International Journal of Science and Research (IJSR)
ISSN: 2319-7064
SJIF (2022): 7.942

Training and Testing Autonomous Driving Software using Udacity Car Simulator

Attla Vamshi Krishna Reddy[1], Guda Varun Sai[2], Sarvesh Vishwakarma[3], Annam Mani Sai[4], Dantuluri Abhishek[5]

[1]Department of Computer Science and Engineering, Graphic Era (Deemed to be University), Dehradun, India
Email: *attlavamshikrishnareddy.cse190111132[at]gehu.ac.in*

[2]Department of Computer Science and Engineering, Graphic Era (Deemed to be University), Dehradun, India
Email: *gudavarunsai.cse190111151[at]gehu.ac.in*

[3]Department of Computer Science and Engineering, Graphic Era (Deemed to be University), Dehradun, India
Email: *sarveshvishwakarma.cse[at]geu.ac.in*

[4]Department of Computer Science and Engineering, Graphic Era (Deemed to be University), Dehradun, India
Email: *annammanisai.cse190111062[at]gehu.ac.in*

[5]Department of Computer Science and Engineering, Graphic Era (Deemed to be University), Dehradun, India
Email: *dantuluriabhishek.cse190111130[at]gehu.ac.in*

Abstract: ***Deep Learning has been shown to be a successful method for autonomous driving, allowing cars to identify and adapt to their surroundings in real time. Many advances in deep learning-based approaches for autonomous driving have been developed in recent years, including object identification, semantic segmentation, and path planning. Object detection allows the vehicle to distinguish items in its immediate surroundings, such as pedestrians, automobiles, and traffic signs. Semantic segmentation allows the vehicle to distinguish assorted items in the area as well as their borders, offering a more thorough awareness of the surroundings. Path planning is the process of determining a safe and efficient route for a vehicle based on data from object detection and semantic segmentation. Deep learning's capacity to learn from massive volumes of data is one of its primary advantages for autonomous driving, allowing the automobile to adapt to changing events and conditions. Yet, there are still obstacles to overcome, such as assuring the safety and dependability of self-driving vehicles and resolving ethical issues about decision-making.***

Keywords: Autonomous driving, Deep learning, Convolutional neural networks (CNNs), Reinforcement learning, Decision-making, Perception, Accuracy

1. Introduction

The concept of autonomous driving has rapidly gained attention in recent years, with the potential to revolutionize the transportation industry. The development of autonomous vehicles requires the ability to recognize and respond to their environment in real-time, which has led to a surge in the application of deep learning techniques in this field. Deep learning in autonomous driving refers to the use of sophisticated and intricate artificial neural networks to enable autonomous cars to learn and make decisions based on their perception of their surroundings. [1]

To train neural networks to recognize objects in real-time, such as other vehicles, pedestrians, traffic signs, and road markings, deep learning techniques are employed in autonomous driving. This involves using techniques such as object detection and semantic segmentation to accurately identify and locate objects in the environment.

Deep learning is also utilized in route planning, which entails constructing a safe and efficient track for the vehicle to follow based on data from object identification and semantic segmentation. This entails employing techniques like reinforcement learning to allow the vehicle to learn from past events and make better judgements in the future. Deep learning has gotten a lot of attention in recent years due to its ability to learn and adapt to new environments and scenarios, making it a great tool for building autonomous automobiles that can operate safely and productively in real-world settings.

Deep learning has also been utilized in other areas of autonomous driving, such as sensor fusion, decision-making, and mapping. Sensor fusion is the process of combining data from several sensors, such as Lidar, radar, and GPS, to produce a more accurate picture of the environment. Deep reinforcement learning is used in decision-making to allow the vehicle to learn from its experiences and make better judgements in the future. Mapping entails employing deep learning techniques to generate high-definition maps that may be utilized to improve the vehicle's view of its surroundings. While deep learning has shown considerable promise in autonomous driving, its implementation is fraught with difficulties.

Deep learning models such as convolutional neural networks (CNNs) [2], recurrent neural networks (RNNs), and reinforcement learning are among those discussed in the paper's discussion of autonomous driving. CNNs are often employed for image processing tasks such as object recognition and semantic segmentation, whereas RNNs are typically utilized for sequential data processing tasks such as path planning. Reinforcement learning, on the other hand, comprises the agent learning to make decisions based on their environment through trial and error.

