Evolutionary Optimization of Deep Learning Models for Vision based Accident Prevention and Smart Traffic Surveillance Systems

Submitted in partial fulfilment of the requirements of the degree of

Doctor of Philosophy

by

Medipelly Rampavan

(Roll No. 701958)

Under the supervision of

Prof. Earnest Paul Ijjina

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
NATIONAL INSTITUTE OF TECHNOLOGY WARANGAL
WARANGAL - 506004, TELANGANA, INDIA
September, 2024

Dedicated to

Family, Friends & Teachers

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
NATIONAL INSTITUTE OF TECHNOLOGY WARANGAL
WARANGAL - 506004, TELANGANA, INDIA

Thesis Approval Sheet for Ph.D.

This dissertation work entitled **Evolutionary Optimization of Deep Learning Models for Vision based Accident Prevention and Smart Traffic Surveillance Systems** by **Mr. Medipelly Rampavan (Roll No. 701958)** is approved for the degree of **Doctor of Philosophy** at the National Institute of Technology Warangal.

Examiner

Research Supervisor

Prof. Earnest Paul Ijjina
Assistant Professor
Dept. of Computer Science and Engg.
NIT Warangal, India

Chairman

Prof. R. Padmavathy
Professor & Head of the Department
Dept. of Computer Science and Engg.
NIT Warangal, India

Date:

Place: NIT Warangal

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
NATIONAL INSTITUTE OF TECHNOLOGY WARANGAL
WARANGAL - 506004, TELANGANA, INDIA

Certificate

This is to certify that the thesis entitled, **Evolutionary Optimization of Deep Learning Models for Vision based Accident Prevention and Smart Traffic Surveillance Systems**, submitted in partial fulfilment of requirement for the award of the degree of **Doctor of Philosophy** to the National Institute of Technology Warangal, is a bonafide research work done by **Mr. Medipelly Rampavan (Roll No. 701958)** under my supervision. The contents of the thesis have not been submitted elsewhere for the award of any degree.

Research Supervisor

Prof. Earnest Paul Ijjina
Assistant Professor
Dept. of Computer Science and Engg.
NIT Warangal, India

Date:

Place: NIT Warangal

Declaration

This is to certify that the work presented in the thesis entitled, **Evolutionary Optimization of Deep Learning Models for Vision based Accident Prevention and Smart Traffic Surveillance Systems** is a bonafide work done by me under the supervision of Prof. Earnest Paul Ijjina was not submitted elsewhere for the award of any degree.

I declare that this written submission represents my ideas in my own words and where others ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/date/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Medipelly Rampavan
(Roll No. 701958)

Date:

Place: NIT Warangal

Acknowledgments

I would like to express my sincere gratitude and appreciation to my supervisor, Prof. Earnest Paul Ijjina for the invaluable guidance throughout the completion of this work. The continuous support, timely feedback, and constructive discussions have played a pivotal role in helping me achieve my objectives. I am grateful for the ample time the Professor dedicated to review my work and providing insightful suggestions for improvement. The mentorship not only shaped me as a researcher but also as an individual. I have been inspired by the words, actions, and values, which have demonstrated the qualities of a great teacher and a compassionate human being. The Professors unwavering dedication and commitment to excellence have left a profound impact on me. I aspire to embody these remarkable qualities throughout my life.

I would like to express my heartfelt gratitude to all the members of my Doctoral Scrutiny Committee (DSC), namely Prof. R. Padmavathy, Prof. P. RadhaKrishna, Prof. U.S.N. Raju, and Prof. M. Shashi. Their valuable comments and suggestions during the oral presentations have greatly enriched my research work. I am truly fortunate to have had the opportunity to attend lectures by esteemed professors namely Prof. R. B. V. Subramaanyam, Prof. K. V. Kadambari, Prof. E. Suresh Babu, Prof. Earnest Paul Ijjina, and Prof. Madhavi Reddy Kesari. Their knowledge and expertise have been instrumental in broadening my understanding in the field.

I immensely thankful to Prof. R. B. V. Subramaanyam, Prof. P. Radha Krishna, Prof. Ravichandra Sadam, and Prof. R. Padmavathy Heads of the Department of Computer Science and Engineering (CSE) and chairmans of DSC, during my tenure for providing adequate facilities in the department to carry out the oral presentations. I wish to express my thanks to all the esteemed faculty members of the Department of CSE at NIT Warangal. I would also like to extend my heartfelt gratitude to Prof. N.V. Ramana Rao and Prof. Bidyadhar Subudhi, the Directors of NIT Warangal, for their unwavering support and encouragement throughout my research endeavor. I am truly fortunate to have been part of such a remarkable institution and to have received support from all these individuals who have contributed to my academic and personal development.

I would like to express my heartfelt gratitude to my co-research scholars and friends for their invaluable support and selfless cooperation throughout my Ph.D. journey. Their presence and assistance have been truly remarkable, and I am deeply thankful for their unwavering support and readiness to help me at any time. In particular, I would like to express a special thanks to Ch. Rajesh for his unconditional love, support, and constant presence in both the joyful and challenging moments of my personal life and research work. Their friendship has been a source of strength and motivation, and I am forever grateful to have such amazing individuals by my side. Without the support and camaraderie of these remarkable individuals, my Ph.D. journey would have been much more challenging. I am truly blessed to have such incredible friends in my life, and I will cherish their friendship and support forever.

I want to thank my wife Mrs. Mounika and my brother Mr. Anjaneyulu who understood, questioned, and complimented my journey in such a way that it will always remain cherishable. Finally, I would like to thank my parents for being a constant source of motivation and support.

Medipelly Rampavan
September, 2024

Abstract

The increasing use of automobiles as the primary mode of transportation has made a critical focus on safety measures to address the growing traffic-related problems. This thesis presents an exploration of four objectives, contributing to the advancement of vision based accident prevention and the enhancement of vision based smart traffic surveillance through the use of evolutionary optimization techniques and deep learning models.

In the first objective, a Genetic Algorithm (GA) based Neural Architecture Search (NAS) is proposed for constructing a Mask R-CNN based object detection model specifically designed for vehicle brake light detection. By automating the design process, the approach addresses the limitations of manually designed Deep Neural Network (DNN) architectures, leading to superior performance in detecting brake light status for both two-wheeler and four-wheeler vehicles.

The second objective aims to expand the scope of the first objective for vehicle brake light detection task. This approach involves a NAS with an expanded search space encompassing the backbone architecture parameters and training parameters. We employ a modified Differential Evolution (DE) algorithm for the search strategy. This algorithm incorporates evaluation correction based selection for mutation and species protection based selection, aiming to identify an optimal DNN model. The experiments on two-wheeler and four-wheeler vehicle datasets demonstrate the effectiveness of the proposed method. Further, cross-dataset evaluation and experiments on real-world traffic videos demonstrate the proposed approach's generalization capability.

The third objective introduces NAS based approach using a DNN model, designed for vehicle re-IDentification (reID) task useful for smart surveillance systems. The Grasshopper Optimization Algorithm (GOA) is employed to search for the optimal DNN model, considering both architecture parameters and hyperparameters related to the reID task. The experiments on two vehicle reID datasets demonstrate the effectiveness of the proposed method in automatically discovering optimal models for vehicle reID task.

Finally, the fourth objective addresses driver distraction detection task for accident prevention. Recognizing the limitations of manually developed DNN architectures for this task, we employ NAS with an improved GA to design a one-stage object detection model. The proposed approach explores YOLO backbone architecture parameters and training parameters. Experimental results showcase the proposed approach's superiority compared to existing models on driver distraction detection datasets, emphasizing its efficacy in improving driver safety.

In summary, this thesis offers a comprehensive exploration of evolutionary optimization techniques applied to deep learning models for object detection and reID tasks, contributing to vision based accident prevention and smart traffic surveillance systems. The integration of NAS and evolutionary algorithms across these works demonstrates their effectiveness in automating the design process and improving the efficiency of deep learning models for various tasks.

Contents

List of Figures

List of Tables

List of Algorithms

Abbreviations

ADAS	Advanced Driving Assistance Systems
ASS	Attention Search Space
BPReID	Bike Person re-IDentification
CNN	Convolutional Neural Network
DAG	Directed Acyclic Graph
DDDI	Distracted Driver Detection Image
DDCV	Distracted Driving Computer Vision
DE	Differential Evolution
DNN	Deep Neural Network
DPM	Deformable Parts Model
EA	Evolutionary Algorithms
ECS	Evaluation Correction based Selection
ECSM	Evaluation Correction based Selection for Mutation
E-DE	Evaluation correction based Differential Evolution
GA	Genetic Algorithm
GOA	Grasshopper Optimization Algorithm
IoU	Intersection over Union
MoRe	Motorcycle re-IDentification
MSE	Mean Squared Error
mAP	mean Average Precision

MSML	Margin Sample Mining Loss
NAS	Neural Architecture Search
NITW-MBS	NITW Motorcycle Brake Light Status
R-CNN	Region-Based Convolutional Neural Network
RL	Reinforcement Learning
reID	re-IDentification
SGD	Stochastic Gradient Descent
SPS	Species Protection based Selection
SPPS	Species Protection based Population Selection
SVM	Support Vector Machines
TOOD	Task-aligned One-stage Object Detection
WHO	World Health Organization
YOLO	You Only Look Once

Chapter 1

Introduction

The increase in the number of vehicles is leading to a rise in both traffic accidents and vehicle related crimes, posing significant threats to public safety. The World Health Organization (WHO) report mentions that approximately 1.35 million lives are lost each year due to road traffic accidents [1]. To prevent these accidents and vehicle related crimes, industry and academia are focused on developing safety systems as well as smart traffic monitoring systems.

Vehicles with essential safety features can avoid collisions and minimize the risk of catastrophic injuries. Pre-collision sensing and collision avoidance are areas of interest among automobile manufacturers in order to achieve the ultimate goal of accident prevention. Collision avoidance systems are classified as either passive or active safety systems. Passive safety systems use techniques like maintaining high production safety standards and enforcing strict traffic restrictions, such as the use of helmets and seat belts, to ensure a safe driving environment. Active safety systems can use techniques such as a collision avoidance warning system to alert and assist the driver in preventing accidents before they happen. Some existing studies rely on active sensors like radar technology [2], beamforming techniques [3, 4], infrared sensors, etc. With the exponential growth in processing power and the advancement of computer vision techniques, vision based sensing technologies can be used for developing cost-efficient solutions. Vision based techniques have a significant advantage over non-vision based techniques. New and efficient functionality can be incorporated into a solution by only modifying the software without changing the hardware.

Vision based brake light detection and vision based driver distraction recognition are crucial for avoiding collision. The signals of vehicles, in particular the brake lights, are used to effectively communicate a possible reduction of a vehicle's speed to the vehicles following them. As a result, real-time detection and recognition of brake lights can assist in reducing collisions in traffic. Driver distraction detection systems are designed to monitor the driver's behavior and alertness and detect signs of distraction or drowsiness, which can provide warnings to the driver, thereby preventing a possible collision. Vision based smart traffic surveillance is crucial for urban areas, contributing significantly to traffic management, road safety, emergency response, enforcement of regulations, informed decision-making, environmental sustainability, economic prosperity, and public convenience. Object re-IDentification (reID) is a vital component of smart traffic surveillance systems. It enables continuous vehicle tracking, swift incident detection and response, effective traffic law enforcement, investigation and forensics support, traffic flow analysis insights, enhanced public safety and security, and data-driven decision-making for urban management.

The approaches used in vision based accident prevention systems and smart traffic surveillance systems can be categorized into two, namely, traditional and deep learning based methods. The initial work in this domain used traditional techniques, such as color based segmentation, statistical machine learning, employing Support Vector Machines (SVM) [5], AdaBoost [6], etc., for classification based on color, shape, and texture features. In contrast, deep learning based approaches, particularly Convolutional Neural Networks (CNNs) [7], have demonstrated notable efficiency in visual recognition tasks. The CNNs excel in learning hierarchical semantic features and have made significant strides in the detection of vehicle tail light when compared to the traditional approaches. Deep learning methods were found to have better generalization capability for tasks related to object recognition and classification, which can be used in approaches for accident prevention and smart traffic surveillance.

In computer vision, both driver distraction detection and brake light detection are classified as object detection tasks. Object detection is a computer vision task that involves identifying and locating objects of interest within an image or video frame. It goes beyond image classification, which simply classifies the entire image, by providing precise localization of objects through bounding boxes and identifying multiple objects within the same image. In the case of driver distraction detection, the system analyzes video feeds from cameras installed in vehicles to identify instances where the driver's attention is diverted from the road, such as texting, eating, or interacting with electronic devices. This involves detecting and localizing the driver's face and hand regions within the image, often using facial recognition or hand tracking techniques. Similarly, brake light detection focuses on identifying and locating brake lights on vehicles within the scene. This information is crucial for tasks such as monitoring traffic flow, detecting potential collisions, or assessing driver behavior. Both tasks utilize object detection algorithms, such as CNNs, to analyze visual data and extract relevant features for identifying and localizing the objects of interest within the image or video frame.

The one-stage and two-stage models are the most commonly used object detection approaches. On one hand, we have one-stage detectors like the YOLO family [8, 9, 10] and SSD [11], which use a single neural network to learn the bounding box coordinates and the probability of their labels for an input image, treating object recognition as a simple regression task. On the other hand, two-stage detectors, like the R-CNN Family (R-CNN [12], Fast R-CNN [13], Faster R-CNN [14], and Mask R-CNN [15]) use the first stage to extract regions of interest using a Region Proposal Network (RPN) and the second stage to predict bounding boxes of objects along with their class labels. Fig. 1.1 shows the framework of the object detection model.

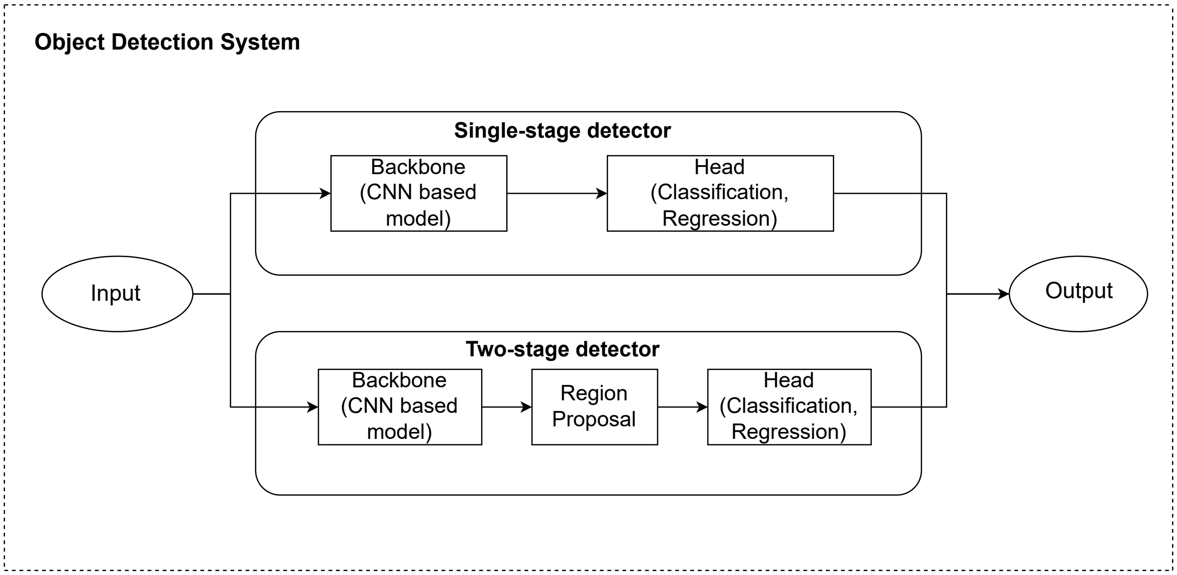

Object Detection System
Single-stage detector
Backbone (CNN based model)
Head (Classification, Regression)
Input
Output
Two-stage detector
Backbone (CNN based model)
Region Proposal
Head (Classification, Regression)

Figure 1.1: Object detection framework

Vision based smart traffic surveillance involves the use of cameras and computer vision algorithms to monitor and analyse traffic situations in real-time. Tracking vehicles over time involves re-identifying the vehicles across locations which is one of the important task of smart traffic surveillance. Object reID task is the process of matching the same object across multiple cameras. An approach for object reID is crucial for identifying objects like people, cars, motorcycles, etc., in video surveillance systems. This is an active area of research in both industry and academia due to the ever-growing population and need for smart surveillance, public safety, and traffic management. Object reID [16] based on CNN typically involves two main phases: feature learning and re-identification. During the feature learning phase, images are used to train a classification network to extract feature vectors. In the re-identification phase, test images are split into gallery images of one camera and probe images of another camera, and these are then fed into the trained classification network to extract their respective feature vectors. A distance measure is used to match objects and identify the same object across different locations. Fig. 1.2 shows this reID process.

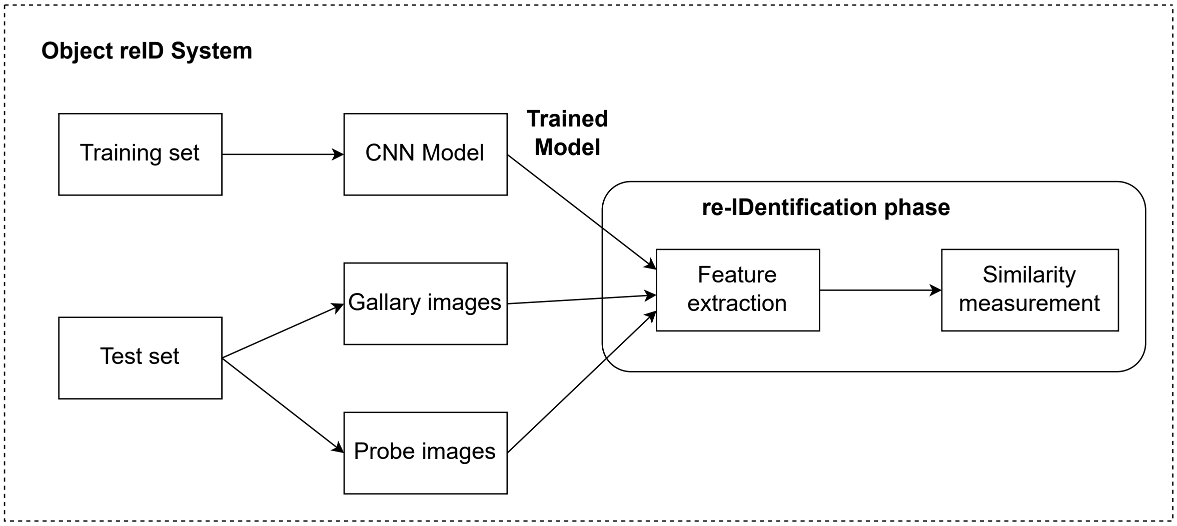


Figure 1.2: Object reID framework

However, the existing approach of using a manually designed Deep Neural Network (DNN) architecture requires expertise and empirical experimentation. Recently, Neural Architecture Search (NAS) based approaches have achieved great success in designing DNN architecture automatically for tasks related to image classification [17], object detection [18], and segmentation [19].

To design a NAS based system, the search space needs to be defined and an efficient search strategy should be used. The first step involves defining the "**search space**" of the DNN architecture. The hyper-parameters of the DNN architecture, such as the type of layer, number of units in each layer, number of kernels, kernel size, etc, are generally considered in the NAS search space. In addition, other parameters, such as the loss function, optimization algorithm, activation function, etc., can also be included in the search space. The inclusion of more parameters in the NAS search space may lead to the identification of a better model at the expense of computational complexity. Finding the best DNN architecture, even for a smaller search space, is a challenging task without a search strategy. As a result, NAS approaches require an efficient "**search strategy**" to identify the optimal values of the parameters in the search space. The commonly used search strategies for finding the best DNN architecture are based on Evolutionary Algorithms (EA) [20], Reinforcement Learning (RL) [21], and Gradient-based optimizers [22]. The candidate DNN models explored by NAS in a generation are evaluated using an objective function to select the optimal DNN models for the next generation. The main goal of this thesis work is to develop an automated NAS based approach for identifying DNN models through optimization by evolutionary algorithms for tasks related to vision based accident prevention and smart traffic surveillance systems. The overall framework of the proposed thesis is given in Fig. 1.3.

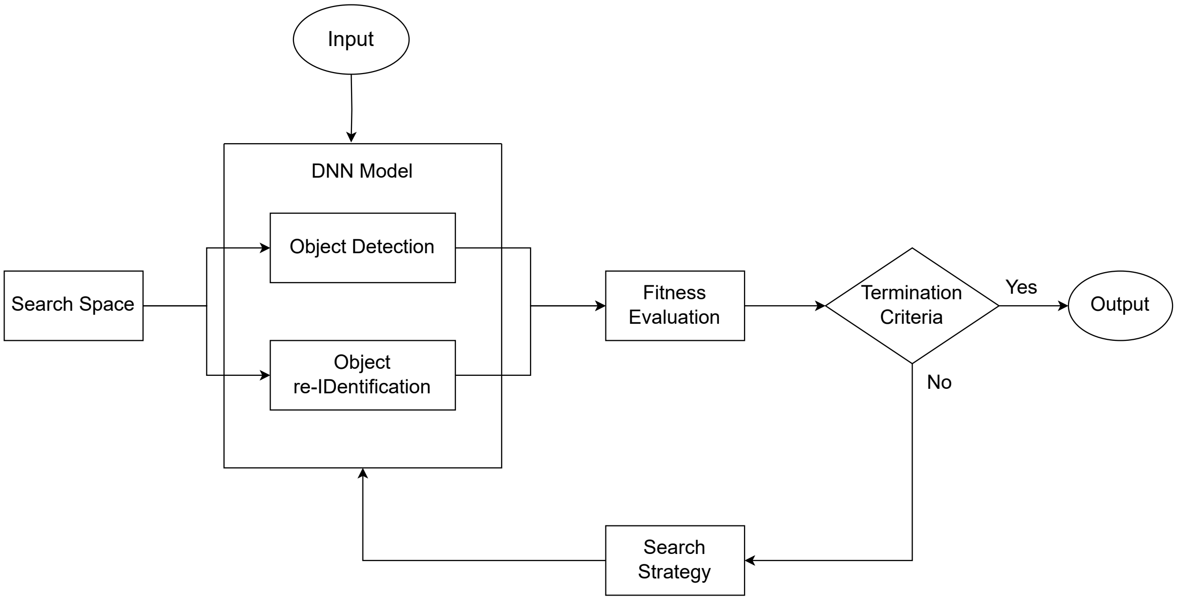


Figure 1.3: Overall proposed NAS framework

1.1 Motivation and objectives

In this work, we aim to identify optimal DNN models to address tasks related to accident prevention and traffic surveillance by using evolutionary algorithms for optimization. Detecting vehicle brake lights is essential for preventing collisions, but the task is challenging due to occlusion, varying light conditions, and variations in vehicle make and model. Most existing research focuses on car brake light detection, but real-world systems must handle a broader range of vehicles, including two-wheelers. Two-wheelers exhibit diverse brake light shapes, sizes, and locations, depending on the make and model, further complicating the detection process. Traditional methods [23, 24], which primarily use color-based and

Harr-like features with machine learning models like SVM, have shown limited effectiveness in real-world conditions due to their reliance on handcrafted features. In contrast, deep learning approaches, particularly CNN based models [25, 26, 27] have demonstrated improved performance for brake light detection. However, these manually designed models face challenges in adapting to real-world environmental conditions and variations.

In the context of driver distraction detection, research has evolved from traditional machine learning techniques [28, 29] that depend on manual feature extraction to more advanced deep learning models such as CNNs and Vision Transformers [30, 31]. Recent innovations, including lightweight models [32] and attention mechanisms [33], have significantly improved the accuracy and efficiency of these systems. Moreover, using self-supervised learning [34] and federated learning [35] frameworks has addressed issues like the need for large labeled datasets and privacy concerns. Despite the progress, manually designed models still have limitations, particularly in adapting to real-world situations.

Despite significant progress in reID task, most research has focused on person reID [36, 37, 38] and four-wheeler reID [39, 40, 41]. These tasks differ significantly due to differences in object characteristics, camera angles, and movement patterns. Motorcycles, which are often overlooked in reID research, present unique challenges because they include both a rider and a vehicle, each with distinct characteristics. Motorcycle riders frequently wear helmets that obscure their faces, and their clothing can hide additional characteristics. Furthermore, motorcycles are smaller and often get occluded, making them more difficult to spot in surveillance cameras. Although some studies have attempted motorcycles reID using traditional [42] and deep learning methods [43, 44, 45], further research is required to develop robust motorcycle reID techniques that can perform well in diverse real-world scenarios.

Given the limitations of traditional methods and manually designed DNN models, we propose an Evolutionary NAS approach to automatically generate optimal DNN models for detecting brake lights status, driver distractions and vehicle re-identification tasks. The exploration mechanism in NAS can be used to facilitate the automatic exploration and identification of optimal parameters for a DNN model. Using evolutionary algorithms to explore the search space offers an effective way to identify models that perform well in diverse conditions. Most of the existing literature used NAS to identify the optimal DNN architecture only (like the type of blocks, kernel size, number of layers, etc.), while it can also be used to identify the optimal training parameters.

The following objectives are formulated in this thesis with respect to the above mentioned research gaps:

1. The first objective of this research is to design an evolutionary based optimization of the DNN model for detecting brake light status on two-wheeler and four-wheeler vehicles for accident prevention. This involves designing a NAS search space for the DNN model to include backbone architecture parameters and training parameters. The search space exploration to identify the optimal model is done using a genetic algorithm. The genetic algorithm will optimize the parameters of the DNN model, thereby enhancing its ability to identify the brake light status accurately. We also proposed a new dataset, NITW Motorcycle Brake Light Status (NITW-MBS), for detecting brake lights in two-wheelers in this objective due to the unavailability of publicly available datasets for this task.

2. The second objective extends the scope of the search space to design a DNN for detecting the brake light status of multiple types of vehicles. The methodology considers a NAS search space with the parameters of the backbone architecture and training parameters. A modified differential evolution algorithm with evaluation correction based selection for mutation and species protection based selection is used to identify an optimal DNN model.

3. The third objective focuses on developing an evolutionary based optimization of the DNN model for vehicle re-identification in smart traffic surveillance systems. The search space is designed to include the parameters of the DNN backbone architecture and the training parameters. The search space is explored using the grasshopper optimization technique to identify an optimal DNN model for vehicle re-identification tasks.

4. The fourth objective is to develop an evolutionary based optimization of the DNN model to detect driver distractions that can be used in an accident prevention system.

The NAS search space is designed to include the parameters of the model backbone architecture and the DNN model's training parameters. The search strategy involves using a genetic algorithm with evaluation correction based selection and species protection based selection to identify an optimal DNN model. This system can be used to alert drivers when they are distracted to prevent accidents.

1.2 Overview of the contributions of the thesis

In this section, an overview of the chapter-wise contributions of this thesis is presented. Each subsection presents a summary of the contributions of the corresponding chapter.

1.2.1 Genetic algorithm based optimization of deep neural networks for vision based vehicle brake light status detection for accident prevention systems

A break light detection system is necessary for vehicles to avoid collisions. In the automobile industry, collision sensing and accident prevention techniques are an active area of research. However, the task of vehicle brake light detection in computer vision is a challenging task due to occlusion, variation in capturing conditions, and the smaller size of brake lights. Thus, it is crucial to detect brake light status to avoid collisions and to ensure safety during driving. To the best of our knowledge, existing research primarily focuses on detecting brake light status in four-wheeler vehicles. However, in the real world, a vehicle may come across a wide range of vehicles, such as motorcycles, four-wheelers, buses, trucks, etc. Motivated to replicate such a scenario, as there was no publicly available dataset for identifying the status of motorcycle brake lights, a new dataset is proposed in this work. We proposed a NAS based DNN technique with a genetic algorithm to construct a DNN model by searching for better backbone and optimal training parameters.
The major contributions of this work are listed below:

- A new dataset (NITW-MBS) for detecting the status of two-wheeler brake lights status is proposed. In this dataset, rear-view tail light images of different types of motorcycles are considered.

- For the detection of brake light status, a DNN based object detection model using Mask R-CNN is considered in this work. A Genetic Algorithm (GA) based NAS approach is used to find the optimal backbone architecture and training parameters.
- Evaluation of the proposed approach against the existing brake light status detection approaches as well as existing state-of-the-art object detection models for both two-wheeler and four-wheeler vehicles is conducted.

Proposed method

According to the vision based object detection literature, single-stage detectors excel at identifying large objects but struggle to recognize smaller ones. Given the small size of the brake light, we chose a two-stage object detection model based on Mask R-CNN in this work. We formulated NAS to find the optimal DNN model for object detection tasks. In this context, we use mean Average Precision (mAP) as the fitness function to maximize validation accuracy.

To enable a comprehensive exploration of DNN architecture and training parameters, a search space is designed with essential components such as *type of block*, loss functions (*class loss*, *BBox loss*), *activation* function, and *optimizer*. For the *activation* function ReLU, Mish are considered, for optimizer SGD, Adam are included in the search space. The study considers cross-entropy loss, Focal loss for *class loss* parameters. We also consider L1 loss, MSE loss for BBox loss parameter.

In this work, GA is used to automatically identify the optimal neural network architecture. GA is based on Darwin's natural evolution hypothesis, which generates a new population with better average fitness than the population in the current generation. In binary encoded genetic algorithms, the GA chromosome is represented by a binary bit string. A chromosome comprises genes that capture the individual's genetic characteristics, thereby representing a solution. The three primary operations of genetic algorithms are selection, crossover, and mutation. The "selection" operation aims to identify individuals, known as parents, with a higher fitness value so that the resulting offspring may inherit the better characteristics of the parents. In this work, tournament selection is used for selecting parents. The diversity in the population is achieved through crossover and mutation. In the "Crossover" operation, a random crossover site is identified, and the bit strings of the selected parents are interchanged to form new offspring. The "Mutation" operation inserts random genes into offspring to achieve population diversity and prevent early convergence.

The object detection model identified by the proposed approach achieved a mean accuracy of 97.14% on the proposed two-wheeler (NITW-MBS) dataset and 89.44% on the four-wheeler (CaltechGraz) dataset, respectively. The proposed model obtained better results than the existing approaches for both two-wheeler and four-wheeler vehicle brake light status detection. This indicates that the proposed approach can explore the search space to identify the optimal object detection model for the brake light detection task.

1.2.2 Differential evolution based optimization of deep neural networks for vision based vehicle brake light status detection for accident prevention systems

This work aims to design a vehicle brake light status detection system that is effective for multiple types of vehicles by considering a strong search space. Existing manually designed DNN architectures for brake light status detection face challenges in accurately estimating the brake light status in diverse real-world conditions. This is because brake lights come in various shapes, color shades, and brightness levels, which differ between cars, motorcycles, trucks, buses, and other vehicle types. In addition, the position of the tail light can vary even within the same category of vehicle. The wear and tear of the vehicle further adds to its diversity. The current solutions can't handle these challenging conditions effectively, resulting in only a few reliable brake light status detection methods. Recently, NAS based approaches have achieved great success in automatically designing DNN models for image classification [17], object detection [18], and segmentation [19] tasks. This is the motivation of the proposed NAS based approach that uses Evaluation correction based Differential Evolution (E-DE) for exploring the search space to find an optimal brake light status detection task.

The major contributions of this work are listed below:

- A modified DE based NAS approach is proposed to optimize the two-stage object detection framework for two-wheeler and four-wheeler vehicle brake light status detection tasks.
- The search space is designed to include the parameters of DNN backbone architecture and training parameters for brake light status detection.
- An Evaluation correction based selection for mutation and species protection based selection is used in the modified DE algorithm to find the optimal DNN network.

Proposed method

In this work, we designed a two-stage Mask R-CNN based object detection model for brake light status detection. A search space is designed to include the parameters of backbone architecture and training parameters for designing a system for brake light status detection.

The proposed search space explores four kinds of blocks: Resnet block [46], ReneXt block [47], ReneSt block[48] and Swin transformer block [49]. Apart from parameters to search for the backbone architecture, this work includes the parameters related to training like *activation function*, *optimizer*, *box loss* and *class loss* in the search space. For the *activation function* parameter {ReLU, GELU, CELU, Mish} are considered, for *optimizer* {SGD, Adam, AdamW} are used, for *box loss* {MSE loss, L1 loss, Smooth L1 loss} is used, and finally for *class loss* {Cross Entropy loss, Focal loss} are used.

A modified version of the DE algorithm named E-DE is used for the search strategy. Traditional mutation strategies use the best vector to choose parents for generating the donor vector. Estimating performance based on validation mAP only may lead to a better network but may not be efficient in terms of computation. Hence, we introduce the use of an Evaluation Correction based Selection strategy to choose individuals for the Mutation operation (ECSM). In the evolutionary process, maintaining the diversity in the population of network architectures is crucial for improving the algorithm's overall performance. To address this issue, this work explores the use of Species Protection based environmental Selection (SPS) operation.

The optimal DNN models discovered using the proposed approach achieved mean accuracy of 89.73 % and 88.90 % on the four-wheeler vehicle datasets CaltechGraz [50, 51] and UC Merced Vehicle Rear Signal [52], respectively. We also evaluated this approach on the proposed two-wheeler NITW-MBS dataset, for which the proposed approach achieved an accuracy of 97.97 %. The comparative study with other existing manually designed DNN approaches and NAS based object detectors on these datasets indicates the effectiveness of the proposed approach. In addition, a comparison of the proposed approach with basic DE suggests the effectiveness of the modified DE approach. Finally, we have tested the proposed method on real-life video sequences to evaluate the effectiveness of the proposed vision based approach for detecting brake light status.

1.2.3 Grasshopper algorithm based optimization of deep neural networks for vision based vehicle re-identification for smart traffic surveillance systems

Object re-identification is an important visual recognition task in computer vision, with applications in security, surveillance, traffic monitoring, retail analytics, robotics, sports analytics, and more. Object reID is a challenging task due to the variations in illumination, occlusions, appearance, and other factors, making it difficult to recognize and track objects/persons across various cameras. Most research on re-identification focuses on identifying persons. This study aims to address vehicle re-identification tasks by developing a vehicle re-identification system based on appearance features crucial for recognizing vehicles. With the existing license plate recognition systems, it is difficult to identify vehicles from different views. Therefore, we have proposed a NAS based DNN optimization approach for vision based vehicle re-identification tasks.

The major contributions of this work are listed below:

- A NAS based DNN optimization approach using Grasshopper Optimization Algorithm (GOA) is proposed for re-identification task.

- A search space is designed to include the parameters of the backbone architecture as well as the training hyperparameters to compute an object re-identification model.
- Experiments are conducted to search and train different architectures for reID without pretraining on two publicly available reID datasets which outperformed the existing approaches.

Proposed method

The proposed object re-identification approach comprises two phases: feature learning and re-identification. In the feature learning phase, a classification network is trained on images to extract feature vectors. The re-identification phase involves dividing test images into gallery images from one camera and probe images from another. These images are then processed through the trained classification network to obtain the feature vectors. Then, a distance measure facilitates the re-identification of the same object across different locations using probe images and gallery images. In this work, we have modeled this entire process as a NAS optimization task to design an optimal deep neural network model for vehicle re-identification.

In this work, we propose the search space with four kinds of blocks: Resnet block [7], EfficientnetV2 block [53], Regnet block [54, 55] and Densenet block [56]. In addition to the parameters of the backbone, we have also included the parameters associated with training such as *Size of input* , *Pooling operation*, *Activation function*, *Optimization algorithm*, *Loss function* and *Distance metric* in the search space.

The Rank-1 accuracy is a key metric for evaluating the performance of object re-identification. The rank-1 accuracy measures the percentage of correctly identified probe images, where the matched highly ranked gallery image is correct. Therfore, in this work, we use rank-1 accuracy as the fitness function.

GOA was first introduced by Saremi et al. in [57]. In the literature, it is found to be effective in solving various optimization problems, including medical image segmentation [58], image enhancement [59], image fusion and feature selection [60, 61]. Its simplicity, efficiency, and robustness make it a popular optimization technique. Therefore, this approach uses GOA as the NAS search strategy for finding an optimum DNN model for the reID task.

The performance of the proposed approach is compared with the existing approaches for vehicle re-identification on the MoRe and BPReID datasets. The experimental results suggest that the proposed approach outperforms the existing methods, indicating its effectiveness for the vehicle re-identification task.

1.2.4 An improved genetic algorithm based optimization of deep neural networks for vision based driver distraction detection for accident prevention systems

The manually designed DNN architectures for driver distraction detection may be ineffective for predicting driver behavior in real-world scenarios with several types of driver distractions such as texting, eyes closed, yawning, talking on the phone, etc. Further, accurate estimation of driver behavior under real-world driving conditions depends on the localization of the driver's facial and hand actions. Current approaches struggle to deal with these practical, diverse driving conditions. In this work, we propose an evolutionary NAS based approach to automatically design a DNN model for detecting driver distraction using an improved GA as the NAS search strategy.

The major contributions of this work are listed below:

- A NAS based approach with improved GA optimization of a single-stage YOLO object detection framework is proposed to localize and classify driver distractions.
- A search space is designed to include the parameters of the backbone and the training parameters of the YOLO object detection network.
- An evaluation correction based selection and species protection based environment selection are used in the Genetic Algorithm to find an optimal DNN model with fewer parameters.

Proposed method

In this work, a one-stage YOLO architecture is considered the base model for NAS based optimization of the DNN model. The search space is designed to include parameters of backbone architecture and training parameters. In the search for backbone architecture, we consider four types of blocks: CSPDarknet53 block [62], RepVGG block[63], CSP-NeXt blocks [64], and CSPResNet block [65]. We consider that the *depth of blocks* and *width of blocks* in each stage of the backbone architecture vary from 0 to 2. In addition to the search for the backbone architecture, the proposed search space also includes training parameters such as the *box loss*, *class loss*, *activation function*, and *optimizer*. For *box loss*, we explored {IoU loss, GIoU loss, SIoU loss, CIoU loss}, and for *class loss*, we considered {Cross Entropy loss, Focal loss, VariFocal loss, QualityFocal loss}. For the *activation function*, we considered {ReLU, GELU, Swish, SiLU}, and finally, for the *optimizer*, the alternatives are {SGD, NAdam, Adamax, AdamW}.

A modified version of the GA algorithm is used as the search strategy. In the standard GA, selection strategies like tournament selection, roulette wheel selection, etc., are used to select parents that are used to generate offspring. However, performance estimation based on only validation mAP may lead to a better network but is inefficient in terms of computation. Hence, we have introduced the Evaluation Correction based Selection (ECS) strategy, which is considered to choose individuals in the modified selection operation. In the evolutionary process, maintaining diversity in the population of network architectures is crucial for improving the overall performance of the NAS based approach. To accomplish this task, this work utilizes Species Protection based environmental Selection operation (SPS).

The obtained DNN model of the proposed approach achieved a mean accuracy of 87.14% on the Distracted Driver Detection Image (DDDI) [66] dataset and 88.87% on the Distracted Driving Computer Vision project (DDCV) [67] dataset. On both datasets, the generated DNN model outperformed the existing methods for detecting distracted drivers, highlighting its effectiveness in addressing the driver distraction detection task.

1.3 Organization of the thesis

This thesis mainly focuses on developing a NAS based DNN optimization approach using evolutionary algorithms. This approach is used to design models for visual recognition tasks such as brake light status detection, detection of distracted drivers to prevent accidents, and vehicle re-identification for smart traffic surveillance. The remainder of the thesis consists of six chapters, including related work, four contributions of the thesis and a conclusion chapter. The content of each of these chapters is described briefly below:

Chapter 2: Related Work

In this chapter, a survey of the recent work on vision based approaches related to accident prevention mechanisms and tasks related to smart traffic surveillance is provided. In particular, it focuses on the works related to the detection of brake light status, vehicle re-identification, and driver distraction detection tasks.

Chapter 3: Genetic Algorithm based Optimization of Deep Neural Networks for Vehicle Brake Light Detection

This chapter covers the optimization of a deep neural network model for the task of vehicle brake light status detection. A new dataset (NITW-MBS) is introduced to evaluate this work for two-wheeler vehicles, and a NAS based DNN optimization approach is presented. The genetic algorithm is used as the NAS search strategy, exploring a search space encompassing both DNN backbone architecture and training parameters. This approach aims to design an optimal two-stage Mask R-CNN based object detection model for two-wheeler and four-wheeler vehicle brake light status detection tasks.

Chapter 4: Differential Evolution based Optimization of Deep Neural Networks for Vehicle Brake Light Detection

This chapter focuses on optimizing DNN models to detect the status of brake lights for multiple types of vehicles simultaneously. The search space is designed to enable the detection of brake light status across different types of vehicles. A modified differential evolution strategy is proposed as the NAS search strategy to find the optimal two-stage Mask R-CNN based object detection model. The experimental study suggests the efficacy of the modified DE based NAS approach for finding an optimal DNN model for detecting the vehicle brake light status.

Chapter 5: Grasshopper Optimization based Deep Neural Networks for Vehicle Re-identification

In this chapter, we propose a NAS based optimization of the DNN model for vehicle re-identification task. The grasshopper optimization algorithm is used as a NAS search strategy. The search space is designed to include parameters of DNN architecture and training parameters for the re-identification task. The proposed approach is evaluated on two publicly available datasets to showcase the effectiveness of the proposed NAS based optimization of deep learning models for vehicle re-identification task.

Chapter 6: Improved Genetic Algorithm based Optimization of Deep Neural Networks for Driver Distraction Detection

This chapter explores the optimization of a DNN model for the task of detecting distracted drivers. A modified GA is used as the NAS search strategy. The search space is designed to cover parameters related to backbone architecture and training parameters for a one-stage object detection model. The proposed approach is evaluated on two publicly available datasets to demonstrate the effectiveness of the proposed NAS based optimization of DNN model for detecting driver distraction.

