 7481 labeled images from 80 classes. Single image size is 800kb to 900kb and total size of 7481 images is 6.2GB. Among these classes, only the frequent objects (car, truck, pedestrian, cyclist, tram, and sitting people) are labeled independently, all the other classes are labeled as 'M isc' or 'DontCare' class. The 7481 images were divided into 70% (5237 images) for training and 30% (2244 images) for testing. Below figure shows the samples from KITTI datasets.

**  
**

The label files contain the following parameters, class type, truncation, occlusion, 32 orientation and bounding box

We trained YOLO dataset on KITTI dataset.

In this section we will give the training configuration, results and discussions and analysis.

**TRAINING CONFIGURATION:-**

KITTI dataset includes 5237 images (70%) for training and 2244 images (30%) for testing.

Our model has achieved 85% overall precision and 62% overall recall with detection speed of 0.03s per images. In this section, we will show samples of detection results of KITTI testing images and wild test results on our captured images.

**Contingency table:**

* True positive = correctly identified
* False positive = incorrectly identified
* True negative = correctly rejected
* False negative = incorrectly rejected

Sensitivity and specificity are statistical measures of the performance of a binary classification test, also known in statistics as classification function:

Sensitivity (also called the true positive rate, or the recall in some fields) measures the proportion of positives that are correctly identified

1. SEN = TP / P
2. SEN = TP / (TP + FN)

Specificity (also called the true negative rate) measures the proportion of negatives that are correctly identified

1. SPE = TN / N
2. SPE = TN / (TN + FP)

Accuracy has two definitions:

More commonly, it is a description of systematic errors, a measure of statistical bias;

Alternatively, the ISO defines accuracy as describing both types of observational error above (preferring the term trueness for the common definition of accuracy).

1. Accuracy = (TP + TN) / (TP + TN + FP + FN)

**Positive predictive value:**

The positive and negative predictive values (PPV and NPV respectively) are the proportions of positive and negative results in statistics and diagnostic tests that are true positive and true negative results. The PPV and NPV describe the performance of a diagnostic test or other statistical measure. A high result can be interpreted as indicating the accuracy of such a statistic. The PPV and NPV are not intrinsic to the test; they depend also on the prevalence. The PPV can be derived using BPN approach.

Although sometimes used synonymously, a positive predictive value generally refers to what is established by control groups, while a post-test probability refers to a probability for an individual. Still, if the individual's pre-test probability of the target condition is the same as the prevalence in the control group used to establish the positive predictive value, the two are numerically equal.

1. PPV = TP / (TP + FP)

**F Score:**

In statistical analysis of binary classification, the F1 score (also F-score or F-measure) is a measure of a test's accuracy. It considers both the precision p and the recall r of the test to compute the score: p is the number of correct positive results divided by the number of all positive results, and r is the number of correct positive results divided by the number of positive results that should have been returned. The F1 score can be interpreted as a weighted average of the precision and recall, where an F1 score reaches its best value at 1 and worst at 0.

The traditional F-measure or balanced F-score (F1 score) is the harmonic mean of precision and recall:

1. F = (2 \* TP) / (2 \* TP + FP + FN)

**Results**

**MAJOR RESULTS:-**

By performing a comprehensive analysis of YOLO over KITTI dataset, we found that YOLO can achieve 85% precision with 62% recall at 30 frames per second. The results are encouraging and suggest that YOLO is an excellent model for detecting objects required for autonomous driving systems. However, YOLO processes the images individually despite the fact that there is continuous information in video stream in real-time driving situation. YOLO needs further modifications to better fit real-time driving system. To fill this gap, the most important contribution of this research lies in proposing a novel technique called memory map to make YOLO suitable for real-time video streaming. The memory map mechanism, which accumulates class-confidence throughout temporal frames, 73 helps increase recall while reduce precision. By comparing with state-of-the-art methods, our revised model with memory map got a little lower detection precision. However, our methods won the first place in detection speed. Our detection speed is 0.03 seconds per image, which is 10 times faster than the best methods. Our modified model is the only one that achieve rea ltime, 30 frames per second, which can be used in driving situation. In the orientation estimation section, we found that YOLO works well in predicting an object's orientation. In an auto-driving system, predicting an object's direction is crucial for the system to be able to make correct decisions.

**CONCLUSION:-**

In an auto-driving system, predicting an object's direction is crucial for the system to be able to make correct decisions. Through our experiments, we proved that YOLO has great By performing a comprehensive analysis of YOLO over KITTI dataset, we found that YOLO can achieve 85% precision with 62% recall at 30 frames per second. The results are encouraging and suggest that YOLO is an excellent model for detecting objects required for autonomous driving systems performance in image level object detection and orientation estimation. Our novel memory map improvement makes YOLO much suitable in driving situation, that makes it the best choice for autonomous driving system.

**FUTURE WORK:-**

In YOLO network, the input images are divided into unique-sized grids to predict objects with various sizes. Thus, the unique-sized grid may affect the bounding box accuracy of smaller or larger objects. In the future, we will try to divide the input images into multiple sizes and select the best results from overlapping grids. From our experiments, we observed an increase in recall as an impact of using memory map. In the future, we can add memory map to training process to learn the optimum weights. KITTI dataset has multiple ground truth data, 2D and 3D. In our experiments, Currently, our system can only run on a single GPU. In the future, we can revise the system to make it run on multiple GPUs to get higher detection speed. That will enable the network to work better in high speed driving situations.