The research also emphasizes the significance of transfer learning in deep learning-based autonomous driving, where pre-trained models are employed to fine-tune neural networks to meet the unique needs of the job at hand. Transfer learning is especially effective when training data is restricted, since it allows the model to employ knowledge learnt from a bigger dataset to enhance performance.

In addition, the article will offer experimental results on real-world datasets to demonstrate the efficacy of deep learning-based systems for autonomous driving. This will feature a comparison of deep learning techniques to traditional approaches, emphasizing deep learning's better performance in several autonomous driving tasks.

Furthermore, the paper will address the significance of ethical issues in the development and deployment of self-driving cars. It will discuss the ethical quandaries related to autonomous car decision-making, such as the trade-off between passenger and pedestrian safety, and how these quandaries may be solved using deep learning.

This study gives a thorough examination of the most recent deep learning techniques utilized for autonomous driving, such as object identification, semantic segmentation, and path planning. It delves deeper into the issues surrounding the deployment of autonomous cars, including safety, ethics, and decision-making, and how deep learning may assist address these issues.

This study also discusses the significance of real-world testing for deep learning model validation, as well as the relevance of different sensors such as Lidar, radar, and GPS in enabling autonomous cars to properly perceive their environment.

The purpose of this article is to offer a thorough review of the use of deep learning in autonomous driving and its potential to alter the transportation sector.

2. Related Work

In 1992, Stefan Neuter, Jos Nijhuis, and Lambert Spangenberg published a research paper titled Advances in Autonomous Vehicle Navigation. This research describes a novel method to autonomous vehicle navigation that uses neural networks to mimic human driving behavior. Because of the task's complicated structure and uncertain parametrization, it is difficult to apply typical system—theoretical methodologies. Neural networks give a solution that may be taught from driving data measurements.

Obstacle Detection and Classification Using Deep Learning for Tracking in High-Speed Autonomous Driving was published in 2017 by Gowtham Prabhakar and Binsu Kailath.[3, 4, 5]The classification and recognition of on-road obstructions is an essential component of a self-driving car's visual system.[6, 5] Because vehicle tracking requires vehicle location and frame connection, vehicle detection and classification are critical. Vision-based approaches are recommended for this purpose due to the cost and importance of the appearance data connected with the vision data. This study employs a region-based convolutional neural network trained on the PASCAL VOC image dataset to build a deep learning system for recognizing and categorizing on-road impediments such as vehicles, people, and animals.[3, 5]

Kichun Jo, Junsoo Kim, Dong Chul Kim, Chulhoon Jang, and Myoungho Sunwoo published A Case Study on the Development of an Automatic Driving System Based on Distributed Architecture in 2015.[7] This article focuses on the deployment of an autonomous driving system as a case study of the proposed development strategy. To simplify the process, the essential autonomous driving algorithms (location, perception, planning, vehicle control, and system management) [7]are rapidly explained and applied to the creation of an autonomous driving system. We may evaluate the benefits of a distributed system architecture and the proposed development approach by conducting a case study on the autonomous system's implementation.[7]

3. Proposed Methods

The simulator contains two courses and two modes, training and autonomous, as well as a controls section where we can regulate things like speed, braking, and steering.

The dataset is generated by the user by running the car in training mode and downloading images from the simulator. This dataset is deemed to include "excellent" driving data. The deep learning model is then put to the test on a racecourse once it has been trained using this user data. Another challenge is translating performance across many tracks. Alternatively, test the model on one track before practicing it on another.