Chapter 7: Conclusion and Future work

This chapter presents the conclusion of the thesis. It also outlines potential future research directions based on the findings presented in this thesis.

Chapter 2

Related Work

A comprehensive literature review of different tasks is presented in this chapter. The literature on vehicle brake light detection is discussed in Section 2.1, while Section 2.2 covers the literature on vehicle re-identification. Section 2.3 focuses on the literature related to driver distraction detection. Finally, a summary is provided in Section 2.4.

2.1 Vehicle brake light detection

This section presents the literature related to vehicle brake light detection and determining the status of the vehicle brake light signal to prevent rear-end collisions. The vision based methods that are used for brake light status classification can be categorized into traditional and deep learning based methods. During the early attempts, researchers predominantly employed traditional techniques, such as segmentation based on colour information, shape, brightness, and other features, along with statistical machine-learning models. Colour based techniques often utilize morphology and colour/intensity thresholds to extract relevant features. Among the machine learning classifiers, SVM [5] and AdaBoost [6] are the two most popular algorithms used to classify tail lights based on colour features.

Chen et al. [68] proposed a vision based technique for brake light detection using a tail light symmetry verification to extract the vehicles, and a combination of radial symmetry traits and luminance was used to determine the brightness of each pixel inside a vehicle's bounding box, to be a Red component. Missed targets are addressed in a detection refining process based on temporal information. Jen et al. [23] used a Harr-like based classifier to detect vehicles and their paired tail lights, then they used a kernelized correlation filter to track the detected RoI regions and finally used colour and brightness analysis of the tail light area to identify the status of the tail light. The tail light detection system, developed by Almagambetov et al. [69] can detect and track a vehicle's tail lights as well as predict the status of tail light signals. In this work, in order to find potential tail regions, the approach identifies Red or White colours first. Then, the symmetry test and Kalman filtering were used to find a sequence of sequential matching tail light pairings. Cui et al. [70] proposed a hierarchical framework with three stages: first, vehicle bounding box detection using the Deformable Parts Model (DPM); second, tail light candidate extraction using Hue-Saturation-Value (HSV) colour space, and the two most significant clusters are then extracted using the OPTICS method; and finally, turn and brake light status is estimated using the brightness values of the tail light region. In order to extract tail light features influenced by external lighting and weather, Weis et al.[71] developed pixel values of interest for tail light areas by processing the input video stream to extract colour, shape, and other information that is closely related to the object of interest. Arun et al. [24] employed HSV colour space to detect brake and turn signals, and an optical flow technique was used to detect moving cars. SVM was then used to classify the brake and the turn signals.

However, traditional methods relying on manually determined features may not be effective in real-world conditions with varying illumination conditions and cluttered environments. In contrast, deep learning based approaches, particularly CNNs, have demonstrated notable efficiency in addressing these challenges. CNNs excel in learning hierarchical semantic concepts and have made significant strides in the detection of vehicle tail light when compared to the traditional approaches. Zhong et al. [72] used a Faster R-CNN based model to identify the vehicles, then used colour information and morphological operations to recognize the brake light regions. An SVM classifier was used to classify the status of the tail lights Vancea et al. [73] employed two strategies for tail light detection. The first strategy uses the L*a*b* colour space and explicit thresholds to detect Red colour regions. The second strategy uses a deep learning model based on FCN-VGG16 to recognize vehicles first and then uses the detected vehicles to segment candidate tail lights. The identified tail lights are then tracked using a Kalman filter. Wang et al. [74] introduced a brake light detection system that uses a modified HoG detector to extract the vehicle's rear light signal. To detect vehicles, CNN based AlexNet model was used. Later, they extended this work to HDR cameras instead of standard colour cameras. Vancea et al. [75] developed a CNN based model to recognize the tail light of vehicles. A Faster RCNN was utilized to detect vehicles, while a sub-network was used to segment tail light pixels and classify their signal status. Rapson et al. [76] compared the performance of various YOLO models. The YOLOv3 model recognizes vehicles in low-resolution images. The tiny YOLO can recognize vehicle tail lights in high-resolution images from the vehicles. Frossard et al. [77] proposed a deep learning model to identify a vehicle's brake light status directly. In this work, a VGG16 was used to predict an attention mask and extract spatial data, and a CNN-LSTM was used to extract temporal information. Nava et al. [78] proposed a model for detecting and classifying brake lights, which was primarily designed for daytime collision warning systems. They used YOLO and a lane detection algorithm in the first phase to detect the vehicles. In the second phase, they used SVM to recognize the status of the brake light. Li et al. [79] developed a one-stage model using modified YOLOv3. They used three techniques for tail light detection: multi-scale detection for detecting objects at varying sizes, the Spatial Pyramid Pooling (SPP) technique for extracting rich information and the focal loss to address the issue of class imbalance. Hsu et al. [25] developed two distinct classifiers to recognize turn signals and brake signals. The CNN-LSTM model receives an image sequence to determine the braking condition of a vehicle from its rear view. To determine the status of a turn signal, SIFT flow alignment is used to determine the difference between succeeding frames for the turn signal state. The RoI regions of turn signals are given as input to the CNN-LSTM model for prediction. Lee et al. [26] combined a spatial attention model and a temporal attention model into a CNN-LSTM framework to recognize vehicle tail lights. From the above discussion, it can be concluded that these approaches do not localize tail light regions in each frame; instead, they learn spatiotemporal properties from a sequence of frames. As a result, they do not distinguish between the tail light signal and other light signals, such as traffic lights. It is crucial to know the location of the tail lights and the status of the vehicles that are going in front of the vehicle.

2.2 Vehicle re-identification

In this section, we outline some of the existing approaches to re-identification tasks. The deep learning methods have been successfully used for many computer vision tasks, and many deep learning methods have been developed in the literature for person reID and vehicle reID. The state-of-the-art methods typically employ a deep neural network to extract features from the visual representation of persons/vehicles. For object reID, two popular approaches were proposed in the literature, which are based on ranking and classification concepts. In ranking-based methods, three images are input: two depict the same identity, while the third belongs to a different identity. The model is trained using a loss function such as Triple Loss [80], Quadruplet Loss [81], or Margin Sample Mining Loss (MSML) [82]. The objective of these loss functions is to bring together samples with the same ID while simultaneously pushing apart those with different IDs. On the other hand, classification-related approaches [83, 84] use classification loss with a carefully designed special structure for object reID.

Luo et al. [16] introduced a baseline model for person reID incorporating a BNNeck technique. This approach aims to improve performance by combining ID loss and triplet loss in the training process. Pu et al. [85] presents a methodology for person image re-identification in a multi-camera setup. It proposes a multi-scale feature fusion network model that combines global and local features. The network is made up of four stacked building blocks, each of which processes multi-scale features with different weights before fusing them based on output conditions. Furthermore, a multi-head attention mechanism network is used to model relationships between input images, which improves feature aggregation from neighbouring images. Huynh et al. [41] presented a model for vehicle reID that introduces the concept of multi-head attention combined with Supervised Contrastive Loss [86]. Chen et al. [87] introduced an end-to-end distance-learning deep network for vehicle reID. This network integrates global features and local features at a more detailed level, aiming to improve the performance of vehicle reID systems. Khorramshahi et al. [88] employed a key point detection approach to segment vehicle images and extract detailed information. They then utilized deep learning techniques to refine these features and effectively combine both coarse and fine features, resulting in more discriminative and efficient representations. Cheng et al. [89] introduced a multi-granularity deep feature fusion method for vehicle reID. Their approach involved designing two distinct branches to extract and fuse global features and local features, ultimately utilizing the fused feature representation for vehicle image representation. Wang et al. [90] introduced an efficient deep convolution method for re-identification tasks. Their approach involved learning deep features guided by essential attributes, which collectively contributed to improving the overall re-identification performance. Zheng et al. [91] proposed a method to enhance feature representation using a multi-head architecture to extract multi-scale information. He et al. [92] investigated the utilization of vision transformers for re-identification tasks. They explored several domain adaptations and proposed a robust baseline called ViT-BoT, which served as the backbone network. To address the specific characteristics of re-identification data, the researchers introduced two modules: side information embedding and jigsaw patch module. Zhang et al. [93] explored the use of transformers for person re-identification in videos. However, they noted a challenge related to training transformers: the requirement for a large amount of data. Insufficient data can lead to a higher risk of overfitting. Yuan et al. [42] have created BPReID, a large-scale dataset that focuses on bike person re-identification collected in a campus environment. An approach based on handcrafted features of the bike and person parts of the image is used for bike-person re-identification. While this approach worked on the campus dataset, the challenges faced by real-time urban monitoring systems can’t be handled by this approach. The MoRe dataset, which was first proposed by Figueiredo et al. [43], is the first large-scale dataset that focuses entirely on motorcycles and is captured from urban traffic surveillance cameras in real-time. A strong baseline approach is developed using a deep learning model with a combination of triple loss [94, 80], quadruplet loss [81], and MSML [82], which are metric learning losses. They have also used techniques like warmup learning rate and label smoothing to increase the performance of motorcycle re-identification. Li et al. [44] proposed the pyramid attention mechanism to enhance the strong baseline introduced by Figueiredo [43] for capturing the essential information of the rider’s images. The effectiveness of this approach was evaluated on BPReID and MoRe datasets.

In the literature, few NAS approaches have been specifically designed for the reID task. Auto-reID [95] differs from expert-designed neural networks [36, 43, 44] as it focuses on automating the search for neural network architectures. Zhou et al. [96] introduced NAS based attention modules to learn spatial and channel attention feature maps. These feature maps were then combined to enhance the model's feature representation capabilities without the need for pretraining. To optimize the efficiency of attention placement in the re-identification module, an Attention Search Space (ASS) was proposed. Fu et al. [97] proposed a cross-domain deep network architecture search method that specifically explores Batch Normalization (BN) layers to discover the optimal architecture. Their work emphasizes the importance of learning shared information between different modalities in existing representation learning methods, which primarily aim to improve feature extraction. Chen et al. [98] introduced a NAS approach for person reID task. This method involved searching for an optimal cell structure by making greedy decisions during the search process. Further, they introduced a triplet loss with batch hard mining [80] as the retrieval loss, aimed at enhancing the feature representation capability of the backbone network and improving the overall generalization performance. Unlike these methods, which primarily concentrate on the search for backbone architectures in DNN, this work aims to automate the search process for the optimal backbone architecture of DNN and its hyperparameters for vehicle reID tasks.

2.3 Driver distraction detection

Various studies have been conducted to detect and mitigate the effects of driver distraction, improve road safety and reduce accidents. The survey by [99, 100] reviews existing studies on driver distraction, examining various methodologies, experimental setups, and their results. These categorize the impact of different distractions on drivers' physiological responses, visual signals, and performance. Another review by [101] follows the PRISMA guidelines and assesses various technologies such as eye tracking systems, and cardiac sensors to monitor driver distraction.

Earlier research on identifying driver distraction behavior relied mainly on manually collected features and conventional machine learning approaches. Zhang et al. [102] created a dataset that included four unique categories of driving activities: safe driving, gear-stick manipulation, talking on the phone, and eating. They also evaluated the performance of their proposed Hidden Conditional Random Fields model by evaluating the recognition accuracy with different classifiers. Zhao et al [103] proposed a method for improving detection model performance by integrating the pyramid histogram of oriented gradients with spatial scale feature extractors. This methodology was evaluated using a self-created distracted driving database, which included four driver activities: shifting, gripping the steering wheel, chatting on the phone, and eating. Artan et al. [104] used SVM to detect driver behavior, with a particular focus on mobile phone usage. This was achieved by integrating a near-infrared camera system aimed at the vehicle's front screen. Berri et al. [28] manually extracted features and used an SVM model to detect driver attention. In addition, they used a genetic algorithm to optimize the model parameters. The primary objective was to identify cell phone usage based on front-facing images of drivers. Craye and Karray [105] started by extracting four features from the driver using a Kinect camera, which provided a comprehensive representation of the driver's entire range. Head orientation, facial expression, eye focus and closure, and arm position were among the characteristics. The combination of these characteristics resulted in a representation capable of detecting driver distraction. Following that, an AdaBoost classifier was used to classify distraction behavior. Yan et al. [29] created a pyramidal histogram of gradients using the driver's motion history images, taking into account the data's temporal features. The collected features were fed into a random forest classifier to classify distraction behavior. However, manual feature extraction is time-consuming, requires specialized knowledge, and frequently produces substandard results.

In recent years, researchers have explored employing a deep learning approach to address driver behaviour detection. Baheti et al. [106] introduced an improved VGG-16 model that replaced two convolutional layers with fully connected layers. They modified the activation function of LeakyReLU and added a dropout layer. Eraqi et al. [107] created the American University in Cairo Driver Distraction Dataset (AUCD2), which has ten classes in it. A pre-trained AlexNet and InceptionV3 models were used for face image and hand image segmentation. A genetic algorithm was used to fine-tune the models. However, because of the large scale size of the model, real-time detection is difficult. Ghizlene et al. [108] introduced a method for detecting driver drowsiness that combines the Haar cascade with the YOLO algorithm for fast recognition of the driver's eyes. Lu et al. [109] proposed dilated and deformable Faster R-CNN as a driver action recognition system. This system detects driver actions by detecting items with motion specificity that show both inter-class and intra-class distinctions. The use of dilated and deformable residual blocks facilitates the extraction of irregular and tiny characteristics such as cell phones and cigarettes. The authors collected a dataset of images, which contains diverse driver activities. Masood et al. [110] used the VGG16 and VGG19 models to detect driver distraction and efficiently classify the distinct driving actions. Alkinani et al. [111] used deep learning techniques to detect both inattentive and aggressive driving behaviors in drivers. They classified inattentive driving into subtypes such as driver fatigue, drowsiness, distraction, and other risky behaviors such as aggressive driving. These risky behaviors were discovered to be related to factors such as driving age, experience, health conditions, and gender. CNNs, Recurrent Neural Networks (RNNs), and LSTMs were used in the research for their task. Li et al. [112] created a two-stage system for detecting driver distraction. The YOLO deep learning object detection model is used in the first module to detect the bounding boxes associated with the driver's right ear and right hand from RGB images. These bounding boxes are then used as input for the second module, a multi-layer perceptron, which predicts the type of distraction based on the information provided by the bounding boxes. Shahverddy [113] used a recursive graph technique to analyze driver behavior. Driving signals such as acceleration, gravity, and throttle were converted into images. After that, a CNN was used to detect driver distractions. Based on the analysis of signals such as acceleration, gravity, throttle, speed, and Revolutions Per Minute (RPM), the system classified driving styles into five categories: normal, aggressive, distracted, drowsy, and drunk. Methuku et al. [114] presented a deep learning model for identifying and classifying drivers' behaviors and actions while traveling. The model divides driver actions into ten categories, with the first representing safe driving and the remaining nine representing various unsafe actions such as fixing makeup and texting. CNNs were used in the training and detection processes, with ResNet50 serving as the backbone of DNN architecture. For classification, a dense net architecture was used after the ResNet50 architecture. The State Farm dataset was used for training, which included images depicting various driver actions associated with distracted driving. Li et al. [115] developed a deep learning research approach aimed at investigating the relationship between typical driving behavior and instances of distracted driving. The researchers used advertisements to recruit 24 volunteers for their study and conducted signal collection experiments. The results were derived from their gathered dataset. Zhao et al. [116] proposed a system for detecting driver behavior based on an adaptive spatial attention mechanism. To extract features, the model employs an adaptive discriminative space. A multi-scale feature representation is extracted, and classification is performed using a k-Nearest Neighbour (K-NN) classifier. Zhang et al. [117] developed an unsupervised multimodal fusion network for detecting driving distractions. The model is made up of three major modules: multimodal representation learning, multiscale feature fusion, and unsupervised driver distraction detection. Qin et al. [118] presented an improved eye-tracking object detection dataset based on driving videos. Following that, the Increase-Decrease YOLO network was developed to simulate the driver's selective attention mechanism, with a focus on identifying key objects on the driver's face.

With the advancement of deep learning, Vision Transformers (ViT) [119] have gained popularity in distraction detection tasks. They can capture long-range dependencies in visual data, making them suitable for tasks that require high-level semantic understanding. [30] proposes a lightweight vision transformer-based method for detecting distracted behavior using a pseudo-label-based semi-supervised learning approach. This method allows accurate detection by generating strong and weak augmented versions of the input data. Another approach, CaTNet, introduced in [32], integrates self-attention with convolutional layers to capture local and global features efficiently. This lightweight model achieves good accuracy with fewer parameters. Swin Transformers has also been explored in [31], where a driver distraction detection model incorporates a high-discrimination feature learning strategy to improve the separation between different classes of distractions. This method, evaluated on two public datasets SFDDD and AUCD2, achieved superior performance over CNN-based approaches.

Attention mechanisms have been applied to driver distraction detection to focus on relevant parts of the visual input and improve classification performance. In [33], a constrained attention (CA) mechanism is introduced for real-time distraction detection. The CA mechanism includes an internal constraint, with concentrative regularization to prevent excessive or ambiguous attention and orthogonal regularization to differentiate attention across classes. It also uses an intersample constraint to optimize image representations within a batch. The method has been tested on SFDDD and AUCD2 datasets and demonstrated significant gains in accuracy. [120] proposed a three-level attention mechanism, combining channel, spatial, and batch-level attention, to enhance the generalization ability of a lightweight vision transformer model. This method strikes a balance between performance and efficiency. The technique incorporates a lightweight feature extraction backbone with a dual-stream structure, enhancing training efficiency. It employs contrastive learning based on similarity calculations, with attention visualization at different depths demonstrating the method's effectiveness.

Self-supervised learning has emerged as an efficient approach to reducing the reliance on large labeled datasets. The work presented in [34] introduces a self-supervised learning framework based on Masked Image Modeling (MIM) for driver distraction detection. Using the Swin Transformer as an encoder, the model achieves high accuracy while maintaining lightweight architecture. Various data augmentation strategies are applied to improve recognition and generalization, making the model highly effective in large-scale driver distraction datasets. Representation learning has also been explored to enhance driver distraction detection. A multi-modal ViT method, ViT-DD, presented in [121], integrates emotion recognition with distraction detection to enhance the model's performance. The semi-supervised learning algorithm allows the integration of driver distraction images without emotion labels into the multi-task training process, resulting in improved accuracy on standard benchmarks.

The challenge of protecting user privacy while collecting driver data has prompted the exploration of federated learning. In [35], the Asynchronous Federated Meta-learning framework (AFM3D) is proposed to address data heterogeneity, privacy concerns, and the inefficiencies of centralized learning paradigms in driver distraction detection. The framework bridges data islands using federated learning and meta-learning to quickly adapt to new driver data. At the same time, an asynchronous mode ensures efficient learning even in the presence of delays. The model outperforms existing methods in terms of accuracy, recall, and learning speed.

2.4 Summary

This chapter provides an overview of the literature on tasks related to the detection of vehicle brake lights, vehicle re-identification, and driver distraction detection, particularly in the context of vision based accident prevention and smart traffic surveillance applications. It commences by discussing traditional approaches and subsequently explores various techniques based on deep learning. Furthermore, it explores the use of NAS techniques for optimizing DNN models. Subsequent chapters (Chapter 3, 4, 5, and 6) will delve into more elaborate solutions focusing on NAS based optimization of DNN models for these tasks.

Chapter 3

Genetic Algorithm based Optimization of Deep Neural Networks for Vehicle Brake Light Detection

This chapter proposes a NAS based optimization of deep neural networks using a genetic algorithm for the task of vehicle brake light detection.

Chapter Organization:

Section 3.1 presents the proposed methodology, Section 3.2 discusses the experimental results and analysis and finally, Section 3.3 provides the summary of the work.

3.1 Proposed approach

This section introduces the proposed use of NAS to optimize the object detection network, including both the backbone and the training attributes, specifically designed for the brake light detection task. The details of the proposed model are shown in Fig. 3.1. The initial population is set randomly. A GA chromosome in the population is encoded using the scheme shown in Fig. 3.1 (c), to create a corresponding Mask R-CNN based object detection network. Then, the model is trained on vehicle data to evaluate its fitness value. A GA based search strategy is used to generate a new population. After each generation, a new population is generated, and the GA cycle is terminated when certain criteria are met.

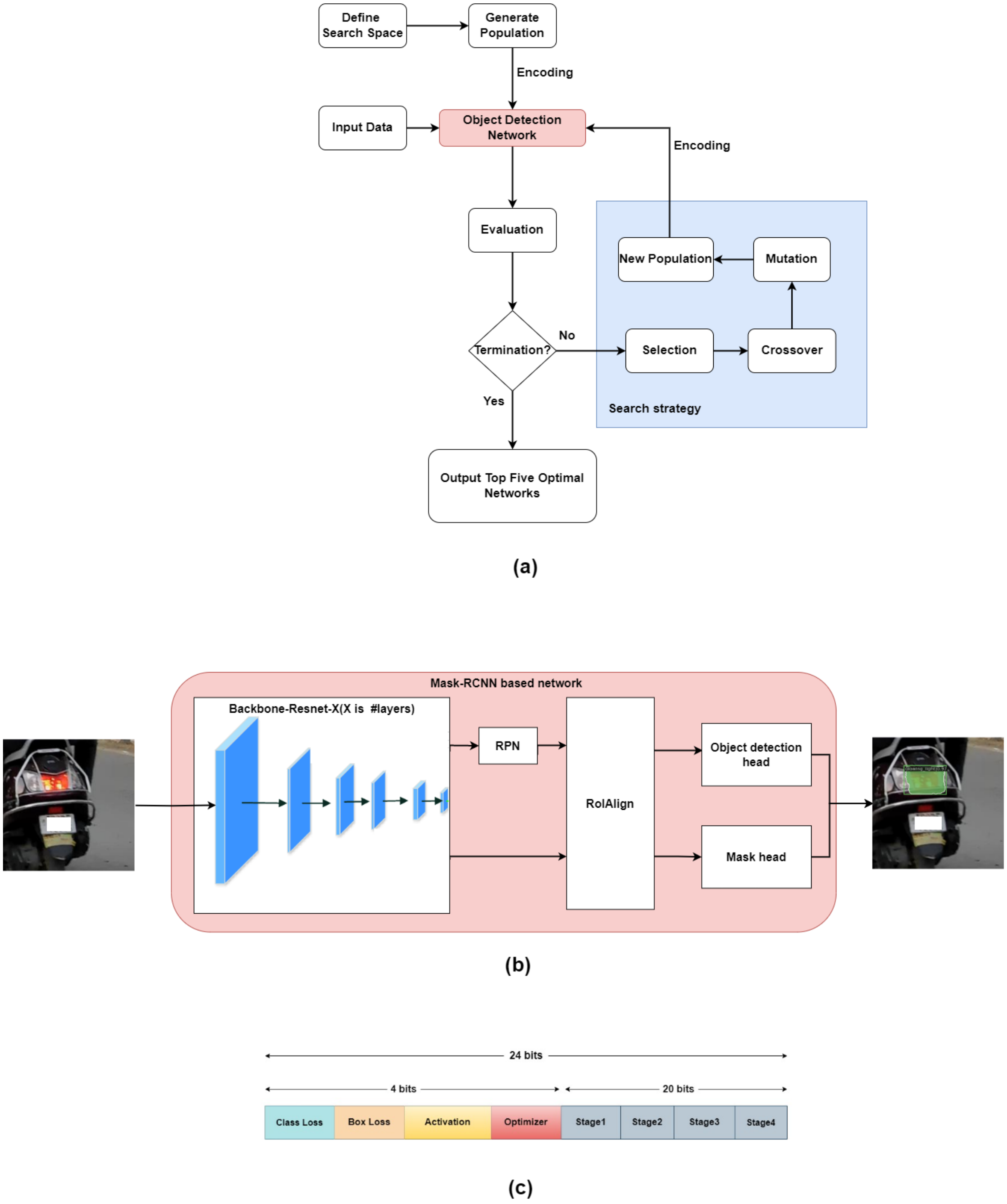


Figure 3.1: (a) The proposed GA based NAS for brake light detection (b) Mask R-CNN based object detection network (c) Encoding of GA chromosome.

Neural Architecture Search(NAS):

A NAS approach is composed of three parts: a search space, a search strategy, and a performance estimation method. The first step in using a NAS is to define the “**search space**" for the design of the neural network architecture. The neural network hyper-parameters generally set using NAS are the number of filters, type of layer, number of units in each layer, kernel size, etc. Other parameters can also be included in the search space, such as the loss function, optimization algorithm, activation function, etc., which are related to training a neural network architecture. More versatility can be obtained by adding more parameters to the search space. However, as the search space grows, the cost of finding the best deep neural network architecture increases. Finding the best DNN architecture, even for a simple search space, is a challenging task without a proper search strategy. As a result, NAS approaches require an efficient “**search strategy**". The commonly used search strategies for finding the best DNN architecture are based on evolutionary algorithms, reinforcement learning, and gradient-based optimizers. After training the deep learning models using the NAS approach, the objective function used for optimization is used to find the best neural network. We can formulate NAS as an optimization problem, as given in Eq. 3.1.

$$max_{a \in S}\ \ \mathcal{M}_{val}(\mathcal{A}(a, w^*(a))) \tag{3.1}$$

Here, S denotes the search space of the DNN models that can be represented by a Directed Acyclic Graph (DAG), in which each path in the DAG corresponds to a specific DNN model denoted by a, which is an element in S, i.e. $a \in S$. Here, the DNN model includes both the backbone architecture and training parameters. $\mathcal{A}(a, w)$ refers to the DNN model a with weight w. Similarly, $\mathcal{A}(a, w^*(a))$ refers to the DNN model a with optimal weights w^*, that is obtained through training as given in Eq. 3.2. Here, $\mathcal{L}_{train}$ is the training loss. $\mathcal{M}_{\text{val}}$ represents the validation criteria based on mean Average Precision (mAP). Eq. 3.1 identifies the optimal architecture a with optimal weights w^* that maximizes validation accuracy, which is the formulation of NAS based optimization of the DNN model.

$$\mathcal{A}(a, w^*(a)) = min_{w}\ \ \mathcal{L}_{train}\mathcal{A}(a, w) \tag{3.2}$$

The performance of the object detectors depends on the features extracted by the backbone. In image classification, a significant performance improvement can be achieved by replacing a ResNet-50 backbone with deeper neural networks, such as ResNet-101, ResNet-152, etc. However, using NAS for identifying a backbone used in the object detector, is a challenging task. A backbone is pre-trained on ImageNet for image classification. Pretrained models have two limitations in the standard object detector training process. The first limitation is that they explore predefined architecture as the backbone, which may not be the best architecture for a given object detection task. The second limitation is that each candidate architecture requires training on ImageNet before it is fine-tuned for the given detection dataset, which is computationally expensive. In this work, we presented an approach for finding optimal DNN architecture for object detection, by using GA based NAS to search for optimal backbone and training attributes.

3.1.1 Search space design

The search space will have a significant impact on the performance of the architectures generated by the NAS approach. In this work, six parameters are considered in the search space, including the backbone architecture and the training attributes as given in Table 3.1. From the [7] it is observed that the type of blocks and depth of the neural network determines the performance of the DNN recognition model. In [15], He et al. explored the use of fixed number blocks in each stage with a fixed number of layers (e.g. Resnet-50, Resnet-101 and Resnet-152), which may not give an optimal model. Hence, we considered the type of block and the number of blocks as parameters in the search space to identify the best backbone architecture. In [7], He et al. considered two blocks (Basic block and Bottleneck block shown in Fig. 3.2) in which, the Bottleneck block is used to reduce the number of parameters of the model as the depth of the model increases. In this work, the Bottleneck block is used in each stage, if the total number of blocks is greater than or equal to 40; else, the Basic block is used. The number of blocks in each stage is set to vary from 1 to 31 to explore up to Resnet- 374 (When the Bottleneck block is chosen, each block consists of three convolutional layers. For example, if a stage has 31 blocks, it contains 31 × 3 = 93 layers. Since the network has four stages, the total number of layers across all stages is 93 × 4 = 372 layers. Adding two more layers, one for the input convolution and one fully connected layer at the end, gives a maximum depth of 374 layers, creating the ResNet-374 architecture). This parameter for four stages is considered in the search space to search for the optimal number of blocks in each stage, thereby finding the optimal number of layers.

Recent literature [122, 123] is focused on optimization of the neural network's backbone architecture but does not consider training attributes such as loss function, activation function, and optimization algorithms that have a significant impact on the model's performance [124, 125]. As a result, we have incorporated the training attributes and parameters related to backbone architecture in the search space of this work. ReLU [126] is a generic activation function that is used in models capable of performing a variety of tasks. Mish [127], a recently proposed activation function with ReLU-like properties, adds continuous differentiability, non-monotonicity, and other behaviour, which outperformed ReLU in some tasks. As a result, these functions are included in the search space as values for the activation function parameter. For the optimization algorithms, Stochastic Gradient Descent (SGD) and Adam are considered. The class loss and bounding box loss also have a major impact on the performance of the object detection model. For bbox loss, L1 loss and Mean Squared Error (MSE) loss were used. For class loss, cross-entropy loss and Focal loss [128] are used in recent literature [79], the focal loss is found to be effective in dealing with class imbalance problems.

A 24-bit chromosome is used to encode the architecture and the training attributes, as shown in Fig. 3.1 (c). The first bit encodes the class loss, the second bit encodes box loss, the third bit encodes the activation function, the fourth bit encodes the optimization algorithm, and bits from 5 to 24 encode the number of blocks in each of 4 stages (5 bits per stage).

Table 3.1: Parameters in the search space, considered in this work

Type of parameter	**Range**
Type of block	{Basic Block, Bottleneck Block}
Number of blocks in each stage	[1-31]
Activation	{ReLU, Mish}
Optimizer	{Adam, SGD}
Class Loss	{Cross Entropy loss, Focal loss}
BBox loss	{L1 loss, MSE loss}

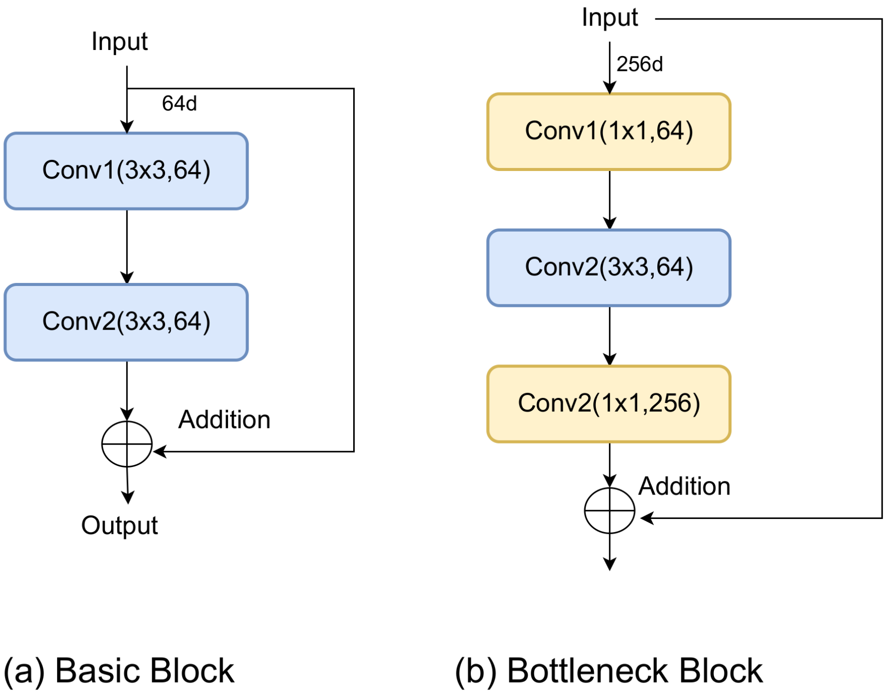


Figure 3.2: Type of block (a) Basic Block (b) Bottleneck Block

3.1.2 Search strategy

We use the evolutionary algorithm as the search strategy for the proposed NAS based optimization approach. In contrast to RL based and gradient-based NAS approaches, the evolutionary search strategy can consistently meet hard limitations, such as inference speed. RL based approaches require a carefully tuned reward function to optimize inference speed, whereas gradient-based methods require a well-designed loss function. GA is the most commonly used evolutionary algorithm for exploring neural network architectures. In this work, GA is used to automatically identify the optimal neural network architecture. GA is based on Darwin's natural evolution hypothesis, which generates a new population with better fitness than the existing population. In binary encoded genetic algorithms, the GA chromosome is represented by a binary bit string. A chromosome comprises of genes that capture the individual's genetic characteristics, thereby representing a solution. The individual GA chromosome is a candidate in the population. The three primary operations of genetic algorithms are selection, crossover, and mutation. The “selection" aims to identify individuals, known as parents, with a higher fitness value and hence may result in offspring that have a better chance of surviving in the following generation. Tournament selection is used in this work to select parents. The diversity in the population is achieved through crossover and mutation. In “crossover" a random crossover site is identified and the bit strings of the parents identified in the selection are interchanged, to form a new offspring. “Mutation" inserts random genes into offspring, to achieve population diversity and prevent early convergence. It simply means changing 0 to 1 and 1 to 0. The process of population generation begins with creating a random population of candidate solutions. These GA operations will produce a new population from the existing population. The selection operation identifies parents from the existing population. Then, to create new offspring, the crossover and mutation operations are applied to the parents and offspring, respectively. Finally, new individuals are evaluated and included in the next generation's population. The flowchart of the GA process is given in Fig. 3.3.

3.2 Experimental results and analysis

In this section, we first discuss the evaluation metric and implementation details. Section 3.2.1 presents the evaluation of the proposed approach for two-wheeler vehicle brake light detection on the proposed NITW-MBS dataset. The evaluation of the proposed approach for four-wheeler vehicles on the CaltechGraz dataset [50, 129] is discussed in Section 3.2.2. The effectiveness of the proposed approach against existing approaches for the brake light

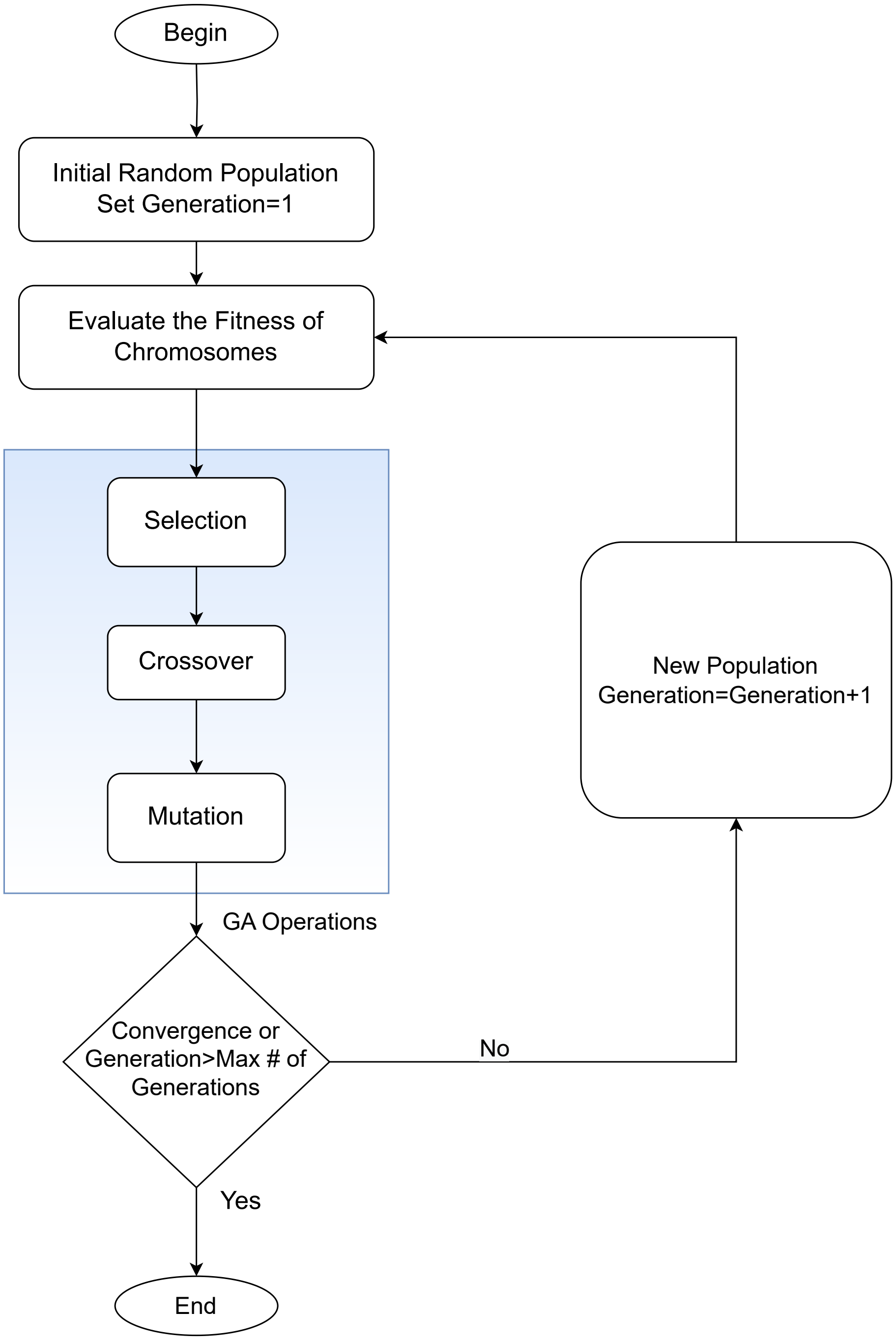

Begin
Initial Random Population
Set Generation=1
Evaluate the Fitness of Chromosomes
Selection
Crossover
Mutation
GA Operations
Convergence or Generation>Max # of Generations
No
New Population
Generation=Generation+1
Yes
End

Figure 3.3: Flowchart of Genetic Algorithm

detection task is presented in Section 3.2.5.

Experimental Settings:

The experimental study in this work is performed on a computer with an Intel Xeon(R) Silver 4110 CPU running at 2.10GHz, 64GB of RAM, and one NVIDIA GeForce RTX

2080 Ti GPU, with CUDA 11.1 in Linux platform with Pytorch framework. During the exploration of the search space, if SGD is selected as the optimizer then the weight decay of 0.005, momentum of 0.9 and the initial learning rate of 0.02 are considered. Similarly, if Adam is selected as the optimizer then the weight decay of 0.005, momentum of 0.9 and the initial learning rate of 0.0003 are considered. The input images are scaled to 227 × 227 pixels, before giving them as input to the object detection model.

Evaluation Metrics: Accuracy based on Intersection over Union (IoU) and mAP are commonly used evaluation metrics to assess the performance of the object detection model. To compute Accuracy, the IoU threshold is set to determine if a prediction is correct or not. If the IoU between the predicted and ground truth bounding boxes exceeds the threshold, it is considered a true positive; otherwise, it is a false positive. Accuracy is the percentage of correct detections (true positives) out of the total predictions made by the model, indicating its ability to localize objects accurately. mAP calculates the average precision across multiple object classes. It involves generating precision-recall curves for each class by varying the IoU thresholds (0.5, 0.55, 0.95). The formulas for calculating IoU, Precision, Recall, and Accuracy are provided in Eq. 3.3, 3.4, 3.5, and 3.6, respectively.

$$IoU = \frac{Area\ of\ Overlap}{Area\ of\ Union} \tag{3.3}$$

$$Precision = \frac{True\ Positive}{True\ Positive\ +\ False\ Positive} \tag{3.4}$$

$$Recall = \frac{True\ Positive}{True\ Positive\ +\ False\ Negative} \tag{3.5}$$

$$Accuracy = \frac{True\ Positive\ +\ True\ Negative}{Total\ number\ of\ predictions} \tag{3.6}$$

3.2.1 Experimental study for Motorcycle (two-wheeler) brake light detection

In this section, we discuss the details of the evaluation of the proposed approach for detecting brake lights of two-wheelers. A new dataset (NITW-MBS) for detecting motorcycle brake lights is presented. The evaluation of the proposed approach against various existing approaches on this dataset is presented.

