CUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCE

MSc THESIS

Çağrı DÜKÜNLÜ

SMART TRAILER SYSTEM WITH DRIVER BEHAVIOUR RECOGNITION AND 360-DEGREE SURROUND VIEW CAMERA

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

ADANA-2022

ABSTRACT

PhD THESIS

A SENTENCE BOUNDARY DETECTION SYSTEM FOR TURKISH TEXTUAL DOCUMENTS

Çağrı DÜKÜNLÜ

ÇUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES
DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

Supervisor : Assoc. Prof. Dr. M. Uğraş Cuma
Year: 2022, Pages: 66
Jury : Assoc. Prof. Dr. M. Uğraş Cuma
: Assoc. Prof. Dr. Murat Mustafa SAVRUN
: Asst. Prof. Dr. Üyesi Oğuzhan TİMUR

Driver safety has always been an essential subject for humanity, and technology has improved continuously in the twenty century. New technologies for driver safety have started to develop in-car technologies. Even though the technologies are increasing and significant motor-vehicle fatalities are decreasing on motorways, driver safety still plays a big part in the automotive industry.

This study intends to provide a new dataset and an effective method for driver behavior recognition based on driver behavior with a machine learning model named YOLOV5. Furthermore, a 360-degree blank spot surround-view camera system was also integrated into the study to increase driver safety. Due to advanced technology giving a chance to imply a better plan, using these systems may prevent traffic accidents much more rather than not using them at all.

The image recognition model used in this study is YOLOV5 (You Only Look Once Version 5). This model is implemented the driver image data provided from KOE (Koluman Otomotiv Endüstri). Different parameters are used to determine the results. The selected metrics are mean averaged precision, precision, recall, classification, and essential other metrics. With the evaluation of the object behavior model's accuracies, this paper aims to determine the effectiveness of driver behavior recognition using the YOLOV5 model.

Image processing methods used in this study are from OpenCV libraries. This library function is implemented to the taken image from fisheye lens cameras, and calibration methods are used for better results.

Key words: Machine Learning, Image Processing, Driver Behavior Recognition, Surrond View, Artificial Intelligence

ÖZ

YÜKSEK LİSANS TEZİ

TÜRKÇE METİN BELGELERİ İÇİN CÜMLE SINIRI TESPİT SİSTEMİ

Çağrı DÜKÜNLÜ

ÇUKUROVA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

Danışman : Doç. Dr. M. Uğraş Cuma
Yıl: 2022, Pages: 66
Jüri : Doç. Dr. M. Uğraş Cuma
: Doç. Dr. Murat Mustafa SAVRUN
: Dr. Öğr. Üyesi Oğuzhan TİMUR

Modern dünyada insalık için sürücü güvenliği her zaman önemli bir konu olmuştur ve teknoloji yirminci yüzyılda sürekli olarak gelişme göstermiştir. Sürücü güvenliği için yeni teknolojiler, araç içi teknolojiler modern teknolojinin gelişmesi ile çok yol kat etmiştir. Otoyollarda artan kaza ve motorlu araç ölümlerinin artma ise beraber , otomotiv endüstri Gelişmiş sürücü asistanı destek modüllerini geliştirmeye başlamış ve kaza oranlarının yüksek olması geliştirme aşamasında verilen öncelikte büyük rol oynamıştır.

Bu çalışma, YOLOV5 adlı makine öğrenme modeli ile sürücü davranıına dayalı hataları tespit etmek için geliştirilmiş entegre bir ADAS sistemini oluşturmaktadır. Ayrıca sürücü güvenliği arttırmak ve araç dışı kör noktaların görüşünün sağlanarak, sürücü çevre farkındalığını arttırmak amacı ile ADAS'a entegre 360 derecelik bir çevre görüş kamera sistemini de modele entegre etmiştir. Gelişmiş ADAS sisteminin aktif olarak kullanılması ile trafik kazalarını daha olmadan önce tespit edilmesini sağlamak ve ölüm oranlarını azaltmayı hedefleyen bir çalışmadır.

Bu çalışmada kullanılan görüntü tanıma modeli YOLOV5 olmakla beraber görüntü tanıma modelleri arasında en hızlı çalışan model olarak bilinir. Model eğitimi için kullanılacak olan dataseti KOE'den(Koluman Otomotiv Endüstri) sağlanan sürücü imajı verilerini kullanmaktadır. Sonuçları belirlemek için birden fazla farklı parameter kullanır. Seçilen metrikler, ortalama kesinlik, kesinlik, geri çağırma, sınıflandırma ve diğer temel metriklerden olumaktadır.

Bu çalışmada kullanılan görüntü işleme yöntemleri OpenCV kütüphanelerinin fonksiyonlarını barındırmaktadır. Bu kütüphane fonksiyonları yardımı ile balıkgözlü kamera lensleri kalibre edilir. Kuş bakışı görüntü oluşturulur. Maskeleme işlemi uygulanır ve fotoğraf birleştirilerek, pixel manipülasyonuna tabi tutulur. Sonuç olarak 360 derece entegre bir çevre görüş sistemi elde edilir.