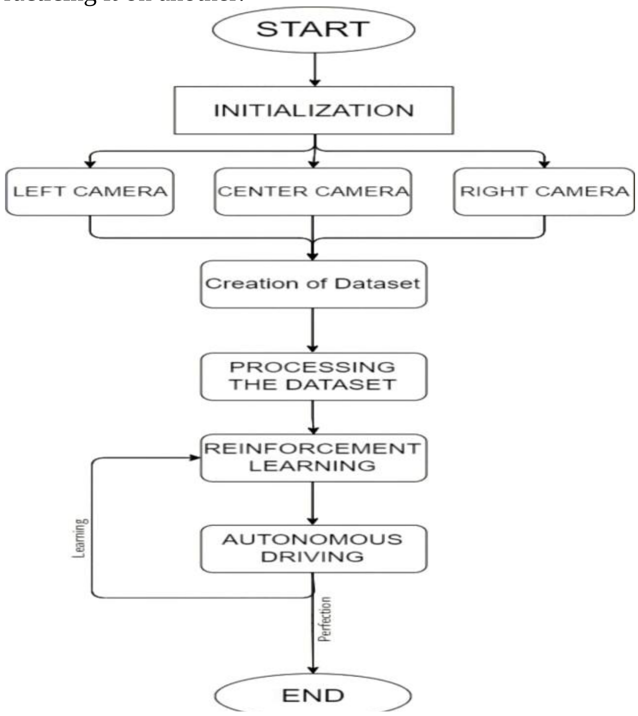


Figure 1: Flow chart for complete execution

Figure1 is a flow chart of the complete project which gives a brief overview of this paper as well as the project.

Initialization is the step taken to start the simulator for the training mode through which dataset is generated.

We have a three headed camera set on three sides of the car (Left, Center, Right) by which gets the sensor dataset. By using the above sensor data, we will get a dataset which will have paths for all the images stored and as well as the controls data. Which is mentioned in Figure4.

Now the main phase starts here, we need to load the data set into the autonomous mode by which it gets run. Using reinforcement learning it learns its mistakes and moves into autonomous driving platform and if in case any learning needed it heads back to reinforcement learning and if all goes well it reaches the destination it's the end.

Now we will be learning about the main three phases of the above flowchart in detail view:

A. Training Mode

In Training mode, we will drive the car manually, recording the path and angles we take. The simulator will also take images of us driving and save them in the drive images folder, which will be used to train the autonomous mode for its deployment, and it will correct itself based on its mistakes.

Figure 1: Recoding and capturing the manual driven.

If we look deeply, we can see the two main parameters in Figure1 are.

- Steering angle: the angle of rotation of the vehicle steering through which the complete car rotation takes place.
- Speed meter: the measurement of the speed travelled by the car is denoted in meter per hour (mph). We have set maximum speed in the training mode to 50mph, for autonomous mode we can set speed in the code itself.

B. Dataset of Images

All the photographs that the simulator took while in training mode are now stored in the drive-images folder. At least 2000 photos are most likely taken by the simulator when it is in full training mode. It creates both a datalog.csv file and an image dataset. This file includes the picture pathways together with the current steering angle, throttle position, brake pressure, and speed of the vehicle.

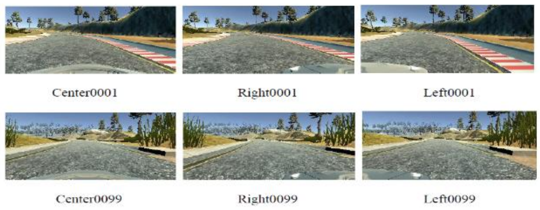
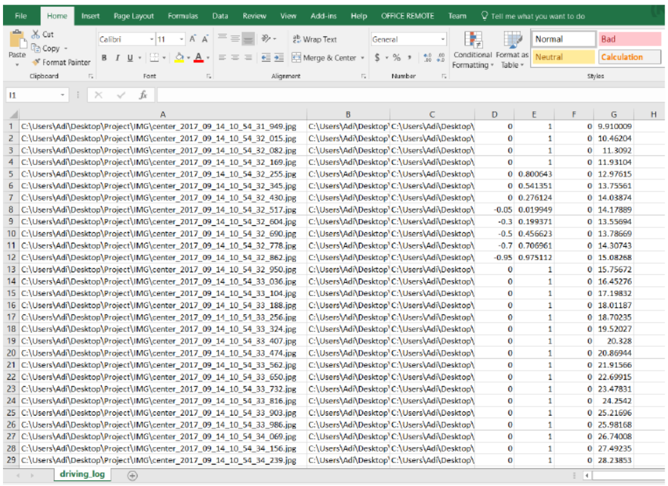


Figure 2: Saved Data From three different angles of camera

The dataset of the images is shown in the excel format which consists of 6 columns. For each one there is a unique identification.