NITW-MBS dataset:

To our knowledge, no publicly available dataset for detecting motorcycle brake lights exists. For the task of brake light detection, we proposed the NITW-MBS dataset to evaluate the proposed approach. We have recorded several videos of vehicles on different roads and at different times using a handheld camera. The dataset is built to cover different capturing & environmental conditions along with different shapes and sizes of brake lights. 2125 images are selected and annotated for brake light detection in the COCO data format [130]. Fig. 3.4 and Fig. 3.5 show some of the observations in this dataset. To train a model, the dataset should be split into three parts: training, validation, and test data. The training data is used to train the model, and the validation data is used to adjust the model's hyper-parameters to avoid over-fitting, and the test data is used to compute the model's performance on unseen data. The dataset has two classes: glowing light and non-glowing light, with 1650 images for training, 210 images for validation, and 265 images for test data. The statistics of the dataset are given in Table 3.2

Table 3.2: Number of images in each class, for the training, validation and test data of NITW-MBS dataset

Data	Glowing light	Non-Glowing light	Total
#training	1036	614	1650
#validation	138	72	210
#test	157	108	265

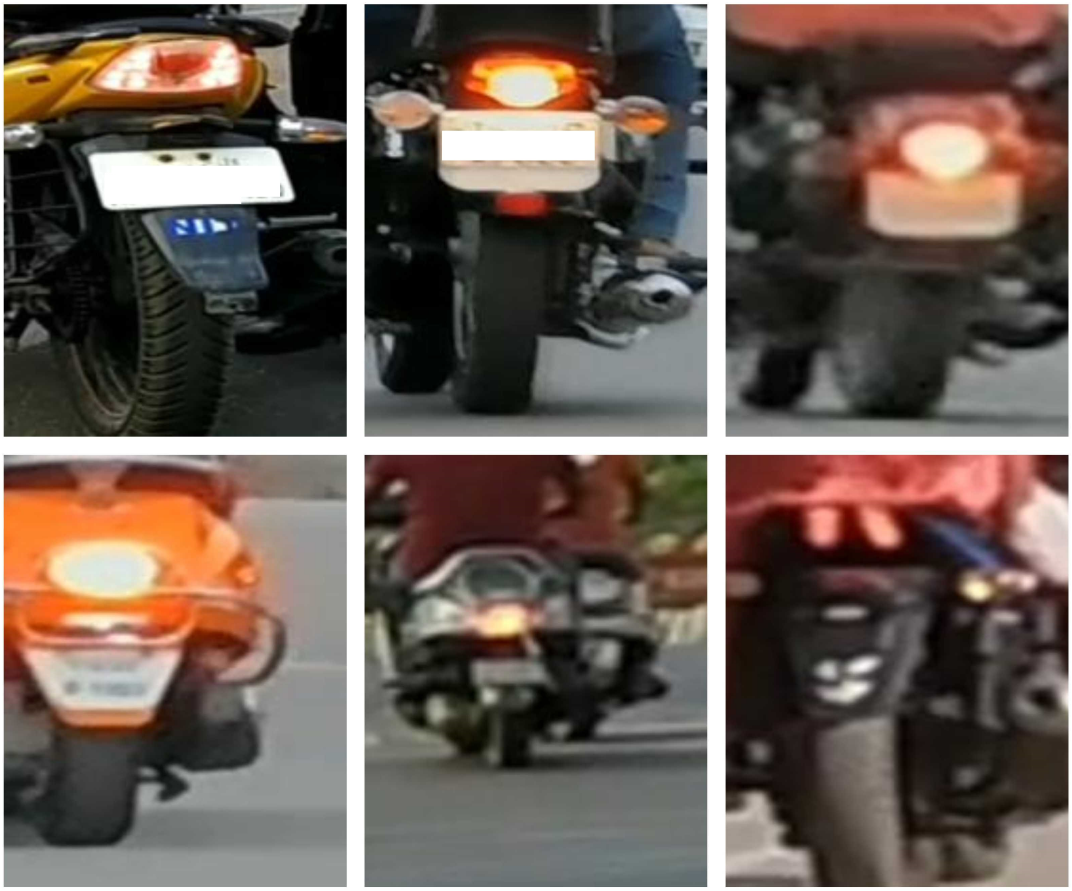


Figure 3.4: Images from NITW-MBS dataset, with the status of brake light as glowing

We evaluate the existing object detectors, including one-stage YOLOv3 [10], TOOD [131] and two-stage Faster R-CNN [14], Mask R-CNN [15] and Sparse R-CNN [132]. The results are given in Table 3.3. It can be observed that the model identified by the proposed approach performs better than the existing benchmark object detection models. Fig. 3.6 and Fig. 3.7 show the brake light status detection results using the proposed approach on sampled images from the test data. Fig. 3.9 shows the change in validation mAP of the methods given in Table 3.3 against the number of training epochs.

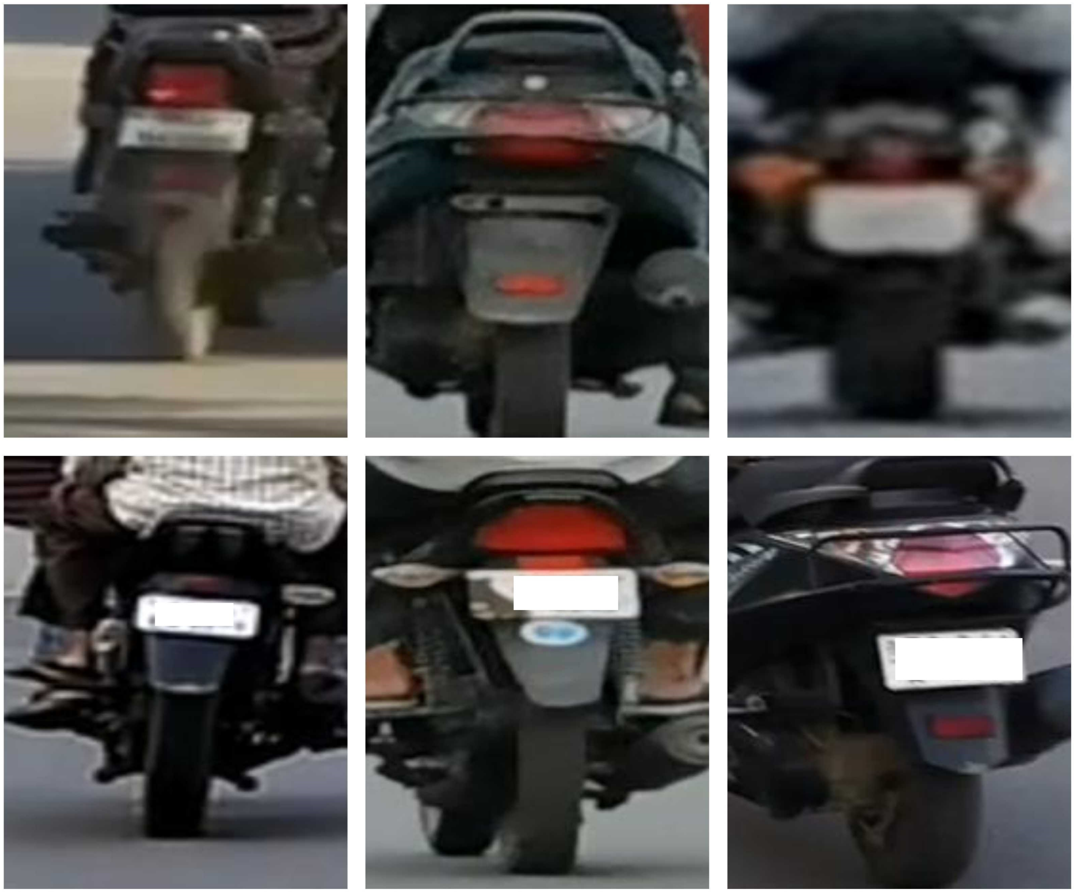


Figure 3.5: Images from NITW-MBS dataset, with the status of brake light as non-glowing

Table 3.3: Comparison with state-of-the-art models for two-wheeler brake light detection on NITW-MBS dataset

Method	mAP(%)		Params
	Valid	Test	
YOLOv3 [10]	35.6	37.1	61.5 M
Faster R-CNN [14]	52.1	48.2	63.6 M
Mask R-CNN [15]	52.7	48.7	31.6 M
TOOD [131]	53.9	49.5	22.4 M
Sparse R-CNN [132]	53.8	51.4	96.6 M
Mask R-CNN-Resnet131 (**Ours**)	**54.5**	**52.2**	98.3 M

Note: The best values are highlighted in bold.

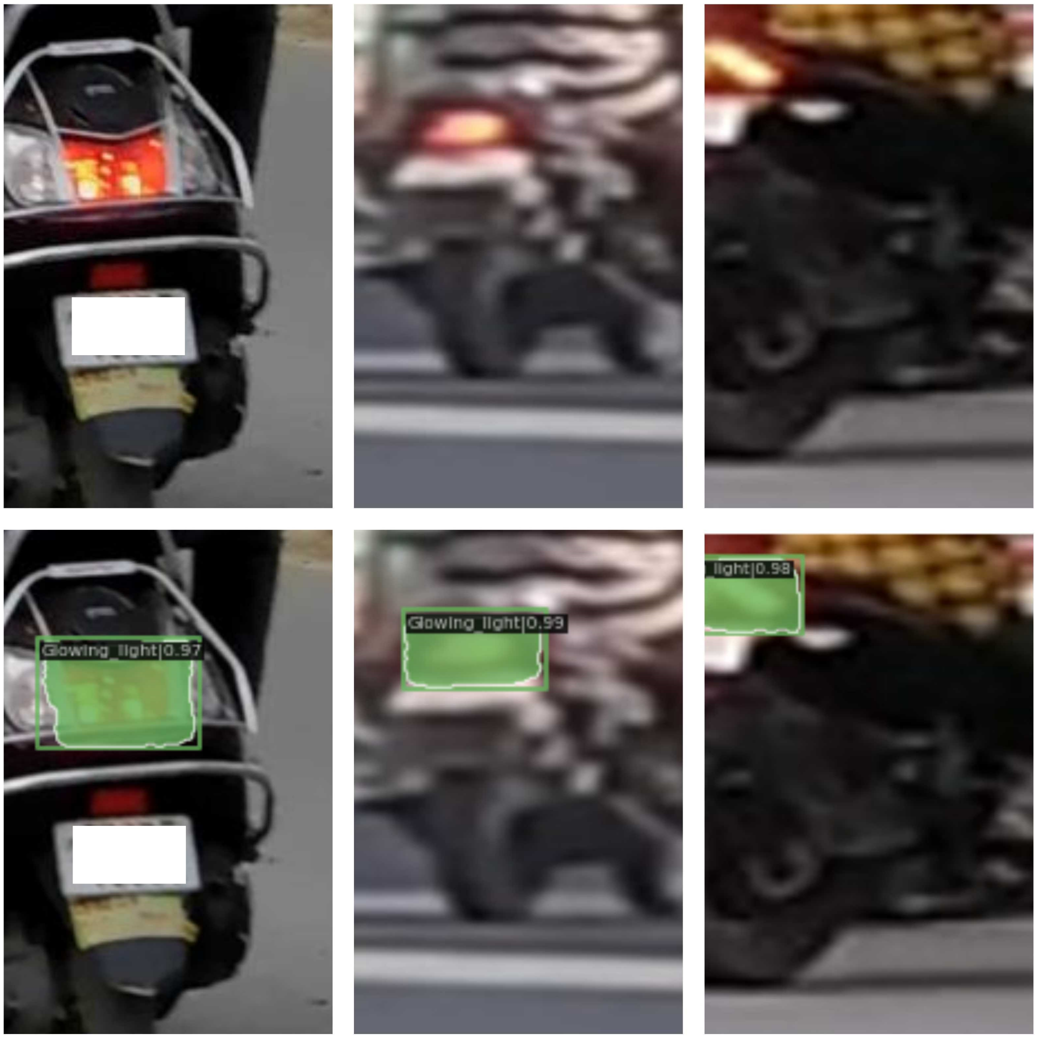


Figure 3.6: Glowing brake light status detection results of the proposed approach on NITW-MBS dataset

We compared the performance of the top-5 distinct architectures generated by the proposed approach, results are given in Table 3.4. It can be observed that in most architectures, the ReLU activation function outperforms the Mish activation function, Cross entropy loss outperforms Focal loss, and MSE loss outperforms L1 loss. Furthermore, when it comes to optimizers, Adam sometimes outperforms SGD while at other times SGD outperforms Adam. The Mask R-CNN is compared with the modified backbone with basic and bottleneck blocks. The generated optimal architecture has a detection result of 54.5 % mAP on validation data and 52.2 % mAP on test data, indicating that the identified architecture is more effective than the existing models for the Motorcycle brake light detection task, as

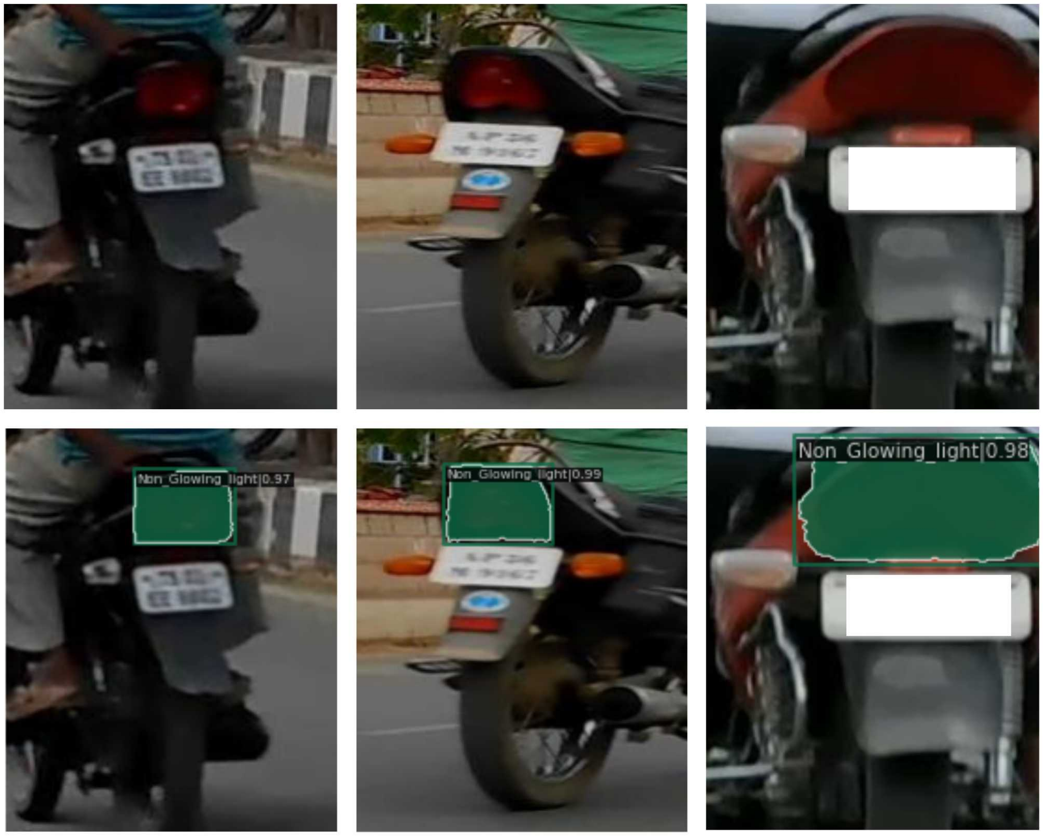


Figure 3.7: Non-Glowing brake light status detection results of the proposed approach on NITW-MBS dataset

shown in Table 3.4.

We analyzed the various values used while exploring the search space by the proposed approach across GA generations. We first calculate the fitness of the initial random population, considering a population size of 20. Now, the top 10 chromosomes are selected and subjected to selection, cross-over, and mutation to generate the population for the next generation. If the maximum fitness value across three generations does not differ significantly, the algorithm is set to terminate. In this study, the proposed approach converged after seven generations. Fig. 3.8 shows the number of times a parameter value is used in the population across generations. The analysis between the parameter exploration process and the top-1 model identified by the proposed NAS based approach reveals both consistencies and deviations. The exploration favoured Cross Entropy loss over Focal loss, and MSE loss was preferred over both L1 and Smooth L1 losses, with the final model choosing Cross Entropy

Table 3.4: Comparison of top-5 models, generated by proposed approach on NITW-MBS dataset

Method	#Blocks	Block type	Class loss	Bbox loss	Optimizer	Activation	mAP (%)	
							Valid	Test
Mask R-CNN-Resnet62	[7,1,1,21]	Basic Block	Cross Entropy loss	MSE loss	Adam	Mish	52.0	45.4
Mask R-CNN-Resnet152	[29,1,5,15]	Bottleneck Block	Focal loss	L1 loss	Adam	ReLu	50.1	46.8
Mask R-CNN-Resnet155	[23,6,1,21]	Bottleneck Block	Focal loss	MSE loss	SGD	ReLu	51.1	49.5
Mask R-CNN-Resnet62	[13,10,1,6]	Basic Block	Cross Entropy loss	L1 loss	SGD	ReLu	54.1	50.6
Mask R-CNN-Resnet131	[15,8,5,15]	Bottleneck Block	Cross Entropy loss	MSE loss	SGD	ReLu	**54.5**	**52.2**

Note: The best values are highlighted in bold.

and MSE, confirming their suitability for the task. The ReLU activation function, which dominated during exploration, was also selected in the final model. While Adam was initially common, SGD became the preferred optimizer as generations progressed, aligning with its selection in the final model. Despite the exploration favouring the basic block, the choice of the bottleneck block for final model, indicates a strategic adjustment recognizing the bottleneck design's enhanced performance potential for the task. Overall, the final model reflects key exploration trends while also incorporating strategic choice for block type.

3.2.2 Experimental study for four-wheeler brake light detection

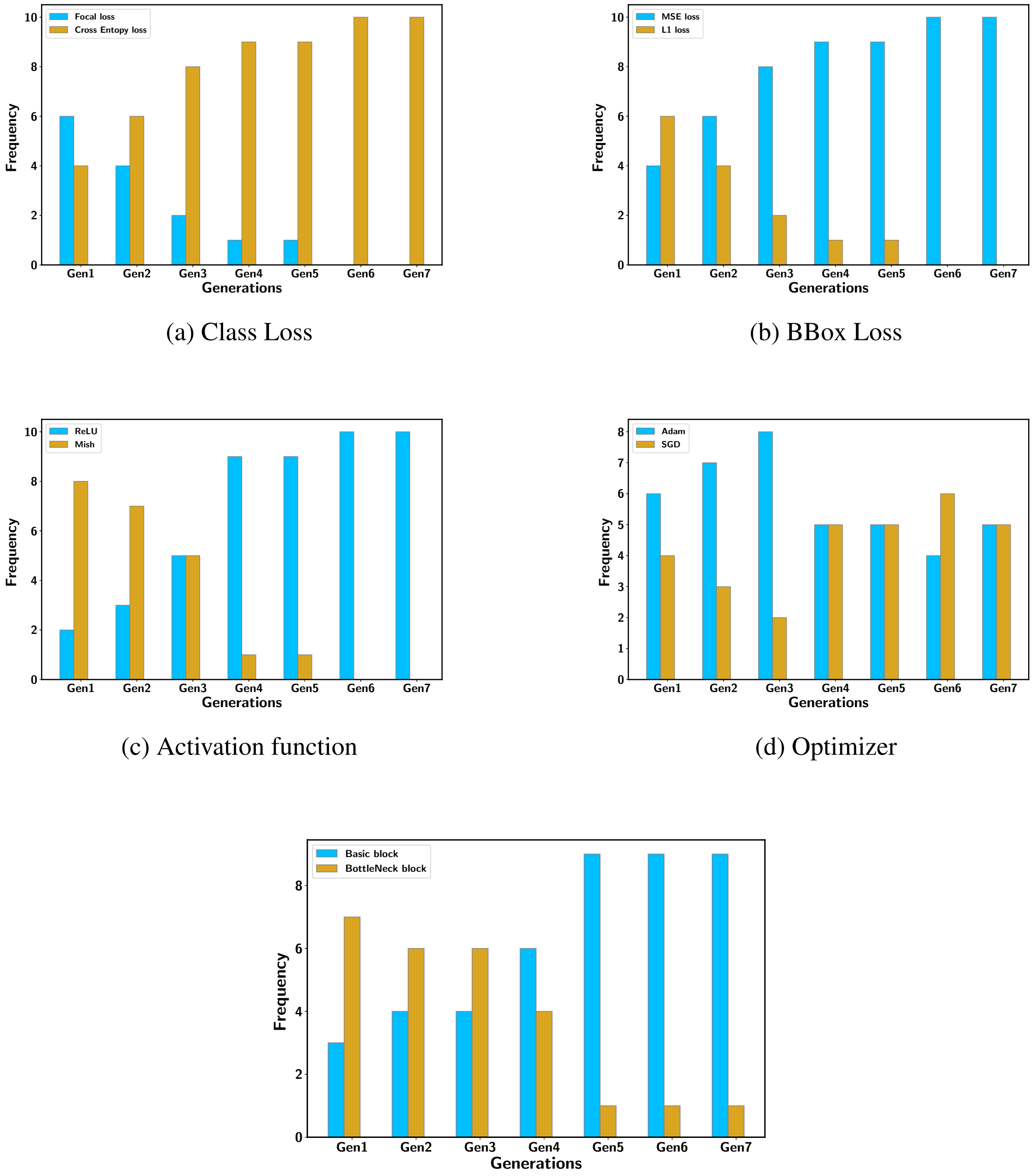
In this section, we evaluate and analyse the effectiveness of the proposed approach for four-wheeler vehicles on the CaltechGraz [50] dataset. The comparison of the performance of the proposed approach against the existing object detection approaches is presented.

We considered CaltechGraz dataset [50] from the Caltech database [51, 133] to evaluate the proposed approach. We selected 490 images and annotated them in the COCO data format. Fig. 3.10 and Fig. 3.11 show some of the observations in this dataset.

To train the model, the dataset is split into three parts: training, validation, and test data. CaltechGraz dataset has two classes: glowing light and non-glowing light, with 350 images for training, 50 images for validation and 90 images for test data. The statistics of the CaltechGraz dataset considered in this work are given in Table 3.5.

Table 3.5: Number of Images, Bounding boxes for each class in the training, validation, and test data of CaltechGraz dataset considered in this study

Data	Glowing light Bounding boxes	Non-Glowing light Bounding boxes	#Images
#training	400	324	350
#validation	54	51	50
#test	116	80	90



(a) Class Loss

(b) BBox Loss

(c) Activation function

(d) Optimizer

(e) Blocks Types

Figure 3.8: The number of times a parameter value is used across generations in the population

We evaluate the existing object detectors, including one-stage models such as YOLOv3 [10], TOOD [131], and two-stage models such as Faster R-CNN [14], Mask R-CNN [15] and Sparse R-CNN [132] and the results are given in Table 3.6. It can be observed that the proposed model outperforms the existing object detection models by a considerable

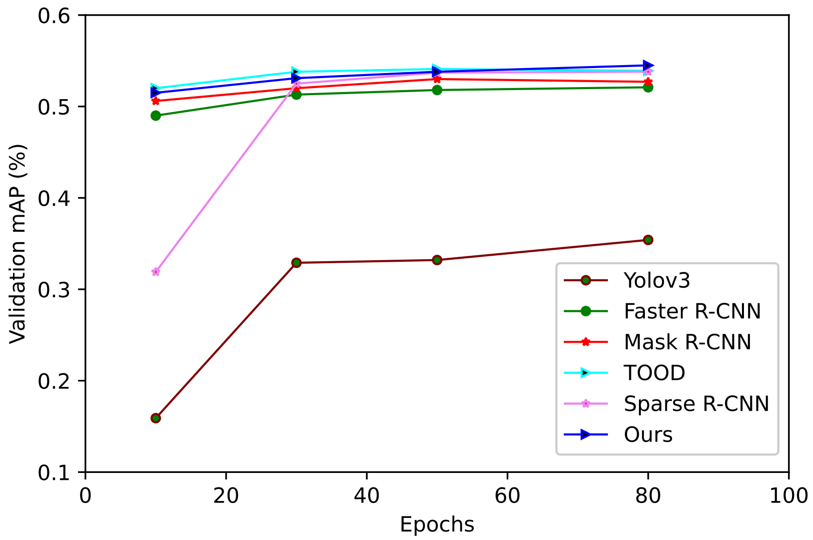


Figure 3.9: Change in validation mAP against training epochs for YOLOv3, Faster R-CNN, Mask R-CNN, TOOD, Sparse R-CNN and proposed model on NITW-MBS dataset

margin. Fig. 3.12 and Fig. 3.13 show the detection of the brake light status by the proposed approach on some test images in the CaltechGraz dataset. Fig. 3.14 shows the plot of variations of mAP for the object detectors in Table 3.6 against training epochs.

Table 3.6: Comparison with the state of the art models on CaltechGraz dataset

Method	mAP(%)		Params
	Valid	Test	
YOLOv3 [10]	18.9	20.9	61.5 M
TOOD [131]	29.4	24.4	22.4 M
Faster R-CNN [14]	28.6	31.1	63.6 M
Mask R-CNN [15]	29.7	33.1	31.6 M
Sparse R-CNN [132]	32.5	36.2	96.6 M
Mask R-CNN-Resnet64 (**Ours**)	**35.3**	**36.9**	27.7 M

Note: The best values are highlighted in bold.

The comparison of the performance of the top-5 architectures generated by the proposed approach is given in Table 3.7. The optimal architecture has a detection result of 35.3 %

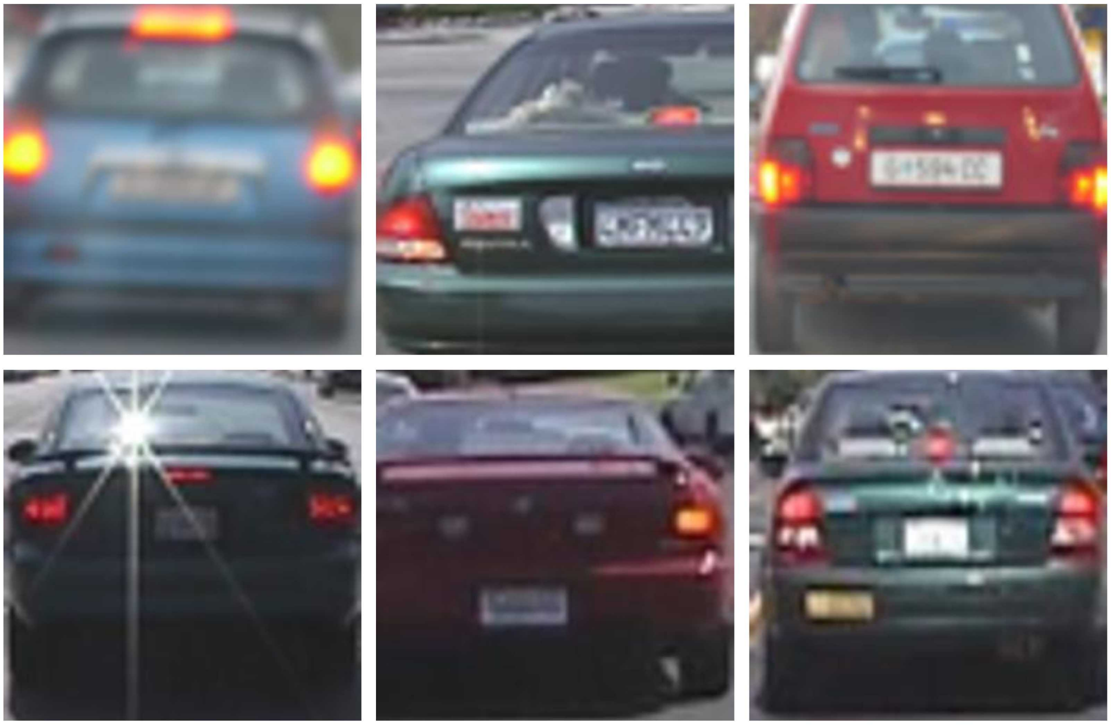


Figure 3.10: Images from CaltechGraz dataset, with the status of the brake light as glowing

mAP on validation data and 36.9 % mAP on test data, indicating that the proposed approach is effective for four-wheeler vehicle brake light detection on the CaltechGraz dataset.

3.2.3 Network architectures identified by the proposed approach

The proposed approach explored various deep neural network architectures and their corresponding parameters across four-wheeler and two-wheeler vehicles. The details of the top-1 identified DNN architectures are shown in Tables 3.4 and 3.7. The tables show that the Bottleneck block combined with Cross Entropy loss, MSE loss, SGD, and ReLu activation performed well for two-wheeler vehicles. Meanwhile, the Basic block with Cross Entropy loss, MSE loss, Adam optimizer, and ReLu activation performed well for the four-wheeler vehicles. For the two-wheeler vehicles, the identified architecture is Mask R-CNN with a ResNet-131 backbone. The configuration includes 43 bottleneck blocks, with each stage comprising 15, 8, 5, and 15 blocks, respectively. Since each bottleneck block consists of 3 convolutional layers, this results in 129 layers (43 × 3 = 129). There are two extra layers: an input layer with a 7 × 7 convolution (64 kernels with stride 2) and a fully con-

Table 3.7: Comparison of top-5 different models, generated by the proposed approach on CaltechGraz dataset

Method	#Blocks	Block type	Class loss	Bbox loss	Optimizer	Activation	mAP (%)	
							Valid	Test
Mask R-CNN-Resnet122	[15, 8, 5, 12]	Bottleneck Block	Cross Entropy loss	MSE loss	SGD	ReLu	24.6	32.0
Mask R-CNN-Resnet64	[15, 11, 1, 4]	Basic Block	Cross Entropy loss	L1 loss	SGD	ReLu	32.1	33.3
Mask R-CNN-Resnet140	[30, 11, 1, 4]	Bottleneck Block	Focal loss	MSE loss	Adam	ReLu	28.1	34.3
Mask R-CNN-Resnet48	[7, 11, 1, 4]	Basic Block	Cross Entropy loss	L1 loss	Adam	ReLu	31.5	36.1
Mask R-CNN-Resnet64	[15, 11, 1, 4]	Basic Block	Cross Entropy loss	MSE loss	Adam	ReLu	**35.3**	**36.9**

Note: The best values are highlighted in bold.

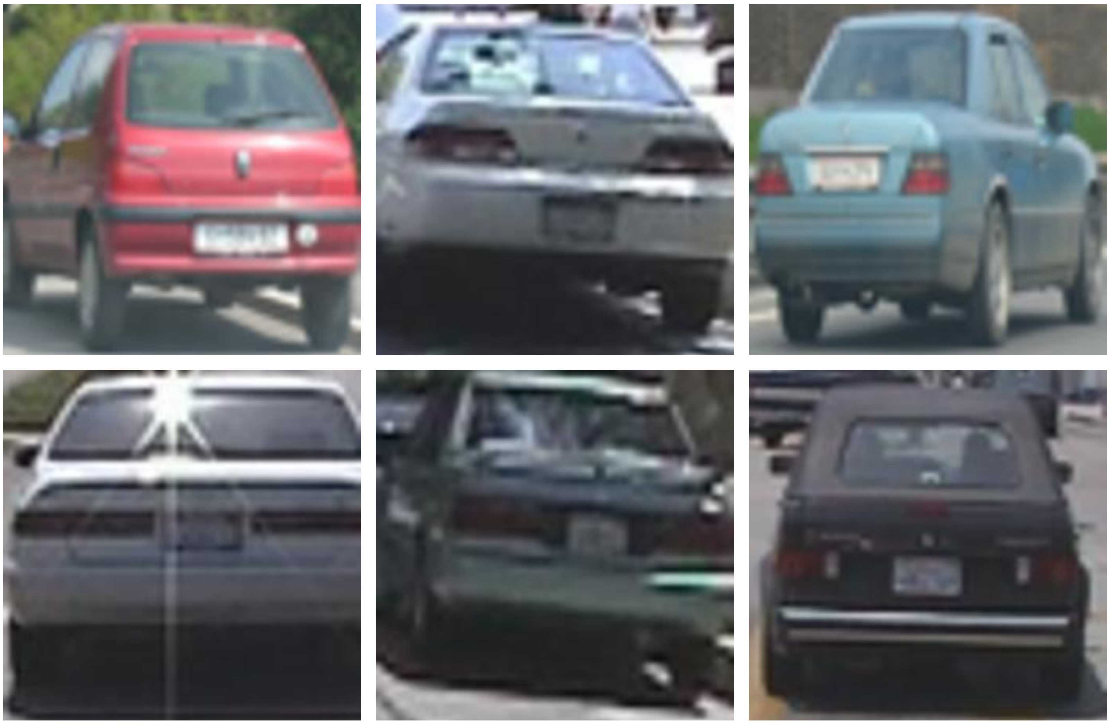


Figure 3.11: Images from CaltechGraz dataset, with the status of the brake light as non-glowing

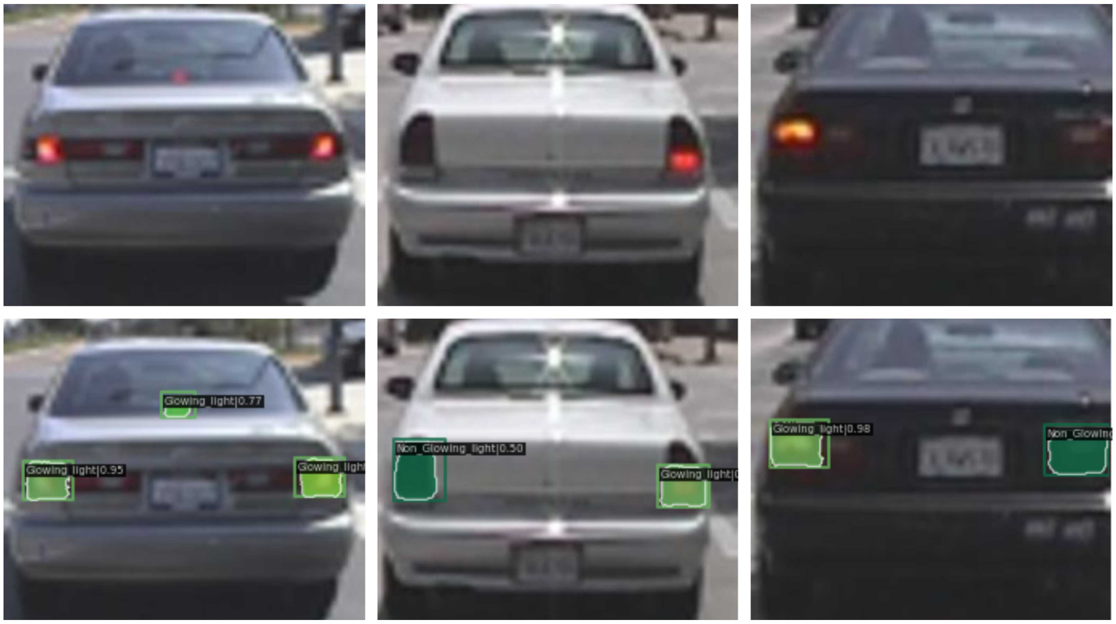


Figure 3.12: Glowing brake light status detection results of the proposed approach on CaltechGraz dataset

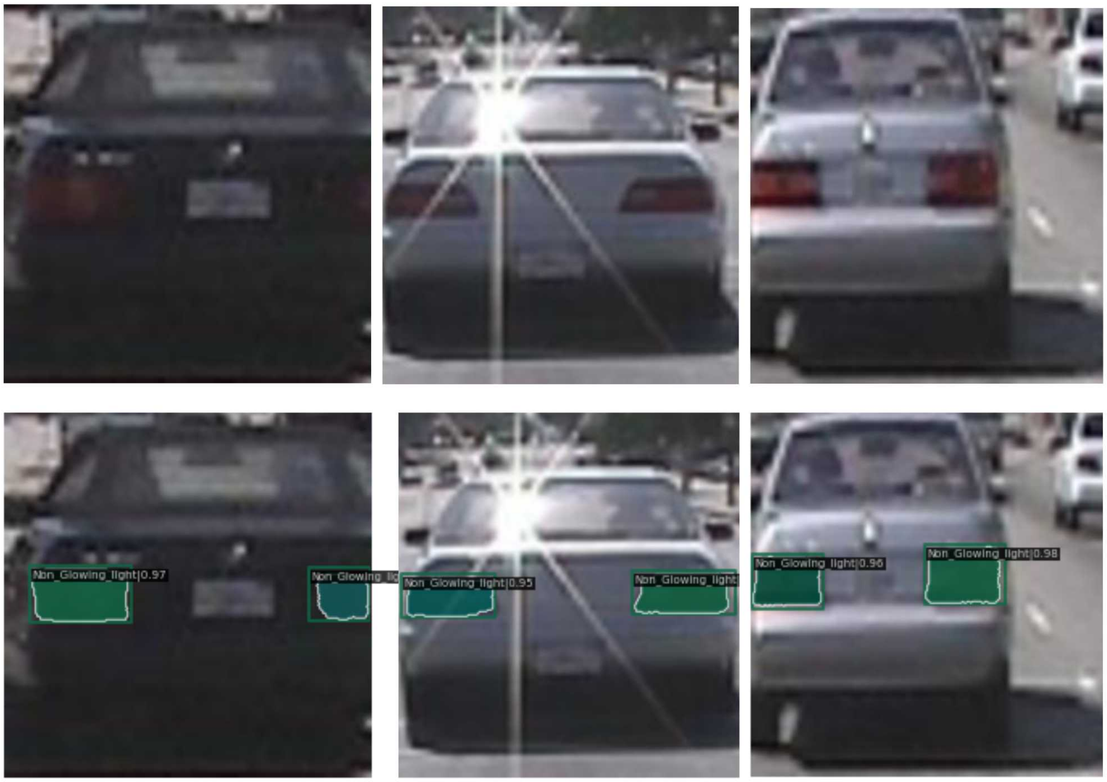


Figure 3.13: Non-Glowing brake light status detection results of the proposed approach on CaltechGraz dataset

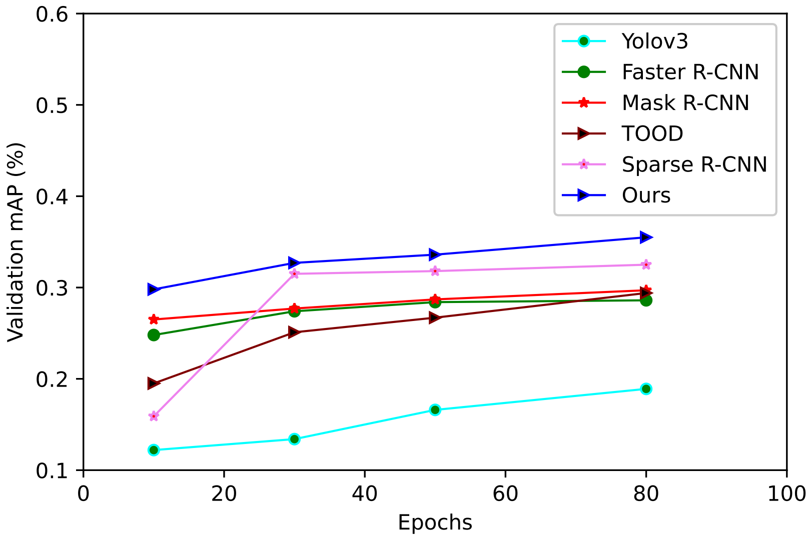


Figure 3.14: Change in validation mAP against training epochs for YOLOv3, Faster R-CNN, Mask R-CNN, TOOD, Sparse R-CNN and proposed model on CaltechGraz dataset

Algorithm 3.1 Pseudocode to calculate accuracy

```
Input: $Model$, input_images $(I)_{i=1}^{N}$, ground_truth_labels $(G)_{i=1}^{N}$, IoU_threshold $(T)$.
Output: $accuracy$
Create $(P)_{i=1}^{N}, (E)_{i=1}^{N}$    ▷ $P_i$, $E_i$ stores predicted and expected ground truth labels
for $i = 1, 2, ...., N$ do    ▷ Here, $N$ is number of images given as input
    Initialize $S_1, S_2 = \emptyset$    ▷ Temporary lists to store class labels of bboxes

    Predict the bboxes along with their class labels and scores in $I_i$ using $Model$
    $S_1 \leftarrow$ predicted classes of bboxes with scores greater than T
    if $Glowing\_light \notin S_1$ then
        $P_i \leftarrow Non\_Glowing\_light$    ▷ $I_i$ has no glowing light
    else
        $P_i \leftarrow Glowing\_light$    ▷ $I_i$ has glowing light

    $S_2 \leftarrow$ ground truth labels of bboxes given in $G_i$
    if $Glowing\_light \notin S_2$ then
        $E_i \leftarrow Non\_Glowing\_light$    ▷ $I_i$ has no glowing light
    else
        $E_i \leftarrow Glowing\_light$    ▷ $I_i$ has glowing light


Calculate $accuracy$ from $(P)_{i=1}^{N}, (E)_{i=1}^{N}$ using Eq. 3.6 .
```

nected layer, bringing the total to 131 layers. For the four-wheeler vehicles, the identified architecture is Mask R-CNN with a ResNet-64 backbone. The configuration comprises 31 basic blocks, distributed across stages as 15, 11, 1, and 4 blocks, respectively. Since each basic block contains two convolutional layers, the total becomes 62 layers (31 × 2 = 62). As with the two-wheeler model, the additional two layers include an input layer with a 7 × 7 convolution (64 kernels with stride 2) and a fully connected layer, resulting in 64 layers. Furthermore, the analysis of the combination of parameter values for the best-performing models of both two-wheeler and four-wheeler vehicles suggests that Cross Entropy loss, MSE loss, and ReLU activation effectively optimise performance across both cases. However, the choice of optimizers and block types differs between the two models, reflecting differences in their NAS exploration process. Specifically, SGD was used for the two-wheeler model and Adam for the four-wheeler model, while the two-wheeler model employed the Bottleneck block, and the four-wheeler model used the Basic block. These distinctions suggest that different architectures and optimizers were suited for different types of vehicles, leading to optimal performance in their respective type of vehicles.

3.2.4 Computational complexity analysis

The time complexity of a GA is determined by the number of generations (T), population size (N), and the fitness evaluation cost. The most computationally expensive task is evaluating the fitness of each individual, which involves training the neural network models and takes $O(N \times E_{\text{train}})$, where E_{train} is the time required to train a model on the dataset. Selection, crossover, and mutation operations each take $O(N)$ per generation. Since these steps are repeated for T generations, the overall time complexity of the proposed GA based NAS approach is $O(T \times (N \times E_{\text{train}} + N))$, with fitness evaluation being the dominant factor.

3.2.5 Effectiveness of brake light status detection

To assess the effectiveness of the proposed approach, we compare the proposed approach with the existing brake/tail light detection models. In [79], YOLOv3-tiny is used as the baseline model with an SPP module and focal loss for brake/tail light detection task. The top-2 models identified by the proposed approach and the YOLOv3-tiny-spp-focal [79] model are evaluated on two-wheeler vehicles (NITW-MBS dataset), and four-wheeler vehicles (CaltechGraz dataset) and their performance are given in Table 3.8. From the table, it can be observed that the proposed approach achieves 54.5% and 52.5% mAP for validation and test data, respectively, for the NITW-MBS dataset. It can also be observed that the proposed approach achieves 35.3% and 36.9% mAP on validation and test data, respectively, on the CaltechGraz dataset. It can be concluded that the proposed approach outperforms the existing approaches for two-wheeler and four-wheeler brake light status detection.

