Machine Learning in Automotive Software

Bharath Bangalore Somashekar
Matriculation Number: 3375923
Email: st164288@stud.uni-stuttgart.de

***Abstract*—The software in automotive is ever evolving, the inclusion of intelligence to automotive software is revolutionizing the transport sector. All the sensors and ECUs in the automobile produce massive amount of data and this fosters the advent of machine learning and deep learning approaches in automobile. Machine learning and deep learning have many applications in automobiles such as in Advanced driver assistance, automatic cruise control, driver activity monitoring, autonomous driving, etc. Autonomous driving has become the buzzword in automobile and this is catered by the rise in machine learning and deep learning methodologies. Deep convolutional network is extensively adopted in the field where image processing and computer vision is involved and this play a major role in autonomous driving. This paper provides an introduction to machine learning and deep learning approaches and describes an approach of autonomous driving using deep convolutional neural network, discuss on the verification of the system with changes to the traditional V-model, touch up on an approach to enhance the comfort of the driver, examine the efficient validation of the system considering rare failure rate and also the challenges faced in autonomous driving and its non conformity with the automotive functional safety standard ISO 26262**

I. Introduction

Machine Learning and Deep learning have become the rapidly growing technologies of the decade, with the advent of this, tasks which seemed impossible a few years back are now done with ease. Deep learning approaches have become a major part in autonomous driving. CNN is one such approach that has revolutionized pattern recognition [1]. Before the rise of CNNs, most of the pattern recognition tasks were done by hand-crafted feature extraction and classification. In CNNs, features are learned automatically from training examples and relatively fewer parameters need to be learnt, this methodology is now widely used for image processing and computer vision. Owing to all these benefits, machine learning and deep learning have extensively been applied to solve real world problems in fields such as medicine, biology, industry, manufacturing, security, education, and games

Automotive software has experienced tremendous innovations in recent years [2]. Various technologies such as airbags, anti lock braking system, autonomous parking, lane following, platooning and many more have made it easy and safe for consumer to drive a car. Then came the notion to integrate machine Learning in the automotive industry. Machine learning has many potential applications in the automotive domain. For example, machine learning is used during the development, manufacturing and sales of the automobile as well as in technologies like advanced driving assistance system (ADAS) and autonomous driving. When it comes to autonomous driving, machine Learning based processes, especially, deep learning is extensively used to solve increasingly complex tasks in autonomous driving [3].

In this paper, I briefly discuss the foundations of machine learning and deep learning and introduce self driving car in section 5. In section 6, we see a of methodology to enhance the automotive driving experience by the effective usage of convolutional neural network to reduce the steps involved in training the autonomous car and also discuss on the usage of a modified V-model for deep learning based approaches. In the section 7, we see how to ensure the comfort of the driver by mimicking the driving style of the driver using deep learning. In section 8, we see an approach to validate the deep learning based system statistically by finding the rare error cases with comparatively lesser data requirement.

Injecting intelligence in the driving related tasks requires expensive modifications to the overall hardware and skill set in the automotive industry and also usage of deep learning doesn't adhere to the ISO 26262 [4], a safety standard for automobiles proposed by the International Organization for Standardization, section 9 focuses on this and discuss some of the challenges faced by deep learning in automobiles

II. Machine Learning

Machine learning is the study of statistical models that can perform a task or action without explicitly mentioning the way a task has to be done. The machine learning system will decide the output based on the inferences and patterns found in the given input

In simple terms, in traditional programming, the system will receive the input, process the input with the already known programming logic and output the data In machine learning, specifically supervised learning, the input to the system will be the given input and the expected output, the system runs the machine learning algorithm on the input data and output the programming logic

A. Types of Machine Learning

As shown in fig 1, there are three main types in machine learning.

1) Supervised Learning: In this type, along with the given input the system requires the expected output to be given as the input to the system. The system will progress the input data and find the connection between the given input and the expected output.

As an example, a spam filter is trained with many emails along with the labels as spam or not spam, the system learns to classify new emails as spam or not spam

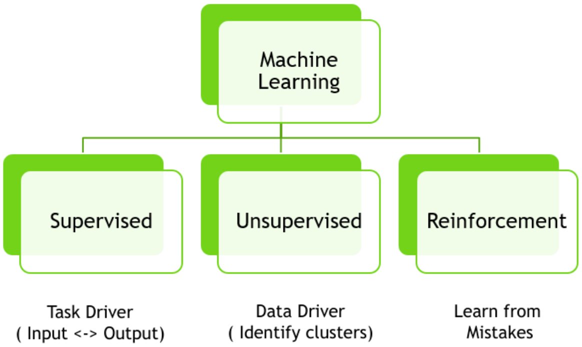

Fig. 1: Types of Machine Learning

2) Unsupervised Learning: In this type, only the given input becomes the overall input to the system. The system will classify the output based on the similarity and pattern among the input data

3) Reinforcement Learning: This is a type of learning when software agents observe and perform action in the environment to get a reward. It learns the best strategy called the policy over time to get the most reward. The policy defines what action the agent should at a particular situation [5]

For this paper, the main focus will be on supervised learning. The tasks performed by supervised learning can be subdivided into linear regression and classification. Linear regression is to find a linear or nonlinear connection between the input and the output. Classification is to assert the given set of inputs belong to a particular class of output

B. Training

To train a machine learning model is to find a statistical connection between the input and the desired output. This usually involves finding the unknown parameters of the equation which is the minimal of the cost function. The cost function is usually the squared difference between the actual output and the desired output and is scaled by some factors

C. Algorithms in Supervised Learning

There are various algorithms in machine learning, the main focus is on the classification part of the supervised learning. As shown in fig 2, there are many classifications algorithms for supervised learning - Random forest, Bayesian, Gaussian Mixture Model, Support vector machines [5], Neural Networks. Among them, Neural Network is an intriguing algorithm as it is a global machine learning algorithm which can be used on any dataset

D. Neural Networks

Analogous to biological neural present in the brain, a simple neuron is neural network is an element which takes an input, scales the input, add it to a bias value and pass this summation to a nonlinear function to produce the output.