Anahtar Kelimeler: Makine Öğrenmesi, Görüntü İşleme,Sürücü Davranış Tanımlama, Çevre Görüntüsü, Yapay Zeka

EXTENDED ABSTRACT

The development of the automotive industry is mainly a driver-oriented issue. Thus, driver safety is critical and several studies have been performed continuously. In this study, advanced driver assistance system (ADAS) algorithms including 360-degree surround view and driver behavior recognition have been developed for a trailer in order to improve safety. To detect the driver's inattention/misbehavior, the You Only Look OnceV5 (YOLO_V5) method has been adopted to ADAS because of its fast response provided by taking region proposal inside an image multiple class probabilities and processing instantaneously. In addition, the developed ADAS has been equipped with an image processing-based surround view system using OpenCV. To evaluate the performance of the proposed algorithms for the trailer system, experimental studies have been conducted. The driver's behavior recognition has been tested using different images captured from the camera located inside the vehicle. Also, the surround view performance of the proposed system has been verified with different driving scenarios such as routine driving, parking, and parking near-to-line. The results prove the viability and effectiveness of the proposed system.

The main motivation is to reduce the rate of accidents caused by driver inattention/misbehavior and blind spots by equipping vehicles with smart features. Thus, the recognition of driver activity and detection of blind spots have become key issues for safer vehicles. As a matter of fact, the intelligent vehicle industry is expected to improve gradually in the following decades. The outside mirrors used in vehicles to change lanes have visibility restrictions. The blind-spot detection studies have been carried out to prevent these limitations from causing fatal traffic accidents as a result of reckless lane departures. The related system is one of the critical functionality of conventionally named advanced driver assistance systems (ADAS). Several studies have been carried out regarding the ADAS and the authors stated that the automotive industry market intention is increasing for it as it increases control of

vehicles for the driver by providing advanced visualization in real-time. Studies within the scope of ADAS involve lane departure monitoring, adaptive cruise control, speed limit monitoring, forward collision warning, emergency brake assistance, and surround-view monitoring, which have been accelerated in literature. Studies highlight that the lane departure warning and forward collision warning systems provide 9.3-33.3% and 23-50% crash prevention rates respectively. One of the real-time systems that are adapted to the ADAS is Bird's-eye Surround View (SV) which integrates several cameras to produce a virtual bird's-eye view in real-

SV is a sensor fusion problem in which many sensors, in this case, cameras are merged and processed together to create a complete image of the location surrounding a vehicle and to assist a machine or human driver in making quick choices. The other problem is defined as image stitching. Several studies including pairing the corresponding detected key points and interpolation of image elements captured by vehicle mirror cameras and stereo pairs of cameras at the rear of the vehicle, have been performed in order to solve the stitching problem. The other way to enhance road safety for a vehicle driver is to adopt driver behavior recognition systems. Intelligent (ie. highly automated) vehicles are classified into categories considering the autonomy ranges. Some vehicles have no driving automation, while some vehicles allow drivers for taking over vehicle control under emergency to take the control of vehicle considering the driver; behavior/activity. Several studies have been performed regarding real-time driver behavior/activity in the literature. While previous studies have focused on drivers' attention/distraction, driver intention, driver styles, and driver fatigue detection, the studies based on artificial intelligence have been accelerated.

This study contains two different solutions for ADAS systems, a new fast driver behavior recognition machine learning model and an original manual surround view method. The main superior aspect of the proposed system is the recognition of the driver's behavior fast in comparison with existing methods owing to the fast prediction capability of YOLO_V5. The improved ADAS which is equipped with

methods with the aforementioned advantages has been experimentally tested with a trailer truck.

The arrangement of the current work is described in the following manner: The need for equipment and materials is presented in Section 4. The proposed methods and experimental setup are described in Section 5. The details of the results of the method are presented in Section 6. Finally, the references are presented in Section 7.

ACKNOWLEDGEMENT

I would like to thank the following people, without Assoc. Prof. Mehmet Uğraş Cuma and I would not have been able to complete this research, and without my parents support I would not have made it through my masters degree!

The Electricity team at Koluman Automotive Industry, especially my colleague Emre Aksakal, whose insight and knowledge into subject matter steered me through this research. And special thanks to Metehan Kuşdemir whose support make my studies to go the extra mile.

Special thanks to the white collars from Koluman Automotive Industry which took their time to generate an original dataset for this study and allowed me to use this dataset for my thesis without them I would have no content for my thesis.

Lastly again my biggest thanks to my family for all the support you have shown me through this research the culmination of four years of distance learning. For my mother thanks for all your support without which I would have stopped these studies a long time ago. You have been amazing, and I will now be able to more experienced engineer than ever.