	A	B	C	D	E	F	G	H
1	C:\Users\Adi\Desktop\Project\IMG\center_2017_09_14_10_54_31_949.jpg	C:\Users\Adi\Desktop\	C:\Users\Adi\Desktop\	0	1	0	9.910009	
2	C:\Users\Adi\Desktop\Project\IMG\center_2017_09_14_10_54_32_015.jpg	C:\Users\Adi\Desktop\	C:\Users\Adi\Desktop\	0	1	0	10.46204	
3	C:\Users\Adi\Desktop\Project\IMG\center_2017_09_14_10_54_32_082.jpg	C:\Users\Adi\Desktop\	C:\Users\Adi\Desktop\	0	1	0	11.3092	
4	C:\Users\Adi\Desktop\Project\IMG\center_2017_09_14_10_54_32_169.jpg	C:\Users\Adi\Desktop\	C:\Users\Adi\Desktop\	0	1	0	11.93104	
5	C:\Users\Adi\Desktop\Project\IMG\center_2017_09_14_10_54_32_255.jpg	C:\Users\Adi\Desktop\	C:\Users\Adi\Desktop\	0	0.800643	0	12.97615	
6	C:\Users\Adi\Desktop\Project\IMG\center_2017_09_14_10_54_32_345.jpg	C:\Users\Adi\Desktop\	C:\Users\Adi\Desktop\	0	0.541351	0	13.75561	
7	C:\Users\Adi\Desktop\Project\IMG\center_2017_09_14_10_54_32_430.jpg	C:\Users\Adi\Desktop\	C:\Users\Adi\Desktop\	0	0.276124	0	14.03874	
8	C:\Users\Adi\Desktop\Project\IMG\center_2017_09_14_10_54_32_517.jpg	C:\Users\Adi\Desktop\	C:\Users\Adi\Desktop\	-0.05	0.019949	0	14.17889	
9	C:\Users\Adi\Desktop\Project\IMG\center_2017_09_14_10_54_32_604.jpg	C:\Users\Adi\Desktop\	C:\Users\Adi\Desktop\	-0.3	0.199371	0	13.55694	
10	C:\Users\Adi\Desktop\Project\IMG\center_2017_09_14_10_54_32_690.jpg	C:\Users\Adi\Desktop\	C:\Users\Adi\Desktop\	-0.5	0.456623	0	13.78669	
11	C:\Users\Adi\Desktop\Project\IMG\center_2017_09_14_10_54_32_778.jpg	C:\Users\Adi\Desktop\	C:\Users\Adi\Desktop\	-0.7	0.706961	0	14.30743	
12	C:\Users\Adi\Desktop\Project\IMG\center_2017_09_14_10_54_32_862.jpg	C:\Users\Adi\Desktop\	C:\Users\Adi\Desktop\	-0.95	0.975112	0	15.08268	
13	C:\Users\Adi\Desktop\Project\IMG\center_2017_09_14_10_54_32_950.jpg	C:\Users\Adi\Desktop\	C:\Users\Adi\Desktop\	0	1	0	15.75672	
14	C:\Users\Adi\Desktop\Project\IMG\center_2017_09_14_10_54_33_036.jpg	C:\Users\Adi\Desktop\	C:\Users\Adi\Desktop\	0	1	0	16.45276	
15	C:\Users\Adi\Desktop\Project\IMG\center_2017_09_14_10_54_33_104.jpg	C:\Users\Adi\Desktop\	C:\Users\Adi\Desktop\	0	1	0	17.19832	
16	C:\Users\Adi\Desktop\Project\IMG\center_2017_09_14_10_54_33_188.jpg	C:\Users\Adi\Desktop\	C:\Users\Adi\Desktop\	0	1	0	18.01187	
17	C:\Users\Adi\Desktop\Project\IMG\center_2017_09_14_10_54_33_256.jpg	C:\Users\Adi\Desktop\	C:\Users\Adi\Desktop\	0	1	0	18.70235	
18	C:\Users\Adi\Desktop\Project\IMG\center_2017_09_14_10_54_33_324.jpg	C:\Users\Adi\Desktop\	C:\Users\Adi\Desktop\	0	1	0	19.52027	
19	C:\Users\Adi\Desktop\Project\IMG\center_2017_09_14_10_54_33_407.jpg	C:\Users\Adi\Desktop\	C:\Users\Adi\Desktop\	0	1	0	20.328	
20	C:\Users\Adi\Desktop\Project\IMG\center_2017_09_14_10_54_33_474.jpg	C:\Users\Adi\Desktop\	C:\Users\Adi\Desktop\	0	1	0	20.86944	
21	C:\Users\Adi\Desktop\Project\IMG\center_2017_09_14_10_54_33_562.jpg	C:\Users\Adi\Desktop\	C:\Users\Adi\Desktop\	0	1	0	21.91566	
22	C:\Users\Adi\Desktop\Project\IMG\center_2017_09_14_10_54_33_650.jpg	C:\Users\Adi\Desktop\	C:\Users\Adi\Desktop\	0	1	0	22.69915	
23	C:\Users\Adi\Desktop\Project\IMG\center_2017_09_14_10_54_33_732.jpg	C:\Users\Adi\Desktop\	C:\Users\Adi\Desktop\	0	1	0	23.47831	
24	C:\Users\Adi\Desktop\Project\IMG\center_2017_09_14_10_54_33_816.jpg	C:\Users\Adi\Desktop\	C:\Users\Adi\Desktop\	0	1	0	24.2542	
25	C:\Users\Adi\Desktop\Project\IMG\center_2017_09_14_10_54_33_903.jpg	C:\Users\Adi\Desktop\	C:\Users\Adi\Desktop\	0	1	0	25.21696	
26	C:\Users\Adi\Desktop\Project\IMG\center_2017_09_14_10_54_33_986.jpg	C:\Users\Adi\Desktop\	C:\Users\Adi\Desktop\	0	1	0	25.98168	
27	C:\Users\Adi\Desktop\Project\IMG\center_2017_09_14_10_54_34_069.jpg	C:\Users\Adi\Desktop\	C:\Users\Adi\Desktop\	0	1	0	26.74008	
28	C:\Users\Adi\Desktop\Project\IMG\center_2017_09_14_10_54_34_156.jpg	C:\Users\Adi\Desktop\	C:\Users\Adi\Desktop\	0	1	0	27.49235	
29	C:\Users\Adi\Desktop\Project\IMG\center_2017_09_14_10_54_34_239.jpg	C:\Users\Adi\Desktop\	C:\Users\Adi\Desktop\	0	1	0	28.23853	