This single neuron can be augmented with multiple inputs, each scaled by difference scaling values and added along with the bias and transformed non-linearly to produce the output.

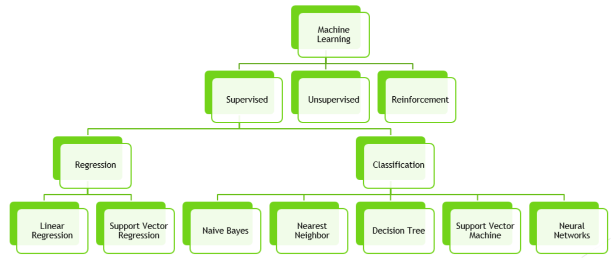

Fig. 2: Algorithms of Machine Learning

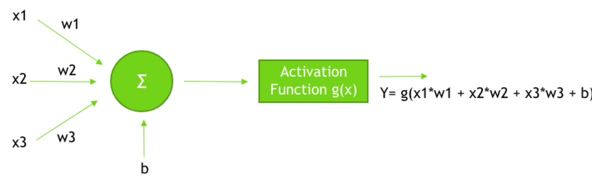

Fig. 3: Single Neuron

By stacking up multiple neurons in a layer, the network can perform harder classification task. This is the network structure for the neural network, the layer with the neurons is called the hidden layer, the rest of the layers at the input and the output are called input layer and output layer respectively.

Neural network is a global algorithm which can perform decent tasks with just one hidden layer, this intrigued the possibility of solving high complexity task with multiple hidden layers and hence gave rise to Deep Learning

III. Deep Learning

Deep Learning is an extension of neural network by adding multiple hidden layers to make the overall network deep.

The difference between machine learning and deep learning is shown in fig 4, machine learning algorithms need an external feature extractor whereas these features are extracted implicitly by deep learning algorithms.

A. Training in General

Training the network is to find all the scaling factors and bias parameters which can minimize the error or loss of the network. Backpropogation and gradient descent algorithms are used in training a deep neural network

B. Main networks in Deep Learning

1) Fully connected neural network: The figure 5 shows the fully connected neural network, in this network, each unit in one layer is connected to each unit in the next layer.

2) Recurrent Neural Network: This is a deep learning network with spatial feed-forward and temporal feed-backward network, it is used for processing sequential data. Because of the temporal feature, this is used in speech recognition and translation

3) Convolutional neural network: CNNs are a neural network that is used in processing of data that is of grid-like topology. they use the convolution operation instead of general matrix multiplication in at least one of their layers [6].

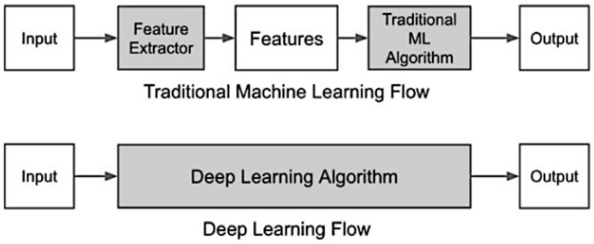

Fig. 4: Difference between Machine Learning and Deep Learning [6]

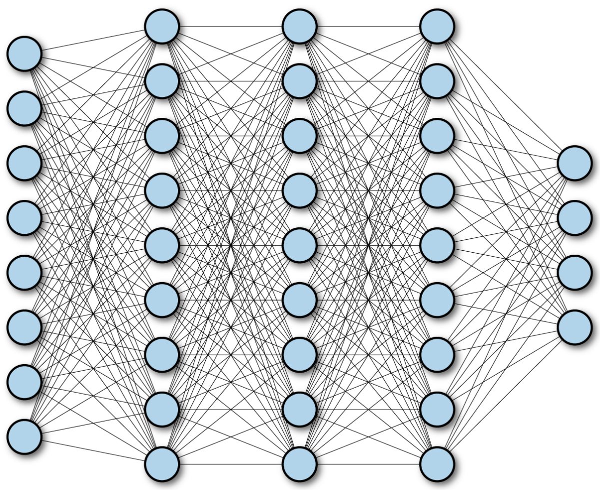

Fig. 5: Fully Connected Neural Network [7]

C. Convolutional neural network

As shown in figure 6, the whole system usually consists of CNN layers along with pooling, fully connected layers and softmax layer

CNN is a widely used deep learning network used in image processing and computer vision. It is used for object detection, lane detection, driver activity detection and more

1) Convolution Layer: In this layer input image is convoluted with filters to find features such as edges, boundaries and more. The values in the filters are unknown and are determined while training the network.

2) Max Pooling Layer: These layers usually follows the convolutional layers and is used to shrink the output from the convolutional layers.

3) Fully Connected Layer: These layers are at the end of the network, where the output from Convolutional layers and max pooling layers are small so that they can be flattened into a single array and fed to this network.

4) SoftMax Layer: This layer is the final layer and it takes the output of a Fully connected layer as input and output the class to which the input image belongs to.