TABLE OF CONTENTS **PAGE**

LIST OF TABLES **PAGE**

LIST OF FIGURES **PAGE**

ABBREVIATIONS

ADAS : Advanced Driver Assistance Systems
AV : Autonomous Vehicle
AVI : Audio Video Interleave
CCTV : Video Surveillance
CDS : Central Depository System
CNN : Convolutional Neural Network
Concat : Concatenate Function
ConvNet : Convolutional Neural Network
CSI : Channel State Information
Descriptor : Description
DVR : Digital Video Recorder
EXIF : Exchangeable Image F
GBP : British Pund Futures
GPU : Graphical Process Unit
HMI : Human User Interface
IEEE : Institute of Electrical and Electronics Engineers
InterCNN : Interwowen Deep Convolutional Neural Network
IR : Infrared
KNN : K-Nearest Neighbour
KOE : Koluman Otomotiv Endüstri
mAP : Mean Average Precision
MIL-STD : Military Standard
MLT : Military
NATO : North Atlantic Treaty Organization
NN : Neural Network
NTSC : National Television Standards Committee
OpenCV : Open Source Computer Vision Library

PAL : Phase Alternate Line
PNG : Portable Network Graphics
R-CNN : Region Based Convolutional Neural Network
ResNet :Residual Network
SGD : Stochastic Gradient Descent
SV : Surround View
SVM : Support Vector Machine
TF : Tensor Flow
TSK : Türk Silahlı Kuvvetleri
YOLO : You Only Look Once

1. INTRODUCTION

This chapter gives information about simple machine learning types. It provides a theoretical background with respect to the research goals outlined before. The pre-given reports will help to understand the study because it is related. Our analysis starts with a machine learning term explanation that aims to provide our work's broader context. We mainly focus on this subject when dealing with the driver behavior recognition model. Having obtained a theoretical path towards our research objectives about machine learning, we then explain the algorithms of machine learning algorithms, each one separately. After machine learning algorithms are described, we forward to the subject of Image Processing which we especially focus on this subject when we are dealing with surround view system. Having obtained a theoretical path towards of surround view, finally we explain a brief information about smart vehicle techonologies. Our study generally based on this subject having a machine learning model to decrease the accidents via an alarm system and surround view system to increase awareness to decrease incidents is basically a smart vehicle Technologies and in this study we explain how can such a researcher can achive this via given study.

1.1. Machine Learning

An application of artificial intelligence is machine learning that gives computers the capacity to automatically pick up knowledge and information and get better with practice without being explicitly programmed. Machine learning can access data and use it for learning for themselves.

The process of learning begins with gathering data, direct experience, or instructions, in order to look for patterns in data and make the best decisions in the future based on the latest examples that we provide to machine learning. The first thing is to allow the computer maket he system learn automatically without explicitly

programmed without assistance from humans and adjusting actions in order to that algorithms

Some Machine Learning Methods

- A. Supervised machine learning algorithms
- B. Unsupervised machine learning algorithms
- C. Semi-supervised machine learning algorithms
- D. Reinforcement machine learning algorithms

1.2. A Supervised Machine Learning

It could be applied what has been learned in the past to new data using labeled examples for the future examples. A function for making predictions about the values of the results is produced by the learning algorithm. The system has the capacity to offer targets for fresh inputs. After required sufficient training time and examples algorithm can make decision based on the latest values.

1.3. B Unsupervised Machine Learning

When the training data is unclassified or unlabeled, unsupervised machine learning algorithms are applied.Unsupervised learning algorithms studies Show that how system can infer a function to describe a hidden structure from unlabeled data.The system explores the data and can draw interferences from datas and sets to describe hidden structures from unlabeled or classified data, but we should know that the system doesn't figure out the right output but as we said, it explores data and try to label it.

1.4. C Semi-supervised machine learning algorithms

Semi-supervised machine learning algorithms stays somewhere around supervised and unsupervised learning. As an illustration, a small amount of labeled data and a large amount of unlabeled data are provided to both learning algorithms for training. Systems that employ this technique can increase learning accuracy and usually, semi-supervised learning is chosen when acquired labeled data needs relevant resources in order to learn from it. Otherwise unlabeled data doesn't require any additional resources to use in the system.

1.5. D Reinforcement machine learning algorithms

A learning technique that engages with its surroundings is reinforcement machine learning. This method allows that automatically determine the ideal behavior in order to maximize its performance. Feedback for this system is essential and required for the agent to learn which action is best for the spesified situations; this is known as the reinforcement signal.

Machine learning gathers massive quantities of data andand analysis the data in order to needs. While it generally delivers more accurate and faster results, it may also require additional time and external resources to train it properly.

1.6. Image Processing

Basically, image processing involves the following three steps:

1. Importing the image via image acquisition tools
2. Analysing and manipulating the image
3. A report based on image analysis

Image processing is a technique used to extract information from an image. It is a form of signal processing in which an image serves as the input, and the output

could be an image that is related to that image. Image processing is one of the great rising technologies today.

OpenCV (Open Source Computer Vision Library)

OpenCV is a software library for computer vision and machine learning. The purpose of OpenCV was to create a standard infrastructure for computer vision applications. Over 2500 optimized algorithms are available in the library, and these algorithms may be used to identify faces, categorize human behaviors in movies, find and extract 3D models of objects, follow camera motions, stitch together photos to create high-resolution images of complete scenes, and more.