Vehicles typically have one or more brake lights/tail lights. For computing accuracy, we assume that the vehicle has applied the brakes if at least one of the detected brake lights is glowing; otherwise, we consider it non-glowing. The Pseudocode used for computing accuracy is given in Algorithm 3.1. This pseudocode uses the model identified by the proposed approach, denoted by $Model$, the N input images, denoted by $(I)_{i=1}^{N}$, the cor-

Table 3.8: Performance comparison of the top-2 models identified by the proposed approach and the existing approach on the NITW-MBS and CaltechGraz datasets for brake light detection task

Method	Dataset	mAP(%)		Params
		valid	test	
YOLOv3-tiny-spp-focal [79]	NITW-MBS (two-wheeler)	40.5	37.8	8.9 M
Mask R-CNN-Resnet62 (**Our top-2 model**)		54.1	50.6	36.7 M
Mask R-CNN-Resnet131 (**Our top-1 model**)		**54.5**	**52.2**	98.3 M
YOLOv3-tiny-spp-focal [79]	CaltechGraz (four-wheeler)	22.7	30.3	8.9 M
Mask R-CNN-Resnet48 (**Our top-2 model**)		31.5	36.1	27.1 M
Mask R-CNN-Resnet64 (**Our top-1 model**)		**35.3**	**36.9**	27.7 M

Note: The best values are highlighted in bold.

responding ground truth labels, represented by $(G)_{i=1}^{N}$ and IoU threshold, represented by T as inputs. These inputs are used to predict the bounding boxes and their assigned class labels (i.e. $Glowing\_light/Non\_Glowing\_light$) for a given IoU threshold (The formula in Eq. 3.3 is used to calculate IoU). The input image I_i is considered to be $Glowing\_light$ if at least one of the bounding boxes is assigned the glowing class label; otherwise, we consider it to be $Non\_Glowing\_light$, and we store the corresponding class label in P_i. Similarly, from the ground truth labels, we consider image I_i as $Glowing\_light$ if at least one of its bounding box class labels is glowing, and $Non\_Glowing\_light$ if none are glowing, and we store the corresponding class label in E_i. The efficiency is determined from P and E by computing accuracy using Eq. 3.6. The performance of the proposed and existing approaches for different values of IoU thresholds is given in Table 3.9. From Table 3.9, it can be observed that the proposed approach gives better results than the existing approach on the NITW-MBS dataset and on the CaltechGraz dataset.

3.3 Summary

In this chapter, a genetic algorithm based Neural Architecture Search approach for vehicle brake light detection is proposed based on the Mask R-CNN object detection model. A genetic algorithm is used as a search strategy to explore the search space, thereby identifying the optimum backbone architecture and training attributes for the brake light detection task. The object detection model identified by the proposed approach achieves a mean accuracy of 97.14% on the proposed two-wheeler (NITW-MBS) dataset and 89.44% on the four-wheeler (CaltechGraz) dataset, respectively. The resulting model exhibits significant improvement over the existing approaches for both two-wheeler and four-wheeler vehicle brake light status detection. This indicates that the proposed approach can explore the search space to identify the optimum architecture of the object detection model and its training attributes for the brake light detection task.

Table 3.9: Accuracy of brake light status detection of the proposed and existing approaches, for IoU threshold range of 0.3 to 0.9, for two-wheeler and four-wheeler vehicles. The best values are highlighted in bold

Method	Dataset	Accuracy(%) for various IoU thresholds							Mean Accuracy(%)
		0.3	0.4	0.5	0.6	0.7	0.8	0.9	
YOLOv3-tiny-spp-focal [79]	NITW-MBS (two-wheeler)	96.21	97.77	97.77	95.07	55.68	40.90	40.90	74.90
Mask R-CNN-Resnet62 (**Our top-2 model**)		93.56	96.96	96.96	98.86	98.86	97.34	97.34	97.12
Mask R-CNN-Resnet131 (**Our top-1 model**)		92.80	96.21	98.10	98.10	98.86	98.86	97.34	**97.14**
YOLOv3-tiny-spp-focal [79]	CaltechGraz (four-wheeler)	87.50	89.77	90.90	92.04	53.40	43.18	43.18	71.42
Mask R-CNN-Resnet48 (**Our top-2 model**)		82.95	87.50	90.90	92.04	92.04	93.18	87.50	**89.44**
Mask R-CNN-Resnet64 (**Our top-1 model**)		84.09	86.36	88.63	89.77	89.77	90.90	85.22	87.82

Note: The best values are highlighted in bold.

Chapter 4

Differential Evolution based Optimization of Deep Neural Networks for Vehicle Brake Light Detection

This chapter proposes a NAS based optimization of deep neural networks using a differential evolution algorithm for the task of vehicle brake light detection.

Chapter Organization:

Preliminaries for the proposed NAS based approach is covered in Section 4.1. The proposed approach is described in Section 4.2. Section 4.3 presents the experimental results and analysis. Finally, Section 4.4 presents the summary of the work.

4.1 Preliminaries

This section covers the background of the Differential Evolution algorithm, which is used as the NAS search strategy in this objective.

4.1.1 Differential Evolution (DE)

DE is an evolutionary optimization method first proposed by Storn and Price [134]. DE has several advantages over other evolutionary algorithms, such as obtaining global optima with few control parameters and fast convergence [135]. DE can also be applied to difficult optimization problems in various spaces, including multi-modal, noisy, and multi-dimensional spaces [136]. In DE, domain knowledge or constraints can be incorporated into the search methodology for discrete or continuous optimization of neural network models. In the last few years, it has been successfully applied in a broad range of industrial and academic research areas, including machine learning [137], signal processing [138], and pattern recognition [139]. Despite its ease of implementation and extensive use in engineering, it is still uncommon to utilize it to solve neural network optimization problems like image classification, object detection, and image segmentation.

The four phases of DE's working principle are initialization, mutation, crossover, and selection. First, an initial random population of target vectors of length M is generated. The target vectors are represented as shown in Eq. 4.1.

$$X_i^t = \{X_{i,1}, X_{i,2}, X_{i,3}, ..., X_{i,M}\} \tag{4.1}$$

Here, X_i^t denotes the target vector at t^{th} generation, where $i = 1, 2.., N$; N denotes the population size, and t denotes the generation number.

Then, for each target vector X_i, a mutation operation is carried out to produce a donor vector using Eq.4.2

$$H_i^t = X_{r1} + F(X_{r2} - X_{r3}) \tag{4.2}$$

Here, H_i^t denotes the donor vector at t^{th} generation, $X_{r1}, X_{r2}, X_{r3} \in X_i^t$ are randomly chosen target vectors, where $r1 \neq r2 \neq r3 \neq i$ and F is the scaling factor.

To add diversity to the population, a crossover operation is applied on the donor vector and target vector to get trial vector U_i^t, using Eq. 4.3

$$U_i^t = \begin{cases} H_i^t, \;\; if \;\; rand \leq C_r \;\; or \;\; j = rand(i) \\ X_i^t, \;\; otherwise \end{cases} \tag{4.3}$$

Here, C_r is a crossover rate and $rand(i)$ is randomly selected index, where $j = 1, 2, ...M$.

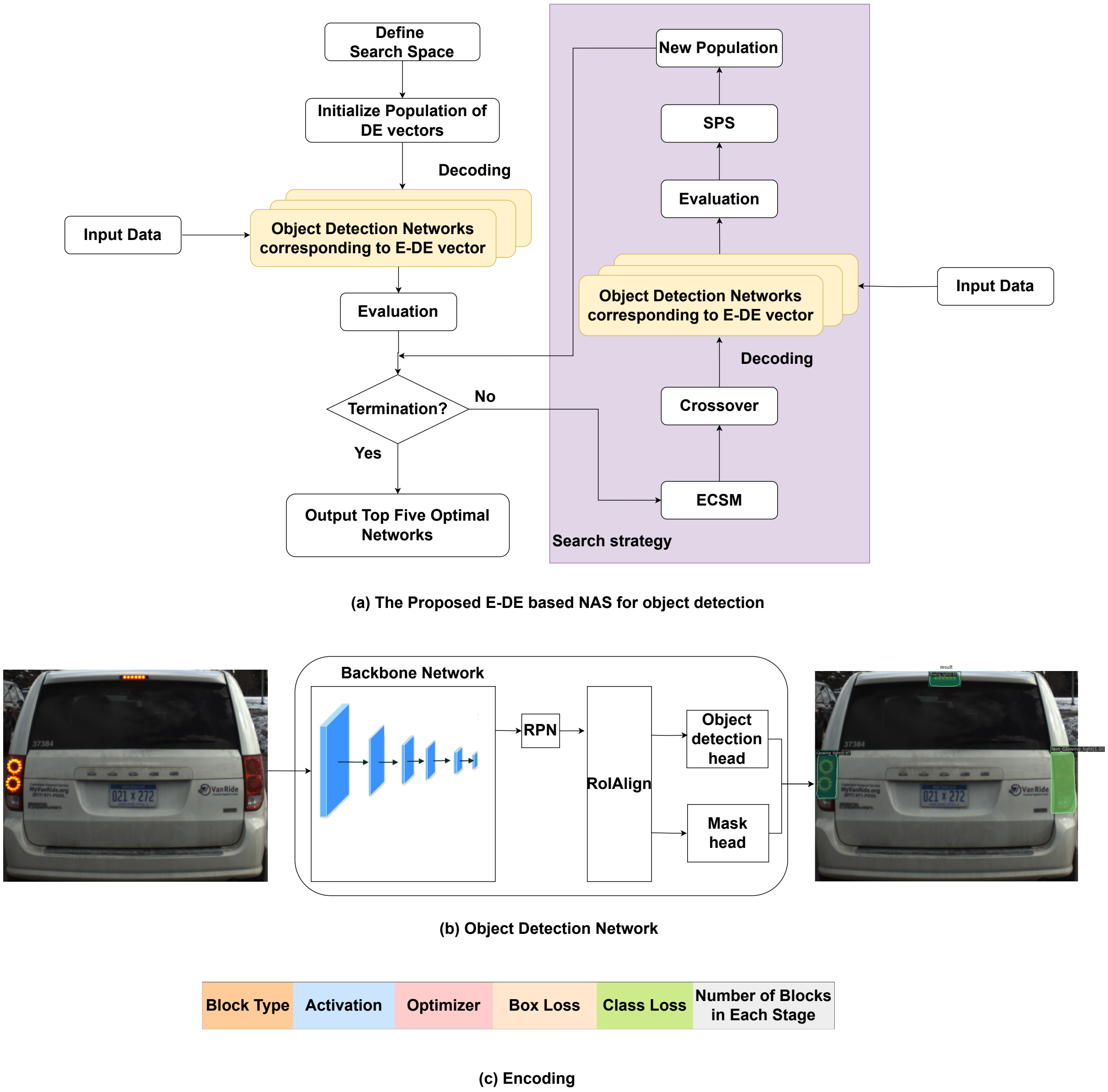
Next, greedy selection strategy is performed on the trial vector and on the target vector using Eq. 6.6. Here vector with a higher fitness value will be selected for the next generation.

$$X_i^{t+1} = \begin{cases} U_i^t, \ \ if \ \ f(U_i^t) > f(X_i^t) \\ X_i^t, \ \ otherwise \end{cases} \tag{4.4}$$

Where $f(U_i^t)$ and $f(X_i^t)$ are the fitness values of the trial vector and target vector, respectively.

4.2 Proposed approach

This section first discusses the proposed search space designed for the target object detection network, including the backbone architecture and training parameters for brake light status detection in Section 4.2.1. Then, a search strategy based on a modified version of the DE algorithm, named E-DE, is discussed in Section 4.2.2. Fig.4.1 (a) illustrates the overall E-DE based NAS framework used for designing the brake light detection system in the proposed approach. The framework begins by generating an initial population that is randomly initialized. Each DE vector in the population undergoes an encoding process, as depicted in Fig.4.1 (c). This encoded vector is then decoded, resulting in a two-stage object detection network represented in Fig. 4.1 (b). The model's performance is evaluated on vehicle brake light detection data, and its fitness value is computed. The search for the optimum DNN model continues until the termination of the DE cycle.



(a) The Proposed E-DE based NAS for object detection

(b) Object Detection Network

Block Type	Activation	Optimizer	Box Loss	Class Loss	Number of Blocks in Each Stage

(c) Encoding

Figure 4.1: (a) The proposed E-DE based NAS for optimizing an object detection network (b) Two-stage object detection network (c) Encoding of E-DE Vector.

4.2.1 Search space design for object detection

Similar to the work done in Chapter 3, to detect the brake lights that are small in size, we consider a modified two-stage Mask R-CNN object detector in this work. We considered different types of blocks and varying sizes of depth to search for a better backbone. In addition, different types of training parameters like activation functions, optimization techniques, and loss functions are explored in the search to find the optimum architecture for brake light detection task.

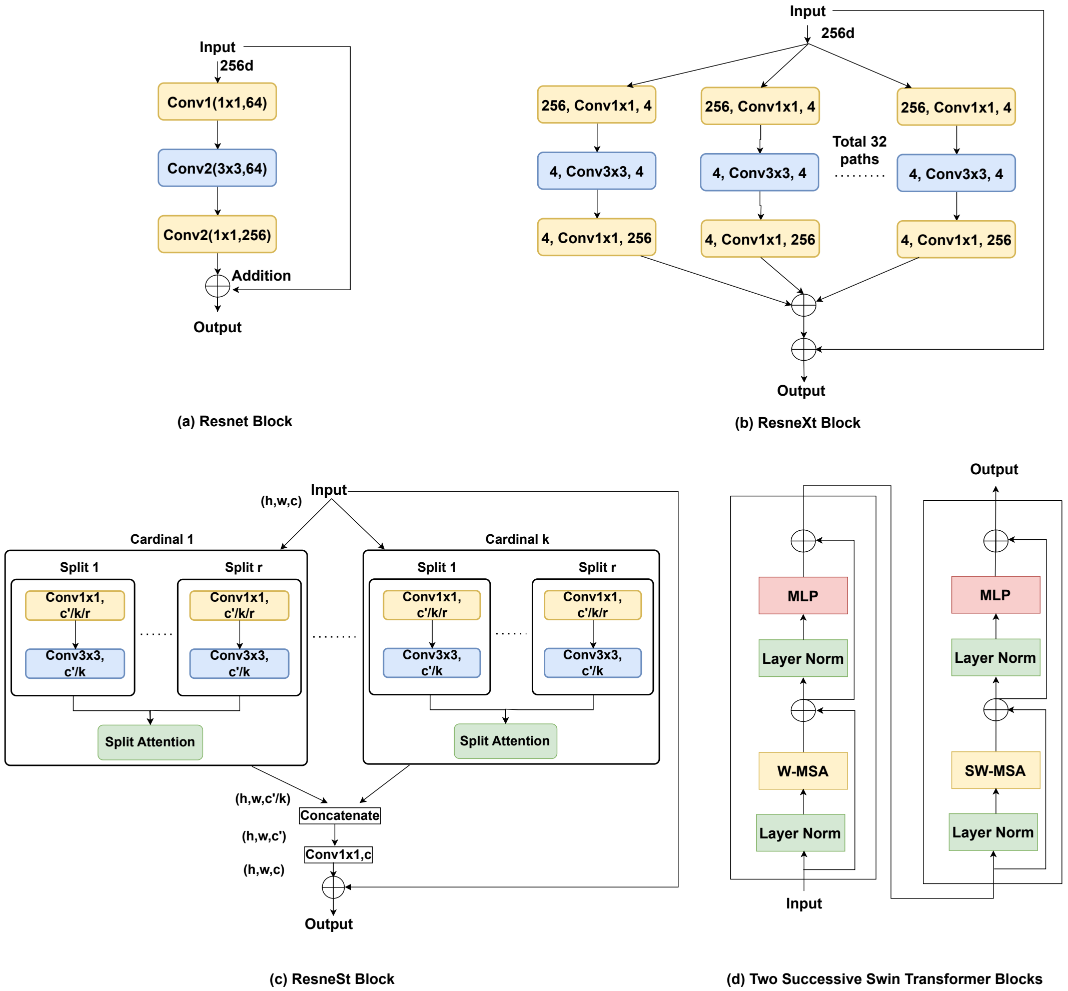
The backbone architectures of object detectors are crucial for the effective recognition of objects. The effectiveness of object detectors heavily depends on the features extracted by the backbone. Therefore, we included parameters related to the backbone in the search space. Backbone search space consists of a sequence of blocks. Each part of the backbone could be divided into several stages according to the resolution of the output features, where the stage refers to a number of blocks fed by the features with the same resolution. In this work, the proposed search space is based on four kinds of blocks: Resnet block [46], ReneXt block [47], ReneSt block[48] and Swin transformer block [49], as shown in Fig. 4.2. The *number of blocks* in each stage of the backbone varies from 1 to 16. The number of stages considered is 4. We allow the same number of blocks in each stage except for the last stage, where three blocks are used in the last stage.

Apart from backbone search, we have also included training parameters like *activation function*, *optimizer*, *box loss*, and *class loss* in the proposed search space. For the *activation function*, we have used ReLU, GELU, CELU, Mish, for the *optimizer*, we have used SGD, Adam, AdamW; for the *box loss*, we have used MSE loss, L1 loss, Smooth L1 loss, and finally for the *class loss*, we have used Cross Entropy loss, Focal loss. The details of the parameters and their values explored in the search space are described in Table 4.1.

A complete architecture is encoded as a vector of length six. The first placeholder encodes a block type, the second placeholder encodes activation function, the third placeholder encodes box loss, the fourth placeholder encodes optimizer, the fifth placeholder encodes class loss, and the last placeholder encodes the number blocks as shown in Fig. 4.1 (c).

Table 4.1: Parameters in the search space, considered in this work

Type of parameter	**Range**
Type of block	{Swin,ResneSt,ResneXt,Resnet}
Activation	{ReLU, Mish, GELU, CELU}
Optimizer	{Adam, AdamW, SGD}
BBox loss	{L1 loss, MSE loss, Smooth L1 loss}
Class Loss	{Cross Entropy loss, Focal loss}
Number of blocks in each stage	[1-16]



(a) Resnet Block

(b) ResneXt Block

(c) ResneSt Block

(d) Two Successive Swin Transformer Blocks

Figure 4.2: Type of blocks considered in this work (a) Resnet Block (b) ResneXt Block (c) ResneSt Block (d) Two Successive Swin Transformer Blocks

4.2.2 E-DE based NAS optimization

This section introduces the adapted mutation and selection strategies used within the proposed E-DE based NAS framework.

Evaluation Correction based Selection for Mutation (ECSM): Traditional mutation strategies use random selection or the best vector as parents to generate donor vectors. However, estimating performance only based on validation mAP may lead to a better network, but it is not efficient in terms of computation. Therefore, we have adapted the evaluation correction based selection strategy from [17] to choose individuals for mutation operation. In this method, the network architecture is evaluated based on the validation mAP. If there is a noticeable difference in the validation mAP, the network with the higher mAP is chosen. However, if the validation mAP scores are similar, the network with fewer parameters is preferred. This method ensures that the network with superior validation mAP, as well as the network with less number of parameters, is selected. The logic for this method is given in Algorithm 4.2.

Algorithm 4.2 Evaluation Correction based Selection for Mutation

```
Input: Population ($X_i^t$), population size $N$, number of parameters $z$ and fitness $\mu$ of each
    individual in $X_i^t$, threshold $\alpha$ in individual fitness, scale factor $F$.
Output: Donor vector ($H_i^t$)
for $i = 1, 2, ...., N$ do
    $E \leftarrow$ Select three individuals at random from the population
    while $|E| > 1$ do
        $X_1, X_2 \leftarrow$ Select 2 individuals from $E$
        $E \leftarrow E - \{X_1, X_2\}$
        $\mu_1, \mu_2 \leftarrow$ Fitness of $X_1, X_2$
        $z_1, z_2 \leftarrow$ Number of parameters of $X_1, X_2$
        if $|\mu_1 - \mu_2| < \alpha$ then
            Put the individual with fewer number of parameters in $\{X_1, X_2\}$ back into $E$
        else
            Put the individual with greater accuracy in $\{X_1, X_2\}$ back into $E$
    $X_{\text{best}} \leftarrow$ Return the best individual in $E$
    $X_{r1}, X_{r2} \leftarrow$ Return the remaining individual from $(E - X_{\text{best}})$
    $H_i^t \leftarrow X_{\text{best}} + F \times (X_{r1} - X_{r2})$
Output the donor Vector $H_i^t$
```

The ECSM initially chooses three individuals randomly from the population. The selection process begins by comparing two vectors, and the better one is determined. This selected vector is then compared with the last individual from the set to identify the optimal architecture. In cases where the mAP scores of the two vectors show no significant difference (i.e., the difference is below the threshold α), ECSM considers the network with fewer parameters. After the selection process, the best vector is chosen as the base vector, and the other two vectors are used to create difference vectors. This difference vector is added to the base vector to generate a donor vector, which further contributes to the population's evolution and the search for optimal network architecture.

Species Protection based Selection (SPS):

Traditional DE algorithms typically employ a greedy selection strategy, which favors exploitation but tends to reduce the diversity of the population over time. However, in the evolutionary process, maintaining diversity in the population of network architectures is crucial for improving the algorithm's overall performance. To address this issue, this work explores the use of Species Protection based environmental Selection operation (SPS) [17]. The process for this SPS strategy is described in Algorithm 4.3.

SPS starts by dividing the population P into different species, denoted as P_{class}. From P_{class}, species Φ is selected with uniform probability. An individual P_{best} is randomly chosen from Φ to promote diversity. However, a challenge arises regarding the lack of competition among different species. To overcome this challenge, SPS introduces a random number r that controls whether to employ the SPS strategy to balance competitive pressures within and outside species. Additionally, SPS incorporates an elite retention strategy to protect the optimal individuals in the population from being eliminated during evolution. The strategy involves retaining the top $2N \times \gamma$ individuals in the population before species division, where N represents the population size and γ is a retention parameter. When dividing the species into P_{class}, SPS prioritizes the network's building block type. The block type significantly influences the performance when compared to the other parameters within the search space.

Algorithm 4.3 Species Protection based Selection

```
Input: Population $P = X^t \cup U^t$, population size $N$, elite rate $\gamma$ in the population.
Output: The new population $X^{t+1}$.
$X_i^{t+1} \leftarrow$ Select $2N \times \gamma$ individuals with the highest fitness from population $P$ using the elite retention strategy.
$P$ is divided into 4 distinct species, denoted as $P_{\text{class}}$, based on the first field of DE vector representing the type of block, which ranges from 0 to 3.
while $|X_i^{t+1}| < N$ do
    $r \leftarrow$ Generate a random number.
    if $r < 0.5$ then
        Randomly select a species $\Phi$ from $P_{\text{class}}$.
        Select the best individual $P_{\text{best}}$ from $\Phi$ by using Algorithm 4.2 lines 3 to 15.
    else
        Select the best individual $P_{\text{best}}$ from $P$ by using Algorithm 4.2 lines 3 to 15.
    $X_i^{t+1} \leftarrow X_i^{t+1} \cup P_{\text{best}}$.
Return the new population $X^{t+1}$.
```

The overall framework of proposed E-DE based NAS optimization:

The proposed E-DE based NAS framework is given in Algorithm 4.4, which incorporates modified mutation (ECSM) and selection (SPS) operations. The proposed approach begins by randomly generating an initial population X^0. Fitness values are then calculated for the N individuals in X^0. The framework proceeds with T rounds of iterative evolution.

During the t^{th} generation evolution process, three vectors are randomly selected, and the ECSM is applied to generate donor vectors for each target vector. Subsequently, a trial vector is generated by performing a binomial crossover between the target vector and the donor vector. This process continues until a set of trial vectors (U^t) is generated with the same size as that of the population. Finally, the proposed approach utilizes an SPS operation to select the next generation population X^{t+1} from the union of the current target vectors X^t and the trial vectors U^t. This selection process ensures diversity and balances competitiveness. The framework then proceeds to the next round of the evolutionary process. After completing the T-round evolutionary process, the top 5 optimal network architectures are selected.

Algorithm 4.4 The overall framework of proposed E-DE based NAS optimization

```
Input: train_images $I_{train}$, number of evolutionary iterations $T$, population size $N$.
Output: The top-5 optimal network architectures.
$X^0 \leftarrow$ Generate a randomly initialized population.
Decode the vectors of $X^0$ into object detection networks
Calculate the fitness and parameters of each decoded network using $I_{train}$.
$t \leftarrow 1$.
while $t < T$ do
    $U^t \leftarrow \{\}$.
    while $|U^t| < N$ do
        Apply mutation on each vector to generate donor vector $H_i^t$ using Algorithm 4.2.
        Generate trial vector $U_i^t$ with binomial crossover operation using Eq. 4.3.
        Decode the trial vector $U_i^t$ into object detection networks.
        Calculate the fitness and parameters of the decoded network using $I_{train}$.
        $U^t \leftarrow U^t \cup \{U_i^t\}$.
    $X^{t+1} \leftarrow$ Generate next-generation populations from $X^t \cup U^t$ using Algorithm 4.3.
    $t \leftarrow t + 1$.
Return the top-5 optimal network architectures based on their fitness.
```

4.3 Experimental results and analysis

In this section, we present the experimental evaluation of the proposed approach on the four-wheeler and two-wheeler brake light detection datasets. We first discuss the experimental settings. Section 4.3.1 presents the evaluation of the proposed approach on four-wheeler vehicles on the CaltechGraz dataset [50, 129] and UC Merced Vehicle Rear Signal dataset [25]. Section 4.3.2 presents the evaluation of the proposed approach on the proposed NITW-MBS dataset. Finally, we analyze the effectiveness of the proposed approach for brake light detection task on these datasets.

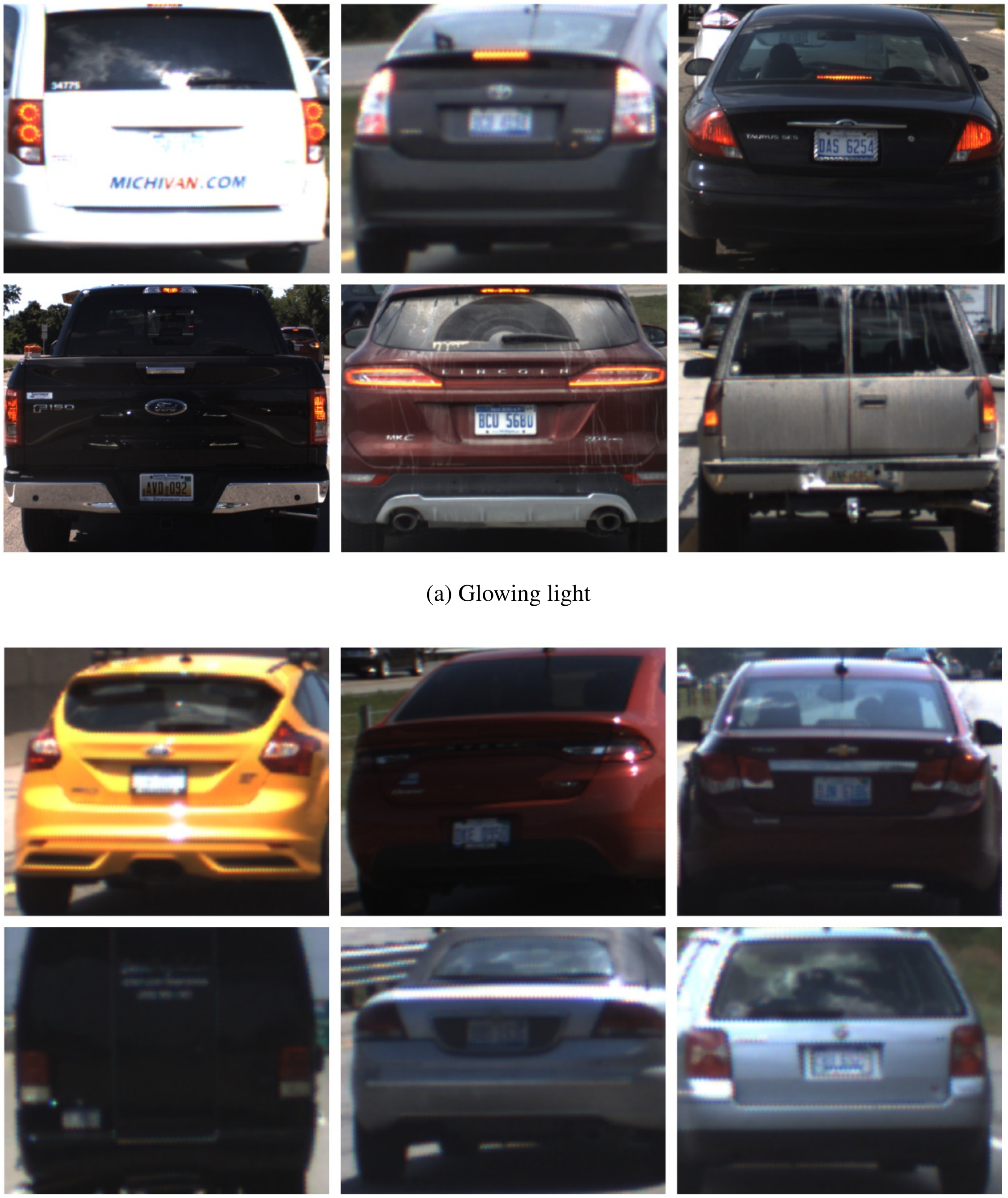
Experimental Settings:

The experimental study in this work is performed on a computer with an Intel Xeon(R) Silver 4110 CPU running at 2.10GHz, 64GB of RAM, and one NVIDIA GeForce RTX 2080 Ti GPU, with CUDA 11.1 in Linux platform with PyTorch framework. During the search space exploration, if Adam or AdamW are selected as the optimizer, then the weight decay of 0.0001 and the initial learning rate of 0.0001 are considered. Similarly, if SGD is selected as the optimizer, then the momentum of 0.9 and the initial learning rate of 0.02 are considered. In the evolutionary process, selecting the DE parameters is critical for the algorithm's success. A trial-and-error approach generally determines the optimal values of these parameters and can vary depending on the specific problem being addressed. In this study, a scaling factor of $F = 0.5$ and a crossover rate of $C_r = 0.5$ were considered similar to existing literature [140]. The other parameters of the proposed E-DE, elite rate γ of 30 $\%$ and a threshold α of 2, were considered. The vector length M was set to six since the search space consisted of six parameters, and the population size N was set to 20. The experiments were conducted for a total of $T(= 10)$ generations. Initially, all models were trained for 20 epochs in the evolutionary process. Subsequently, the top-5 distinct models generated by the proposed approach were trained for an additional 80 epochs.

4.3.1 Experimental study for four-wheeler brake light detection

This section evaluates the effectiveness of the proposed approach for four-wheeler vehicles. The comparison of the performance of the proposed approach against the existing manually designed object detection approaches, as well as NAS based object detection approaches, is presented in this section. The performance of the top-5 distinct models identified by the proposed approach is also discussed in this section.

In addition to the CaltechGraz dataset [50], which is used in the Chapter 3, we considered the UC Merced Vehicle Rear Signal dataset [25] to evaluate the proposed approach in this contribution. For the CaltechGraz dataset, we selected 490 images and annotated them in the COCO data format. Fig. 3.10 and Fig. 3.11 show some of the observations in this dataset. The statistics of the CaltechGraz dataset considered in this work are given in Table 3.5. For the UC Merced Vehicle Rear Signal dataset, we selected 6375 frames and annotated them in the COCO data format. Fig. 4.3 shows some of the sample images in this dataset. In this dataset, we have selected every tenth frame from a given sequence of frames from each observation. The statistics of the UC Merced Vehicle Rear Signal dataset considered in this work are given in Table 4.2



(a) Glowing light

(b) Non-glowing light

Figure 4.3: Images from UC Merced Vehicle Rear Signal dataset with the status of the brake light

Table 4.2: Number of Images, Bounding boxes for each class in the training, validation, and test data of UC Merced Vehicle Rear Signal dataset considered in this study

Data	Glowing light Bounding boxes	Non-Glowing light Bounding boxes	#Images
#training	6644	3823	4145
#validation	1171	1360	960
#test	1877	1368	1270

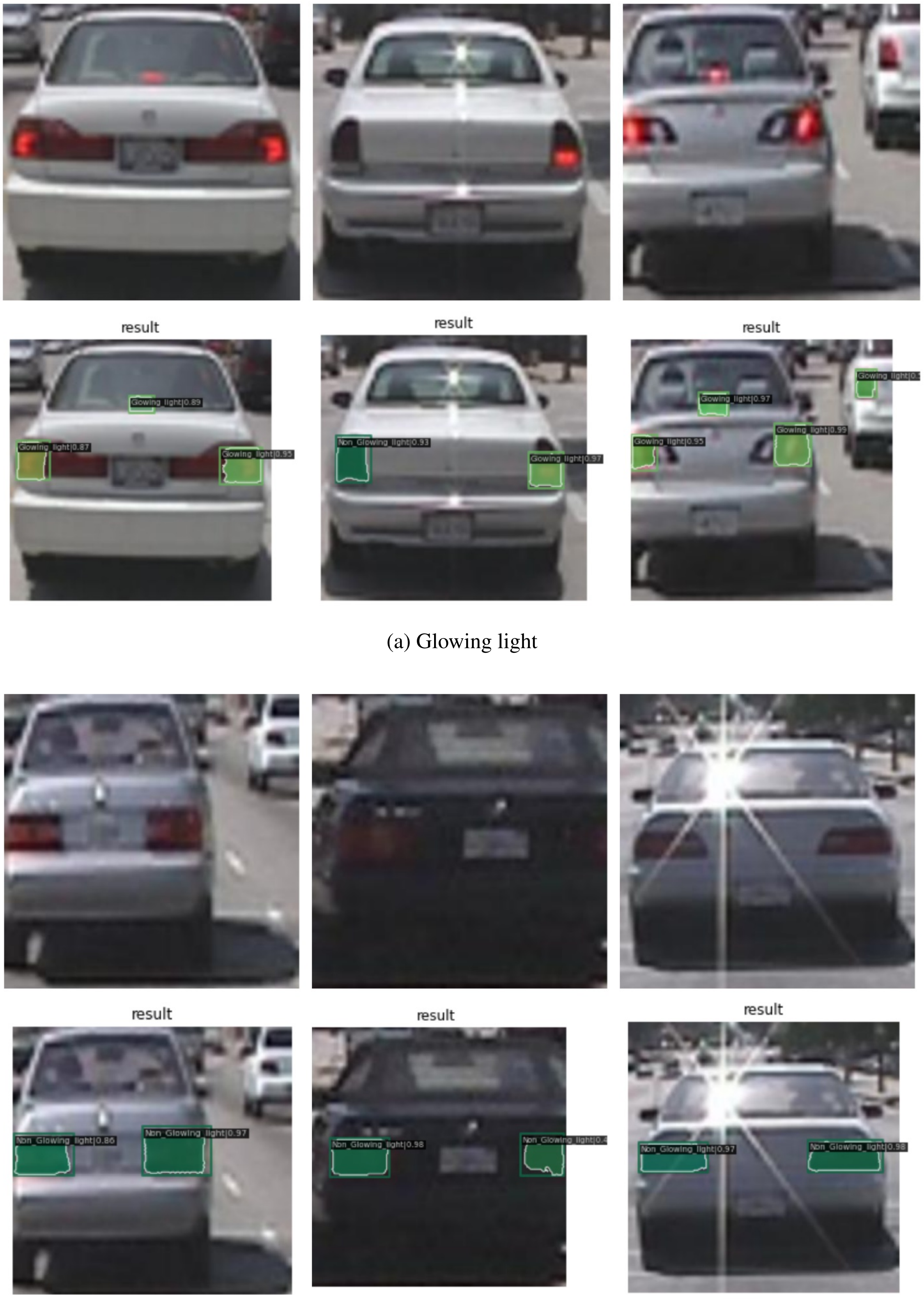
We conduct a comprehensive evaluation of various manually designed object detectors and propose an automated approach for brake light detection task. The evaluated models include both one-stage models, such as YOLOv3 [10], YOLOF [141], and TOOD [131], as well as two-stage models, such as Faster R-CNN [14], Mask R-CNN [15], Sparse R-CNN [132], and a NAS based object detection model MAE-DET [18]. The evaluation results are summarized in Table 4.3. From the table, it can be observed that the proposed model outperforms the existing manually designed and NAS based object detection models by a considerable margin. The optimal model identified by the proposed approach on the CaltechGraz dataset is trained on the UC Merced Vehicle Rear Signal dataset for cross-dataset evaluation. Fig. 4.4 and Fig. 4.5 show the detection of the brake light status by the proposed approach on some test images in the CaltechGraz dataset and UC Merced Vehicle Rear Signal dataset. Fig. 4.7 shows the plot of variations of mAP for the object detectors in Table 4.3 against training epochs.

Table 4.3: Comparison of mAP of the existing object detection models with proposed approach on CaltechGraz dataset and UC Merced Vehicle Rear Signal dataset

Method	Dataset				Params
	CaltechGraz		UC Merced		
	Valid (%)	Test (%)	Valid (%)	Test (%)	
YOLOv3 [10]	18.9	20.9	31.8	31.6	61.5 M
TOOD [131]	29.4	24.4	40.5	36.3	22.4 M
YOLOF [141]	36	36.4	36.5	34.4	32.8 M
Faster R-CNN [14]	28.6	31.1	40.2	36.4	63.6 M
Mask R-CNN [15]	29.7	33.1	39.1	35.9	31.6 M
Sparse R-CNN [132]	32.5	36.2	37.9	34.3	96.6 M
MAE-DET [18]	36.4	38.6	41.6	37.2	52.9 M
Ours	**40.4**	**39.3**	**42.2**	**40.4**	54.3 M

Note: The best values are highlighted in bold.

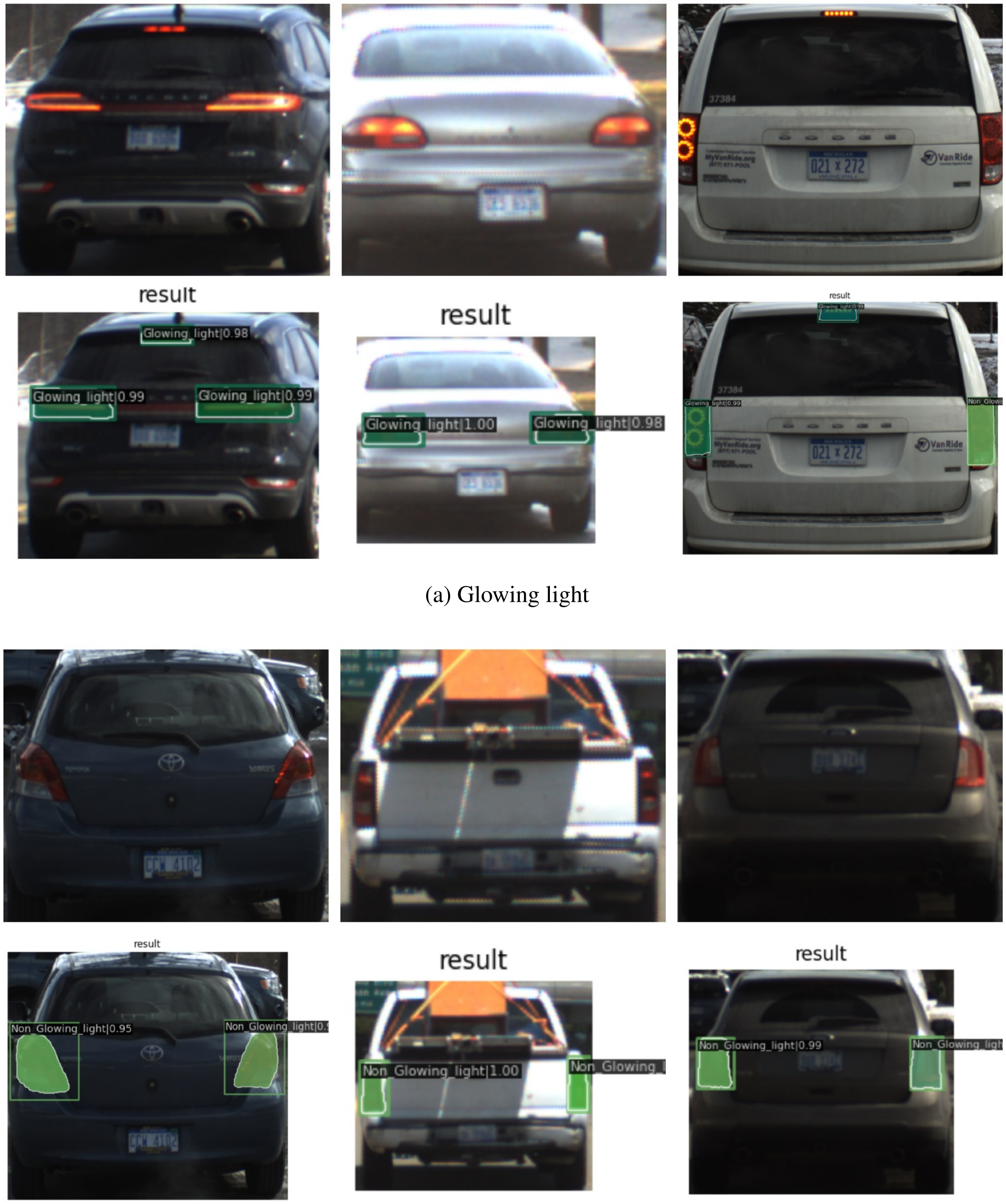
The performance of the top-5 different architectures generated by the proposed approach on the CaltechGraz dataset and the same architectures trained on the UC Merced Vehicle Rear Signal dataset is given in Table 4.4. The optimal architecture has a detection of 40.4 % , 42.2 % mAP on validation data and 39.3 % mAP, 40.4 % mAP on test data respectively, for the CaltechGraz dataset and UC Merced Vehicle Rear Signal dataset respectively, indicating that the proposed approach is effective for four-wheeler vehicle brake light detection on both datasets.