IV. Applications

Machine learning and deep learning are widely used in online advertisement to advertise products based on browsing industry. In marketing and sales ML and DL are used to

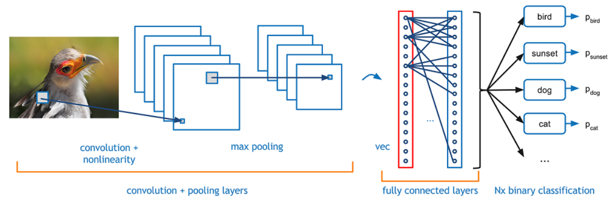

Fig. 6: Convolutional Neural Network [8]

Level (SAE)	Type	Dynamic driving task	Monitoring of Driving Environment	Request to intervene (When fallback performance of Dynamic driving task)	Driving mode
0	No automation	*Human driver*	*Human driver*	*Human driver*	*n/a*
1	Driver assistance	*Human driver and system*	*Human driver*	*Human driver*	*Some driving modes*
2	Partial automation	*System*	*Human driver*	*Human driver*	*Some driving modes*
3	Conditional automation	*System*	*System*	*Human driver*	*Some driving modes*
4	High automation	*System*	*System*	*System*	*Some driving modes*
5	Full automation	*System*	*System*	*System*	*All driving modes*

Fig. 7: Levels of autonomy in autonomous driving [9]

determine the customer base for a product and pricing. In social media, to suggest friends and events based on the user's friend circle. Recurrent neural networks are widely used in speech recognition and translation. Convolutional neural networks are widely used in object detection and segmentation

One of the main usage is in autonomous driving where a machine learns to drive a vehicle. Autonomous driving has many underlying application of ML and DL such as lane detection, parking space detection, object detection, driver activity detection and more

V. Self Driving car

Self-driving means the driving a vehicle autonomously to a specific destination in real traffic without the intervention of a human driver.

There are various levels of autonomy in the car. Advanced driver assistance functions supports driver by taking care of the automatic cruise control and parking assistance systems . In case there is an unavoidable situation, the system alerts the driver to take over the driving. Therefore, human intervention is involved in such operations. Autonomous driving, which can perform driving tasks without human intervention, belongs to higher levels of autonomy 4 or 5. The figure 7 shows the autonomy as per Society of Automotive Engineers (SAE).

Self Driving cars have a rich history of research and experimentation, but there are two which significantly stood out to make a big impact in this field. Firstly, in 1995, Dr Pomerleau, a professor in Carnegie melon university, used neural network to train a car and drove autonomously across the streets of America. This is famously known as "No hands across America", shown in figure 8. This showed the possibility that a neural network can drive a car. Later in the year 2005, the DARPA challenge for Autonomous Driving

Fig. 8: Dr. Pomerleau's No Hands across America [10]

Fig. 9: Stanford Stanley, winner of the DARPA challenge 2005 [11]

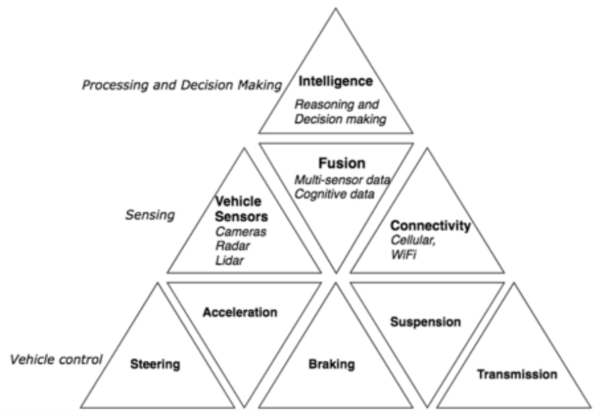

Fig. 10: Automated Driving building blocks [9]

gave rise to a lot of research in this field. The challenge was to build an autonomous car that can navigate from one place to the destination with no human intervention. Stanley, as shown in figure 9, is the car from Stanford University that won the challenge. This gave rise for the possibility of the Autonomous cars

Presently, there are many automobile manufacturers working on self driving cars, also companies like Waymo, Tesla and Uber are already testing their self driving cars on the local streets.

The Sensors that are required for perceiving the eniviron-ment are LIDAR, RADAR and Camera. LIDAR is a very accurate sensor whose data can be used to perceive the environment with little or no machine learning involved. But LIDARs are expensive and it is not viable on a commercial vehicle so to percieve the environment most of the cars resort to RADAR and Camera image. RADARs are similar to LIDAR but with lesser accuracy, this is used in sensor fusion along with Camera data.

Camera data are widely used in autonomous driving, the images of the street are searched for objects such as people or other vehicles and the vehicle takes the decision. Traditionally, the image from the camera was processed using image processing techniques and Computer Vision, this involved in running complex and bulky algorithms on the image for object detection. With the advent of Convolutional Neural Networks, this task of image processing and object detection can be done with high accuracy and non-bulky algorithms using CNN

The prism view of intelligent driving functions is shown in figure 10. The data captured from the sensors mentioned above are combined, pre-processed and passed to a machine learning or deep learning algorithm. The output from the algorithm can control steering, accelerating or braking etc.

VI. End to end learning

CNN has revolutionized Pattern Recognition and with the advent of Large Scale Visual Recognition Challenge (ILSVRC), it has become widely popular. CNN is now is widely used for mainly object detection in the autonomous driving. In autonomous cars, the object is first detected and then the controller will decide to take further actions over the detected object. This method involves controller to know, before hand, an action to perform when the object is detected. This gave rise to the question if this can be made more effective and hence the following solution [12].

This solution is inspired by 1989 built the Autonomous Land Vehicle in a Neural Network (ALVINN) and also the DARPA challenge vehicle DARPA Autonomous Vehicle (DAVE)

A. *Proposed Solution*

The primary goal for this work is to avoid any need to recognize specific human-designated features, such as lane markings, guard rails, or other cars, and to avoid if, then, else rules, based on observation of these features.

The solution involves taking images of the road along with the corresponding steering angle and mapping a connection between the both so that the car can drive itself just based on the image of the road

The solution involves three cameras mounted behind the windshield of the data-acquisition car to collect the time stamped data of the road ahead. Along with this, the steering angle at the corresponding time is also obtained by tapping into the CAN bus of the steering controller. The collected time stamped images are taken as input and the steering angles as the desired output to a system running Convolutional Neural Network.