It supports Windows, Linux, Android, MAC OS, and Python, Java, and MATLAB interfaces. The majority of its applications are in real-time vision. About ten times as many functions make up the more than 500 algorithms that exist. C++ is the native language of OpenCV.

1.7. Smart Vehicles Techonologies

Nowadays one of the biggest research development Projects are smart vehicle(AV) systems. Smart vehicles combine a variety of sensors to perceive their surroundings.The smart vehicle industry expected to reach 900 billion GBP market value in 2025.When we are designing an smart vehicle,we must avoid accidents.This is one of the most important feature for lifes.Rather is causing peoples lifes of making huge Money losses on peoples bank accounts we need to make an exception for that case.In this case smart vehicles comes the priority.Driving assistance systems could be controlled by a driver to decrease the distractions and accidents.There are low chances of crashes rather the humans because computers can calculate the visual datas thousand time faster than the human and it makes machines more trustable than the humans.

Of course there are risky ways of accepting such tehnology.There are trolley problems humans asking about that robots can’t easily answer to it.Let me explain.A trolley problem consist of 2 ways that needed to be choosed.Think it as there are 5 little childrens in one way and 10 old people in other way.The car can’t be stopped at this moment cause brakes are doesn’t replying to commands which you needed to choose to kill 5 little children or 10 old people otherwise kill your self with crashing to a wall.Since a machine can’t decide an ethic problem it self humans are not trusting too much to devices but it still a great calculatin machine thats reply’s faster the humans.Thats why there are 2 different communities in this industry but it doesn’t change that smart vehicles industry is a very big market in this time and will make it bigger in the next featured times.

2. AIM OF THE STUDY

In this chapter the aim of the study is explained briefly. The reason of the study and development stages are talked about. Difference from other studies are indicated with methods and results. Also the reasons for choosing this solution for specified problem is explained in this section.

Nowadays smart vehicle technologies are increasing and automotive companies are giving their full effort to decrease the accidents that happen because of the drivers and other objects. Thus increasing awareness of the driver and warning the driver at some point is very efficient to achive this success. That is why we are focused on this problem to solve it. Many accidents are happen from driver side and the reasons for accidents are generally distracted driver or blind spots. So we had researched the best methods to imply in this study. Distracted driver is studied with machine learning models in general. Different machine learning applied in related fields but our method not have too many examples. Our algorithm is applicable to real-time projects for the best efficiency and in traffic one or a little more second could might cause you crash someone. That is why our method is chosen to apply. Also it's accuracy is generally good.

Surrond view system is another real life problem to be solved. There are different methods of producing automatic surround view but calibration and manipulation to pixels with an automatic way is not always grants you the best results due to environment lights or depth of cameras differ. That is why we bring a manual calibration and fixing method that grants you the stable surrond view system to produce. Due to it is fixed and not calibrate every frame it is fast and high quality.

3. RELATED STUDIES

Characterization. In chapter 1 we address the introduction part of machine learning models and different tools for achive smart vehicle technologies. Chapter 2 discuss the related studies about the driver behaviour recognition via different machine learning algorithms and achive surround view with different computer vision tools. Thus having knowledge about how to success about the selected real life problem will be easier to solve and make us choose the right method to start the project with.

University of Edinburgh, UK resarchers Chaoyun Zhang, Ruı lı and their coworkers aimed to reduce the incidence of car accidents rooted in cognitive distraction [1]. Prior work has largely relied on a single sensing modality, which can be a single point of failure, to achieve these runtime/accuracy requirements. In this paper, they use deep learning's extraordinary feature extraction capabilities to propose an InterwovenDeep Convolutional Neural Network (InterCNN) architecture to address the problem of accurate real-time classification of driver actions. The suggested system integrates abstract features extracted from multi-stream inputs, such as in-vehicle cameras with diverse fields of view and optical flows estimated based on recorded images, using several fusion layers. This creates a tight assembling mechanism, which greatly increases the model's robustness. In order to improve classification accuracy, they also create a temporal voting mechanism based on past inference cases. Experiments using a dataset collected in a mock-up car show that the proposed InterCNN with MobileNet convolutional blocks can accurately categorize 9 individual behaviors with 73.97 percent accuracy and 5 'aggregated' behaviors with 81.66 percent accuracy. They also show that their architecture is computationally efficient, as it performs inferences in under 15 milliseconds, meeting the real-time requirements of intelligent vehicles. Despite this, their InterCNN is resistant to lossy input, as the classification is correct even when two input streams are blocked.

University of North Carolina, researchers Jane C. Stuts, Donald W. Reinfurt and their coworkers researh the reason of the crashes due to having look at the CDS data when is 1995-1999 where 48.6% of the drivers were identified as attentive at the time of their crash, 5.4% as “looked but did not see” and 1.8% as sleepy or asleep, the other 8.3% were identified as distracted and the remaining 35.9% were either as unknown or no driver present.[2] Yearly trends in specific driving distractions based on weighted CDS data also taken and the distracting events are ordered. These activities were was object event, adjusting radio/casette/CD, other occupant, moving object in vehicle, other device/object, vehicle/climate controls, eating, drinking, using/dialing cell phone, smoking related, other distraction and unknown distraction. Due to these events yearly crashes happened and in this study given trendly accident reasons will taken to study and try a machine learning algorithm to detect these activities in real time.