(a) Glowing light

(b) Non-glowing light

Figure 4.4: Brake light status detection results of the proposed approach on CaltechGraz dataset



(a) Glowing light

(b) Non-glowing light

Figure 4.5: Brake light status detection results of the proposed approach on UC Merced Vehicle Rear Signal dataset

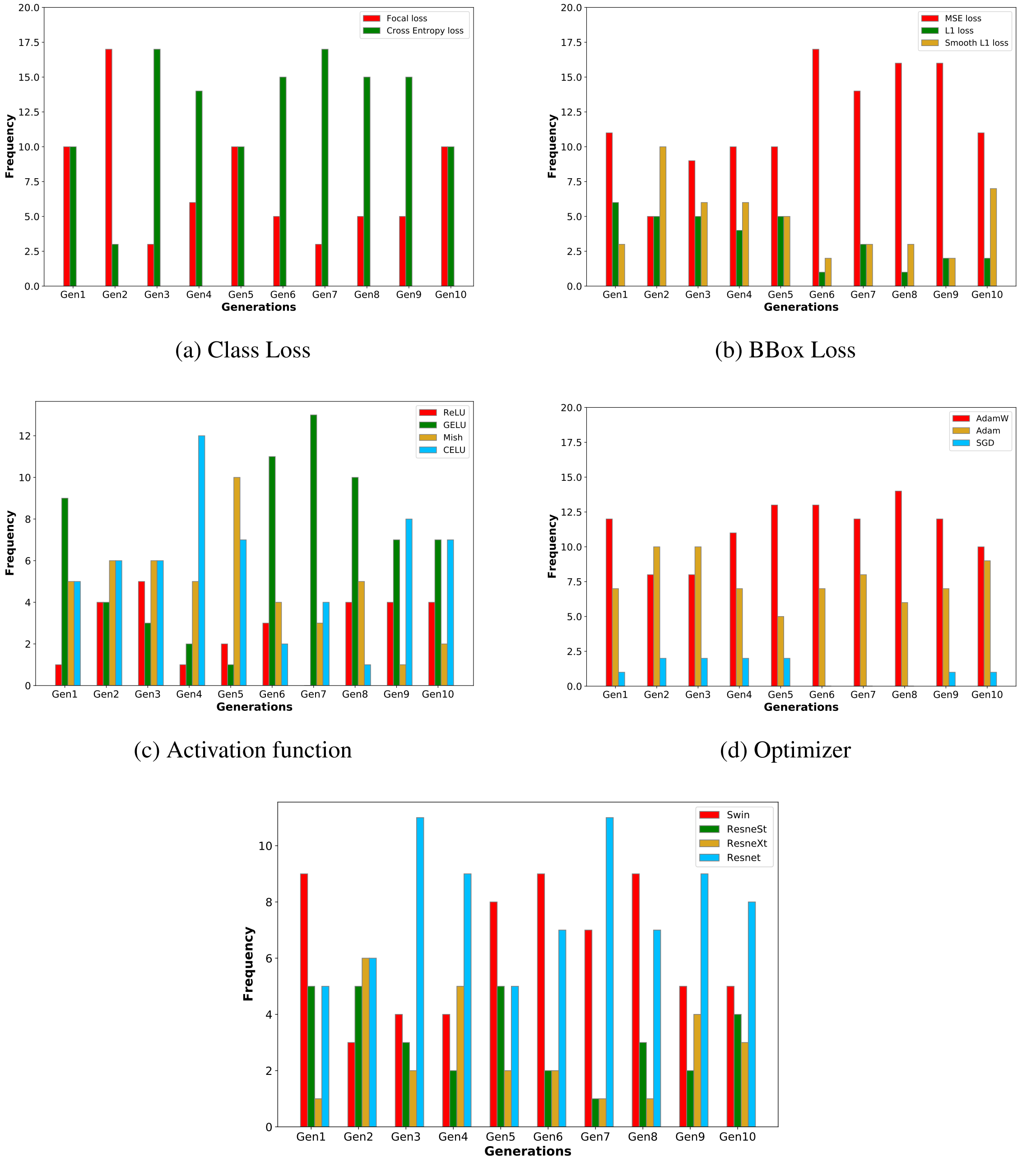
We have analyzed the various values explored by the proposed approach for the parameters of the search space, given in Table 4.1, across E-DE generations in the evolutionary process on the CaltechGraz dataset. Fig. 4.6 shows the number of times a parameter value

Table 4.4: Comparison of mAP of top-5 different models, generated by the proposed approach on CaltechGraz dataset and evaluated on UC Merced Vehicle Rear Signal dataset

Method	#Blocks	Block type	Class loss	Bbox loss	Optimizer	Activation	CaltechGraz		UC Merced	
							Valid (%)	Test (%)	Valid (%)	Test (%)
Top-5	1	ResneSt Block	Focal loss	Smooth L1 loss	AdamW	Mish	30.3	39.9	34.2	29.8
Top-4	2	ResneXt Block	Focal loss	MSE loss	AdamW	ReLU	34.1	**42.2**	39.4	35.1
Top-3	2	Swin Block	Focal loss	MSE Loss	AdamW	Mish	38.5	38.6	41.2	39.2
Top-2	4	Resnet Block	Cross Entropy loss	L1 loss	AdamW	CELU	38.6	41	40.7	38
Top-1	5	Swin Block	Focal loss	L1 loss	Adam	GELU	**40.4**	39.3	**42.2**	**40.4**

Note: The best values are highlighted in bold.

is used in the population across generations. From the figure, it can be observed that in the exploration process for the four-wheeler brake light detection task, the Cross Entropy loss dominated Focal loss; MSE loss dominated L1 loss and Smooth L1 loss; the GELU and CELU activation functions dominated the other activation functions; the AdamW and Adam optimizer dominated SGD, and finally, Resnet and Swin blocks dominated the other types of blocks. The comparison of the explored parameters and the final identified model parameters for the four-wheeler brake light status detection task reveals both agreement and divergence in parameter selection. The Swin block, which was frequently used alongside Resnet during exploration, was included in the final model. Although Cross Entropy and MSE loss were the dominant loss functions during the search process, the final model chose Focal loss and L1 loss, indicating that these less frequently explored options performed better for this task. The GELU activation function, which was frequently used during exploration, was also used in the final model, highlighting its importance to accuracy. Similarly, using the Adam optimizer in the final model corresponds to its frequent use during exploration, further validating its effectiveness in achieving better convergence.



(a) Class Loss

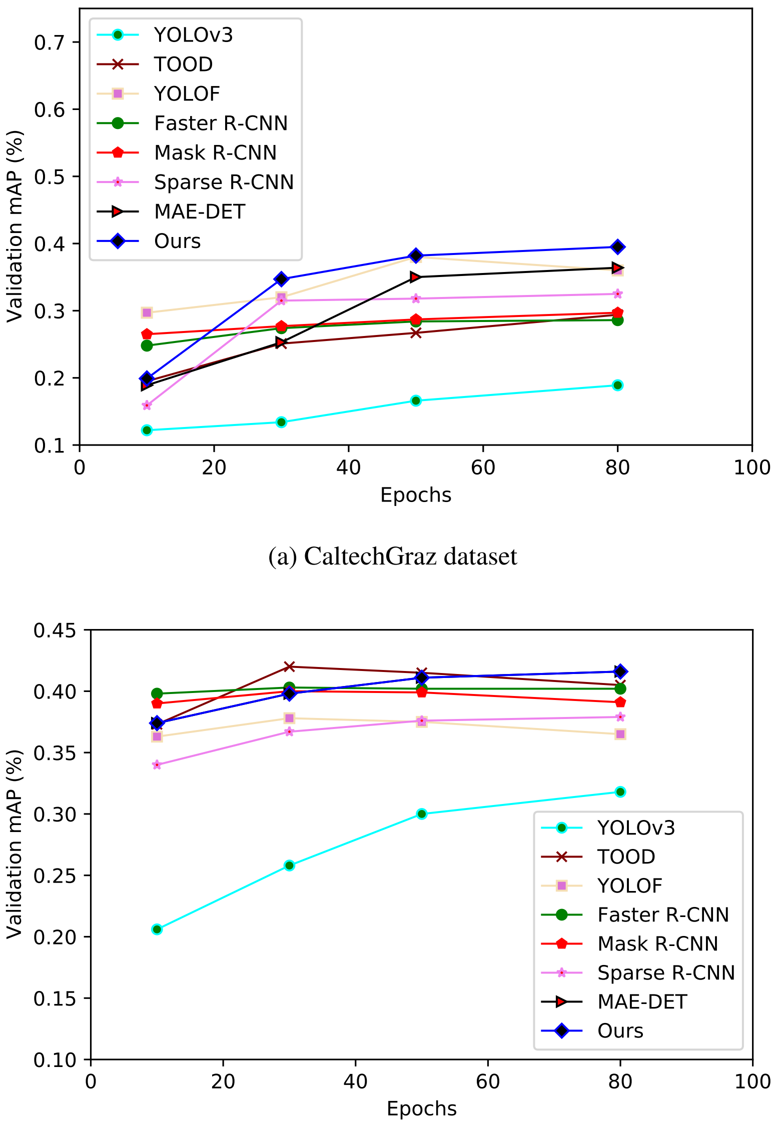
(b) BBox Loss

(c) Activation function

(d) Optimizer

(e) Block Type

Figure 4.6: The number of times a parameter value is used across generations in the population on CaltechGraz dataset



(a) CaltechGraz dataset

(b) UC Merced dataset

Figure 4.7: Change in validation mAP against training epochs for YOLOv3, TOOD, YOLOF, Faster R-CNN, Mask R-CNN, Sparse R-CNN, MAE-DET, and the proposed model for four-wheeler datasets

4.3.2 Experimental study for Motorcycle brake light detection

In this section, we discuss the details of the evaluation of the proposed approach to detect and classify the brake light status of two-wheeler vehicles. A new dataset (NITW-MBS) proposed in Chapter 3, is used for detecting two-wheeler brake lights is discussed. The performance comparison of the proposed approach against various existing approaches on this dataset is presented. The performance of top-5 models generated by the proposed approach is also discussed.

We considered the proposed two-wheeler dataset, which is discussed in Section 3.2.1, to evaluate the proposed approach. 2125 images are selected and annotated in the COCO data format. Fig. 3.4 and Fig. 3.5 show some of the sample images in this dataset. To train a model, the dataset should be split into three parts: training, validation, and test data. The dataset has two classes: *glowing light* and *non-glowing light*, with 1650 images for training, 210 images for validation, and 265 images for test data. The statistics of the dataset are given in Table 3.2

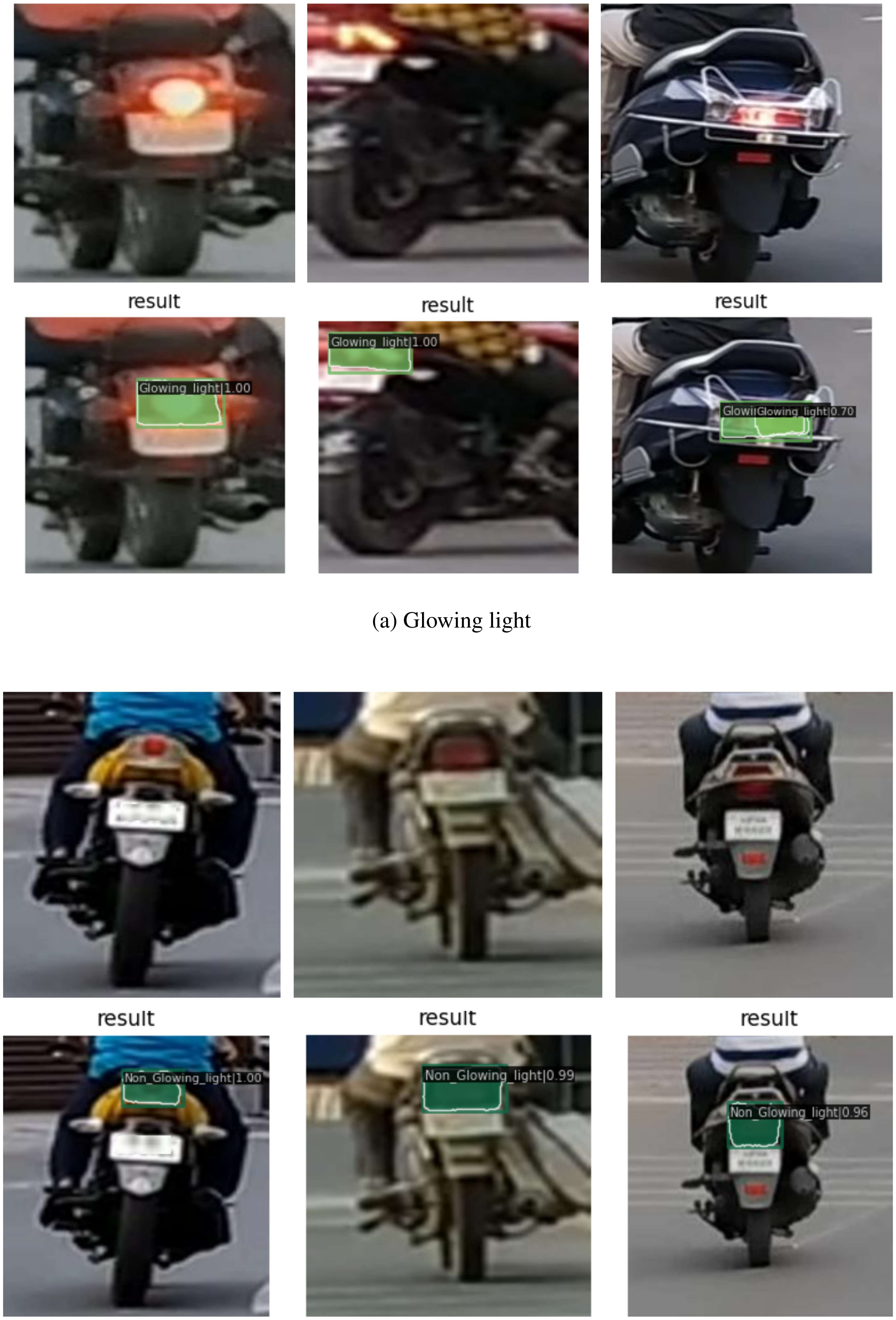
We evaluated the existing manually designed object detectors, including one-stage YOLOv3 [10], TOOD [131], YOLOF [141] and two-stage Faster R-CNN [14], Mask R-CNN [15] and Sparse R-CNN [132] and NAS based object detection model MAE-DET [18] on this dataset. The results are given in Table 4.5. From the table, it can be observed that the model identified by the proposed approach performs better than the existing benchmark models. Fig. 4.8 shows some of the brake light status detection results, predicted by the proposed approach on test data images. Fig. 4.10 shows the change in validation mAP of the methods given in Table 4.5 against the number of training epochs.

Table 4.5: Comparison of mAP of existing object detection models against the proposed approach on NITW-MBS dataset

Method	NITW-MBS Dataset		Params
	Valid (%)	Test (%)	
YOLOv3 [10]	35.6	37.1	61.5 M
TOOD [131]	53.9	49.5	22.4 M
YOLOF [141]	56.1	50.8	32.8 M
Faster R-CNN [14]	52.1	48.2	63.6 M
Mask R-CNN [15]	52.7	48.7	31.6 M
Sparse R-CNN [132]	53.8	51.4	96.6 M
MAE-DET [18]	56.6	50.1	52.9 M
Ours	**57.9**	**53.3**	46 M

Note: The best values are highlighted in bold.

The comparison of the performance of the top-5 different architectures generated by the proposed approach is given in Table 4.6. The optimal architecture has a detection performance of 57.9 % mAP on validation data and 53.3 % mAP on test data, indicating that the proposed approach is effective for two-wheeler vehicle brake light detection on the NITW-MBS dataset.



(a) Glowing light

(b) Non-glowing light

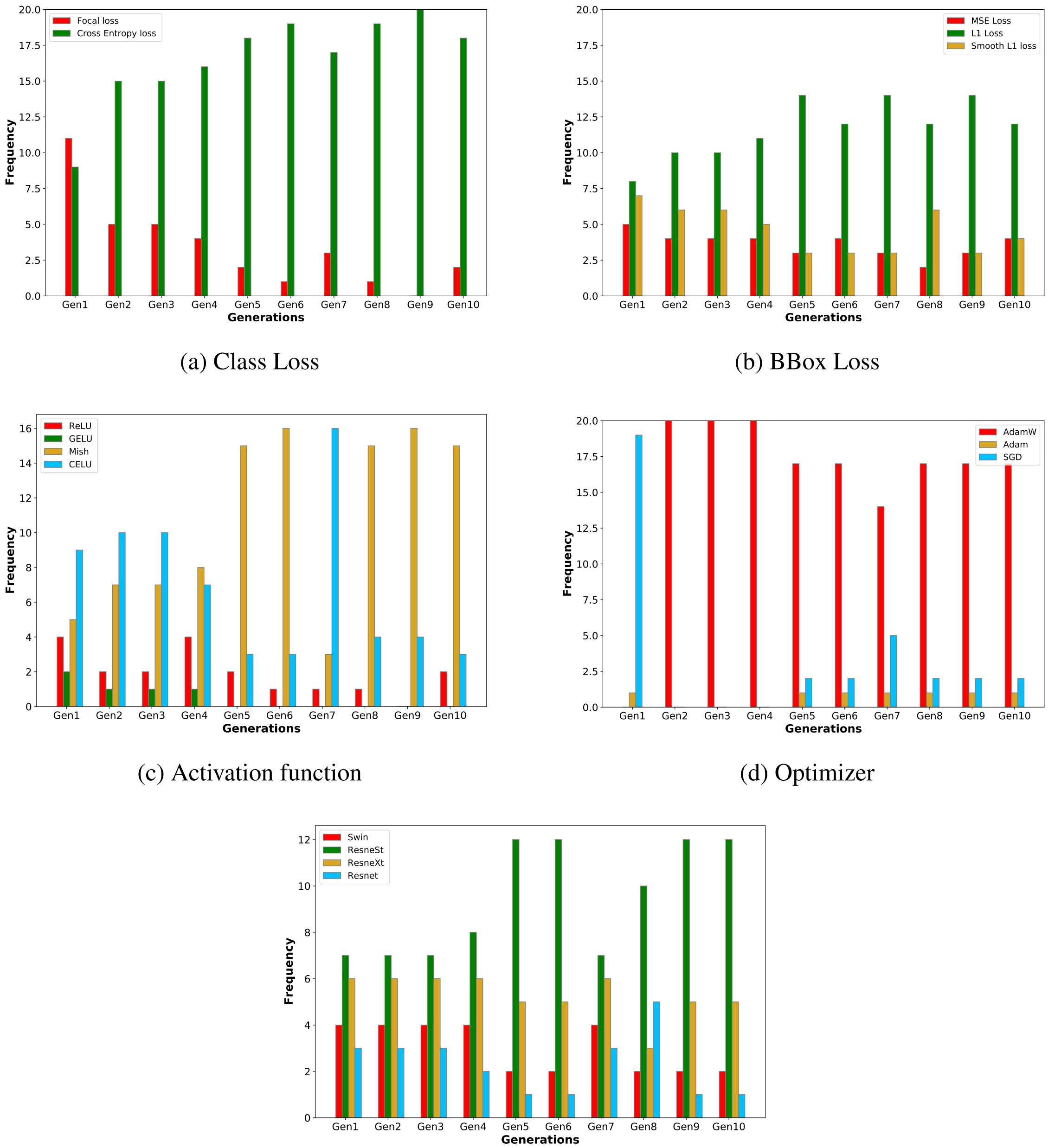
Figure 4.8: Brake light status detection results of the proposed approach on NITW-MBS dataset

We have analyzed the various values explored by the proposed approach for the parameters of the search space given in Table 4.1 across DE generations. Fig. 4.9 shows the number of times a parameter value is used in the population across generations. In the exploration process to find the optimal DNN model for the two-wheeler brake light detection task, it can be observed from the figure that the Cross Entropy loss dominated the Focal loss, the L1 loss dominated the MSE loss & the Smooth L1 loss; the Mish activation function dominated the other activation functions; the AdamW optimizer dominated the Adam and SGD; and finally, ResneSt dominated the other type of blocks. The comparison of the parameter explored for the two-wheeler brake light detection task and the final identified model by the proposed approach reveals both consistency and divergence. During the exploration process, Cross Entropy loss was preferred over Focal loss, and the final model's choice of Cross Entropy loss is consistent with this trend, indicating its suitability for the task. While L1 loss was preferred over both MSE loss and Smooth L1 loss during exploration, the final model includes Smooth L1 loss as well as Cross Entropy loss, indicating that this combination may improve performance by balancing regression accuracy with classification tasks. In terms of optimization, the AdamW optimizer dominated the exploration phase, and its inclusion in the final model demonstrates its effectiveness in improving convergence. Finally, the exploration favoured the ResneSt block, which was eventually chosen for the final model, demonstrating a clear alignment between the exploration findings and the architecture chosen by the proposed approach.

Table 4.6: Comparison of mAP of top-5 different models, generated by the proposed approach on NITW-MBS dataset

Method	#Blocks	Block type	Class loss	Bbox loss	Optimizer	Activation	NITW-MBS Dataset	
							Valid (%)	Test (%)
Top-5	7	Swin Block	Cross Entropy loss	L1 loss	AdamW	Mish	55.3	52.2
Top-4	2	Resnet Block	Cross Entropy loss	Smooth L1 loss	AdamW	CELU	56.7	53.3
Top-3	4	ResneXt Block	Cross Entropy loss	MSE loss	AdamW	Mish	57	53.6
Top-2	1	ResneSt Block	Cross Entropy loss	L1 Loss	AdamW	Mish	57.3	**53.9**
Top-1	6	ResneSt Block	Cross Entropy loss	Smooth L1 loss	AdamW	ReLU	**57.9**	53.3

Note: The best values are highlighted in bold.



(a) Class Loss

(b) BBox Loss

(c) Activation function

(d) Optimizer

(e) Block Type

Figure 4.9: The number of times a parameter value is used across generations in the population on the NITW-MBS dataset

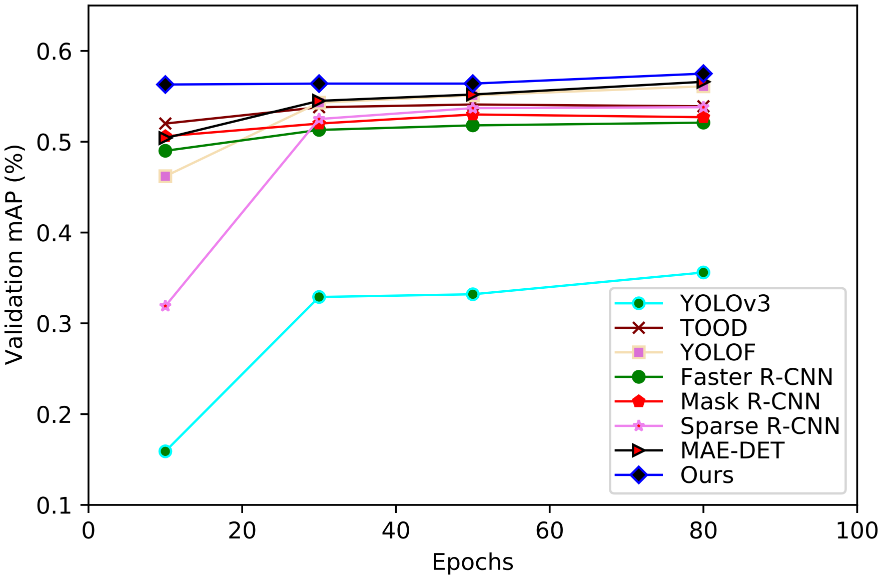


Figure 4.10: Change in validation mAP against training epochs for YOLOv3, TOOD, YOLOF, Faster R-CNN, Mask R-CNN, Sparse R-CNN, MAE-DET and the proposed model on NITW-MBS dataset

We have also compared the proposed approach E-DE with traditional DE using mean and standard deviation metrics whose results are given in Table 4.7. The results indicate that the proposed method outperforms the traditional DE in terms of performance. The proposed approach has achieved better performance with fewer parameters when compared to DE, specifically for the two-wheeler dataset. The best fitness model of DE requires 59 million parameters, while E-DE achieved better performance with 46 million parameters only.

Table 4.7: Comparison of mean, standard deviation and best fitness value of mAP for E-DE and DE algorithms

Dataset	Algorithm	Mean	Standard deviation	Best fitness value
CaltechGraz	DE	36.1	1.26	39.5
	E-DE	36.83	0.28	40.4
NITW-MBS	DE	56.43	0.42	57.5
	E-DE	56.71	0.52	57.9

4.3.3 Network architectures searched by proposed approach

The proposed approach explored network architectures and their corresponding parameters on two four-wheeler datasets and one two-wheeler dataset. The details of the identified architectures is given in Tables 4.4 and 4.6. From the tables, we can observe that the Swin block with Focal loss, L1 loss, Adam, and GELU performed well on the four-wheeler dataset, while the ResneSt block with Cross Entropy loss, Smooth L1 loss, AdamW, and ReLU excelled on the two-wheeler dataset. The Swin block demonstrated its capability to effectively extract complex patterns, making it suitable for the four-wheeler dataset. On the other hand, the ResneSt block showed strong performance in capturing relatively simple patterns, making it well-suited for the two-wheeler dataset. It is worth noting that the proposed algorithm was able to optimize the number of parameters compared to a simpler DE approach for the two-wheeler dataset. Further, we have analyzed the top-5 distinct architectures because focusing only on the top architecture may not fully capture the impact of different parameters. We have selected the top-5 distinct architectures based on the type of block used to analyze the impact of SPS, which is based on the block type used. Additionally, certain block types may yield high validation accuracy but not necessarily high test accuracy. By considering the top-5 models, we can observe how the different architectures perform in terms of both validation and test accuracy. This allows us to assess the effectiveness of different block types across different datasets and identify architectures that demonstrate superior performance on the test set compared to the top-1 model.

4.3.4 Computational complexity analysis

The time complexity of the ECSM algorithm is $O(N)$where N is the population size. The SPS algorithm has a time complexity of $O(NlogN)$ due to sorting in the elite retention step, combined with $O(N)$ for selecting individuals from species. The overall E-DE based NAS framework has a total time complexity of $O(T \times (N \times E_{train} + NlogN)$, where T is the number of generations, and E_{train} is the time complexity of evaluating a network architecture, which is the most computationally expensive part.

4.3.5 Effectiveness of brake light status detection

Vehicles typically have one or more brake lights. If the front vehicle applies the brakes and at least one of the brake lights is glowing, we consider it as a braking alert ($Glowing\_light$). Otherwise, it is not considered as a braking alert ($Non\_Glowing\_light$). The pseudocode used for computing accuracy is given in Algorithm 3.1. This pseudocode uses the model identified by the proposed approach that is denoted by $Model$. The performance of the discovered optimal model for different values of IoU thresholds is given in Table 4.8.

Table 4.8: Accuracy of brake light status detection of the discovered optimal model for IoU threshold range of 0.3 to 0.9, for a four-wheeler and two-wheeler vehicle brake light detection datasets

Dataset	Accuracy(%) for various IoU thresholds							Mean Accuracy(%)
	0.3	0.4	0.5	0.6	0.7	0.8	0.9	
NITW-MBS	95.07	96.59	97.77	98.48	99.24	99.62	98.48	97.97
CaltechGraz	86.09	88.63	88.63	89.77	92.04	94.31	88.63	89.73
UC Merced	87.48	88.18	89.16	89.16	89.68	89.68	88.93	88.90

4.3.6 Cross-dataset evaluation

To evaluate the effectiveness of the proposed method for the brake light status detection task, we have performed cross-dataset evaluation, i.e. the model is evaluated on the test data from a different dataset. Based on our knowledge, the sole publicly available dataset designed for field testing of brake light detection systems in the context of predictive braking is given by J. Pirhonen et al. in [142, 143]. To conduct cross-dataset evaluation, we used the proposed top-1 model, which is identified in Section 4.3.1 on the UC Merced Vehicle Rear Signal dataset. This top-1 model is used to compute the performance of the test data given in [143]. We followed the same evaluation scheme given in [143]. The performance comparison of the proposed method against existing approaches is given in Table 4.9. From the table, it can be observed that the proposed method achieved an accuracy of 93.92 %, which is a 20.52 % improvement over the existing approach. Fig. 4.11 shows the input and output of the proposed method for some observations in this dataset. The figure shows that the proposed method effectively identifies brake light status across various types of vehicles, even when the location of brake lights is different across vehicles.

Table 4.9: The performance of brake light status detection methods on raw test images of dataset given in [143]

Model	Accuracy(%)
J. Pirhonen et al. [143]	73.40
Ours	**93.92**

Note: The best values are highlighted in bold.

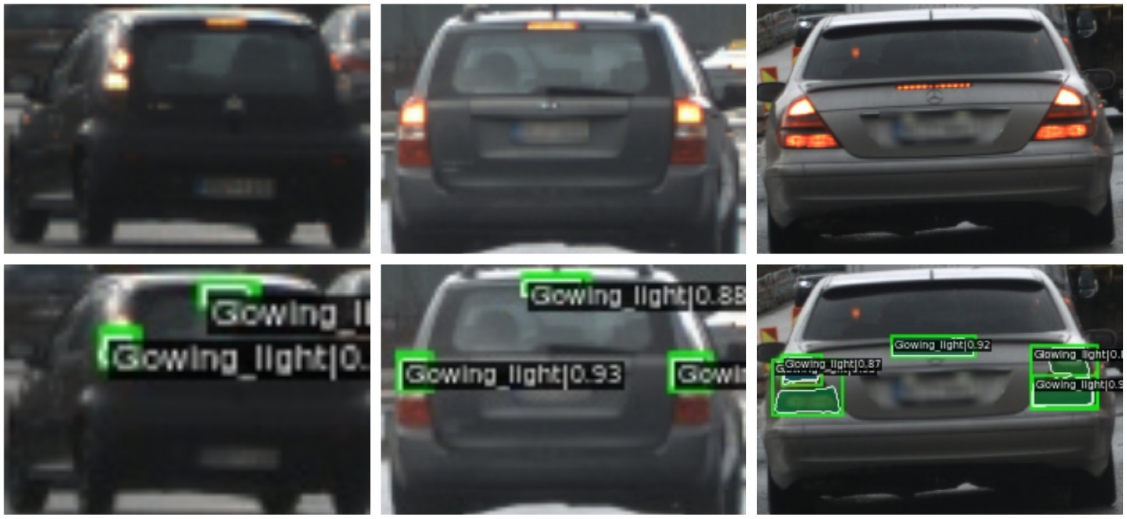


Figure 4.11: Brake light status detection results of the proposed approach on test data given in [143]

4.3.7 Experiments on real-world videos

The proposed model is also evaluated on real-world traffic videos. We selected three videos in Driving Event Camera Dataset [144], which cover various real-world driving scenarios: street view (street1.mp4), back view (back6.mp4), and early morning view (sun13.mp4). In these videos, we sample every 10^{th} frame as the difference in visual information between adjacent frames is less. The number of occurrences of different vehicles that appear in the selected frames of these videos is given in Table 4.10. The evaluation process comprises two phases: the pre-trained YOLOX [145] model is employed to detect the region of vehicles during the first phase. In the second phase, the detected vehicle is given as input to the proposed model to determine the status of the brake lights. For two-wheeler vehicles, we use the top-1 model identified in Section 4.3.2 on the NITW-MBS dataset, and for the four-wheeler vehicles, we consider the top-1 model identified in Section 4.3.1 on UC Merced Vehicle Rear Signal dataset for the prediction of brake light status. To estimate the braking status of a vehicle, if at least one of the detected brake lights has $Glowing\_light$ status, we consider it as a braking alert (shown as ⚠ in Fig. 4.12, 4.13, 4.14); otherwise, there is no braking alert. The experimental results, presented in Table 4.11, convey the effectiveness of the proposed method for the detection of the braking status of vehicles in real-world videos.

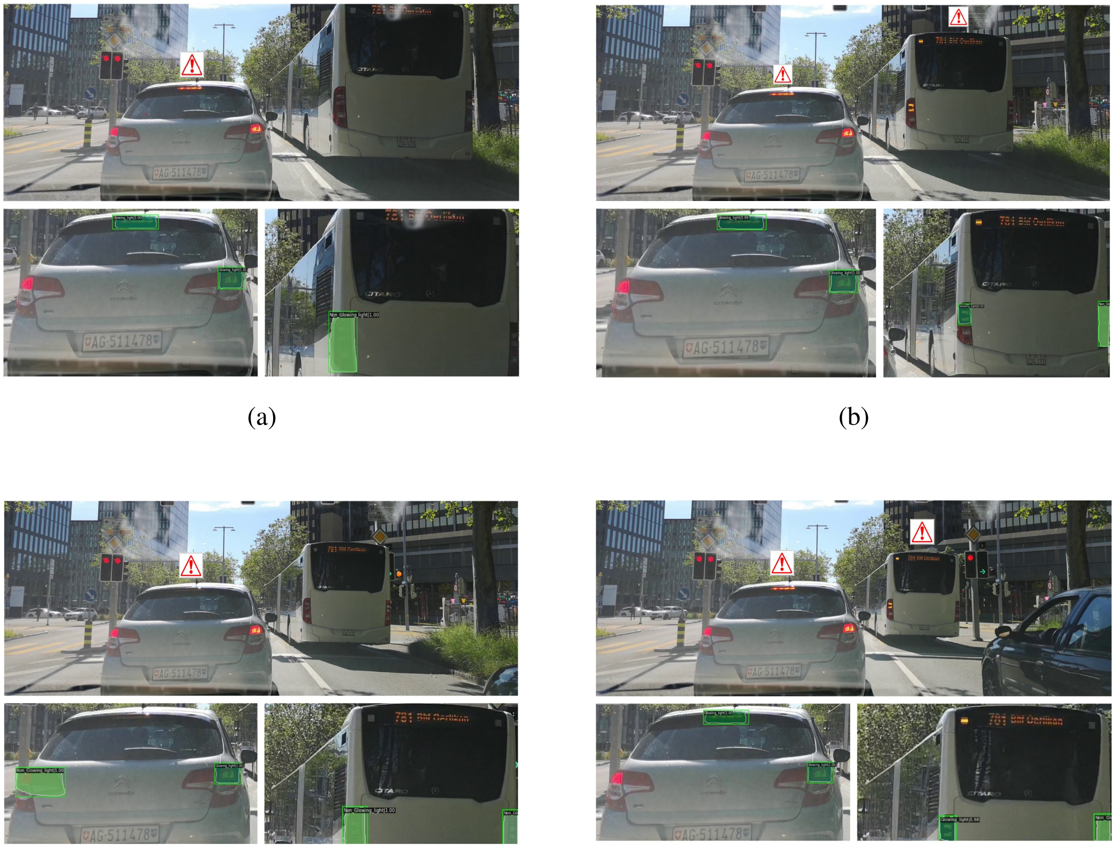
Table 4.10: The frequency of appearance of vehicles in videos

Video	Number of frames a vehicle appeared							Total number frames in video
	car1	car2	car3	car4	truck	bus	motorcycle	
street1.mp4	61	0	0	0	0	33	-	80
back6.mp4	46	6	28	18	10	-	10	46
sun13.mp4	42	95	-	-	24	-	-	145

Table 4.11: The detection accuracy of vehicles braking status on real-world videos

Video	Accuracy(%)							Mean accuracy(%)
	car1	car2	car3	car4	truck	bus	motorcycle	
street1.mp4	90.16	-	-	-	-	81.82	-	87.23
back6.mp4	100	100	100	77.8	100	-	80	95.00
sun13.mp4	83.33	93.68	-	-	100	-	-	91.93

The predicted braking status and the visualization of the braking alert for four consecutive video frames in street1.mp4 video are shown in Fig. 4.12. In each subfigure of Fig 4.12, the bottom half shows the individual vehicles and the braking status predicted by the proposed model; the top half shows the visualization of the braking alert for corresponding vehicles in the video frame. From the figure, it can be observed that the brake lights of the bus are glowing in the frames associated with Fig. 4.12 (b) and Fig. 4.12 (d), while the brake lights of the car are glowing in all the frames. The proposed method is able to identify the glow associated with braking and classify the individual brake lights accurately. The use of two-phased processing in the proposed workflow prevents misinterpretation of traffic lights as brake lights.



(a)

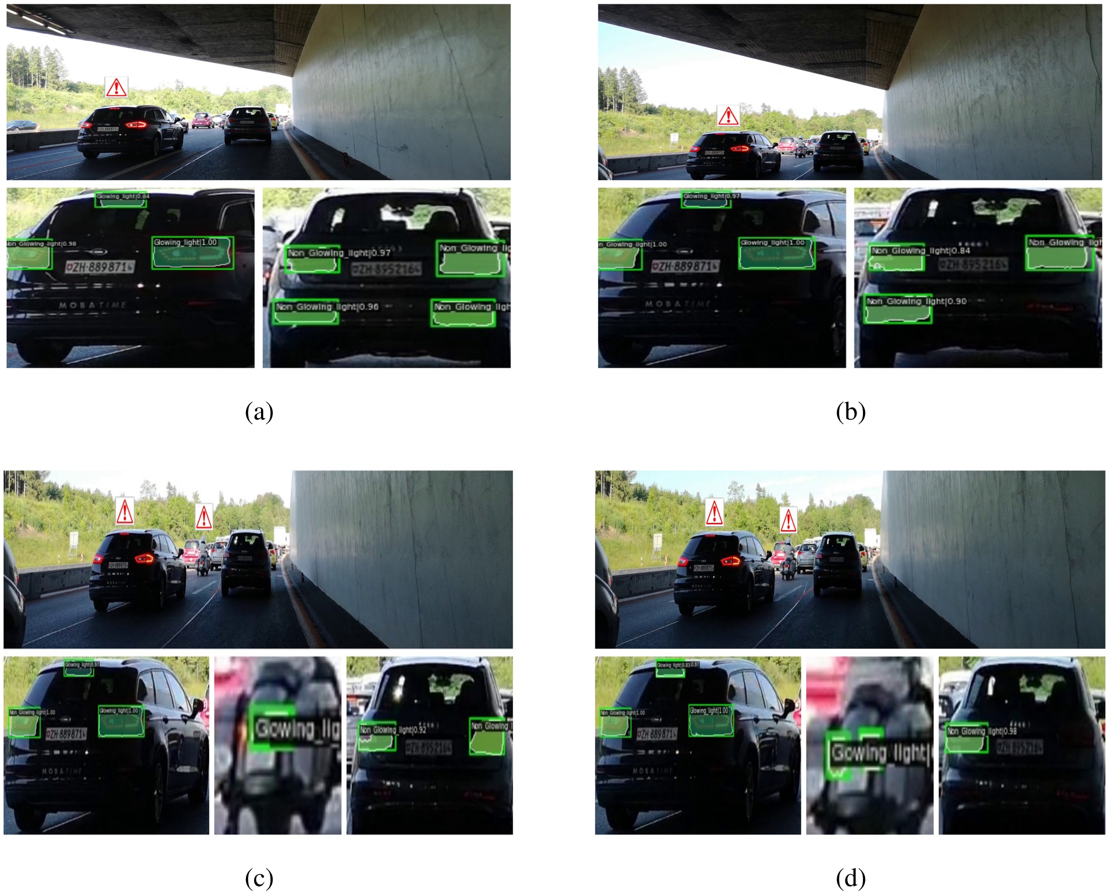
(b)

(c)

(d)

Figure 4.12: Visualization of the predicted brake light status in four consecutive frames of street1.mp4 video, along with braking alert for each vehicle

The predicted braking status and the visualization of the braking alert for four consecutive video frames in back6.mp4 video are shown in Fig. 4.13. Similar to Fig. 4.12, Fig 4.13 shows the status of the brake lights of individual vehicles and the braking status of the vehicles in each subfigure. The proposed method successfully detects cars and motorcycles in these frames, even in scenarios with densely crowded vehicles.