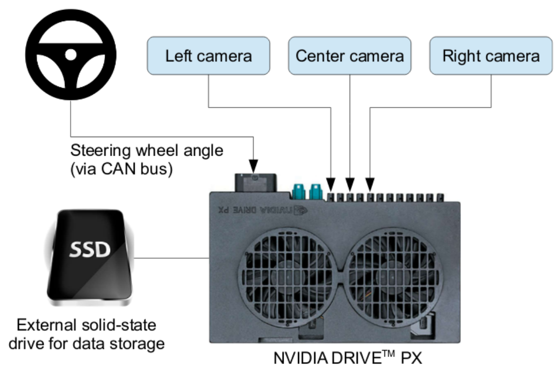

Fig. 11: Hardware overview [12]

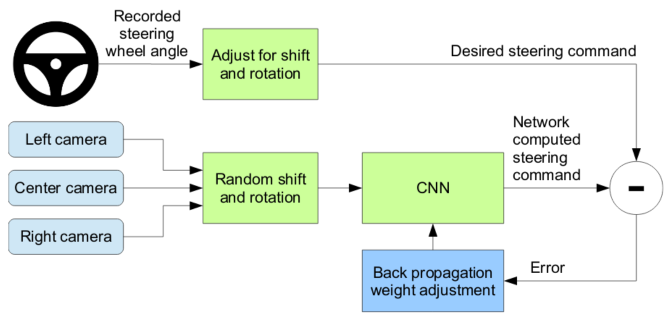

Fig. 12: Training the network [12]

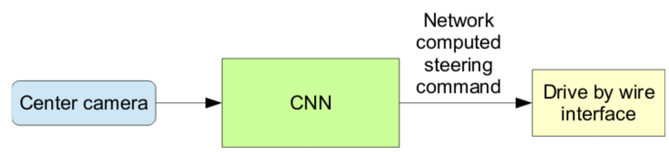

Fig. 13: Trained network [12]

B. System overview

As shown in figure 11, the system consists of one center camera and two off center cameras connected to a NVIDIA DRIVE PX box , which is the Artificial Intelligence platform developed by NVIDIA. The steering angle of the car, received by tapping into the CAN bus of the controller, is also fed into the the NVIDIA DRIVE PX box. An external drive is connected to store the data.

C. Training network

The system is trained using a Convolutional Neural Network with input as the time stamped images from the center and off center cameras and desired output as the corresponding steering angle.

The system has to be independent of the geometry of the car, hence the steering command is taken as 1/r instead of r, where r is the radius of turning in meters. 1/r is used instead of r to prevent a singularity when driving straight where r would become infinity.

As shown in figure 12, the images from all the three cameras are random shifted and rotated and then fed as input to the Convolutional Neural Network. The steering angle is also random shifted and rotated and fed to the Convolutional Neural Network as desired output. The whole network is trained via backpropogation and weight adjusted to minimize the error.

The network is trained to minimize the mean squared error between the predicted steering output and the desired steering output.

After training the network, the trained model is deployed onto the system with only center camera as input, as shown in figure 13, and the corresponding steering angle is received as output

D. Network architecture

The network architecture is shown in figure 14. It is a 9 layered network which includes a normalization layer, 5 convolutional layers and 3 fully connected layers. The input image is converted to YUV planes and passed to the first layer in the network to normalize the input image. The normalizer parameters are hard coded and does not change in the learning process. The next 3 layers are the convolutional layers with filter size 5x5x3, the next two layers are convolutional layers with filter size 3x3x3. The remaining 4 layers are the fully connected layers. The first three convolutional layers are strided with a stride 2x2 and the next two layers are non strided.

E. Data Collection and Augmentation

Data collected consists of diverse conditions such as foggy, cloudy, rainy sunny and lowlight weather conditions,

The data is augmented by adding artificial random, chosen from normal distribution, shifts and rotations so that the network can recover from poor orientation. The distribution has zero mean and the standard deviation is twice as that of a human driver. Additionally, Images of two specific off-center shifts can be obtained from the left and the right camera. An example of the images after augmentation is shown in figure 15.

F. Simulation

Before testing the system on the actual vehicle, it is tested on a simulated environment. As shown in figure 16 , a recorded video footage of the road from the car is fed to the simulator, this frames are passed shifted and rotated and passed on to the Convolutional Neural Network. The CNN will output the corresponding steering angle, this is later translated in to update the position and orientation of the car and the next frame corresponding to the new orientation is passed on to the Convolutional Neural Network

G. Visualization

The figure 17 and 18 shows the visualization of the internal states of the CNN. It shows the activation of the feature maps of the first two layers. Figure 17 shows the feature maps of the image of a road on which the system was trained, the

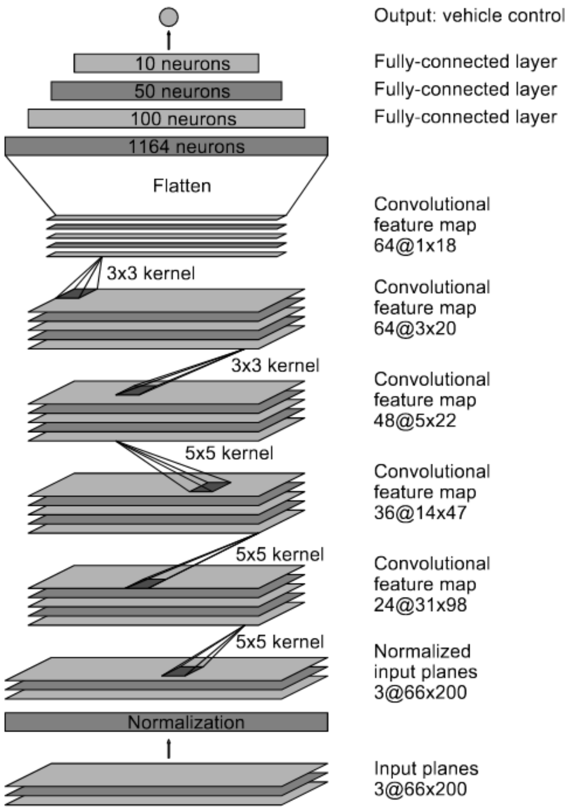

Fig. 14: Convolutional Neural Network architecture [12]