driving_log

Figure 3: Dataset of all the stored images

Columns 1, 2, and 3 include paths to the dataset pictures of the center, right, and left, respectively.

The steering angle is found in column 4. If 0 denotes a straight line, a positive number denotes a right turn, and a negative number denotes a left turn.

The vehicle's acceleration at that time is shown in Column 5.

The vehicle's brake at that time is in Column 6.

The vehicle's speed at that moment is shown in Column 7

C. Autonomous Mode

The Autonomous mode will run the car for three laps to reduce errors and increase reinforcement learning capability. Using the dataset that had been generated by the training mode in which we had manually driven the car.

Figure 4: Autonomous mode Driving test.

4. Results and Analysis

For each of the procedures that were previously discussed, the outcomes that follow were seen. I had to come up with two separate performance indicators to compare them.

1) Accuracy or Value Loss (computed during training phase)
2) Generalization (drive performance)

1) Value Loss:

We evaluated the values of loss rate during the autonomous mode during each lap and we can observe that the loss rate at the beginning of the validation is high when compared with end. It is because the autonomous car is getting trained with the dataset and by its mistakes.

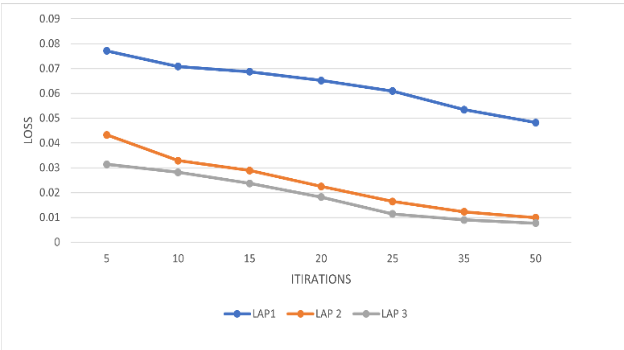


Figure 5: Loss rate vs Iterations.

We have completed 3 laps (lap1, lap2, lap3) and we have recorded the readings after a certain number of iterations. As we can observe, the autonomous car is learning from the dataset and mistakes.