(a)

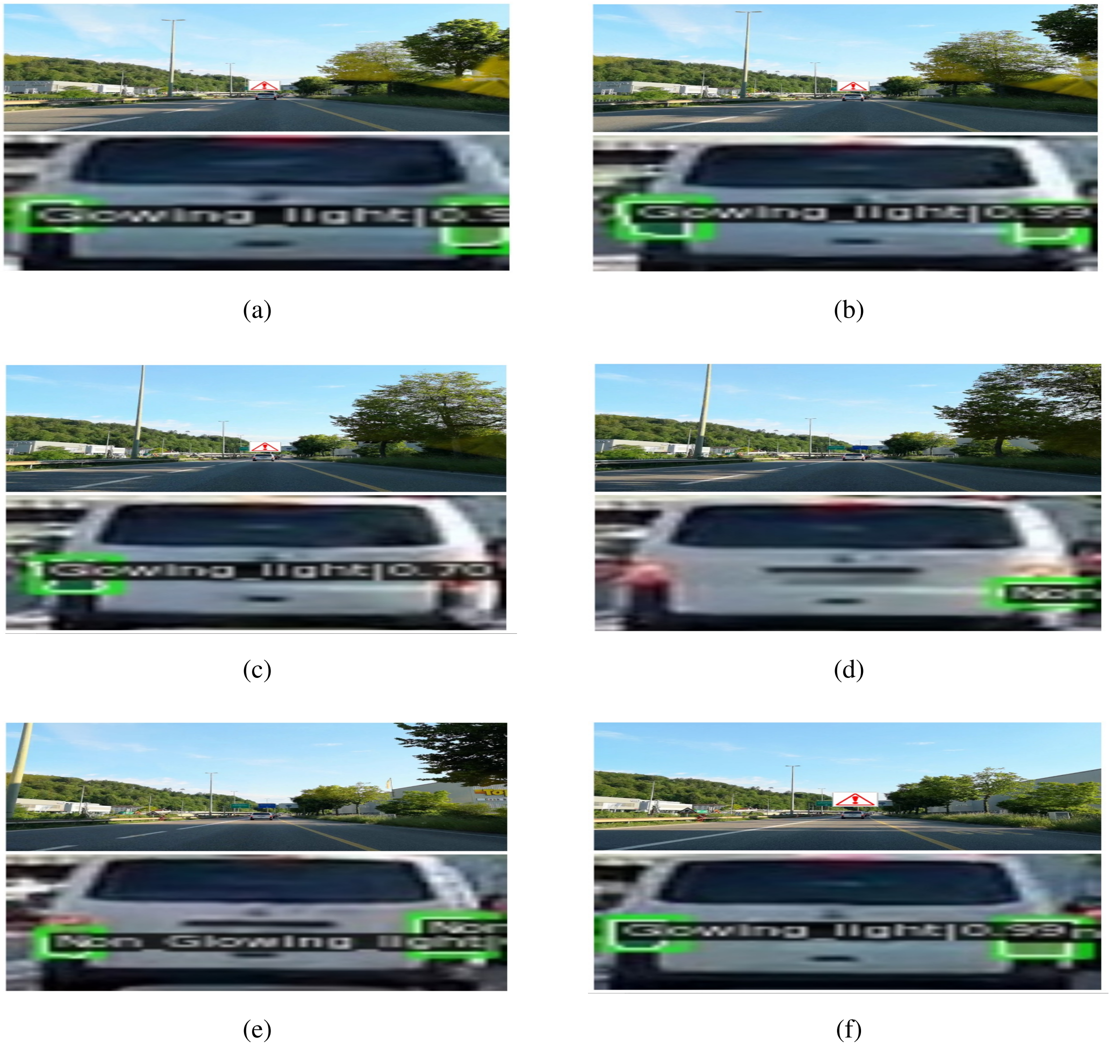
(b)

(c)

(d)

Figure 4.13: Visualization of the predicted brake light status of vehicles in four consecutive frames of back6.mp4 video, along with braking alert for each vehicle

Fig. 4.14 shows the visualization of the predicted braking status of vehicles in six consecutive video frames from the sun13.mp4 video. The car's brake lights remained glowing throughout all six frames. However, the proposed method is unable to predict the correct status of brake lights for the frames associated with Fig. 4.14 (d) and Fig. 4.14 (e). This discrepancy may be due to the blur in the visual information of the detected vehicle due to various factors like the distance of the vehicle from the camera, blur due to the camera motion, blur due to the speed of the vehicle, etc.



(a) (b)

(c) (d)

(e) (f)

Figure 4.14: Visualization of the predicted brake light status in six consecutive frames of sun13.mp4 video, along with braking alert for each vehicle

4.4 Summary

In this chapter, we proposed an automated approach for designing a deep neural network model to detect brake lights in both four-wheeler and two-wheeler vehicles. The proposed approach utilizes strong search space to identify the optimal backbone architecture and training parameters, resulting in an approach capable of identifying efficient DNN models. We employ a modified Differential Evolution based search strategy, which includes evaluation correction based selection for a mutation to find architectures with high fitness and a reduced number of parameters. Additionally, species protection based selection is introduced to maintain population diversity and to achieve global optima. The optimal models discovered using the proposed approach have achieved mean accuracies of 89.73 % and 88.90 % on the four-wheeler datasets CaltechGraz and UC Merced Vehicle Rear Signal, respectively. On the proposed two-wheeler NITW-MBS dataset, the proposed approach has achieved an accuracy of 97.97 %. The comparative study with other existing manually designed and NAS based object detectors on these datasets indicates the effectiveness of the proposed approach. In addition, the comparison of the proposed approach with basic DE highlights the effectiveness of the proposed approach. We have used cross-dataset evaluation to assess the effectiveness of the proposed method for unseen data. We further explored the possibility of practical use by evaluating the proposed method on real-world traffic videos.

Chapter 5

Grasshopper Optimization based Deep Neural Networks for Vehicle Re-identification

This chapter proposes a NAS based optimization of deep neural networks using a grasshopper optimization algorithm for the task of vehicle re-identification.

Chapter Organization:

Preliminaries for the proposed approach is covered in Section 5.1. The proposed approach is described in Section 5.2. Section 5.3 presents the experimental results and analysis. Finally, Section 5.4 summarizes the proposed work.

5.1 Preliminaries

This section provides the fundamentals of the GOA, which is used as the NAS search strategy for this objective.

5.1.1 Grasshopper Optimization Algorithm (GOA)

GOA is a swarm intelligence algorithm inspired by the foraging and swarming behaviour of grasshoppers in nature. It was first introduced by Saremi et al. in [57]. It has been shown to be effective in solving various optimization problems, including medical image segmentation [58, 146], image enhancement [59], image fusion and feature selection [60, 61]. Its simplicity, efficiency, and robustness make it a popular optimization technique. From our review of the existing literature, we noticed that there is no existing approach utilizing GOA for the reID task. This motivated us to use GOA as the search strategy in this work for finding an optimum DNN model for motorcycle reID task.

In GOA, the grasshoppers move based on the attractiveness of the food source and the distance to other grasshoppers. The swarming behaviour of grasshoppers is mathematically modelled in Eq. 5.1

$$Y_i = So_i + Gf_i + Wa_i \tag{5.1}$$

Here, Y_i indicates the i^{th} grasshopper position, So_i represents the social interaction, Gf_i denotes the gravity force on the i^{th} grasshopper, and Wa_i is the wind advection.
The social interaction So_i is defined as follows:

$$So_i = \sum_{j=1, j\neq i}^{N} s(d_{ij}) \frac{Y_j - Y_i}{d_{ij}} \tag{5.2}$$

where N denotes the number of grasshoppers, $d_{ij} = |Y_j - Y_i|$ denotes the Euclidean distance between the i^{th} and the j^{th} grasshopper $s(r)$ represents the social forces that can be computed from the formula in Eq. 5.3.

$$s(r) = F * e^{-r/L} - e^{-r} \tag{5.3}$$

where F and L denote the attraction intensity and attraction length scale, respectively. The social interaction among grasshoppers can be denoted as attraction and repulsion. The gravity force Gf_i can be computed using the formula in Eq. 5.4.

$$Gf_i = -g * \hat{e}_g \tag{5.4}$$

Here, g represents the gravitational constant, and $\hat{e}_g$ denotes a unit vector toward the centre of the earth.

The wind advection Wa_i is defined by the formula in Eq. 5.5.

$$Wa_i = -u * \hat{e}_w \tag{5.5}$$

where u represents the drift constant and $\hat{e}_w$ is a unit vector in the wind direction.

According to Saremi et al., [57], Eq. 5.1 is modified to suit the actual conditions of the grasshopper's movement, using the formula in Eq. 5.6.

$$Y_i^d = C\left(\sum_{j=1, j\neq i}^{N} C\frac{ub-lb}{2}s(|Y_j^d - Y_i^d|)\frac{Y_j - Y_i}{d_{ij}}\right) + T_d \tag{5.6}$$

Where lb, ub is defined as the lower-bound and upper-bound in the d^{th} dimension. T_d is defined as the best value of the d^{th} dimension in the target.
The coefficient C is defined in Eq. 5.7.

$$C = C_{max} - t\frac{C_{max} - C_{min}}{t_{max}} \tag{5.7}$$

5.2 Proposed approach

This section presents the proposed use of NAS with GOA for finding the optimal model for the re-IDentification (reID) task.

Object re-identification typically involves two main phases: feature learning and re-identification. During the feature learning phase, images are used to train a classification network to extract feature vectors. In the re-identification phase, test images are split into gallery images of one camera view and probe images of another camera view, and these are then fed into the trained classification network to extract their respective feature vectors. Once these feature vectors have been extracted, a distance measure is used to match objects and identify the same object across different views. In this work, we have modelled this entire process as a NAS framework to design optimal deep neural network structure for re-identification. Fig. 5.1 illustrates the flow diagram of different stages of the proposed approach. In Fig. 5.1, the process initiates with the random initialization of grasshoppers.

These grasshoppers represent potential solutions or configurations of the DNN models. Each grasshopper configuration is then transformed into a DNN model, which is subsequently trained on a dataset containing training images. During the feature learning phase, the DNN models learn to extract discriminative features from the input images. These learned features are crucial for accurately identifying and matching individuals across different images in the re-identification phase. In the re-identification phase, the performance of the DNN models is evaluated using a metric such as rank-1 accuracy. This measures the model's ability to correctly match probe images and gallery images. Subsequently, the Grasshopper Optimization Algorithm is applied to optimize the DNN models further. Unlike GA and DE, which depend on a limited set of individuals to produce offspring, GOA utilizes the collective behavior of the entire population to explore the search space more thoroughly, enhancing the balance between exploration and exploitation. The GOA optimization process involves several phases: Normalizing the distances between grasshoppers, Updating the position of grasshoppers, and bringing current grasshoppers back if they go outside the boundary. The entire process, from grasshopper initialization to grasshopper updations, is repeated iteratively until a predetermined stopping criterion, typically a maximum number of iterations, is met. This iterative approach allows for the continuous updation and enhancement of the DNN models, ultimately leading to improved performance in re-identification tasks. We can formulate NAS as an optimization problem for the reID task, as given in Eq. 5.8.

$$max_{x \in S} \; \mathcal{M}_{r1}(\mathcal{A}(x, w^*(x)), P, G) \tag{5.8}$$

$$\mathcal{A}(x, w^*(x)) = min_w \; \mathcal{L}_{train}\mathcal{A}(x, w) \tag{5.9}$$

Here, S represents the search space of the DNN architecture denoted by a Directed Acyclic Graph (DAG), $x \in S$ is a specific path in the DAG corresponding to a DNN architecture, w represents the weights of the DNN architecture, $\mathcal{A}(x, w)$ denotes a DNN model with architecture x weight w and $\mathcal{L}_{train}$ is the training loss. Now, Eq. 5.9 represents the computation of the DNN model $\mathcal{A}$ with optimal weights $w^*(x)$ for the DNN architecture x, that mini-

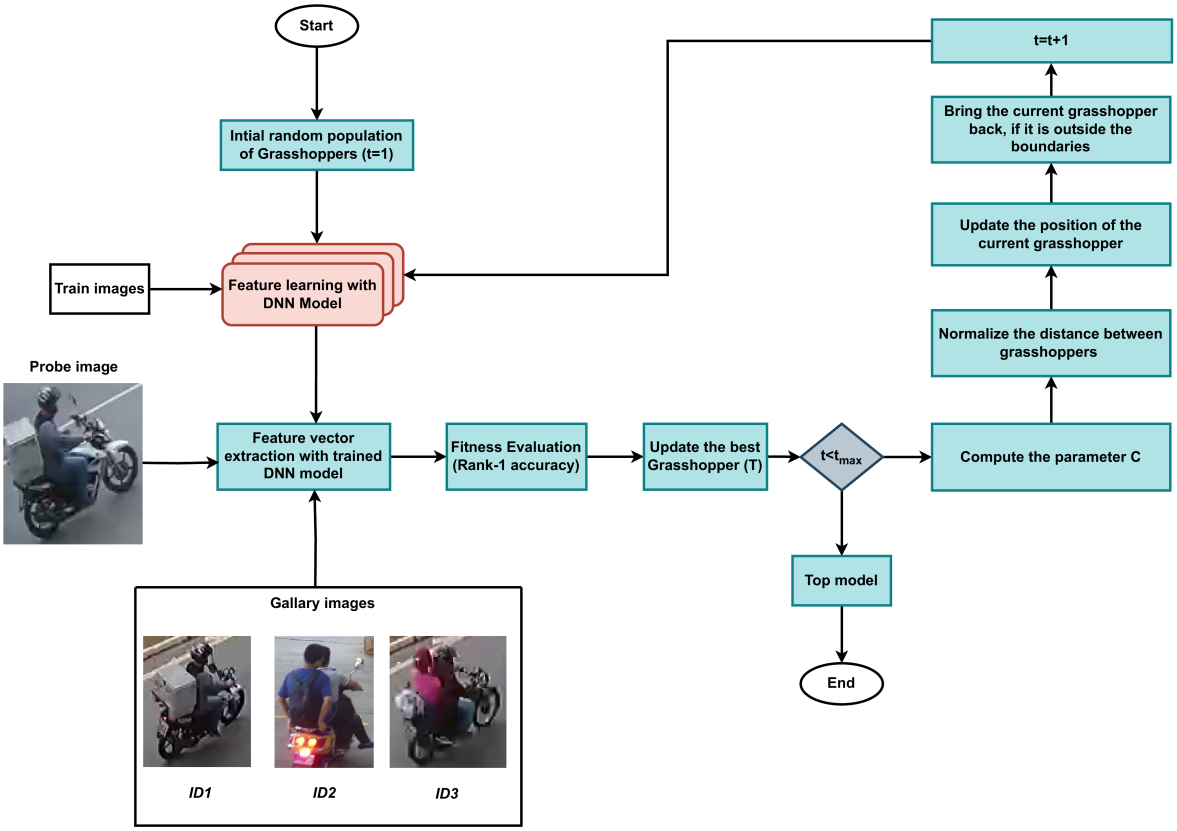

Start
Intial random population of Grasshoppers (t=1)
Train images
Feature learning with DNN Model
Probe image
Feature vector extraction with trained DNN model
Fitness Evaluation (Rank-1 accuracy)
Update the best Grasshopper (T)
t<tmax
Compute the parameter C
Normalize the distance between grasshoppers
Update the position of the current grasshopper
Bring the current grasshopper back, if it is outside the boundaries
t=t+1
Top model
End
Gallary images
ID1
ID2
ID3

Figure 5.1: The flow diagram of the proposed GOA based NAS approach for finding an optimum DNN model

mizes the training loss $\mathcal{L}_{train}$. In Eq. 5.8, $\mathcal{M}_{r1}$ is the rank-1 accuracy, and G, P represents a set of gallery and probe images, respectively. Eq. 5.8 represents the identification of DNN architecture $x \in S$, such that its rank-1 accuracy, $\mathcal{M}_{r1}$ is maximum, corresponding to the optimum DNN model $\mathcal{A}$ identified by Eq. 5.9.

5.2.1 Search space design for re-identification

The performance of the reID task is dependent on the features extracted by the CNN backbone architecture. Due to this reason, we include the parameters related to the backbone in the search space. The search space of the backbone architecture consists of different kinds of blocks. Backbone architecture could be divided into several stages according to the resolution of the output features, where the stage refers to a number of blocks fed by the features with the same resolution.

In addition to the parameters of the backbone, we have also included parameters associated with training like *Size of input* , *Pooling opeation*, *Activation function*, *Optimization algorithm*, *Loss function* and *Distance metric* in the proposed search space. The details of the search space used in this work are described in Table 5.1.

Table 5.1: The Parameters in the search space and their values considered in this work

Parameter	**Range**
Block type	{Regnet, EfficientnetV2, Densenet, Resnet}
Size of input	{320×320, 256×256, 224×224, 128×128}
Pooling operation	{Average, Max}
Activation function	{ReLU, Swish}
Optimization algorithm	{Adam, SGD}
Loss function	{Cross Entropy Loss, Triplet Loss, Quadruplet Loss, MSML}
Distance metric	{Euclidean, Cityblock, Minkowski, Sqeuclidean}

Backbone architecture directly impacts feature representation and extraction. By incorporating diverse *Block type* such as ResNet, DenseNet, EfficientNetV2, and RegNet, we aim to explore various architectural paradigms, from skip connections to dense connections, to capture discriminative features from input images effectively. This diversification enables us to explore various complex architectures and improve the reID performance. The *Size of input* images directly influence the receptive field of the network and its ability to capture spatial information. By considering multiple input sizes such as 320×320, 256×256, 224×224, and 128×128, we included variations in image resolution commonly encountered in reID datasets. *Pooling operation*, such as max pooling and average pooling, are pivotal for feature down-sampling and spatial aggregation. We aim to identify optimal methods for preserving discriminative information by exploring different pooling strategies within the search space. *Activation function*, such as ReLU and Swish, introduce non-linearity essential for learning complex patterns. By incorporating both ReLU and Swish, we leverage their respective strengths. *Optimization algorithm*, including Adam, and to improve superior generalization performance for reID tasks. *Loss function*, such as Cross Entropy Loss, Triplet Loss, Quadruplet Loss, and Margin Sample Mining Loss, are used for metric learning losses. Through the inclusion of diverse loss functions, we aim to use their unique properties to effectively model class imbalance issues, which is essential for discriminative feature learning in reID. Finally, the choice of *Distance metric*, encompassing Euclidean, Cityblock, Minkowski, and Sqeuclidean distances, whose equations are given in Eqs. 5.10, 5.11, 5.12, 5.13, which directly impacts similarity computation and rank1 accuracy in reID evaluation.

$$\text{Euclidean distance} = \sqrt{\sum_{i=1}^{n}(a_i - b_i)^2} \tag{5.10}$$

$$\text{City block distance} = \sum_{i=1}^{n}|a_i - b_i| \tag{5.11}$$

$$\text{Minkowski distance} = \left(\sum_{i=1}^{n}|a_i - b_i|^p\right)^{\frac{1}{p}} \tag{5.12}$$

$$\text{Squared Euclidean distance} = \sum_{i=1}^{n}(a_i - b_i)^2 \tag{5.13}$$

Grasshoppers are encoded from the search space using a proposed encoding scheme. This scheme uses variable length encoding to represent various *Block type* and their depths, allowing for flexible architecture selection. 2 bits are allocated to encode each of the parameters: *Size of input*, *Loss function*, and *Distance metric*. 1 bit is used to each parameter representing *Pooling operation*, *Activation function*, and *Optimization algorithm*.

5.2.2 Search objective

Rank-1 accuracy is a key metric for evaluating the performance of object reidentification. The rank-1 accuracy measures the percentage of correctly identified probe images where the top-ranked match is correct. The rank-1 accuracy can be defined as follows:

Let P be the set of probe images and G be the set of gallery images. For each probe image p in P, the reidentification algorithm returns a ranked list of gallery images g_1, g_2, ..., g_n in G, where n is the total number of gallery images. The rank-1 accuracy denoted as $r1$, which can be defined as:

$$r1 = \frac{1}{m}\sum(p \in P)[f(p, g_1)] \tag{5.14}$$

Where, m is the total number of probe images, $f(p, g_1)$ is an indicator function that equals 1 if the top-ranked gallery image g_1 is the correct match for the probe image p, and

0 otherwise.

Eq. 5.14 represents the average number of correct top-ranked matches across all probe images, expressed as a percentage of the total number of probe images.

5.2.3 GOA based NAS optimization

This section discusses the proposed NAS approach's comprehensive framework, MNAS-reID, utilizing the GOA. The proposed Nature-inspired Optimization based NAS algorithm explained in Algorithm 5.5, begins by generating a diverse population of Grasshoppers, each representing a unique neural network architecture within the search space. These architectures are then encoded and trained using the provided training images (I_{train}), along with specified encoding schemes and search space. The fitness of each trained model is then evaluated using a test dataset (I_{test}), and measures the rank-1 accuracy within Eq. 5.14. The model with the highest fitness, denoted as T_{best}, represents the best-performing architecture discovered in that iteration.

Throughout the iterative optimization process, the algorithm adjusts the positions of Grasshoppers based on their fitness evaluations. This includes updating parameters like C using Eq. 5.7, normalizing distances between Grasshoppers, and updating their positions with Eq. 5.6. The iterative loop runs until it reaches the maximum number of iterations (t_{max}). The proposed approach uses the GOA to efficiently explore the proposed NAS search space, resulting in superior performance for the given task.

Algorithm 5.5 Proposed GOA based NAS approach

```
Input: train_images $(I)_{train}$, test_images $(I)_{test}$ = (P,G), $N$.
Output: rank-1 accuracy of the best model ( $T_{best}$)
Generate the initial random population of Grasshoppers $Y_i$ of size $N$
Initialize $C_{max}$,$C_{min}$,$t_{max}$,$Ia$,$La$,t=1.
while t<$t_{max}$ do
    Encode the DNN models from initial Grasshopper population, using encoding scheme
    Train DNN model with given input data $(I)_{train}$
    Evaluate the fitness (rank-1 accuracy given in Eq. 5.14) of each trained model, using $(I)_{test}$ corresponding to its grasshopper($Y_i$)
    The grasshopper with the highest fitness is assigned to $T_{best}$
    for $i = 1, 2, ...., N$ do
        Update the value of $C$ by using the computation given in Eq. 5.7
        Normalize the distance between grasshoppers
        Update the position of the current grasshopper using Eq. 5.6
        Bring the current grasshopper back if it is outside the boundaries
        Update t=t+1
```

5.3 Experimental results and analysis

In this section, we present the experimental evolution of the proposed approach on two motorcycle reID datasets, i.e., MoRe [43] and BPReID [42]. We first discuss the implementation details and evaluation metrics. The comparison of the performance of the proposed approach against the existing reID also approaches is presented in Section 5.3.1. Finally, an analysis of the results of the proposed approach is presented in Section 5.3.4.

Implementation details:

In this work, we use the TensorFlow framework for implementing the proposed approach. The experimental study is performed on a computer with an Intel Xeon(R) Silver

4110 CPU running at 2.10GHz, 64GB of RAM, and one NVIDIA GeForce RTX 2080 Ti GPU, with CUDA 11.1 in the Linux platform. The Rotation, Translation, and Horizontal flipping operations are used for data augmentation. Attraction intensity Ia is considered 0.5, and attraction length scale La is considered 1.5. C_{max} is 1, C_{min} is 0.00004 and maximum number of iterations t_{max} is considered to be 20. The batch size is 24, and each batch is sampled with randomly selected 6 identities and 4 images per identity.

Datasets and evaluation metrics:

We evaluate the proposed approach on two motorcycle reID datasets, i.e., MoRe and BPReID. The MoRe dataset [43] consists of 14,141 images of 3,827 identities. As per the standard evaluation scheme, 1913 identities are used for training, and 1914 identities are used for testing. The BPReID dataset [42] contains 18,763 motorcycle images of 940 identities. The details of these datasets are given in Table 5.2. The Rank-1 accuracy given in Eq. 5.14 and mAP [87] are considered in this work to evaluate the reID task. The mAP is widely used to evaluate the performance of the convolutional networks for reID. We compare the performance of the proposed approach against the existing approaches by using these two evaluation metrics on the two datasets.

Table 5.2: Number of Images, number identities in MoRe and BPReID datasets

Dataset	#Cameras	#IDs	#Images
Motorcycles BPReID	6	940	18,763
MoRe	10	3,827	14,141

5.3.1 Experimental results

We evaluate the performance of existing Motorcycle re-identification algorithms against our proposed approach on the MoRe dataset and BPReID dataset and summarize these in Table 5.3. Since we specifically focus on motorcycles, the number of images our approach considers differs from the rider reID approach mentioned in [44]. Therefore, we cannot directly compare our results with theirs for the BPReID dataset.

In the MoRe [43] method, a pre-trained ResNet50 model serves as the backbone for feature extraction in classification task. A Fully connected layer is added on top of the backbone to map features to the N identities for classification. Various training tricks are used, including Label Smoothing, to prevent overfitting by introducing slight modifications to training labels and the Warmup Learning Rate, which aids the convergence of the model through a gradual increase in learning rate during initial epochs. Additionally, BNNeck integrates a Batch Normalization layer aligned with classification and metric learning losses, while Last Stride adjusts the final down-sampling operation to enhance spatial resolution in learned representations. Center Loss minimizes intra-class variability by minimizing the distance of all samples within a class from its centroid. Metric learning losses such as Triplet Loss construct triplets of anchor, positive, and negative samples, encouraging the anchor to be closer to the positive sample than the negative one by a margin. Quadruplet Loss extends this by introducing an additional negative pair, further promoting inter-class separation. MSML Loss selects hard positive and negative samples within each batch to maximize the margin between them. These techniques collectively contributed to improving experimental outcomes in the re-identification task.

The Rider reID [44] utilized a ResNet50 as the backbone and Pyramid Attention Network (PANet) to integrate attention computation within both spatial and channel dimensions. This PANet module guides the network to progressively emphasize crucial regions by incorporating pyramid attention after multiple stages. Operating across multiple scales, the pyramid structure captures attention efficiently. Spatial attention directs focus to discriminative regions in input feature maps, while channel attention prioritizes channels with stronger responses facilitated by the use of a Squeezed Excitation (SE) block. The BNNeck structure is employed for batch normalization. Training the network involves employing triplet loss and ID loss. To prevent overfitting and to maintain stability during training, Label Smoothing is applied to the ID loss. This proposed training loss comprises the sum of triplet loss and ID loss, ensuring a balanced optimization approach.

The AANet [45] used EfficientNetV2 as the backbone and integrated custom Atrous Attention blocks into the network, creating a feature-pyramid network capable of capturing both global and local information effectively. The AANet, used for feature extraction, incorporates EfficientNet architecture, optimizing depth, width, and resolution for enhanced visual recognition performance. Global average pooling as the final layer improves efficiency and reduces feature dimensionality, refining the model to increase robustness and reduce overfitting. The Atrous Attention block, a modification of the local branch in DOLG, comprises multi-atrous convolution layers and a self-attention module, facilitating information extraction over image regions. Atrous Convolution widens the receptive field with different dilated rates to consider scale variations. Following Multi-Atrous Convolution, outputs from various dilated rates are concatenated and forwarded to subsequent stages. The researchers also integrated Supervised Contrastive loss into the training pipeline by combining it with Arcface loss to achieve superior re-identification results, effectively improving both inter-class distinctions and intra-class differentiations.

Table 5.3 shows that the proposed approach achieved 90.24% and 92.14% in performance for r1 and mAP metrics, respectively, on the MoRe dataset. For the BPReID dataset, the proposed approach achieved a performance of 43.76% and 52.64% on the r1 and mAP metrics, respectively. From Table 5.3, we can conclude that our proposed approach surpasses the baseline model as well as the recent models in terms of both r1 and mAP metrics. Furthermore, we compared the sizes of existing models and the proposed approach in terms of the number of parameters. The proposed approach has fewer parameters than existing models, highlighting its effectiveness on both the MoRe and BPReID datasets.

5.3.2 Network architecture identified by the proposed approach

The parameter values for the final network architecture identified by the proposed GOA based NAS approach is summarized in Table 5.4. From the Table, it can be observed that EfficientV2 is the *Block type*, which serves as the backbone network. For *Pooling operation*, average pooling is selected, and ReLU is chosen as the *Activation function*. The *Size of input* is chosen as 256×256 pixels. The city block distance is selected as the *Distance metric*. Quadruplet Loss is adopted as the *Loss function*, while the Adam optimizer is chosen as the *Optimization algorithm*.

Table 5.3: The performance comparison of rank1 accuracy and mAP of the top-1 model obtained from the proposed approach against existing re-identification approaches on the MoRe dataset and BPReID dataset

Method	Dataset				#params (M)
	MoRe		BPReID		
	r1 (%)	mAP (%)	r1 (%)	mAP (%)	
MoRe [43]	83.41	86.38	16.94	23.08	23.58
Rider reID [44]	89.1	90.9	-	-	27.77
AANet [45]	86.60	88.32	-	-	-
MNASreID	**90.24**	**92.14**	**43.76**	**52.64**	**20.33**

Note: The best values are highlighted in bold.

Table 5.4: Final network architecture parameter values obtained by proposed approach

Parameter	**Value Obtained**
Block type	EfficientnetV2
Size of input	256×256
Activation function	ReLU
Pooling operation	Average
Optimization algorithm	Adam
Loss function	Quadruplet Loss
Distance metric	Cityblock

5.3.3 Computational complexity analysis

The proposed GOA based NAS framework involves several steps, including initial population generation, encoding, training, fitness evaluation, and updating grasshopper positions. The time complexity of Algorithm 3.1 primarily stems from: (i) Model training and fitness evaluation (E_t) given in steps 5 and 6, (ii) Updating grasshopper positions (E_u) given in steps 9 to 13, (iii) Number of iterations (t_{max}) and the number of grasshoppers (N). E_t depends upon the generated network structures and their size. Hence, we can express the time complexity of the proposed algorithm to be O$(t_{max} \times (E_t + (N \times E_u)))$. The exact computation of the various parameters in the time complexity is reliant on its dependent factors.

To streamline the computationally expensive process of NAS, we adopted a strategy to accelerate the optimization. Firstly, we incorporated pre-termination criteria, which involved monitoring the model's loss after 20 epochs. If the loss does not decrease significantly within this period, we terminate the training early. Otherwise, we continued training the model for a total of 80 epochs and recorded the results. Additionally, to optimize the NAS process further, we implemented a caching strategy to avoid retraining the same model if it had been generated in previous iterations of the GOA. This approach allowed us to skip redundant training of previously generated model architectures during optimization, thereby speeding up the search for the best architecture.

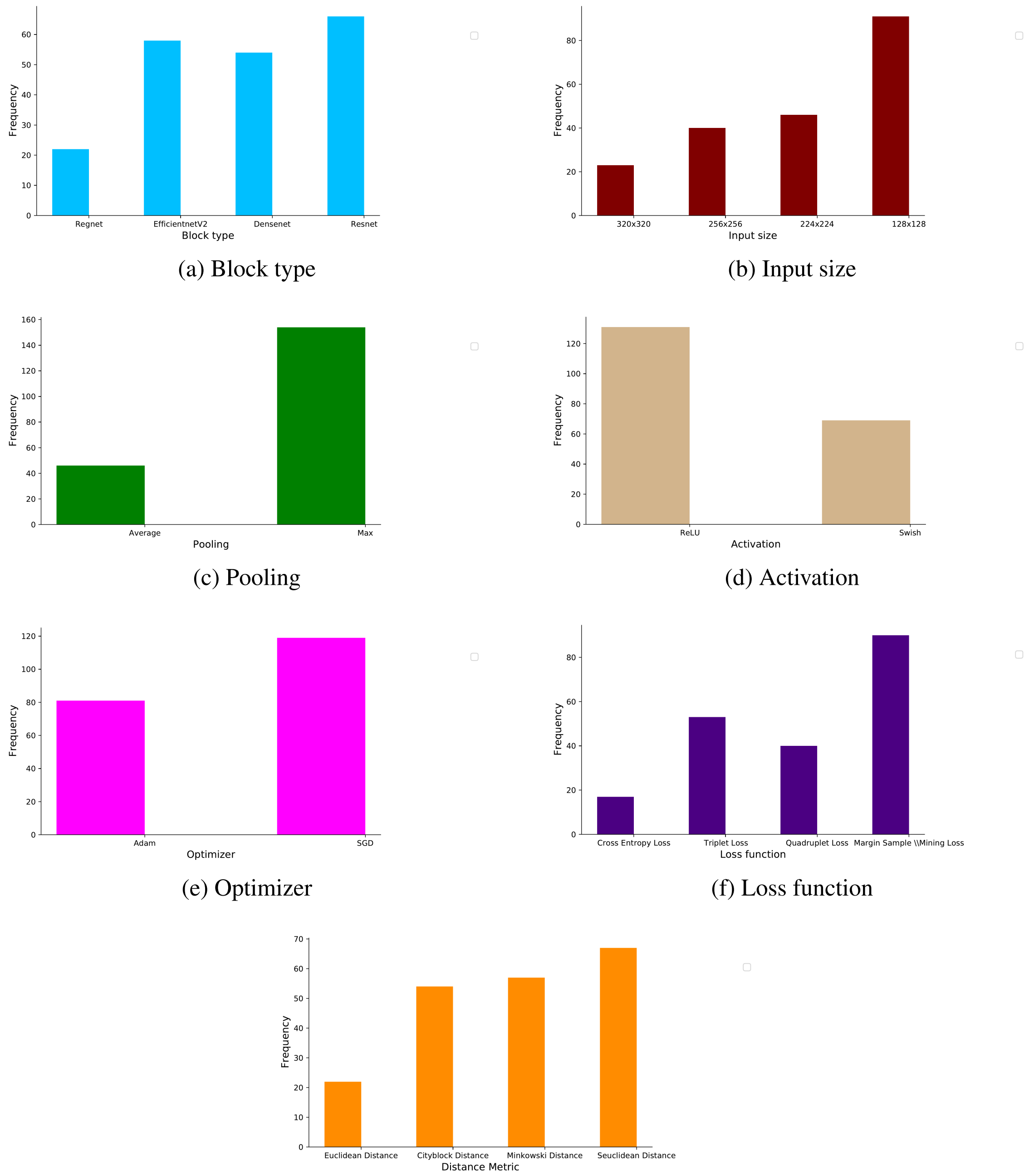
5.3.4 Experimental analysis

The number of times the NAS search space parameter value is used by the NAS exploration process to identify the optimal DNN model for the reID task is analyzed. The results of this analysis depicting the frequency of usage of each parameter value by the proposed MNASreID is shown in Figure 5.2. This reveals the trends in using parameter values by the NAS exploration process. Among the block types, 'Resnet' and 'EfficientnetV2' are the most commonly chosen block types, surpassing other block types. For pooling operation, 'Max' pooling is more favored than 'Avg' pooling. The 'ReLU' activation function emerged as the dominant choice, outperforming 'Swish'. The 'SGD' optimizer is preferred over the 'Adam' optimizer. Among the distance metrics, the 'Sqeuclidean distance' metric is selected most frequently compared to other distance metrics. As per the size of the input, 128×128 pixels is more prevalent compared to other sizes, and for the loss function, 'Margin Sample Mining Loss' is the dominant choice. These observations reveal the preference for these particular parameter values during the NAS optimization process. The analysis of parameter values used during NAS exploration for the reID task reveals both agreement and disagreement with the final identified model by the proposed approach. EfficientnetV2, which was frequently selected during the search, was chosen as the block type in the final model, indicating its high performance. Despite Max pooling being preferred during exploration, the final model employs Average pooling. ReLU's frequent selection is directly related to its use in the final architecture. Although SGD was more commonly explored, Adam was used in the final model for better optimization. While the Sqeuclidean distance was preferred during the exploration, the City Block distance was chosen for the final model. The exploration leaned towards a smaller 128×128 pixel input size, but the final model opted for a larger 256×256 size for better feature representation. Lastly, although Margin Sample Mining Loss was commonly selected, Quadruplet Loss was chosen for its superior performance in proposed reID task.

The rank-1 accuracy of Grasshoppers in each iteration is analysed by evaluating the performance of the best grasshoppers. Fig. 5.3 shows the trajectory details of the MNASreID rank-1 accuracy achieved by the best grasshopper after each iteration. The figure shows that, as the iterations progress, the accuracy steadily increases and eventually reaches a convergence point at subsequent iterations.

Analysis of occlusion and collision scenarios:

Addressing “occlusion" presents a significant challenge in object re-identification, particularly in surveillance or tracking scenarios where objects may become obscured by obstacles or other objects. In our proposed approach, to evaluate performance in occlusion scenarios, we segregated occluded images from the test set of both the MoRe and BPReID datasets. We identified 158 occluded images in the MoRe dataset and 177 occluded images in the BPReID dataset. Sample images with occlusion from MoRe and BPReID datasets are shown in Fig. 5.4 and Fig. 5.5, respectively, where the first row shows gallery images and the second row shows corresponding probe images. The model identified by the proposed approach on the MoRe dataset used the City block distance metric as given in Table 5.4. The performance of this model on occluded images of MoRe and BPReID datasets is shown in Table 5.5. The model identified by the proposed approach and trained on the BPReID dataset is evaluated on occluded images, whose results are shown in the second row of Table 5.4. From the cross-dataset evaluation results shown in Table 5.4, it can be observed that the best performance can be achieved when the model is trained and tested on the same dataset. The performance (mAP) of the model identified by the proposed approach on occluded images of the MoRe dataset using Euclidean, Sqeuclidean, Minkowski, and City block distance metrics is 96.11 %, 96.11 %, 96.01 %, and 95.95 %, respectively.



(a) Block type

(b) Input size

(c) Pooling

(d) Activation

(e) Optimizer

(f) Loss function

(g) Distance metric

Figure 5.2: The frequency of parameter values appearing in the population during 20 iterations of the MNASreID optimization process on the MoRe dataset.

Similarly, on occluded images of the BPReID dataset, the performance (mAP) of the proposed approach using Euclidean, Sqeuclidean, Minkowski, and Cityblock distance metrics is 88.94 %, 88.94 %, 88.27 %, and 88.69 %, respectively. From this analysis, it can be ob-

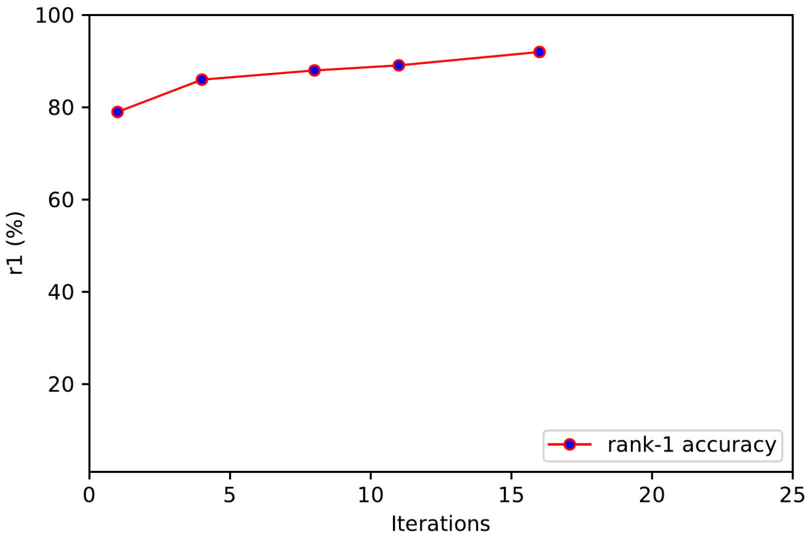


Figure 5.3: Iteration wise rank-1 accuracy of the MNASreID approach for the best Grasshopper

served that Euclidean and Sqeuclidean distances give better results than Cityblock distance on occluded images of both datasets. When a similar experiment using various distance metrics was conducted on the entire MoRe dataset, the Cityblock distance metric achieved the best results. When this study was repeated for the entire BPReID dataset, the Cityblock distance metric achieved the best results. From this analysis, it can be concluded that the Cityblock distance metric gives optimal results in all possible scenarios, while the Euclidean/Sqeuclidean distance metric gives better results for occlusion scenarios. The details of the computation of these distances are given in Eq. 5.10 to Eq. 5.13.

In object re-identification, “collision" refers to the situation where two or more objects physically interact or overlap in a scene, posing challenges in identifying and tracking individual objects, particularly when they share similar appearance features. To evaluate the performance of the proposed approach in collision scenarios, we identified 14 images with similar appearance features from the MoRe test set. Sample images are depicted in Fig. 5.6, where the second row shows similar images with different IDs compared to their corresponding images in the first row. By considering these 14 images in the gallery and probe datasets, we evaluated our proposed approach, which achieved an r1-accuracy of 92.81 % and an r2-accuracy of 100 %. The high r1-accuracy indicates that the proposed

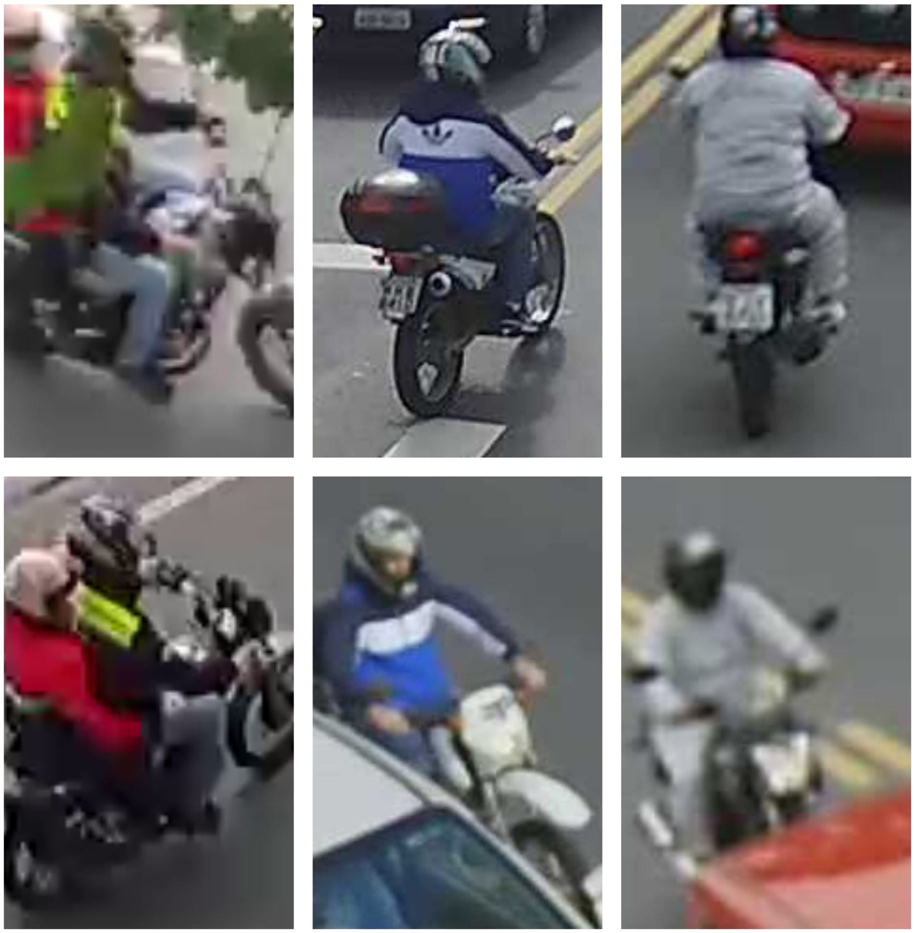


Figure 5.4: Sample occluded images from MoRe dataset

approach was able to correctly identify even under collision conditions. The 100 % r2-accuracy indicates that the proposed approach was able to identify the correct object for all the probing objects using the top-2 distance.