Member of IEEE Matti Kutila, Maria Jokela, Gustav Markkula and Maria Romera Rue studied driver assistance system and electronic devices such as navigators, cell phones, etc. Steal big amount of driver attention [3]. As a result, they research and concluded the automobile industry is working to create a driving environment in which input-output devices are intelligently planned, allowing the driver to focus on the surronding traffic. The driver’s current status must be assesed in order to enable a smart human-machine interface (HMI). Their study describes a system for measuring a driver’s distraction and gives some first results by combining stereo vision and lane tracking data and utilizing both rule-based and support-vector machine (SVM) classification algorithms, the module may asses the driver’s visual and cognitive workload. Data from a truck and passanger automobile were used to evaluate the module.

Singapore IEEE Fifth International Conference had an article in 2019 that is related with the current study that detecting distracted driving in real time [4]. Authors Maitree Leekha, Monotoni Goswami and the resarchers studied distracted driving caused by co-passengers and mobile devices. As a result, they acknowledge

there is a growing demand for a system that can detect a distracted driver in real time and sound an alarm. They describe a simple and resilient architecture based on foreground extraction and a Convolutional Neural Network to achive this goal (CNN). In comparison to state-of-the-art models, their ConvNet model contains much less parameters (0.5M). Their model beats and comparable to the models proposed so far on both their datasets with test accuracy of 98.48 percent and 95.64 percent, respectively.

Martin et al. Provide a driver activity recognition task. As a graph convolution-based approach in [5]. Their method is primarily based on body posture data. They also extracted items and their position information, which they used as inputs in their activity recognition graph convolution-based network model. For picture categorization, Nel and Ngxande used 3 dimensional ResNets [6]. They employed more complex models and demonstrated that they are effective, thanks to residual connections that prevent vanishing or exploding gradients. Deeper models also helped with feature extraction efficiency. For driver activity recognition, Xing et al. [7] used deep neural networks as an image classifier. The authors employed the Gaussian Mixture Model to segment raw photos in order to recover the driver body, and then used these segmented binary images in deep neural networks. They employed a transfer learning strategy in the neural network stage, fine-tuning a pre-trained network using binary pictures. Eraqi et al. [8] also did a study that used segmentation. They also used these photos to create a body segmentation and build a deep learning classifier. They did, however, create face and hand detection models and train deep learning models for each of them. In addition, they constructed a deep learning model to classify the original image. Finally, they weighted each of these models to create an ensemble of deep learning models as their classifier.

Halabi and colleagues [9] created their own virtual reality driving simulator environment. They installed force sensors in the driver's seat and collected sensor inputs from many participants in this simulation environment to create their own training and test data. To categorize a test signal, they employed a basic difference-

based comparison technique with training data. They were able to discern between different driver behaviours and come up with acceptable results. Behera et al. [10] retrieved Convolutional Neural Networks (CNN) characteristics from a driving image using various pre-trained deep learning models. They were able to create three activation maps in this manner: one for the entire image, one for body position features, and one for body-object interaction. They then utilized a merging layer in their own deep fusion network to collect all of these activation maps and used this network to recognize driver activity. Zhao et al. [11] developed a cascading multiple attention-based deep learning model in another investigation. Their entire network is made up of subnetworks. At first, one of the subnetworks takes the original image and creates an activation map. The most active area on the activation map is then cut out of the input image and used as an input in another subnetwork. A K-Nearest Neighbour classifier is used to combine and classify the outputs of each subnetwork. The successive crops from an input image go closer each time, pointing to the most discriminative region in this image, such as the cell phone in the driver's hand or the driver's head looking at a passenger, and this improves the system's performance.

Yang et al. [12] studied the actions of drivers by analyzing their gaze and object detection. The driver's gaze point is approximated using a facial recognition system, and object detection is performed using an object detection system. As a result, non-driving behaviors such as using a cell phone, tablet, or reading a book can be identified. They used deep learning and time series nonlinear system modeling in their research. Zhang and colleagues. [13] created their own deep learning model that was fed data from different sources. These inputs are collected using two in-vehicle cameras, which produce a full body image and a driver's head image, as well as the corresponding optical flow images. They fed these inputs into an interconnected CNN with certain fusion layers, which recognized driver activity.

4. MATHERIALS

In the previous chapter, we discussed different machine learning algorithms for utility material selection. Matherial selection very important due to machine learning networks is require very good Graphical Process Unit and we need the data to provide to the system to training. Firstly we give information for machine learning model equipment selection that which platform it will executed and how can we provide the data to gather. The camera and the training platform is also talked. After the machine learning model, the surrond system equipments given as where the system is executed, which devices is needed and the important equipments to provide surround view system materials informations given.

4.1. DVR Device

CCTV-DVR systems are the devices to take frames from security cameras and record the frames in to their systems and these type of devices called Digital Video Recorder (DVR). DVR devices has a compact structure that takes camera frames and sounds from a far away place take the record of that frames and play the frames from the records.