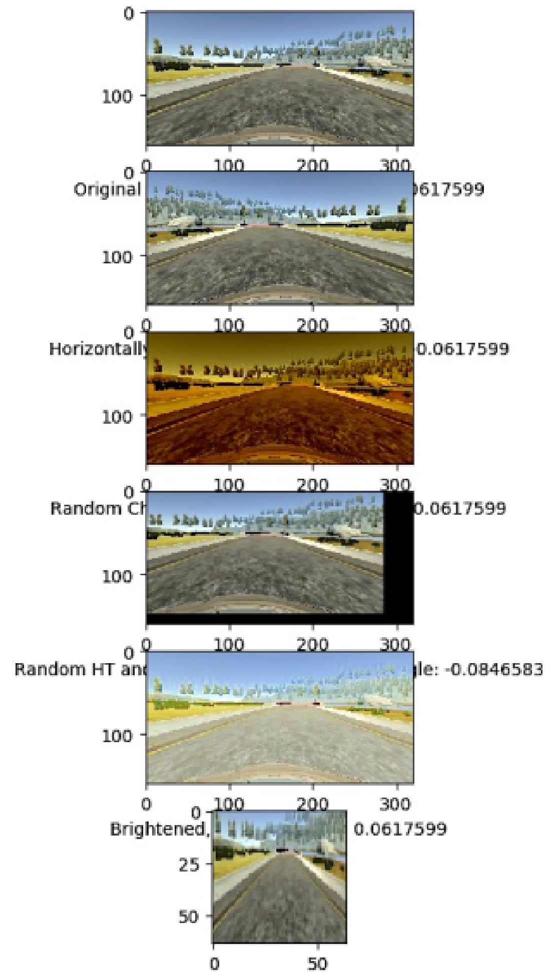

Fig. 15: Augmented data with various shifts and transformations [9]

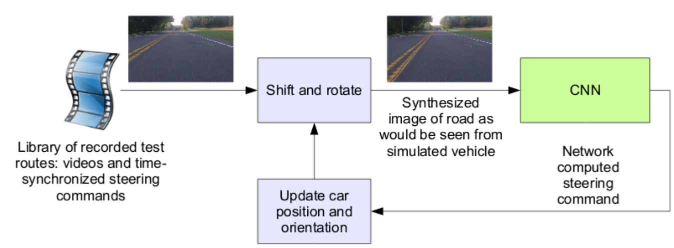

Fig. 16: Simulation environment to train and test the CNN [12]

edges are detected as expected. The same is visualized for the an image of forest, fig 18, which was not trained earlier and hence the feature maps does not provide any useful features.

H. Evaluation

Evaluation of the network is first done in simulation and then in on-road tests.

The metric of evaluation is the number of human intervention required while the autonomous vehicle is driven. The intervention occur when the vehicle departs from the center line by more than a meter. Assuming the actual intervention takes 6 seconds, the formula for autonomy is

$$autonomy = (1 - \frac{(numberofinterventions).(6sec)}{elapsedtime[seconds]}).100$$

On a test run, an autonomy of 98% was achieved by the actual vehicle driven autonomously on the road

Furthermore, a research, extending the above mentioned research, was done [9]. In this, the same Convolutional neural network and training network is being used. But while training and testing the network, a simulator based on Unity 3d is used. One of the main focus of this research is to test the effectiveness of the software validation methodologies for deep learning systems.

There are, in practice, two models for architecting and validating automotive systems.

1) The Standard V-Model
2) The Deep W-Model

The Standard V-Model describes the general SDLC phases of automotive software along with testing and in conformation with the standards and benchmarks as shown in fig 19.

But the traditional V-Model doesn't conform to the architecting and development methodologies of deep learning and hence a new form of validation was required.

Falcini in the year 2017 introduced the W-Model, as seen in

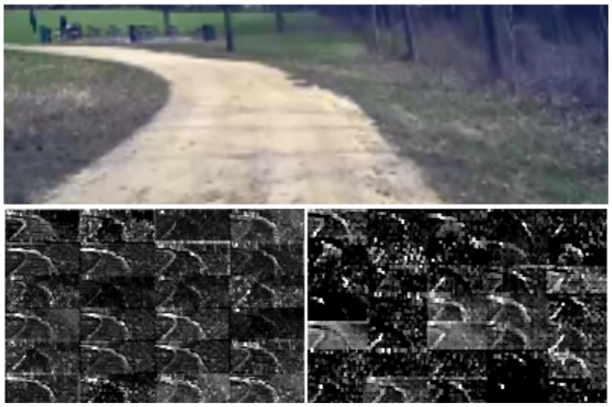

Fig. 17: Image of the road on which the CNN was trained [12]

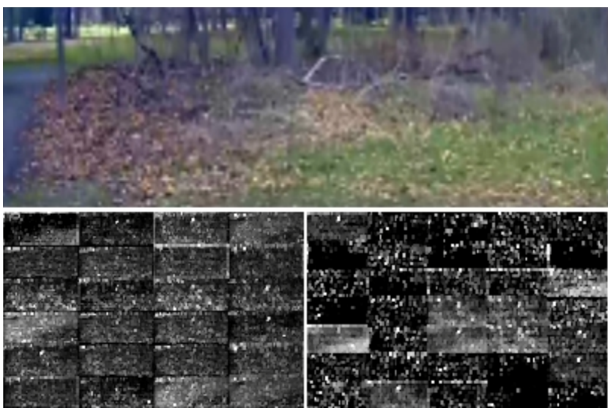

Fig. 18: Image of a forest on which the CNN was not trained [12]

fig 20, which theoretically has the capability to integrate artificially intelligent data driven development with the traditional V-model and hence this methodology was used to develop the deep learning system.