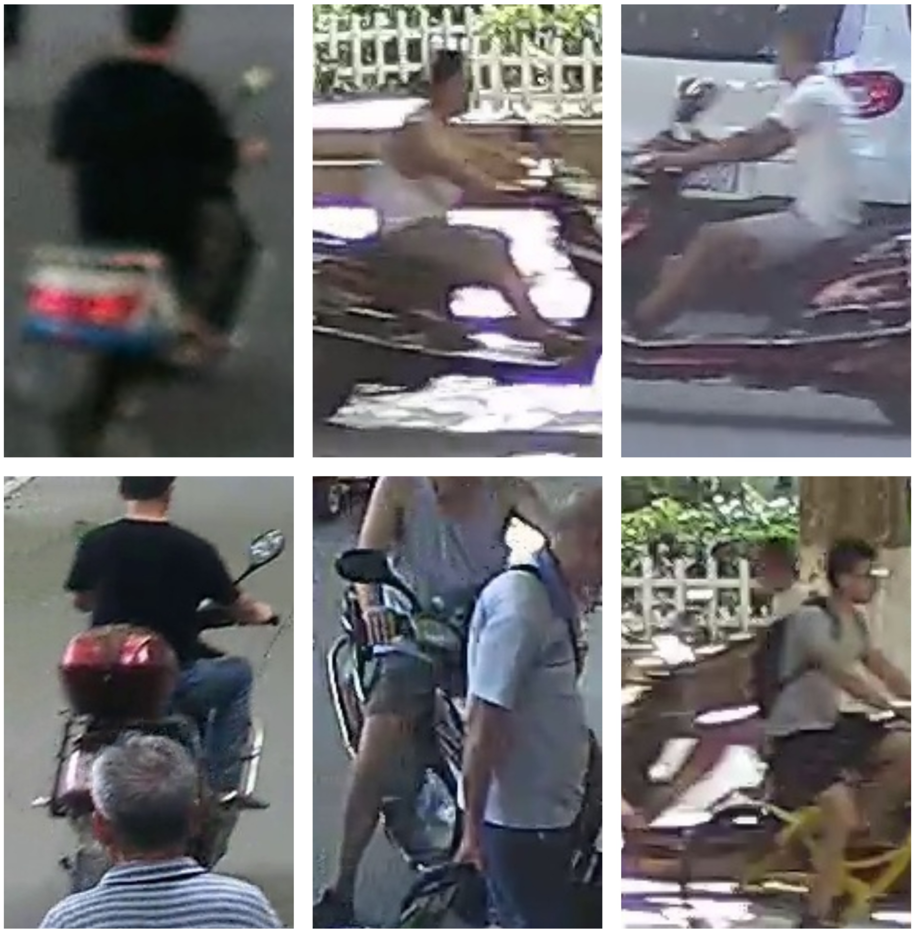


Figure 5.5: Sample occluded images from BPReID dataset

Table 5.5: The performance comparison of rank1 accuracy and mAP of the top-l model obtained from the proposed approach on MoRe dataset and BPReID dataset in occluded scenarios

Method	Dataset			
	MoRe		BPReID	
	r1 (%)	mAP (%)	r1 (%)	mAP (%)
Final model trained on MoRe	**93.57**	**95.95**	37.71	39.01
Final model trained on BPReID	42.14	47.85	**88.57**	**88.69**

Note: The best values are highlighted in bold

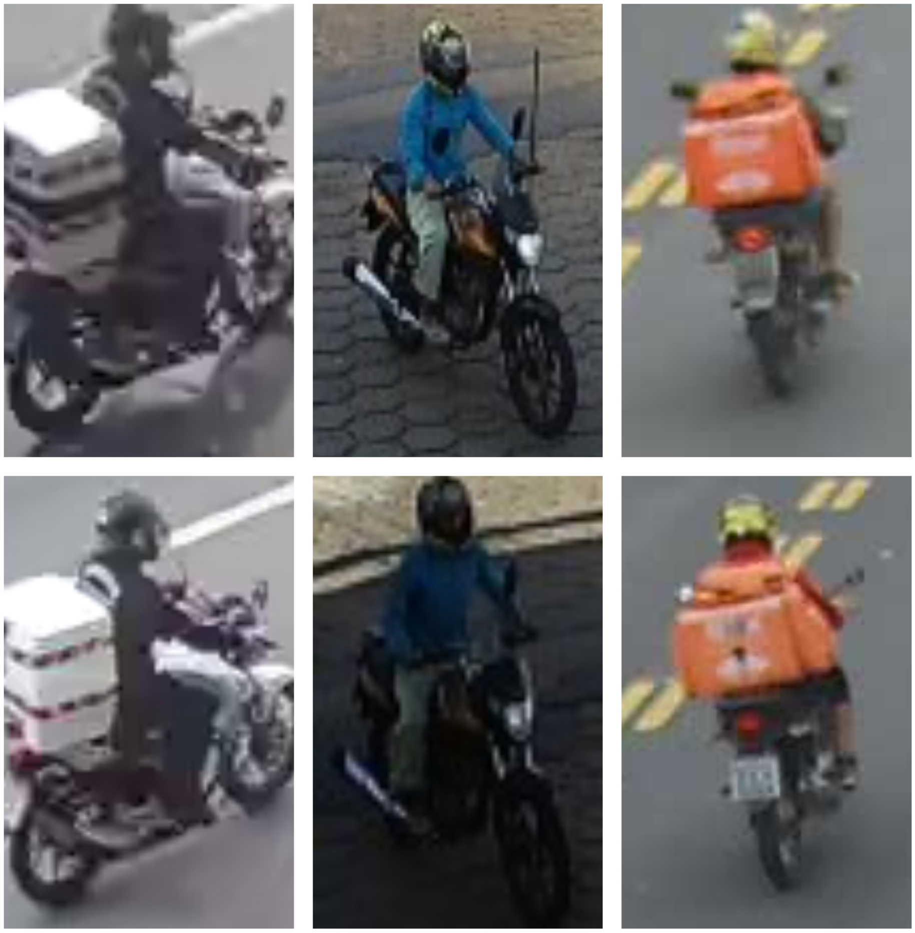


Figure 5.6: Sample images with similar appearance features from MoRe dataset

5.4 Summary

This work presents MNASreID, a novel automated Neural Architecture Search approach utilizing the Grasshopper optimization algorithm as a search strategy specifically designed for motorcycle re-identification. MNASreID efficiently explores both neural network architecture and hyperparameters to find the optimal deep neural network architecture. The performance of MNASreID is compared with the existing approaches for motorcycle re-identification task. On the MoRe dataset, MNASreID exhibits improvements of +1.14 % and +1.24 % in r1 and mAP metrics, respectively, compared to existing methods. Similarly, on the BPReID dataset, MNASreID outperforms existing approaches, exhibiting significant enhancements of +26.82 % and +29.56 % in $r1$ and mAP metrics respectively. These findings show the proposed approach's effectiveness in advancing motorcycle reID performance, establishing clear superiority over existing models. Additionally, an analysis of NAS search space parameter values shows insights on the trends in parameter selection by MNASreID, providing insights into its optimization process.

Chapter 6

Improved Genetic Algorithm based Optimization of Deep Neural Networks for Driver Distraction Detection

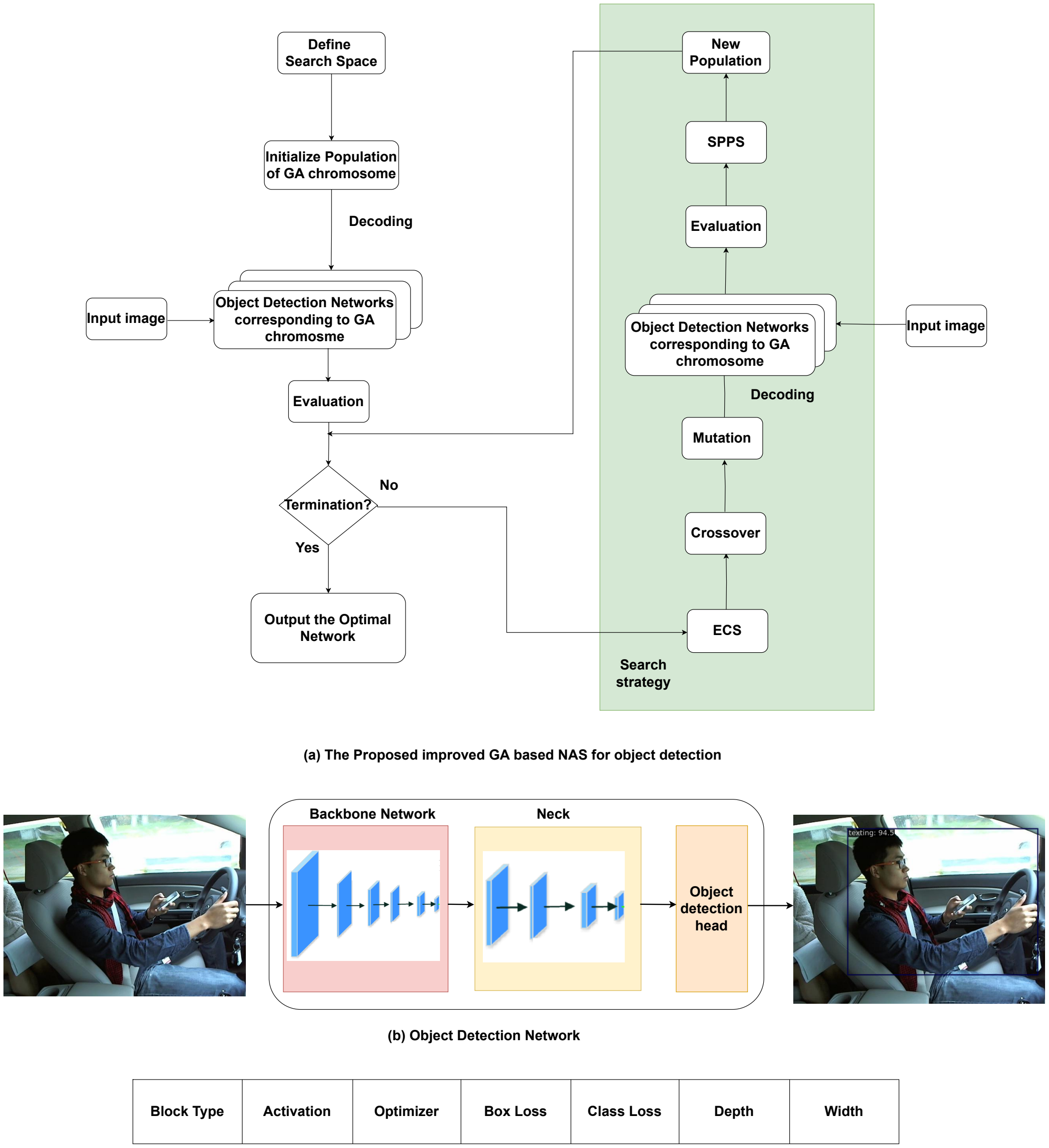
This chapter proposes a NAS based optimization of deep neural networks using a modified genetic algorithm for the task of driver distraction detection.

Chapter Organization:

Proposed approach is described in Section 6.1. Section 6.2 presents the experimental results and analysis. Finally, Section 6.3 summarizes the proposed work.

6.1 Proposed approach

This section discusses the use of NAS to optimize the target object detection network's backbone and training parameters for detecting driver distraction. The components of the proposed model are shown in Fig. 6.1. The initial population is generated at random, and an improved GA that encodes a chromosome, as shown in Fig. 6.1 (c), to represent a YOLO object detection model. The model is trained on driver distraction detection image data to determine the fitness value of each chromosome/YOLO object detection model. A search strategy based on the improved GA is used to generate a new population in each GA cycle, that ends when the termination criteria is met.



(a) The Proposed improved GA based NAS for object detection

(b) Object Detection Network

Block Type	Activation	Optimizer	Box Loss	Class Loss	Depth	Width

(c) Encoding

Figure 6.1: (a) The proposed improved GA based NAS framework for driver distraction task (b) YOLO based object detection network (c) Encoding of GA chromosome.

6.1.1 Search space design for object detection

The efficacy of object detectors is significantly affected by backbone architectures, which play a critical role in extracting essential features from visual data. Due to this reason, we included the backbone related parameters in the search space. The backbone search space is made up of a series of blocks, and each component of the backbone can be divided into different stages based on the resolution of the output. A stage is a collection of blocks that receive features with the same resolution. In this work, the proposed search space consists of four types of blocks: CSPDarknet53 block [62], RepVGG block[63], CSPNeXt blocks [64], and CSPResNet block [65]. We consider 4 stages with the *Depth of blocks* and *Width of blocks* in each stage of the backbone represented with deepen factor and widen factor whose values vary from 0 to 2.

In addition to the backbone search, the proposed search space includes training parameters such as the *Box loss*, and *Class loss*, *Activation function*, and *Optimizer*. For *Box loss*, we explored {IoU loss, GIoU loss, SIoU loss, CIoU loss}, and for *Class loss*, we considered {Cross Entropy loss, Focal loss, VariFocal loss, QualityFocal loss}. For the *activation function*, we considered {ReLU, GELU, Swish, SiLU}, and finally, for the *optimizer*, the choices include {SGD, NAdam, Adamax, AdamW}. The details of the parameters and their corresponding values explored in the search space are outlined in Table 6.1. A 16-bit chromosome, as shown in Fig. 6.1, is used as the encoding mechanism for both architecture and the training properties. It uses 2 bits for the *Type of block*, 2 bits for the *Activation function*, 2 bits for the *Box loss*, 2 bits for *Class loss*, 2 bits for the *Optimizer*, 3 bits for the *Depth of blocks*, and 3 bits for the *Width of blocks*.

Table 6.1: Search space parameters and their values considered in this work

Type of parameter	**Range**
Type of block	{CSPDarknet53, RepVGG, CSPNeXt, CSPResNet}
Activation	{ReLU, GELU, Swish, and SiLU}
Optimizer	{SGD, NAdam, Adamax and AdamW}
Box loss	{IoU loss, GIoU loss, SIoU loss, CIoU loss}
Class loss	{Cross Entropy loss, Focal loss, VariFocal loss, QualityFocal loss}
Depth of blocks	{0.33,0.5,0.67,1.0,1.33,1.5,1.67,2}
Width of blocks	{0.25,0.5,0.75,1.0,1.25,1.5,1.75,2}

6.1.2 An improved GA based NAS optimization

In this section, we will discuss the modified selection strategy and species protection based on next generation population used in the proposed improved GA based NAS framework.

Evaluation Correction based Selection (ECS):

The traditional selection strategy uses various approaches to select the best chromosomes as parents for the generation of an offspring population. However, depending only on the validation mAP for performance estimation may result in a better network, but it is computationally inefficient. Therefore, we have adopted an evaluation correction based selection approach for the “selection" operation, which is inspired by [17]. In this method, the network architecture is assessed based on the validation mAP. If a significant difference exists in the validation mAP, the network with the higher mAP is chosen. In cases where the validation mAP scores are similar, preference is given to the network with fewer parameters. This approach ensures the selection of a network with superior validation mAP as well as fewer parameters. The pseudo-code for this method is outlined in Algorithm 6.6.

Algorithm 6.6 Evaluation Correction based Selection

```
Input: Population ($X_i^t$), population size $N$, number of parameters $p$ and fitness $f$ of each
    individual in $X_i^t$, threshold $\alpha$.
Output: Best individual $X_{best}$
for $i = 1, 2, ...., N$ do
    $L \leftarrow$ Select three individuals at random from the population
    while $|A| > 1$ do
        $X_1, X_2 \leftarrow$ Select 2 individuals from $L$
        $L \leftarrow L - \{X_1, X_2\}$
        $f_1, f_2 \leftarrow$ Fitness of $X_1, X_2$
        $p_1, p_2 \leftarrow$ Number of parameters of $X_1, X_2$
        if $|f_1 - f_2| < \alpha$ then
            Put the individual with fewer parameters in $\{X_1, X_2\}$ back into $L$
        else
            Put the individual with greater accuracy in $\{X_1, X_2\}$ back into $L$
    $X_{\text{best}} \leftarrow$ Return the best individual in $L$
```

Species Protection based Next Generation Population (SPNGP):

Ensuring the diversity of the population, i.e., different types during the evolutionary process, may improve the algorithm's global performance. This work presents a species protection based next generation population, which is inspired by [17], to ensure the diversity of neural network architectures considered in each GA generation. Algorithm 6.7 describes the logic for this approach. SPNGP begins by classifying the population P into k distinct species, denoted by P_k. Species s is chosen with uniform probability from k distinct species $p_1, p_2, ..., p_k$. One individual p_best is picked at random from s and removed from P_k without replacement. However, a problem occurs due to a lack of competition among distinct species. To address this issue, SPNGP includes a random number r that regulates whether the SPNGP approach is used to balance competitiveness within and outside the species. SPNGP also includes an elite retention method to protect the population's best individuals from elimination during evolution. The technique is to preserve the top $2N \times \gamma$ individuals in the population before species division, where N denotes the population size of X^t and O^t, X^t represents the parent population, O^t represents the offspring population and γ, is a retention parameter. In this work, the SPNGP considers the backbone network's *type of block* for identifying the s species when splitting the species into P_k. This is because, when compared to the other parameters in the search space, the *type of block* has a major impact on performance.

Algorithm 6.7 Species Protection based Next Generation Population

```
Input: Population $P = X^t \cup O^t$, population size $2N$, elite rate $\gamma$ in the population.
Output: The new population $X^{t+1}$.
$X^{t+1}$ ← Select $2N \times \gamma$ individuals with the highest fitness from population $P$ using
    the elite retention strategy.
$P$ is divided into k(=4) distinct species, denoted as $P_\text{k}$, based on the first two bits of
    GA chromosome representing the type of block, which ranges from 0 to 3.
while $|X^{t+1}| < N$ do
    $r$ ← Generate a random number.
    if $r < 0.5$ then
        With uniform probability select species $s$ from $P_\text{k}$.
        Select the best individual $p_\text{best}$ from $s$ by using Algorithm 6.6.
    else
        Select the best individual $p_\text{best}$ from $P$ by using Algorithm 6.6.
    $X^{t+1} \leftarrow X^{t+1} \cup p_\text{best}$.
Return the new population $X^{t+1}$.
```

The overall framework of proposed improved GA based NAS:

The proposed improved GA based NAS framework is given in algorithm 6.8, which incorporates modified selection and species protection based selection operations. The proposed approach begins by randomly generating an initial population X^0. Fitness values are then calculated for the N individuals in X^0. The framework proceeds with T rounds of iterative evolution.

During the t^{th} generation evolution process, the ECS operation is initially applied to select the best chromosomes with fewer parameters. Subsequently, a crossover and mutation are applied to generate offspring. Finally, the proposed approach utilizes an SPNGP operation to select the next generation population X^{t+1} from the union of the current population and offspring population X^t. This selection process ensures diversity and balances competitiveness. The framework then proceeds to the next round of the evolutionary process. After completing the T rounds of the evolutionary process, the optimal network architecture will be selected.

Algorithm 6.8 The overall framework of proposed improved GA based NAS optimization

```
Input: train_images $I_{train}$, number of GA generations $T$, population size $N$.
Output: The optimal network architecture.
$X^0 \leftarrow$ Generate an initial population randomly.
Decode the vectors of $X^0$ into object detection networks
Calculate the fitness and parameters of each decoded network using $I_{train}$.
$t \leftarrow 1$.
while $t < T$ do
    $O^t \leftarrow \{\}$.
    while $|O^t| < N$ do
        Select two individuals using using Algorithm 6.6.
        Generate two offsprings $o_1$, $o_2$ with crossover
    and mutation operations.
        Calculate the fitness and parameters of the decoded network
    using $I_{train}$.
        $O^t \leftarrow O^t \cup \{o_1, o_2\}$.
    $X^{t+1} \leftarrow$ Generate next-generation population from $X^t \cup O^t$
    using Algorithm 6.7.
    $t \leftarrow t + 1$.
Return the optimal network architecture
```

6.2 Experimental results and analysis

In this section, we present the evaluation metric and the experimental settings in Section 6.2.1. Subsequently, Section 6.2.2 presents the comparative study of the proposed driver distraction detection approach using the Distracted Driver Detection Image Dataset [66] and the Distracted Driving Computer Vision Project [67] datasets. Section 6.2.5 discusses the analysis of the proposed approach.

6.2.1 Experimental settings

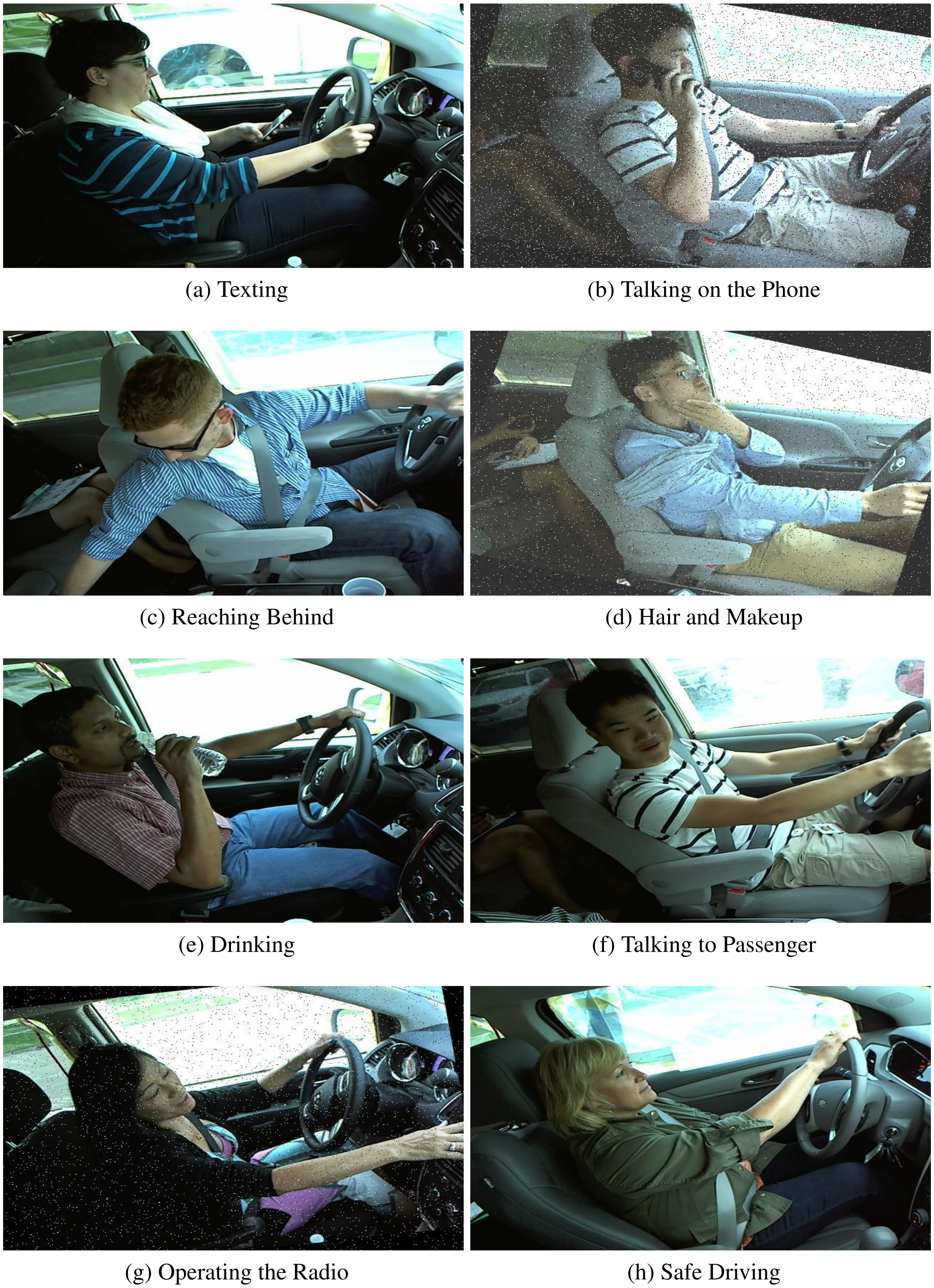
The experiments were carried out on a system with an Intel Xeon(R) Silver 4110 CPU running at 2.10GHz, 64GB of RAM, and a single NVIDIA GeForce RTX 2080 Ti GPU. Which were done on a Linux operating system, utilizing CUDA and the PyTorch framework. While exploring the search space, if the SGD optimizer was used, the parameters chosen were a momentum of 0.9 and an initial learning rate of 0.01. When using the AdamW, NAdam, or Adamax optimizers, We have chosen the weight decay of 0.0005 and an initial learning rate of 0.0001. The input image size is set to 640 × 640 pixels. This study considered a crossover rate of $C_r = 0.5$. The additional parameters for the improved GA are configured with an elite rate (γ) of 0.3 and a threshold (α) of 2. The population size N was set to 20. The experiments were carried out for $T (= 10)$ generations. Initially, all models identified by the NAS were trained for 20 epochs. The best model in the final GA population is trained for an additional 80 epochs and is considered the best model generated by the proposed approach.

Datasets: Popular datasets like SFDDD (State Farm Distracted Driver Detection) [147] and AUCD2 [107] are designed for direct image classification tasks, classifying images into different types of distractions like Texting or Drinking. However, these datasets lack localization capabilities, which means they do not specify where the distraction occurs in the image, such as on the driver's head rotation or hand movements. In contrast, the Distracted Driver Detection Image (DDDI) [66] and Distracted Driving Computer Vision Project (DDCV) [67] datasets provide both classification and localization information. They include annotated data that provides bounding boxes to identify where distractions occur in the image. This is critical for creating models that can accurately identify distractions, thereby increasing the robustness and reliability of driver distraction detection systems. Furthermore, these datasets cover a similar range of classes as SFDDD and AUCDD2 datasets, ensuring that models trained on them can detect the same types of distractions while benefiting from localization of action. This makes them more appropriate for real-world applications, such as video-based analysis, where precise action tracking and localization are critical for effective driver monitoring and safety improvement.

The proposed approach was evaluated using two driver distraction datasets: the Distracted Driver Detection Image (DDDI) dataset [66] and the Distracted Driving Computer Vision Project (DDCV) dataset [67]. The Distracted Driver Detection Image Dataset consists of 2,000 images, encompassing 8 classes (Talking on the Phone, Hair and Makeup, Talking to Passenger, Texting, Operating the Radio, Reaching Behind, Drinking, and Safe Driving). Sample images from the dataset are shown in Fig. 6.2. In this dataset, 1,391 images are available for training, 398 images for validation, and 211 images for testing. The DDCV includes 8,865 images, spanning 12 classes (Eyes Open, Eyes Closed, Yawning, Nodding Off, Talking to Passenger, Talking on the Phone, Drinking, Operating the Radio, Hair and Makeup, Reaching Behind, Texting, and Safe Driving). Sample images from the dataset are shown in Fig. 6.3. In this dataset, 6,860 images are given for training, 1,000 images for validation, and 1,005 images for testing.

6.2.2 Comparision with state of the art models

We evaluated existing object detection models, including one-stage models like YOLOv5 [148] and YOLOv7 [149] as well as two-stage models like Faster R-CNN [14], Mask R-CNN [15], Sparse R-CNN [132] and NAS based model like MAE-DET [18] on the DDDI, DDCV datasets. The optimal architecture generated by the proposed method has an mAP of 86.8 % and 85.1 % on validation and test data, respectively, for the DDDI dataset. Also, for the DDCV dataset, the proposed method achieved an mAP of 76.4 % on validation data and 51.5 % on test data, whose results are presented in Table 6.2. From this table, it can be observed that our proposed approach outperforms these benchmark one-stage and



(a) Texting

(b) Talking on the Phone

(c) Reaching Behind

(d) Hair and Makeup

(e) Drinking

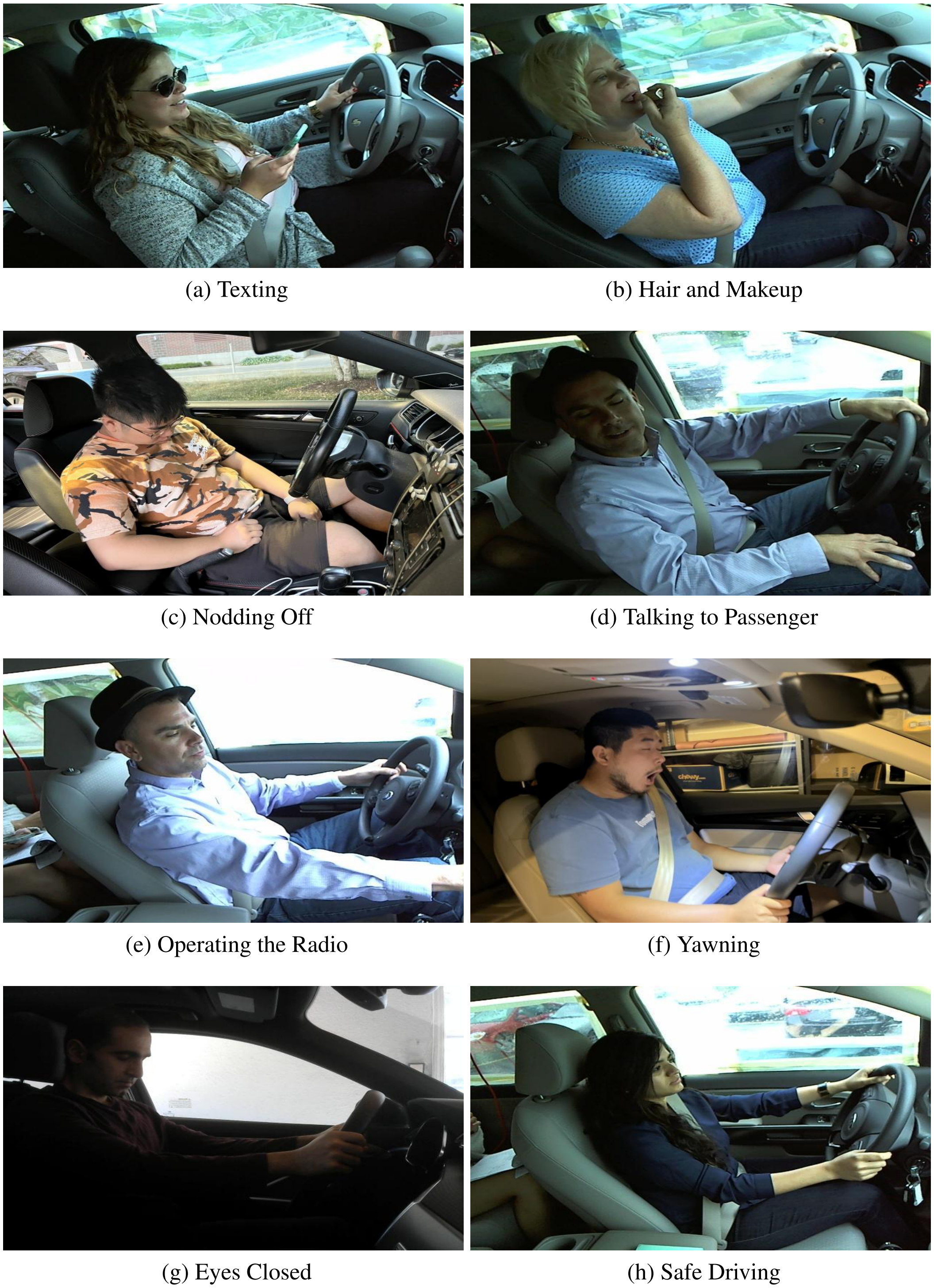
(f) Talking to Passenger

(g) Operating the Radio

(h) Safe Driving

Figure 6.2: Sample images of driver distraction categories in DDDI dataset [66]

two-stage models; on the DDCV dataset, the proposed approach outperformed the existing methods for detecting driver distraction behavior. On the DDDI dataset, the proposed ap-


(a) Texting

(b) Hair and Makeup

(c) Nodding Off

(d) Talking to Passenger

(e) Operating the Radio

(f) Yawning

(g) Eyes Closed

(h) Safe Driving

Figure 6.3: Sample images of driver distraction categories in DDCV dataset [67]

proach gives comparable performance with a recent model with fewer parameters. Fig. 6.4 and Fig. 6.5 demonstrate the detection of driver distraction types by our proposed approach on some test images from the DDDI and DDCV datasets, respectively.

Table 6.2: Comparison of mAP of the existing object detection models with proposed approach on two datasets

Method	Dataset				Params
	Roboflow-DDDI		Roboflow-DDCV		
	Valid (%)	Test (%)	Valid (%)	Test (%)	
Faster R-CNN [14]	72.1	74.5	70.9	42.7	63.6 M
Mask R-CNN [15]	78.4	78.1	72.6	41.8	34.4 M
Sparse R-CNN [132]	78.6	81.7	75.3	44.0	96.6 M
MAE-DET [18]	51.3	55.1	65.1	35.4	34.7 M
YOLOv5 [148]	73.2	75.4	74.1	50.6	46.2 M
YOLOv7 [141]	75.2	77.8	75.6	51.2	37.3 M
Proposed appraoch	**86.8**	**85.1**	**76.4**	**51.5**	**21.1 M**

Note: The best values are highlighted in bold.

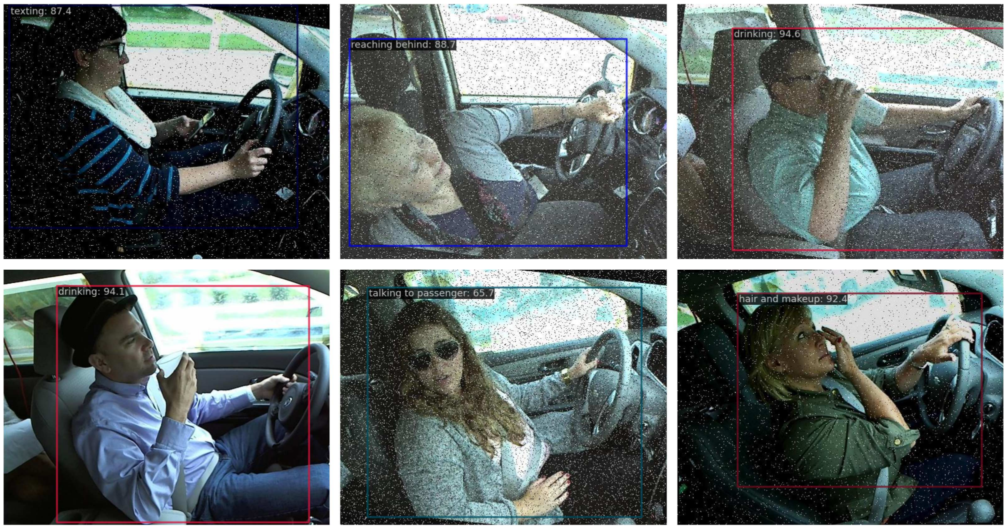


Figure 6.4: Driver distraction detection results of the proposed approach on DDDI dataset

6.2.3 Backbone architectures identified by the proposed approach

The proposed approach minimized the number of parameters and achieved good accuracy. The proposed NAS-based approach effectively searched for the optimal network architecture and parameters for driver distraction detection, whose details are given in Table 6.3. The components of the identified model are: CSPNeXt blocks as *Type of block*, GELU for

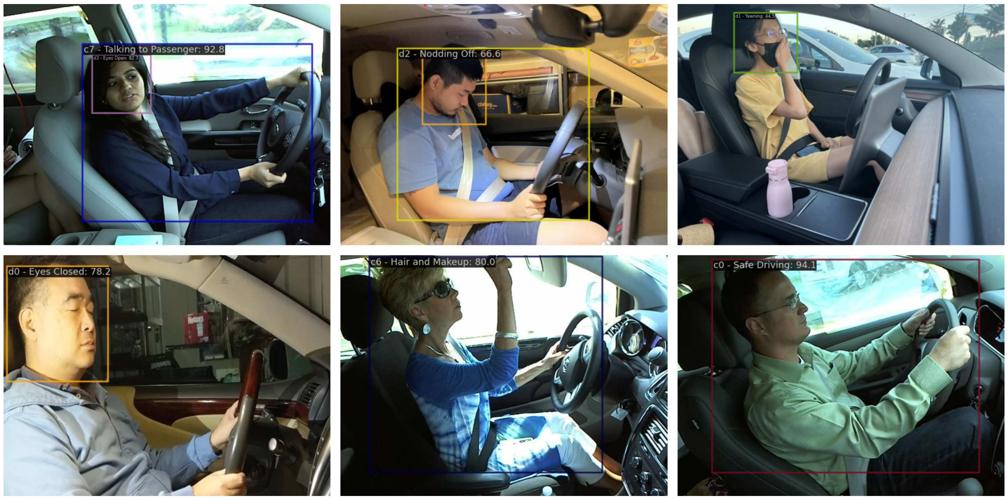


Figure 6.5: Driver distraction detection results of the proposed approach on DDCV dataset

Activation, Cross Entropy Loss for *Class loss*, CIoU loss for *Box loss*, and the AdamW as *Optimizer*, 0.67 as textitDepth of blocks and 0.75 as *Width of blocks*. This approach's primary goal was to balance model complexity and performance by reducing the number of parameters while maintaining high accuracy. The model's depth and width largely determine the number of parameters required. The proposed model achieves high accuracy while reducing depth and width, making it highly efficient and ideal for real-time driver distraction detection.

Table 6.3: The parameter values of the best model identified by the proposed approach

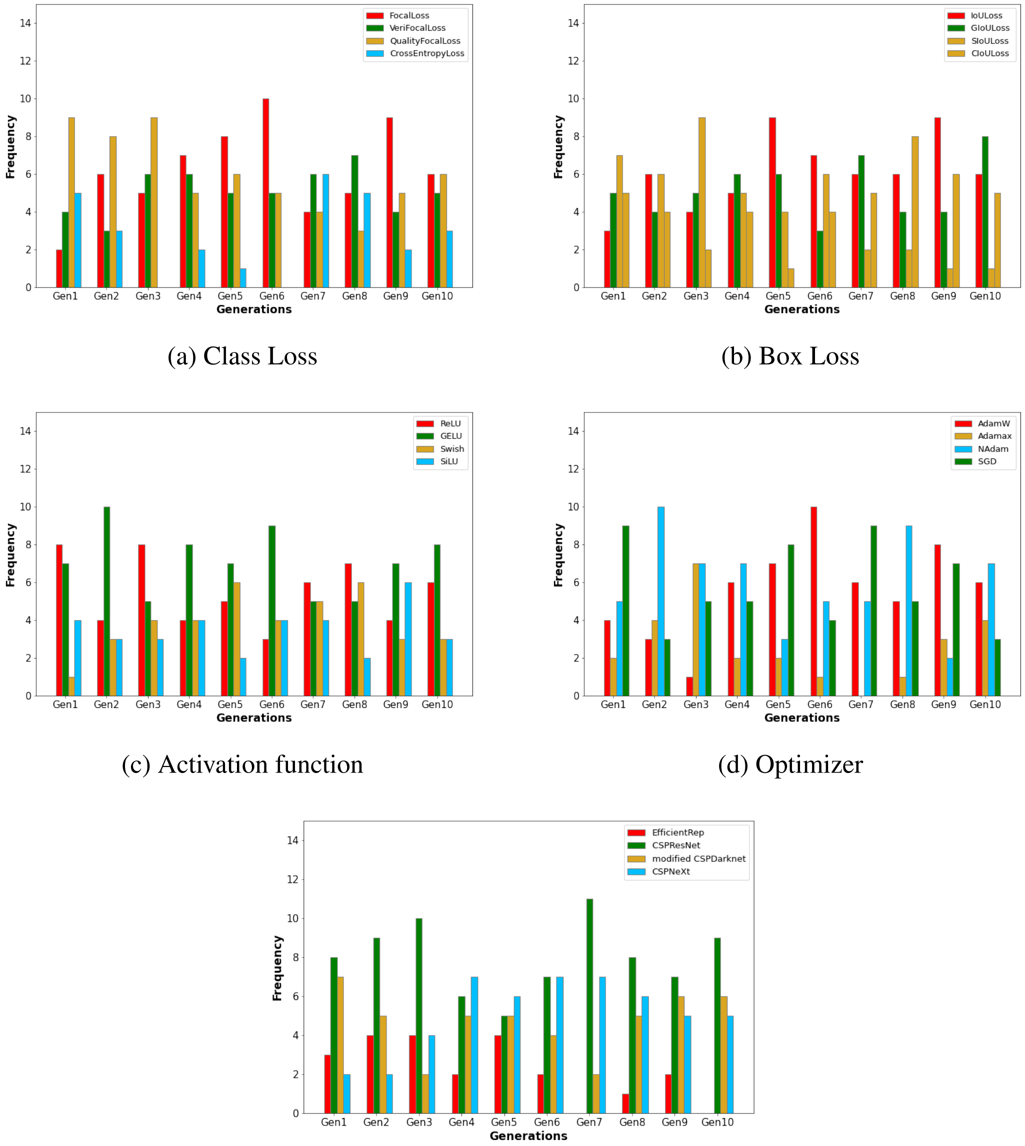
Parameter	**Value Obtained**
Type of block	CSPNeXt
Activation	GELU
Optimizer	AdamW
Box loss	CIoU loss
Class loss	Cross Entropy Loss
Depth of blocks	0.67
Width of blocks	0.75

6.2.4 Computational complexity analysis

The time complexity of the proposed improved GA-based NAS framework is primarily determined by the number of generations (T) and population size (N). The most computationally expensive task is training and evaluating each individual's fitness, which depends on the complexity of the generated DNN architecture and the size of the training dataset, denoted by E_{train}. Other tasks, such as sorting the population based on fitness, take O$(NlogN)$ in Algorithm 6.7 and generating new offspring through crossover and mutation takes O(N) in Algorithm 6.6, which contribute less to the overall time complexity. The overall time complexity of the NAS framework is O$(T \times (N \times E_{\text{train}} + NlogN))$. This denotes that the computational cost scales linearly with the number of generations. Fitness evaluation (E_{train}) increases with the complexity of the DNN model and the size of the training dataset.