The recording devices vary according to the needs and the existing CCTV system. Generally 4, 8, 16, 32 channel DVR types are used in CCTV systems. The channel count is determined by the maximum number of cameras that can be connected to the CCTV system.

Figure.4. 1CCTV 8 channel DVR device

4.2. IR Camera Device

A camera with infrared (IR) technology provides the ability to use night vision, allowing the camera to see objects and people even in total darkness. IR cameras as the known of infrared security cameras are the type of cames that offer improved videos at the times of night or in less lightened places.

In the camera system we design we had used resolution 720 p, 30 fps camera.

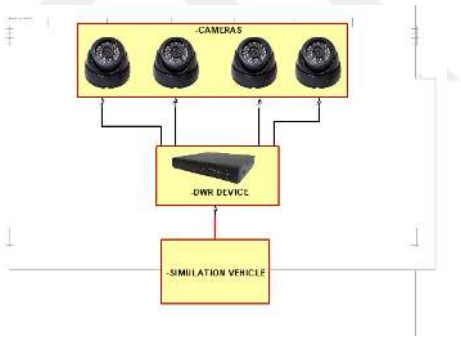


Figure.4.2. 720p IR Camera Device
Figure.4.3Camera System for 360 Vision

4.3. Roboflow AI Platform

Roboflow platform has machine learning models such as we used in our system that YoloV5 model. Roboflow is a computer vision developer framework. This platform has some features for data preprocessing and model training

techniques. For model training some of the necessary libraries already presented in this platform such as Mobile Net, Yolo, TensorFlow, etc.

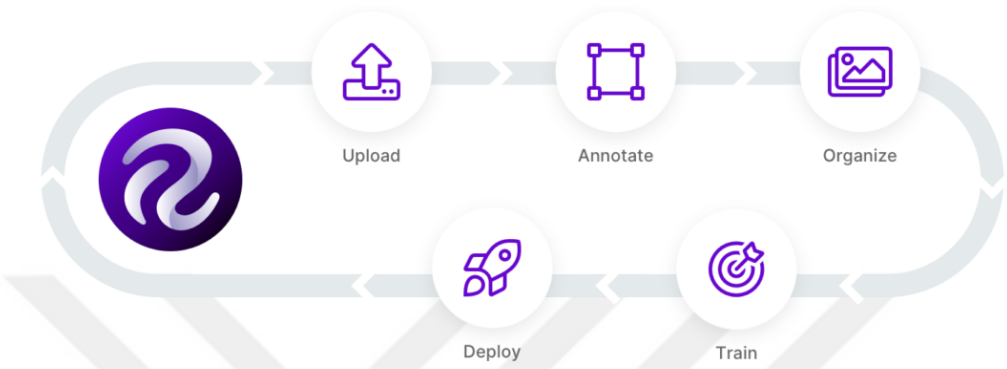


Figure.4.4. Roboflow steps to develop a system

4.4. Google Colabs Platform

The study is conducted on a platform called Google Colab which tehnical information's for developing a model in this platform are n1-highmem-2 instance, 2vCPU @ 2.2 GHz, 12 GB RAM, 100GB Free Space, idle cut-off 90 minutes, maximum 12 hours.

Colaboratory is a product from Google Research. Google Colab allow researchers to write and execute python code through the browser, and is especially well suited for machine learning studies. Moreover to explain the Colab is a hosted Jupyter notebook service that give usage without any deploy in your personal computer including free acces to computing resources including GPUs that Google gives acces to researchers.

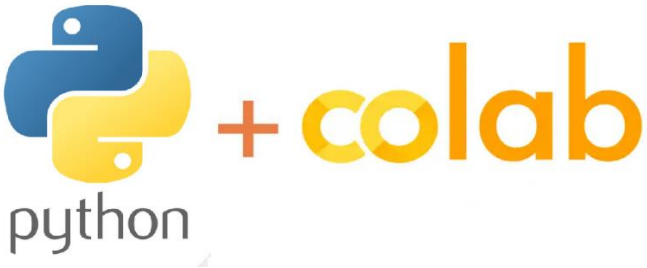


Figure.4.5. Google Colabratory

4.5. A Service Bus

In the study a service bus is used to gather original data to system. The most important thing to developing a detection system accurate model is collecting enough images and the images should be qualified. In the study for driver behaviour Koluman Otomotiv Endüstri (KOE) company has provide qualified data for model training.

The company has its own service trucks inside the factory. For data collecting, dvr recorder, dash cam and a service truck is used.

Smoking Texting

Drinking Talking

Figure.4.6. Driver Activities

4.6. Military Truck

Koluman Automotive Industry, which started R&D studies at the end of 2015 for the Türk Silahlı Kuvvetleri (TSK) need for an 8x8 logistics support vehicle with an armored cabin launched the “DERMAN 8X8” Armored Logistics Support Vehicle with its own brand at the end of 2018, wtih a high localization rate.

With its production infrastructure, KOE has the capacity to meet the 8x8 truck and tractor truck needs of the TSK for various special systems.

Koluman, whose priority in military projects has always been to meet the needs of the TSK, aims to offer the “DERMAN 8x8” vehicle to NATO armies as well.