The data collection was done by using three cameras mounted on the simulated vehicle that record the steering angles, speed, brake and throttle parameters. The simulated vehicle was first driven by a using the keyboard inputs by human driving behavior and using a deep CNN architecture. The CNN model was trained and was able to handle a variety of scenarios, the workflow is depicted in fig 22. The overall flow of the model training and evaluation, are as mentioned below:

1) The Unity 3d simulator used in generation and collection of data
2) Analysis, pre-processing and data augmentation done on the collected data
3) More data collected through re-simulation and augmentation
4) Data divided into training and test set
5) Model is trained for specific number of epochs
6) Testing of the trained model

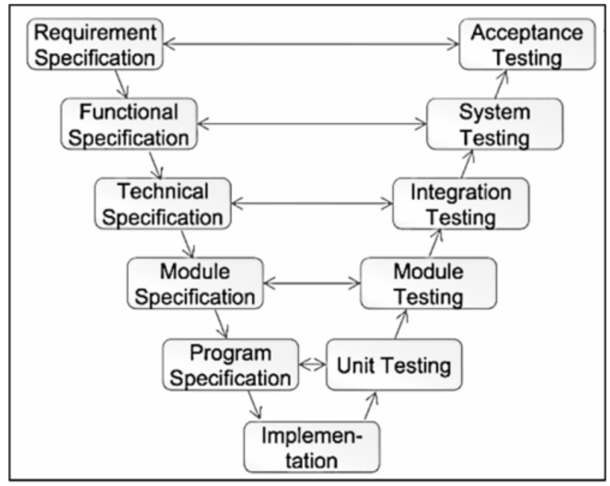

Fig. 19: The standard V-Model [9]

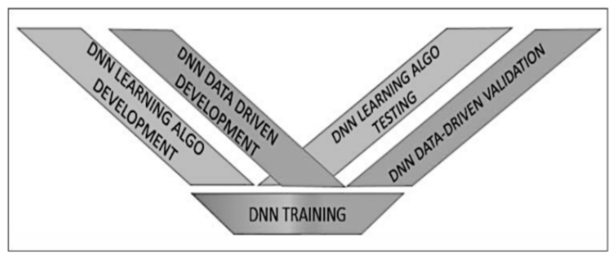

Fig. 20: W-Model of Deep Learning by Falcini [9]

In the training process, the collected data mostly concentrated at one steering angle as most of the time the simulated vehicle was driven straight. This resulted in a large number of 0 degree angle for steering. With data augmentation, the angles were varied and the dataset was normalized such that it's no longer concentrated at one particular angle. Also, data pre-processing is done to ensure that the data is the required data of interest. Three data pre-processing process were done as below

1) Normalization using some of the image augmentation techniques
2) Cropping the regions of interest
3) Scaling the images to improve the efficiency of the model

The histogram depiction of data before and after processing is shown in fig 23.

The network was trained and simulated using the unity based simulator and the result was similar to the previous research.

VII. Learning Driving Style

The Comfort of a driver and passenger in the car is subjective, one might like to drive the car slow and one might like to drive it fast, the acceleration profiles, distance to other cars, speed during lane change and more characterized an human driver. In an autonomous car, various driving styles conforming to the driver can be achieved by changing the

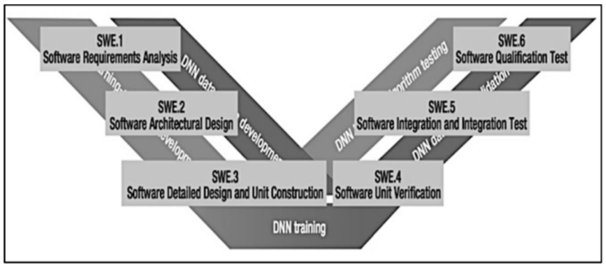

Fig. 21: Data enabled W-model [9]

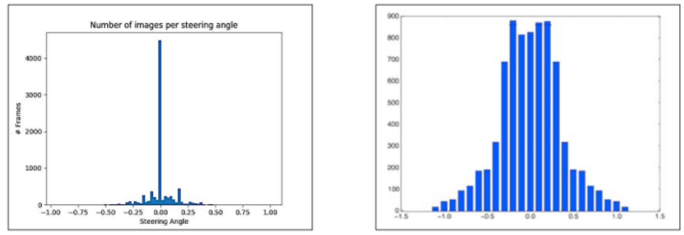

Fig. 23: (a) Highly biased dataset (b) Normalized Data [9]

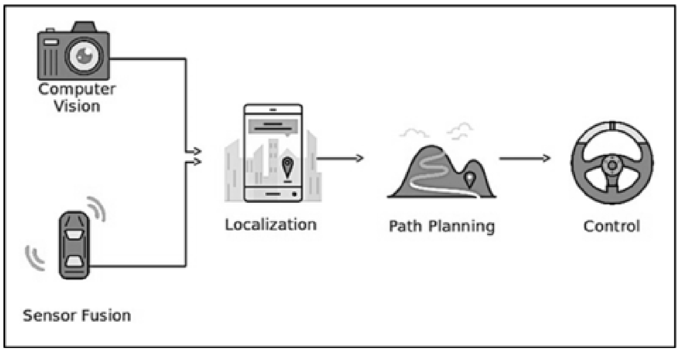

Fig. 22: Workflow from data collection to vehicle control [9]

parameters of the motion planning algorithm in the car. One approach to do this is by manually tuning these parameters if there are smaller set of parameters and smaller set of user preferences, also manual tuning requires expertise in the field. If the parameters are more, then the manual tuning becomes tedious and not feasible.

The earlier solution provides an effective way to utilize the connection between the road image and the steering angle of the driver, but by taking care of the comfort of the driver one can effectively create a connection between the road, steering angle of the driver and parameters such as acceleration, jerk, braking of the car to let the autonomous car mimic the driver's style of driving the car.

In the proposed approach [13], the feature-based inverse reinforcement learning (IRL) [14] is used to learn the parameters for each user from their observed driving style. The system learns to navigate the vehicle on a track within 25 minutes of learning over a real car.