6.2.5 Discussion

This section analyzes the parameter values explored by the proposed approach within the search space across various generations of the GA for the DDDI dataset. Figure 6.6 illustrates the distribution of these explored parameters. Several trends are evident from the figure. CSPResNet and CSPNeXt are more frequently selected than other *block types*, and the GELU *activation* function is favored over other options. In terms of *class loss* functions, Quality Focal Loss and Focal Loss are the most commonly used. As the generation number increases, the usage of IoU Loss, GIoU Loss, and CIoU Loss also rises for *box loss* functions. While NAdam is frequently chosen in most generations, AdamW becomes the preferred *optimizer* as the generations progress. This analysis offers insight into the common choices made by the NAS framework, revealing preferences for block type, activation functions, loss functions, and optimizers across multiple generations of the GA. The generational trends in parameter selection closely align with the identified model architecture in the proposed NAS based approach. CSPResNet and CSPNeXt blocks were frequently explored, with CSPNeXt chosen for the top-performing model. Similarly, GELU was consistently favoured across generations and selcted in the final model, highlighting its importance in achieving high accuracy. While Focal Loss and Quality Focal Loss were commonly selected as class loss functions during exploration, Cross Entropy Loss emerged as the final choice, suggesting it provided a better balance of complexity and performance. As generations progressed, CIoU Loss gained popularity and was incorporated into the final model for box loss, demonstrating its effectiveness in localization tasks. Lastly, although NAdam was often used early on, AdamW became more prominent in later generations and was selected as the optimizer in the identified model, indicating its superior optimization performance.



(a) Class Loss

(b) Box Loss

(c) Activation function

(d) Optimizer

(e) Type of Block

Figure 6.6: The number of times a value for a search space parameter explored across GA generations by the proposed approach for DDDI dataset

6.3 Summary

In this work, we introduce a novel approach for Driver Distraction Detection utilizing an improved GA based NAS, employing the single-stage YOLO object detection model. The proposed approach involves the use of an improved Genetic Algorithm to efficiently explore the search space, aiming to identify the optimal backbone architecture and training parameters. The primary goal is to simultaneously increase the accuracy and reduce the number of parameters of the DNN model. The resulting DNN architecture from this approach showcases superior performance compared to existing one-stage and two-stage models in detecting driver distraction behavior across both DDDI and DDCV datasets, highlighting its effectiveness in addressing the driver distraction detection task.

Chapter 7

Conclusion and Future Scope

This chapter presents the summary of the contributions of this thesis, the conclusion of each objective and the future scope of research for further direction of this thesis is presented.

7.1 Conclusion

This thesis develops techniques for optimizing deep learning models using evolutionary algorithms for vision based applications in vehicle safety and surveillance systems. Through the use of evolutionary optimization techniques, the research achieves enhanced performance in tasks such as vehicle brake light detection, driver distraction detection for accident prevention systems, and vehicle re-identification for smart surveillance systems. This approach facilitates the exploration of various deep learning model architectures and their optimization, resulting in enhanced performance and expanded potential applications across these tasks.

In Chapter 3, the authors introduced a novel dataset for motorcycle brake light detection task. Then, a NAS framework to optimize a two-stage Mask R-CNN object detection model, focusing on finding optimal parameters related to the backbone and training attributes. A Genetic algorithm is used as the search strategy in the NAS search process. The proposed model outperformed the performance of existing one-stage and two-stage object detection models on the proposed two-wheeler dataset and existing four-wheeler dataset.

In Chapter 4, a modified Differential evolution is used as a NAS search strategy to optimize a two-stage object detection network. This chapter expanded the search space compared to the search space proposed in Chapter 3 to enhance the performance of brake light detection for various vehicle types, including both two-wheelers and four-wheelers. The proposed model outperformed the performance of existing one-stage and two-stage object detection models on the proposed two-wheeler dataset and existing 2 four-wheeler datasets.

In Chapter 5, the authors focussed on the optimization of deep learning models using NAS for vehicle re-identification task and designed a search space that included network architecture parameters and training attributes. The Grasshopper optimization algorithm was used as a search strategy, leading to a model that outperformed existing methods on two publicly available datasets for re-identification tasks.

In Chapter 6, the authors proposed a NAS based approach for driver distraction detection, incorporating a search space covering backbone architecture and training parameters. A modified Genetic algorithm was employed as a NAS search strategy, resulting in a model outperforming existing methods for detecting driver distraction behaviors on two publicly available datasets.

7.2 Future scope

In the future, we plan to speed up optimization by exploring parallelism, make NAS approaches handle more computer vision tasks, and improve efficiency by using weight-sharing techniques and incorporating tracking in real-time systems. We also aim to continue enhancing optimization methods and making NAS models work better for different computer vision tasks through ongoing research and development.

Exploring parallelism to speed-up the optimization: Explore techniques for parallelizing training across multiple devices, distributed training, and GPU acceleration to accelerate DNN optimization. This involves training multiple models simultaneously on different machines in each generation of the evolutionary NAS, enabling independent model training within the generation. Assess the impact of parallelism on reducing training time and enhancing optimization efficiency, particularly for large-scale DNN models in vison based tasks.

Expanding NAS approaches for handling different vision based tasks: Expand current NAS approaches to address multiple computer vision tasks, such as object detection, semantic segmentation, and other vision related tasks. Develop new search strategies and objective functions customized for vision based tasks to guide the NAS search process effectively. Evaluate the performance of multi-objective NAS models in terms of accuracy, efficiency, and scalability, comparing them to single-objective NAS models. Continuously improve NAS based optimization methods for vision based tasks, and conduct thorough experiments to assess NAS model capabilities for specific applications.

Improving efficiency through weight-sharing techniques: Study techniques like parameter sharing to lower the computational burden of NAS. Assess how these techniques affect the efficiency and performance of NAS models in vision tasks.

Incorporating tracking to existing systems: Incorporating temporal information alongside spatial information for object detection tasks. Develop algorithms to integrate tracking data into the NAS framework, enhancing the robustness and adaptability of vision based systems. Assess the effectiveness of NAS models with real-world tracking across diverse datasets related to accident prevention and smart traffic surveillance systems.

Integration of explainable AI techniques for DNN optimization: Incorporate explainable AI techniques into the NAS framework to enhance the transparency and trustworthiness of DNN models. This involves exploring methods such as attention mechanisms and visualization tools to provide insights into the decision-making process of NAS optimized models.

Bibliography

[1] World health organization global status report on road safety 2018. https://www.who.int/publications/i/item/9789241565684/. Accessed: 2024-02-12.

[2] Qing Ming and Kang-Hyun Jo. Vehicle detection using tail light segmentation. In *Proceedings of International Forum on Strategic Technology*, volume 2, pages 729–732, 2011.

[3] Vivek Agarwal, N Venkata Murali, and C Chandramouli. A cost-effective ultrasonic sensor-based driver-assistance system for congested traffic conditions. *in: IEEE Transactions on Intelligent Transportation Systems*, 10(3):486–498, 2009.

[4] Aimé Lay-Ekuakille, Patrizia Vergallo, Davide Saracino, and Amerigo Trotta. Optimizing and post processing of a smart beamformer for obstacle retrieval. *in: IEEE Sensors Journal*, 12(5):1294–1299, 2011.

[5] Dario Nava, Giulio Panzani, and Sergio M. Savaresi. A collision warning oriented brake lights detection and classification algorithm based on a mono camera sensor. In *Proceedings of the IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 319–324, 2019.

[6] Duan-Yu Chen and Yang-Jie Peng. Frequency-tuned taillight-based nighttime vehicle braking warning system. *in: IEEE Sensors Journal*, 12(11):3285–3292, 2012.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[8] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, 2016.

[9] Joseph Redmon and Ali Farhadi. YOLO9000: better, faster, stronger. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7263–7271, 2017.

[10] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[11] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 21–37, 2016.

[12] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 580–587, 2014.

[13] Ross Girshick. Fast R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1440–1448, 2015.

[14] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. *in: Advances in Neural Information Processing Systems (NIPS)*, 28:1–9, 2015.

[15] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2961–2969, 2017.

[16] Hao Luo, Youzhi Gu, Xingyu Liao, Shenqi Lai, and Wei Jiang. Bag of tricks and a strong baseline for deep person re-identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1487–1495, 2019.

[17] Ronghua Shang, Songling Zhu, Jinhong Ren, Hangcheng Liu, and Licheng Jiao. Evolutionary neural architecture search based on evaluation correction and functional units. *in: Knowledge-Based Systems*, 251:109206, 2022.

[18] Zhenhong Sun, Ming Lin, Xiuyu Sun, Zhiyu Tan, Hao Li, and Rong Jin. MAE-DET: Revisiting maximum entropy principle in zero-shot NAS for efficient object detection. In *Proceedings of the International Conference on Machine Learning (ICML)*, volume 162, pages 20810–20826, 2022.

[19] Jiahong Wei, Guijie Zhu, Zhun Fan, Jinchao Liu, Yibiao Rong, Jiajie Mo, Wenji Li, and Xinjian Chen. Genetic U-Net: Automatically designed deep networks for retinal vessel segmentation using a genetic algorithm. *in: IEEE Transactions on Medical Imaging*, 41(2):292–307, 2022.

[20] Yuqiao Liu, Yanan Sun, Bing Xue, Mengjie Zhang, Gary G. Yen, and Kay Chen Tan. A survey on evolutionary neural architecture search. *in: IEEE Transactions on Neural Networks and Learning Systems*, 34(2):550–570, 2023.

[21] Gabriel Bender, Hanxiao Liu, Bo Chen, Grace Chu, Shuyang Cheng, Pieter-Jan Kindermans, and Quoc V. Le. Can weight sharing outperform random architecture search? an investigation with tunas. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14311–14320, 2020.

[22] Bichen Wu, Xiaoliang Dai, Peizhao Zhang, Yanghan Wang, Fei Sun, Yiming Wu, Yuandong Tian, Peter Vajda, Yangqing Jia, and Kurt Keutzer. FBNet: Hardware-aware efficient convnet design via differentiable neural architecture search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10734–10742, 2019.

[23] Cheng-Lung Jen, Yen-Lin Chen, and Hao-Yuan Hsiao. Robust detection and tracking of vehicle taillight signals using frequency domain feature based adaboost learning. In *Proceedings of the IEEE International Conference on Consumer Electronics - Taiwan (ICCE-TW)*, pages 423–424, 2017.

[24] J Arunnehru, H Anwar Basha, Ajay Kumar, R Sathya, and M Kalaiselvi Geetha. A vision-based on-road vehicle light detection system using support vector machines. *in: Integrated Intelligent Computing, Communication and Security*, pages 117–126, 2019.

[25] Han-Kai Hsu, Yi-Hsuan Tsai, Xue Mei, Kuan-Hui Lee, Naoki Nagasaka, Danil Prokhorov, and Ming-Hsuan Yang. Learning to tell brake and turn signals in videos using cnn-lstm structure. In *Proceedings of the IEEE International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–6, 2017.

[26] Kuan-Hui Lee, Takaaki Tagawa, Jia-En M. Pan, Adrien Gaidon, and Bertrand Douillard. An attention-based recurrent convolutional network for vehicle taillight recognition. In *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, pages 2365–2370, 2019.

[27] Qiaohong Li, Sahil Garg, Jiangtian Nie, Xiang Li, Ryan Wen Liu, Zhiguang Cao, and M. Shamim Hossain. A highly efficient vehicle taillight detection approach based on deep learning. *in: IEEE Transactions on Intelligent Transportation Systems*, 22(7):4716–4726, 2021.

[28] Rafael A Berri, Alexandre G Silva, Rafael S Parpinelli, Elaine Girardi, and Rangel Arthur. A pattern recognition system for detecting use of mobile phones while driving. In *Proceedings of the International conference on computer Vision theory and Applications (VISAPP)*, volume 2, pages 411–418. IEEE, 2014.

[29] Chao Yan, Frans Coenen, and Bailing Zhang. Driving posture recognition by convolutional neural networks. *in: IET Computer Vision*, 10(2):103–114, 2016.

[30] Adam AQ Mohammed, Xin Geng, Jing Wang, and Zafar Ali. Driver distraction detection using semi-supervised lightweight vision transformer. *in: Engineering Applications of Artificial Intelligence*, 129:107618, 2024.

[31] Ziyang Zhang, Lie Yang, and Chen Lv. Highly discriminative driver distraction detection method based on swin transformer. *in: Vehicles*, 6(1):140–156, 2024.

[32] Xuexi Tang, Yan Chen, Yifan Ma, Wenxuan Yang, Houpan Zhou, and Jingzhou Huang. A lightweight model combining convolutional neural network and transformer for driver distraction recognition. *in: Engineering Applications of Artificial Intelligence*, 132:107910, 2024.

[33] Hang Gao and Yi Liu. Improving real-time driver distraction detection via constrained attention mechanism. *in: Engineering Applications of Artificial Intelligence*, 128:107408, 2024.

[34] Yingzhi Zhang, Taiguo Li, Chao Li, and Xinghong Zhou. A novel driver distraction behavior detection method based on self-supervised learning with masked image modeling. *in: IEEE Internet of Things Journal*, 2023.

[35] Sheng Liu, Linlin You, Rui Zhu, Bing Liu, Rui Liu, Han Yu, and Chau Yuen. Afm3d: An asynchronous federated meta-learning framework for driver distraction detection. *in: IEEE Transactions on Intelligent Transportation Systems*, 2024.

[36] Fengliang Qi, Bo Yan, Leilei Cao, and Hongbin Wang. Stronger baseline for person re-identification. *arXiv preprint arXiv:2112.01059*, 2021.

[37] Hongbin Tu, Chao Liu, Yuanyuan Peng, Haibo Xiong, and Haotian Wang. Clothing-change person re-identification based on fusion of rgb modality and gait features. *in: Signal, Image and Video Processing*, pages 1–10, 2023.

[38] Zhi Xu, Jiawei Yang, Yuxuan Liu, Longyang Zhao, and Jiajia Liu. Staged encoder training for cross-camera person re-identification. *in: Signal, Image and Video Processing*, pages 1–9, 2024.

[39] Zhedong Zheng, Tao Ruan, Yunchao Wei, and Yezhou Yang. Vehiclenet: Learning robust feature representation for vehicle re-identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, volume 2, page 3, 2019.

[40] Yihang Lou, Yan Bai, Jun Liu, Shiqi Wang, and Ling-Yu Duan. Embedding adversarial learning for vehicle re-identification. *in: IEEE Transactions on Image Processing*, 28(8):3794–3807, 2019.

[41] Su V. Huynh, Nam H. Nguyen, Ngoc T. Nguyen, Vinh Tq. Nguyen, Chau Huynh, and Chuong Nguyen. A strong baseline for vehicle re-identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 4142–4149, 2021.

[42] Yuan Yuan, Jian'an Zhang, and Qi Wang. Bike-person re-identification: A benchmark and a comprehensive evaluation. *in: IEEE Access*, 6:56059–56068, 2018.

[43] Augusto Figueiredo, Johnata Brayan, Renan Oliveira Reis, Raphael Prates, and William Robson Schwartz. MoRe: A large-scale motorcycle re-identification dataset. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 4033–4042, 2021.

[44] Jiaze Li and Bin Liu. Rider re-identification based on pyramid attention. In *Proceedings of the Pattern Recognition and Computer Vision (PRCV)*, pages 81–93. Springer, 2022.

[45] Trong-Hieu Nguyen-Mau, Kim-Trang Phu-Thi, Anh-Duy Le-Dinh, Minh-Triet Tran, and Hai-Dang Nguyen. AANet: Motorcycle reid using multi-atrous convolution and self-attention mechanisms. In *Proceedings of the International Conference on Multimedia Analysis and Pattern Recognition (MAPR)*, pages 1–6, 2023.

[46] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[47] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5987–5995, 2017.

[48] Hang Zhang, Chongruo Wu, Zhongyue Zhang, Yi Zhu, Haibin Lin, Zhi Zhang, Yue Sun, Tong He, Jonas Mueller, R. Manmatha, Mu Li, and Alexander Smola. ResNeSt: Split-attention networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2735–2745, 2022.

[49] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10012–10022, 2021.

[50] Caltechgraz dataset. https://www.gti.ssr.upm.es/~jal/Database/CaltechGraz.rar. Accessed: 2024-02-12.

[51] Caltech database. http://www.vision.caltech.edu/datasets/. Accessed: 2024-02-12.

[52] Han-Kai Hsu, Yi-Hsuan Tsai, Xue Mei, Kuan-Hui Lee, Naoki Nagasaka, Danil Prokhorov, and Ming-Hsuan Yang. Learning to tell brake and turn signals in videos using cnn-lstm structure. In *Proceedings of the IEEE International Conference on Intelligent Transportation Systems*, pages 1–6, 2017.

[53] Mingxing Tan and Quoc Le. EfficientNetV2: Smaller models and faster training. In *Proceedings of the International Conference on Machine Learning (ICML)*, volume 139, pages 10096–10106, 2021.

[54] Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10425–10433, 2020.

[55] Piotr Dollár, Mannat Singh, and Ross Girshick. Fast and accurate model scaling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 924–932, 2021.

[56] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269, 2017.

[57] Shahrzad Saremi, Seyedali Mirjalili, and Andrew Lewis. Grasshopper optimisation algorithm: Theory and application. *in: Advances in Engineering Software*, 105:30–47, 2017.

[58] Nassrallah Faris Abdukader Al Shalchi and Javad Rahebi. Human retinal optic disc detection with grasshopper optimization algorithm. *in: Multimedia Tools and Applications*, 81(17):24937–24955, 2022.

[59] Ashish Kumar Bhandari and Kusuma Rahul. A novel local contrast fusion-based fuzzy model for color image multilevel thresholding using grasshopper optimization. *in: Applied Soft Computing*, 81:105515, 2019.

[60] Phu-Hung Dinh. A novel approach based on grasshopper optimization algorithm for medical image fusion. *in: Expert Systems with Applications*, 171:114576, 2021.

[61] Haouassi Hichem, Merah Elkamel, Mehdaoui Rafik, Maarouk Toufik Mesaaoud, and Chouhal Ouahiba. A new binary grasshopper optimization algorithm for feature selection problem. *in: Journal of King Saud University - Computer and Information Sciences*, 34(2):316–328, 2022.

[62] Glenn Jocher, Ayush Chaurasia, Alex Stoken, Jirka Borovec, NanoCode012, Yonghye Kwon, TaoXie, Kalen Michael, Jiacong Fang, imyhxy, Lorna, Colin Wong, Zeng Yifu, Abhiram V, Diego Montes, Zhiqiang Wang, Cristi Fati, Jebastin Nadar, Laughing, UnglvKitDe, tkianai, yxNONG, Piotr Skalski, Adam Hogan, Max Strobel, Mrinal Jain, Lorenzo Mammana, and xylieong. ultralytics/yolov5: v6.2 - YOLOv5 Classification Models, Apple M1, Reproducibility, ClearML and Deci.ai integrations, August 2022.

[63] Chuyi Li, Lulu Li, Hongliang Jiang, Kaiheng Weng, Yifei Geng, Liang Li, Zaidan Ke, Qingyuan Li, Meng Cheng, Weiqiang Nie, et al. YOLOv6: A single-stage object detection framework for industrial applications. *arXiv preprint arXiv:2209.02976*, 2022.

[64] Chengqi Lyu, Wenwei Zhang, Haian Huang, Yue Zhou, Yudong Wang, Yanyi Liu, Shilong Zhang, and Kai Chen. Rtmdet: An empirical study of designing real-time object detectors. *arXiv preprint arXiv:2212.07784*, 2022.

[65] Shangliang Xu, Xinxin Wang, Wenyu Lv, Qinyao Chang, Cheng Cui, Kaipeng Deng, Guanzhong Wang, Qingqing Dang, Shengyun Wei, Yuning Du, and Baohua Lai. PP-YOLOE: An evolved version of YOLO. *ArXiv preprint*, abs/2203.16250, 2022.

[66] new-workspace vrhvx. Distracted driver detection dataset. https://universe.roboflow.com/new-workspace-vrhvx/distracted-driver-detection, dec 2021. Accessed: 2024-02-12.

[67] Ipylot project. Distracted driving dataset. https://universe.roboflow.com/ipylot-project/distracted-driving-v2wk5, jul 2022. Accessed: 2024-02-12.

[68] Hua-Tsung Chen, Yi-Chien Wu, and Chun-Chieh Hsu. Daytime preceding vehicle brake light detection using monocular vision. *in: IEEE Sensors Journal*, 16(1):120–131, 2016.

[69] Akhan Almagambetov, Mauricio Casares, and Senem Velipasalar. Autonomous tracking of vehicle rear lights and detection of brakes and turn signals. In *Proceedings of the IEEE Symposium on Computational Intelligence for Security and Defence Applications*, pages 1–7, 2012.

[70] Zhiyong Cui, Shao-Wen Yang, and Hsin-Mu Tsai. A vision-based hierarchical framework for autonomous front-vehicle taillights detection and signal recognition. In *Proceedings of the IEEE International Conference on Intelligent Transportation Systems*, pages 931–937, 2015.

[71] Tobias Weis, Martin Mundt, Patrick Harding, and Visvanathan Ramesh. Anomaly detection for automotive visual signal transition estimation. In *Proceedings of the IEEE International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–8, 2017.

[72] Guangyu Zhong, Yi-Hsuan Tsai, Yi-Ting Chen, Xue Mei, Danil Prokhorov, Michael James, and Ming-Hsuan Yang. Learning to tell brake lights with convolutional features. In *Proceedings of the IEEE International Conference on Intelligent Transportation Systems (ITSC)*, pages 1558–1563, 2016.

[73] Flaviu Ionut Vancea, Arthur Daniel Costea, and Sergiu Nedevschi. Vehicle taillight detection and tracking using deep learning and thresholding for candidate generation. In *Proceedings of the IEEE International Conference on Intelligent Computer Communication and Processing*, pages 267–272, 2017.

[74] Jian-Gang Wang, Lubing Zhou, Zhiwei Song, and Miaolong Yuan. Real-time vehicle signal lights recognition with HDR camera. In *Proceedings of the IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pages 355–358, 2016.

[75] Flaviu Ionut Vancea and Sergiu Nedevschi. Semantic information based vehicle relative orientation and taillight detection. In *Proceedings of the IEEE International Conference on Intelligent Computer Communication and Processing (ICCP)*, pages 259–264, 2018.

[76] Christopher J Rapson, Boon-Chong Seet, M Asif Naeem, Jeong Eun Lee, and Reinhard Klette. A performance comparison of deep learning methods for real-time localisation of vehicle lights in video frames. In *Proceedings of the IEEE Intelligent Transportation Systems Conference*, pages 567–572, 2019.

[77] Davi Frossard, Eric Kee, and Raquel Urtasun. Deepsignals: Predicting intent of drivers through visual signals. In *Proceedings of International Conference on Robotics and Automation*, pages 9697–9703, 2019.

[78] Dario Nava, Giulio Panzani, and Sergio M Savaresi. A collision warning oriented brake lights detection and classification algorithm based on a mono camera sensor. In *Proceedings of IEEE Intelligent Transportation Systems Conference*, pages 319–324, 2019.

[79] Qiaohong Li, Sahil Garg, Jiangtian Nie, Xiang Li, Ryan Wen Liu, Zhiguang Cao, and M Shamim Hossain. A highly efficient vehicle taillight detection approach based on deep learning. *in: IEEE Transactions on Intelligent Transportation Systems*, 22(7):4716–4726, 2020.

[80] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*, 2017.

[81] W. Chen, X. Chen, J. Zhang, and K. Huang. Beyond Triplet Loss: A deep quadruplet network for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1320–1329, 2017.

[82] Qiqi Xiao, Hao Luo, and Chi Zhang. Margin sample mining loss: A deep learning based method for person re-identification. *arXiv preprint arXiv:1710.00478*, 2017.

[83] Lin Wu, Chunhua Shen, and Anton van den Hengel. Personnet: Person re-identification with deep convolutional neural networks. *arXiv preprint arXiv:1601.07255*, 2016.

[84] Haiyu Zhao, Maoqing Tian, Shuyang Sun, Jing Shao, Junjie Yan, Shuai Yi, Xiaogang Wang, and Xiaoou Tang. Spindle Net: Person re-identification with human body region guided feature decomposition and fusion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 907–915, 2017.

[85] Jiacheng Pu and Wei Zou. Person re-identification based on multi-scale feature fusion and multi-attention mechanism. *in: Signal, Image and Video Processing*, 18(1):243–253, 2024.

[86] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, volume 33, pages 18661–18673. Curran Associates, Inc., 2020.

[87] Xu Chen, Haigang Sui, Jian Fang, Wenqing Feng, and Mingting Zhou. Vehicle re-identification using distance-based global and partial multi-regional feature learning. *in: IEEE Transactions on Intelligent Transportation Systems*, 22(2):1276–1286, 2021.

[88] Pirazh Khorramshahi, Amit Kumar, Neehar Peri, Sai Saketh Rambhatla, Jun-Cheng Chen, and Rama Chellappa. A dual-path model with adaptive attention for vehicle re-identification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6131–6140, 2019.

[89] Yiting Cheng, Chuanfa Zhang, Kangzheng Gu, Lizhe Qi, Zhongxue Gan, and Wenqiang Zhang. Multi-scale deep feature fusion for vehicle re-identification. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1928–1932, 2020.

[90] Huibing Wang, Jinjia Peng, Dongyan Chen, Guangqi Jiang, Tongtong Zhao, and Xianping Fu. Attribute-guided feature learning network for vehicle reidentification. *in: IEEE MultiMedia*, 27(4):112–121, 2020.

[91] Zhedong Zheng, Minyue Jiang, Zhigang Wang, Jian Wang, Zechen Bai, Xuanmeng Zhang, Xin Yu, Xiao Tan, Yi Yang, Shilei Wen, and Errui Ding. Going beyond real data: A robust visual representation for vehicle re-identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2550–2558, 2020.

[92] Shuting He, Hao Luo, Pichao Wang, Fan Wang, Hao Li, and Wei Jiang. TransReID: Transformer-based object re-identification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14993–15002, 2021.

[93] Tianyu Zhang, Longhui Wei, Lingxi Xie, Zijie Zhuang, Yongfei Zhang, Bo Li, and Qi Tian. Spatiotemporal transformer for video-based person re-identification. *arXiv preprint arXiv:2103.16469*, 2021.

[94] De Cheng, Yihong Gong, Sanping Zhou, Jinjun Wang, and Nanning Zheng. Person re-identification by multi-channel parts-based cnn with improved triplet loss function. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1335–1344, 2016.

[95] Ruijie Quan, Xuanyi Dong, Yu Wu, Linchao Zhu, and Yi Yang. Auto-ReID: Searching for a part-aware convnet for person re-identification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3749–3758, 2019.

[96] Qinqin Zhou, Bineng Zhong, Xin Liu, and Rongrong Ji. Attention-based neural architecture search for person re-identification. *in: IEEE Transactions on Neural Networks and Learning Systems*, 33(11):6627–6639, 2022.

[97] Chaoyou Fu, Yibo Hu, Xiang Wu, Hailin Shi, Tao Mei, and Ran He. Cm-nas: Cross-modality neural architecture search for visible-infrared person re-identification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11803–11812, 2021.

[98] Shengbo Chen, Kai Jiang, Xianrui Liu, Kangkang Yang, and Zhou Lei. TGAS-ReID: Efficient architecture search for person re-identification via greedy decisions with topological order. *in: Applied Intelligence*, 53(7):7343–7354, 2023.

[99] Arian Shajari, Houshyar Asadi, Sebastien Glaser, Adetokunbo Arogbonlo, Shady Mohamed, Lars Kooijman, Ahmad Abu Alqumsan, and Saeid Nahavandi. Detection of driving distractions and their impacts. *in: Journal of advanced transportation*, 2023(1):2118553, 2023.

[100] Hong Vin Koay, Joon Huang Chuah, Chee-Onn Chow, and Yang-Lang Chang. Detecting and recognizing driver distraction through various data modality using machine learning: A review, recent advances, simplified framework and open challenges (2014–2021). *in: Engineering Applications of Artificial Intelligence*, 115:105309, 2022.

[101] Eva Michelaraki, Christos Katrakazas, Susanne Kaiser, Tom Brijs, and George Yannis. Real-time monitoring of driver distraction: State-of-the-art and future insights. *in: Accident Analysis & Prevention*, 192:107241, 2023.

[102] Xuetao Zhang, Nanning Zheng, Fei Wang, and Yongjian He. Visual recognition of driver hand-held cell phone use based on hidden crf. In *Proceedings of the IEEE international conference on vehicular electronics and safety*, pages 248–251. IEEE, 2011.

[103] Chihang H Zhao, Bailing L Zhang, Xiaozheng Z Zhang, Sanqiang Q Zhao, and Hanxi X Li. Recognition of driving postures by combined features and random subspace ensemble of multilayer perceptron classifiers. *in: Neural Computing and Applications*, 22:175–184, 2013.

[104] Yusuf Artan, Orhan Bulan, Robert P Loce, and Peter Paul. Driver cell phone usage detection from hov/hot nir images. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 225–230, 2014.

[105] Céline Craye and Fakhri Karray. Driver distraction detection and recognition using rgb-d sensor. *arXiv preprint arXiv:1502.00250*, 2015.

[106] Bhakti Baheti, Suhas Gajre, and Sanjay Talbar. Detection of distracted driver using convolutional neural network. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1032–1038, 2018.

[107] Hesham M Eraqi, Yehya Abouelnaga, Mohamed H Saad, Mohamed N Moustafa, et al. Driver distraction identification with an ensemble of convolutional neural networks. *in: Journal of Advanced Transportation*, 2019, 2019.

[108] Belmekki Ghizlene, Mekkakia Zoulikha, and Hector Pomares. An efficient framework to detect and avoid driver sleepiness based on yolo with haar cascades and an intelligent agent. In *Proceedings of the International Work-Conference on Artificial Neural Networks*, pages 699–708. Springer, 2019.

[109] Mingqi Lu, Yaocong Hu, and Xiaobo Lu. Driver action recognition using deformable and dilated faster R-CNN with optimized region proposals. *in: Applied Intelligence*, 50:1100–1111, 2020.

[110] Sarfaraz Masood, Abhinav Rai, Aakash Aggarwal, Mohammad Najam Doja, and Musheer Ahmad. Detecting distraction of drivers using convolutional neural network. *in: Pattern Recognition Letters*, 139:79–85, 2020.

[111] Monagi H Alkinani, Wazir Zada Khan, and Quratulain Arshad. Detecting human driver inattentive and aggressive driving behavior using deep learning: Recent advances, requirements and open challenges. *in: IEEE Access*, 8:105008–105030, 2020.

[112] Li Li, Boxuan Zhong, Clayton Hutmacher Jr, Yulan Liang, William J Horrey, and Xu Xu. Detection of driver manual distraction via image-based hand and ear recognition. *in: Accident Analysis & Prevention*, 137:105432, 2020.

[113] Mohammad Shahverdy, Mahmood Fathy, Reza Berangi, and Mohammad Sabokrou. Driver behavior detection and classification using deep convolutional neural networks. *in: Expert Systems with Applications*, 149:113240, 2020.

[114] Japesh Methuku. In-car driver response classification using deep learning (cnn) based computer vision. *in: IEEE Trans. Intell. Veh*, 2020.

[115] Guofa Li, Weiquan Yan, Shen Li, Xingda Qu, Wenbo Chu, and Dongpu Cao. A temporal–spatial deep learning approach for driver distraction detection based on eeg signals. *in: IEEE Transactions on Automation Science and Engineering*, 19(4):2665–2677, 2021.

[116] Lei Zhao, Fei Yang, Lingguo Bu, Su Han, Guoxin Zhang, and Ying Luo. Driver behavior detection via adaptive spatial attention mechanism. *in: Advanced Engineering Informatics*, 48:101280, 2021.

[117] Yuxin Zhang, Yiqiang Chen, and Chenlong Gao. Deep unsupervised multi-modal fusion network for detecting driver distraction. *in: Neurocomputing*, 421:26–38, 2021.

[118] Long Qin, Yi Shi, Yahui He, Junrui Zhang, Xianshi Zhang, Yongjie Li, Tao Deng, and Hongmei Yan. ID-YOLO: Real-time salient object detection based on the driver's fixation region. *in: IEEE Transactions on Intelligent Transportation Systems*, 23(9):15898–15908, 2022.

[119] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[120] Zizheng Guo, Qing Liu, Lin Zhang, Zhenning Li, and Guofa Li. L-TLA: A lightweight driver distraction detection method based on three-level attention mechanisms. *in: IEEE Transactions on Reliability*, 2024.

[121] Yunsheng Ma and Ziran Wang. Vit-dd: Multi-task vision transformer for semi-supervised driver distraction detection. In *in: IEEE Intelligent Vehicles Symposium (IV)*, pages 417–423, 2024.

[122] Jianyuan Guo, Kai Han, Yunhe Wang, Chao Zhang, Zhaohui Yang, Han Wu, Xinghao Chen, and Chang Xu. Hit-detector: Hierarchical trinity architecture search for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11405–11414, 2020.

[123] Yukang Chen, Tong Yang, Xiangyu Zhang, Gaofeng Meng, Xinyu Xiao, and Jian Sun. Detnas: Backbone search for object detection. *in: Advances in Neural Information Processing Systems (NIPS)*, 32:1–11, 2019.

[124] Jiahong Wei, Guijie Zhu, Zhun Fan, Jinchao Liu, Yibiao Rong, Jiajie Mo, Wenji Li, and Xinjian Chen. Genetic U-Net: automatically designed deep networks for retinal vessel segmentation using a genetic algorithm. *in: IEEE Transactions on Medical Imaging*, 41(2):292–307, 2021.

[125] Chilukamari Rajesh and Sushil Kumar. An evolutionary block based network for medical image denoising using differential evolution. *in: Applied Soft Computing*, 121:108776, 2022.

[126] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2010.

[127] Diganta Misra. Mish: A self regularized non-monotonic activation function. *arXiv preprint arXiv:1908.08681*, 2019.

[128] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017.

[129] Jon Arróspide, Luis Salgado, and Marcos Nieto. Video analysis-based vehicle detection and tracking using an mcmc sampling framework. *in: EURASIP Journal on Advances in Signal Processing*, pages 1–20, 2012.

[130] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Doll'a r, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014.

[131] Chengjian Feng, Yujie Zhong, Yu Gao, Matthew R Scott, and Weilin Huang. Tood: Task-aligned one-stage object detection. In *Proceedings of IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3490–3499, 2021.

[132] Peize Sun, Rufeng Zhang, Yi Jiang, Tao Kong, Chenfeng Xu, Wei Zhan, Masayoshi Tomizuka, Lei Li, Zehuan Yuan, Changhu Wang, et al. Sparse R-CNN: End-to-end object detection with learnable proposals. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14454–14463, 2021.

[133] TU Graz-02 database. http://www.emt.tugraz.at/~pinz/data/GRAZ_02/. Accessed: 2024-02-12.

[134] Rainer Storn and Kenneth V. Price. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *in: Journal of Global Optimization*, 11:341–359, 1997.

[135] Seng Poh Lim and Habibollah Haron. Performance comparison of genetic algorithm, differential evolution and particle swarm optimization towards benchmark functions. In *Proceedings of the IEEE Conference on Open Systems (ICOS)*, pages 41–46, 2013.

[136] Swagatam Das and Ponnuthurai Nagaratnam Suganthan. Differential evolution: A survey of the state-of-the-art. *in: IEEE Transactions on Evolutionary Computation*, 15(1):4–31, 2011.

[137] Ji-Xiang Du, De-Shuang Huang, Xiao-Feng Wang, and Xiao Gu. Shape recognition based on neural networks trained by differential evolution algorithm. *in: Neurocomputing*, 70(4):896–903, 2007. Advanced Neurocomputing Theory and Methodology.

[138] Wei-Der Chang. Two-dimensional fractional-order digital differentiator design by using differential evolution algorithm. *in: Digital Signal Processing*, 19(4):660–667, 2009.

[139] Swagatam Das and Amit Konar. Automatic image pixel clustering with an improved differential evolution. *in: Applied Soft Computing*, 9(1):226–236, 2009.

[140] Wu Deng, Shifan Shang, Xing Cai, Huimin Zhao, Yingjie Song, and Junjie Xu. An improved differential evolution algorithm and its application in optimization problem. *in: Soft Computing*, 25(7):5277–5298, 2021.

[141] Q. Chen, Y. Wang, T. Yang, X. Zhang, J. Cheng, and J. Sun. You only look one-level feature. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13034–13043, 2021.

[142] Jesse Pirhonen, Risto Ojala, Klaus Kivekäs, and Kari Tammi. Predictive braking with brake light detection—field test. *in: IEEE Access*, 10:49771–49780, 2022.

[143] Jesse Pirhonen, Risto Ojala, Klaus Kivekäs, Jari Vepsäläinen, and Kari Tammi. Brake light detection algorithm for predictive braking. *in: Applied Sciences*, 12(6), 2022.

[144] Henri Rebecq, René Ranftl, Vladlen Koltun, and Davide Scaramuzza. High speed and high dynamic range video with an event camera. *in: IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(6):1964–1980, 2021.

[145] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. Yolox: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430*, 2021.

[146] Chilukamari Rajesh, Ravichandra Sadam, and Sushil Kumar. An evolutionary u-shaped network for retinal vessel segmentation using binary teaching–learning-based optimization. *in: Biomedical Signal Processing and Control*, 83:104669, 2023.

[147] Anna Montoya, Dan Holman, SF data science, Taylor Smith, and Wendy Kan. State farm distracted driver detection, 2016.

[148] Glenn Jocher, Ayush Chaurasia, Alex Stoken, Jirka Borovec, NanoCode012, Yonghye Kwon, TaoXie, Kalen Michael, Jiacong Fang, imyhxy, Lorna, Colin Wong, Zeng Yifu, Abhiram V, Diego Montes, Zhiqiang Wang, Cristi Fati, Jebastin Nadar, Laughing, UnglvKitDe, tkianai, yxNONG, Piotr Skalski, Adam Hogan, Max Strobel, Mrinal Jain, Lorenzo Mammana, and xylieong. ultralytics/yolov5: v6.2 - YOLOv5 Classification Models, Apple M1, Reproducibility, ClearML and Deci.ai integrations, August 2022.

[149] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv preprint arXiv:2207.02696*, 2022.

List of Publications / Preprints

Publications / Preprints Contributing to this Thesis:

1. **Medipelly Rampavan**, Earnest Paul Ijjina, “Genetic brake-net: Deep learning based brake light detection for collision avoidance using genetic algorithm.” in: *Knowledge-Based Systems*, Elsevier, vol. 264, March 2023, pp.110338,
   doi: https://doi.org/10.1016/j.knosys.2023.110338 (**SCI, Q1, Published**)

2. **Medipelly Rampavan**, Earnest Paul Ijjina, “Brake Light Detection of Vehicles Using Differential Evolution Based Neural Architecture Search” in: *Applied Soft Computing*, Elsevier, 147, November 2023, pp.110839,
   doi: https://doi.org/10.1016/j.asoc.2023.110839 (**SCIE, Q1, Published**)

3. **Medipelly Rampavan**, Earnest Paul Ijjina, “MNAS-reID: Grasshopper Optimization based Neural Architecture Search for Motorcycle Re-identification” in: *Signal, Image and Video Processing*, Springer (**SCIE, Q2, Accepted on 9th Sep 2024**)

4. **Medipelly Rampavan**, Earnest Paul Ijjina, “An Improved Genetic Algorithm based Deep Learning Model with YOLO Framework for Driver Distraction Detection” in: *Engineering Applications of Artificial Intelligence*, Elsevier (**SCIE, Q1, First Revision Submitted on 15th Sep 2024**)

Other Publications:

5. Earnest Paul Ijjina, **Medipelly Rampavan**, Beerukuri Santosh Kumar, Sowmya Vinnakota, Vijay Chowdary Nelakurthi, “Person re-Identification using Vision Transformer and Centroid Triplet Loss” in: *Multimedia Tools and Applications*, Springer, vol. 83, August 2024, pp. 73777–73788,
   doi: https://doi.org/10.1007/s11042-024-19911-4 (**SCIE, Q1, Published**)

6. MKumar, Gobind, **Medipelly Rampavan**, and Earnest Paul Ijjina. “Deep Learning based Brake Light Detection for Two Wheelers” in Proceedings of IEEE International Conference on Computing Communication and Networking Technologies (ICCCNT), Nov 2021, pp. 1-4.
   doi: 10.1109/ICCCNT51525.2021.9579918 (**Published**)

7. **Medipelly Rampavan**, Earnest Paul Ijjina “A Fast Garbage Classification Model Based on Deep Learning” in: *IoT-Based Smart Waste Management for Environmental Sustainability*, CRC Press, 2022, pp. 171-182 (**Published**)