Figure.4.7. DERMAN armored support vehicle

4.7 Test Subjects

Camera CSI cable for PC input, Power Adapter and Cable for PC power supply and ground cables for cameras to work under the same ground-Desktop computer connection.

4.8. Fisheye Camera Device

The electronic control unit (ECU) sends the four live images at once, where they are immediately processed, merged, blended, and stitched. Before displaying a crisp, single, smooth, real-time image on the driver's monitor, the distortion from the wide-angle camera lens is also addressed.

Figure.4.8. BN360 Fisheye Camera

4.9. Fanless MLT Standart PC

High-performance graphic processing thanks to the use of the GPU. Used fors mart solutions such as Smart Patrol, Automation Control System and Military Purposes. High Standart and effective at many struggling conditions

Figure.4.9. Fanless PC

4.10. 10.4 Inches Monitor

Industrial grade and military screens, designed for harsh environments with humidity, dirt, dust, snow etc… Designed to work at extreme conditions where temperature is very strong. Operation range is from -35’C to 70’C. MIL-STD 810 is applied.

Figure.4.10. 10.04 Inches Monito

5. METHODS

In this chapter the problem solving methods explained briefly. ML engineer makes decisions is about how he or she intends to train their models. This chapter presents this valid strategy. Chapter 3 discusses a brief recap on what material is avaible to use and Chapter 2 discusses related studies about the methodologies that we can use. As we get information about machine learning and computer vision the next step to make real life project to test the problem solving. This chapter proceed with the selected algorithm, why the algorithm is a good performer and methodologies, how to implement explanation and validation metrics. Then we proceed with the proposed method results and metrics. With metric explanation the results are shown also in this chapter.

5.1. Driver Behaviour Recognition based on Machine Learning Model

5.1.1. Introduction

A significant component of modern technology is driver safety. Renowned companies of automotive industries are increasing their safety protocols for drivers. Due to traffic can become extremely congested at sometimes it is important to stay focused on driving and not become distracted by radio, drinking, talking with the cell phone, talking with the passengers or texting. About 42,000 people died each year on roads and highways in the United States, between 2000 and 2005[14].

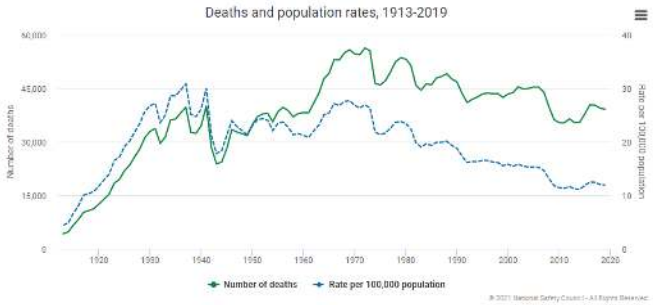


Figure 5.1. Death And Population rates, 1913-2019

Knowing traffic accidents is very significant for society the improvements had been made over time. Since the early 1900s, motor vehicle safety has dramatically improved on all fronts. Driver behaviour and attitudes have changed fairly, as well as vehicle safety technology which makes car travel safer [15]. The new technologies have arrived for driver safety.

One of the technologies are machine learning and artificial intelligence. This incredible technologies aids in the development of computer programs that can automatically access data and complete tasks via predictions and detections, allowing computers to learn and grow as a result of their experiences [16].

The sub branch of this new technology called machine learning is deep learning that teaches computers to learn by example just as we learned as a child [17]. A computer becomes proficient in performing tasks from images, text, or sound using deep learning and can achive state-of-the-art accuracy, often outperforming human implementation.

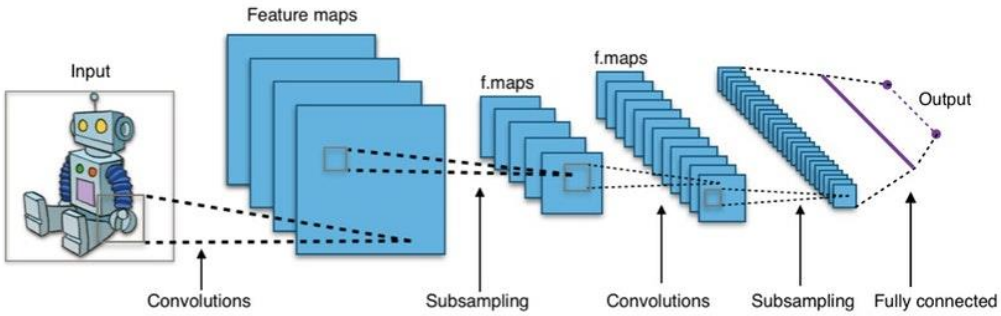


Figure 5.2. Neural Network Architecture

Deep learning usually referred with deep artificial neural networks. In an artificial neural network, the architecture based on neurons and layers which signals travel between neurons just like the brain. But instead of using an electrical signal, neurons has weights in a neural network. A neuron biased great deal to another neuron will impact the effect on the next layer of neurons. The final layer neourons patches these weighted inputs and come up with an answers.