2) Drive Performance

Performance was the second factor evaluated when comparing the results. How successfully the models predict values on a track other than the one on which they were trained can be used to characterize this. The Figures represent the anticipated steering angle, brakes, and throttle in this scenario. The evaluation can be done by counting how long the automobile can travel on the second track before coming to a stop, even if it cannot be plotted. There are a number of variables that can affect this, including speed, curves, track conditions like heights and shadows, and others.

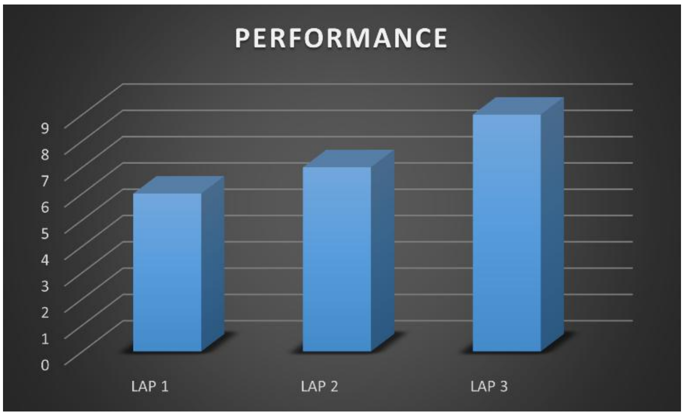


Figure 6: Performance Graph

The above graphical representation is all about the performance measurement out of 10 scale rating for the corresponding laps. According to we can even understand that the lap3 has high performance rate due to the low number of losses in that lap.

By it we can say that Rate of Loss is inversely proportional to the rate of performance.

$$P \propto 1/L$$

Here, L = Rate of Loss & P = Rate of Performance.

5. Conclusion

This article describes how to generate data using the emulator. The simulator's data generation proved to be fairly efficient. It is straightforward to generate a large dataset for testing. It allows for the collecting of real-world data. It necessitates a large number of resources, either in terms of money or personnel. For example, we created 50,000 images in an hour with no further resources. Neural networks can determine the steering angle of a self-driving car using artificial graphics. CNN may collect data from video frames and detect relationships to make predictions.

This article describes how to generate data using the emulator. The simulator's data generation proved to be efficient. It is straightforward to generate a large dataset for testing. It allows for the collecting of real-world data. It necessitates many resources, either in terms of money or personnel. For example, we created 50,000 images in an hour with no further resources. Neural networks can determine the steering angle of a self-driving car using artificial graphics.

This study compares convolutional neural network designs with a limited number of parameters to handle various issues. The final network has three convolutional layers and a total of 26,000 parameters. Overfitting can be prevented, although the dispersion was wide due to a lack of modelled data. In comparison to other common architectures, we have decreased the amount of neural network parameters and RAM utilized by an order of magnitude. These networks are extremely efficient, modified, and sophisticated, according to the results. We have a 78.5 accuracy rating. This is an excellent outcome for computers.

References

[1] *Gospel John, "How Computer Vision Works".*
[2] *"Deep Learning Technology".*
[3] *Kakde, Yogesh, Ssrn-Elsevier, Paul, Aniket, Bothe and Niketan, "Real life implementation of Object Detection and classification Using Deep Learning and Robotic arm.".*
[4] *Sikora, Pavel, Malina, Lukas, Kiac, Martin, Martinasek, Zdenek, Riha, Kamil, Prinosil, Jiri, Leos, Srivastava and Gautam, "Artificial Intelligence-Based Surveillance System for Railway Crossing Traffic," IEEE Sensors Journal, 2020.*
[5] *Talati, Shreya, Vekaria, Darshan, Kumari, Aparna, Tanwar and Sudeep, "An AI-driven Object Segmentation and Speed Control Scheme for Autonomous Moving Platforms," Computer Networks, 2020.*
[6] *Shreyas, Bharadwaj, Skanda, Srinidhi, Ankith and Rajendra, "Self-driving Cars: An Overview of Various Autonomous Driving Systems.," 2020.*
[7] *Kichun Jo, Junsoo Kim, Dongchul Kim, Chulhoon Jang and Myoungho Sunwoo, "Development of Autonomous Car – Part II: A Case Study on the Implementation of an Autonomous Driving System Based on Distributed Architecture," IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS.*