VIII. Efficient Statistical Validation of Machine Learning Systems for Autonomous Driving

In automotive systems, reliability is of utmost importance. After the model is trained, it has be rigorously tested. It is very important to ensure the trained model is highly accurate. As an example, if a vehicle fails to detect a traffic sign on the road, it might move resulting in a catastrophic incident. Also if it raises a false alarm of a false traffic sign it might also result in a non-desirable outcome. So accuracy of prediction in automobile is very important.

In practice, the data required to perform detection is large and to validate small failure rate, a vehicle has to be built and tested on roads which takes several months so this becomes not feasible. To overcome this, the testing can be done on a simulated environment where the car driven data can be fed to the simulator and it runs the intelligence algorithm and the predicted output is validated.

This simulation approach is used even during the design stage to validate the system, the problem that arises with this usage is that the simulator system is 1800 times slower that that physical system. The predicted output from the simulator should be accurate and this in turn demands for a more computationally expensive system than the physical hardware. Hence, to test a large amount of data to capture small failure rate becomes not feasible and inefficient.

Hence, to reduce the computational cost, Subset sampling (SUS) method is used. The proposed solution [15] is to represent the probability of small failure as a product of conditional probabilities of several conditional probabilities.

SUS uses continuous random samples of data but the given dataset contains discrete data and hence an undirected graph mapping is used on discreet datasets to create a conditional dataset and then use Markov Chain Monte Carlo (MCMC) algorithm to walk on the generated graph and create random samples.

Also, while generating the undirected graph with large number of vertices, several dimension reduction techniques are to be used.

The failure rate is defined as the probability P_f that a system takes the input and produces a wrong output.

If the system failure rate is small then a large number of data samples must be generated to determine rare failure probabilities and doing this results in the expensive computation costs. To overcome this, SUS defines intermediate failure regions

$$\Omega = \Omega_k \subset \Omega_{k-1} \subset ..... \subset \Omega_1$$

where Ω_k is identical to the original failure region . An example with k = 3 is shown in figure 24. Each of these intermediate regions are generated by lowering the threshold δ of detection

$$\delta = \delta_k > \delta_{k-1} > ..... > \delta_1$$

After these regions are determined, the failure probability is calculated by the product of failure probabilities in each intermediate region

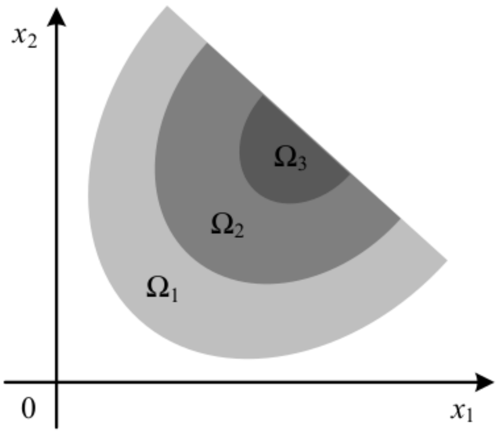

Fig. 24: A plot of intermediate failure regions [15]

$$P_f = P(x \in \Omega_1). \prod_{k=2}^{k} \Omega_{k-1} P(x \in \Omega_k | x \in \Omega_{k-1})$$

Using this approach, the system can be validated for small failures by using lesser data inputs and low computation.

IX. Challenges

In autonomous driving, there are three main building blocks for the development of deep learning approaches.

1) Data preparation
2) Model generation
3) Model deployment

Data preparation prepares the data such as data recording, ground truth labeling, big data storage, etc. to be used for training and testing the neural network. Model generation involves defining the network architecture, train the network and validate the results. Model deployment involves the deployment of a trained network which is optimized for a specific hardware onto the field tests. A network is trained if the difference between the ground truth and the predicted output is minimal. All of this development process must conform to a standard to be used in the automotive industry.

The development of software systems in Automotive industry is rapidly accelerated by establishing various benchmarks and standards. One such standard is the International Organization for Standardization (ISO) proposed ISO 26262 [4] which regulates functional safety for road vehicles. According to this, It includes requirements and recommendations for the entire lifecycle of car manufacturing. ISO 26262 address functional safety issues in a more systematic way and the overall development process purely follows the traditional V-model. With the advent of machine learning and deep learning in the field of autonomous driving, various functions such as environmental perception, path planning, or even steering wheel control is carried out by deep-learning-based approaches and the development process no longer adhere to the V-model. ISO 26262 was defined without considering the usage of deep Learning and machine learning methods and hence the deep learning and machine learning are in violation with the standard. Due to this, many problems are faced in autonomous cars, among them some of the main problems [16] are as follows

A. Dataset Completeness

A supervised model can only do what it has been taught to do, a deep learning network trained only on cars cannot identify pedestrians on the street. Due to the infinite variations in the data, it is not possible to ensure the complete coverage of the real world scenario. So the question arises as to how to verify the completion of the dataset in a way for the trained network to have maximum generalization.

A large dataset is required to train the deep learning network for autonomous driving, but large data of the same type causes redundancy and doesn't add any impact to the weights of the training model. Recording hours of driving data from a single urban place would not help improve the prediction in any other cities. Most of the recording will be redundant such as the traffic scenes and road that doesn't improve the performance of the model during the fine tuning of the pre trained model. To cause the change in the weights, anomalies are required that can impact the weights. The anomalies such as traffic accidents, broken infrastructure, etc. can affect the weights in the network but hard to get. One solution to this is to synthetically generated data that simulates anomalies to tackle the data completeness problem, situations that are important but hardly happen, such as hazardous weather conditions such as, storm, heavy snowfall, etc., wrong way turning, red light running, accidents, animal hazards are generated synthetically. But the question still remains are

1) How much anomalies are still required?
2) How to verify that the anomalies are learnt by the network?
3) How to find a balance between anomalies and normal data during the fine tuning process?

B. Neural Network Implementation

As per the standard ISO 26262, the functional safety requirements are broken down to the code level of the software. The code written for the software should reflect the specific functional safety requirement

But this cannot be achieved in machine learning and deep learning approaches, as most of the information that is required to assess the safety risk resides in a trained model instead of the code.

Example for creating a simple Convolutional neural network

```
model = Sequential()
model.add(Convolution2d(32, (3,3),\
      inputshape=(imageSize, imageSize,3)))
model.add(Activation('relu'))
model.add(MaxPooling2d(poolsize=(2,2)))
```

The whole code for the deep learning is nothing but stacking a set of layers on each other and connecting them to the input

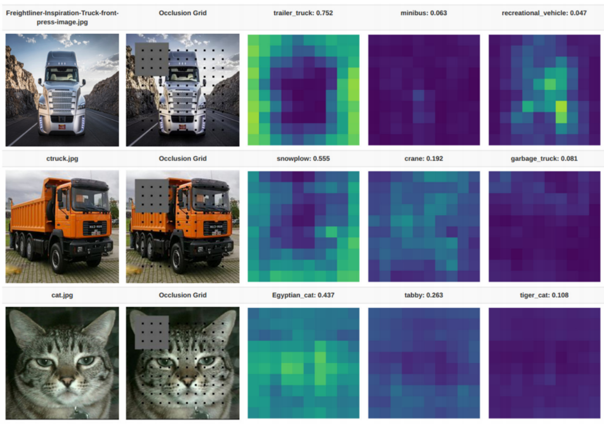

Fig. 25: A screen capture of the Picasso web application after computing partial occlusion figures for various input images [17]

and output. This cannot reflect functional safety requirement such as Collision avoidance with pedestrian or any object on the road. The model becomes a black box and hence the behavior is non-deterministic and verification of the safety requirement becomes hard.

Visual understanding can help provide insights of the functional requirements. Recently an open source visualization software called Picasso [17] was published and it can aid in visualizing the partial occlusion and feature maps in Convolutional neural network. But it still remains non-deterministic for the most of other deep learning and machine learning algorithms.

By using partial specification, one can also interpret the behavior of the neural network. As an example, one can use the physical properties of the object to validate the behavior, if in a road, only pedestrians of height 6 feet or above are allowed, then the system can use this constraint to filter out the false positives. These constraints can be used during the training phase to monitor the behavior of the trained model

C. Transfer learning

Due to different local political restrictions, the data collection fleet could not be operated all around the world and hence training a network from scratch becomes hard. To overcome this problem, transfer learning is used. Transfer Learning refers to training a neural network on one task and use this trained network in another related task. For example, a network trained on various datasets on detection of cats can be be taken and fine tuned to detect dogs instead. Transfer Learning could be applied for reducing the amount of training data required. The idea is to collect data from one part of the world and use it in another part of the world. The pre-trained network is tuned to become compatible for autonomous driving application. The challenge that still remains are:

1) How much additional data is required for tuning the network for Autonomous driving?
2) How to verify the fine tuned network
3) How many layers of the network to be re-trained with the additional data?

X. Conclusion

The advancement in machine learning and deep learning have made the idea of autonomous driving possible. There are various machine learning and deep learning algorithms used in Autonomous driving, the usage of these algorithms depend on the design of the system, but mostly Convolutional neural network and Deep Reinforcement Learning are used widely in autonomous driving.

With the commercial testing of the self driving cars, there are a few accidents involved by Tesla and Uber in the past few years, this is due to the non-deterministic behavior of the learning algorithms so there has to be efforts to make this more deterministic and the output predictable.

With all this into consideration, the future isn't too far. Companies like Tesla, Uber, Daimler, Nisan and more have announced the commercial availability of the respective autonomous cars in early 2020's. What once seemed impossible will now become a part of our lives and the future becomes something to look forward to.

References

[1] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, , and L. D. Jackel, "Backprop-agation applied to handwritten zip code recognition," in *Neural Computation*, 1989, http://yann.lecun.org/exdb/publis/pdf/lecun-89e.pdf.

[2] Watzenig, Daniel, Horn, and Martin, *Automated Driving, Safer and More Efficient Future Driving*, 2017.

[3] S. Srinivas, R. K. Sarvadevabhatla, K. R. Mopuri, N. Prabhu, S. S. S.Kruthiventi, and R. V. Babu, "Taxonomy of deep convolutional neural nets for computer vision," 2016.

[4] I. E. 2011. Road vehicles functional safety. standard. international organization for standardization.

[5] A. Gron, *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly, 2017.

[6] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[7] R. B. Zadeh and B. Ramsundar, *TensorFlow for Deep Learning*. O'Reilly Media, Inc., 2018.

[8] A. Deshpande. A beginner's guide to understanding convolutional neural networks. http://bit.ly/2JKtPBJ.

[9] P. Kaur, R. Sobti, and A. Khamparia, "Simulation and deep cnn based architecture for validation of intelligent automotive functions," in *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2018.

[10] D. R. Baker. Driverless milestone: No hands across america. http://bit.ly/2JFAChj.

[11] Standord. Self driving car. https://stanford.io/2XKE8PO.

[12] N. Corporation, "End to end learning for self-driving cars," 2016.

[13] M. Kuderer, S. Gulati, and W. Burgard, "Learning driving styles for autonomous vehicles from demonstration," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015.

[14] A. Ng and S. Russell, "Algorithms for inverse reinforcement learning."

[15] W. Shi, M. B. Alawieh, X. Li, H. Yu, N. Arechiga, and N. Tomatsu, "Efficient statistical validation of machine learning systems for autonomous driving."

[16] Q. Rao and J. Frtunikj, "Deep learning for self-driving cars:chances and challenges," in *2018 ACM/IEEE 1st International Workshop on Software Engineering for AI in Autonomous Systems*, 2018.

[17] R. Henderson and R. Rothe, "Picasso: A modular framework for visualizing the learning process of neural network image classifiers," in *Journal of open research software*, 2017.