CNN is the most widely utilized technology in this field (Convolutional Neural Network). CNN also known as ConvNet that specializes in processing data has a grid like topology, such as an image [18]. CNN has its core building block called Convolution Layers. Convolutional layers extract this architecture from standard NN (Neural Network). This layers extracting images features on many levels to detect objects and classes if a fast way with convolution, pooling, padding tools.

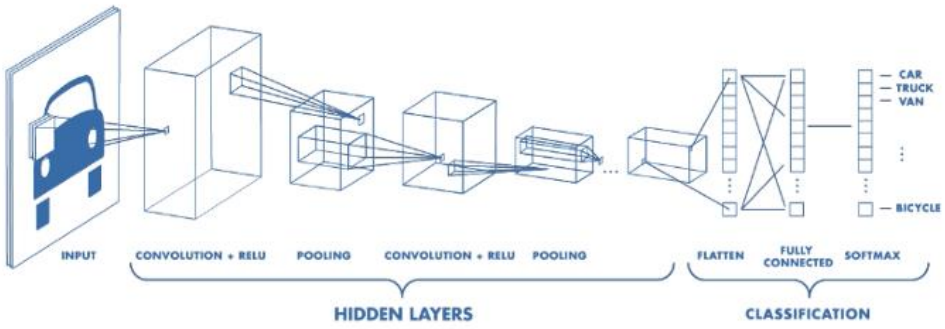


Figure 5.3. Convolution Neural Network Architecture

Automotive industry using this models architectures for detection of objects such as pedestrians, cars, bicycle etc… Use cases in this are generally divided into two branches, one is classifying image or frame in two stages, another one is taking the image in one step for classifying like YOLO.

The most popular object detection algorithms are [19]:

- R-CNN
- Fast R-CNN
- Faster R-CNN
- Mask R-CNN
- Single Shut Detection
- You Only Look Once

The popular object detection algorithms comparison given below.

Table 5.1. Comparing YOLO with other object detection algorithms

Detection frameworks	Train	mAP	fps
Fastest DPM [20]	VOC 2007	30.4	15
R-CNN Minus R [21]	VOC 2007	53.5	6
Fast R-CNN [22]	VOC 2007+2012	70.0	0.5
Faster R-CNN VGG-16 [23]	VOC 2007+2012	73.2	7
Faster R-CNN ZF [24]	VOC 2007+2012	62.1	18
YOLO VGG-16[25]	VOC 2007+2012	66.4	21
YOLO [25]	VOC 2007+2012	63.4	45

YOLO is much faster than other algorithm because it takes region proposal inside an image multiple class probabilities and processes it at the same time to get faster.

The YOLO algorithm works with several steps [26]:

- Diving the image into N grids each region having equal dimension of SxS.
- Using bounded boxes to predict objects
- Use loss function
- Using Non Maximal Suppression to suppresses all bounding boxes that have low probability scores.

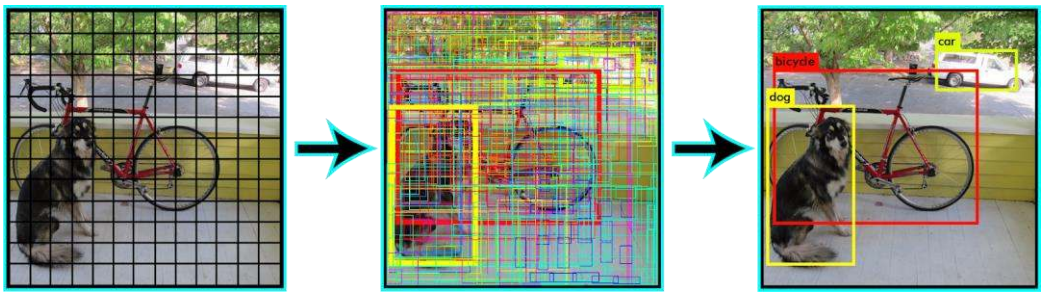


Figure 5.4. Regions remained that contained objects.

The latest comparisons of YOLO versions show that if real time object detection is what developers looking for, should use YOLOv5 for better choice. If looking for more custom configuration and not using it with real time YOLOV4 in Darknet continues to be most accurate.

In our case we are determining the driver behaviour in motorway and real time is a significant factor for modelling due to losing concentrate for a few seconds may cause a major fatal accident. So, we have decided to use YOLOV5 in our project.

5.1.2. Setting Yolov5 Algorithm

There are several options to set a platform use in machine learning model. While developing a yolo algorithm pretrained weights need to be download from git page. In this project we used roboflow.com platform for our Yolov5 model. Roboflow is a computer vision developer framework. This platform has some features for data preprocessing and model training techniques. For model training some of the necessary libraries already presented in this platform such as Mobile Net, Yolo, TensorFlow, etc.

5.1.3. Collecting Images For Driver States

The most important thing to developing a detection system accurate model is collecting enough images and the images should be qualified. In our experiment for driver behaviour KOE company has provide qualified data for model training.
The company has its own service trucks inside the factory. For data collecting, dvr recorder, dash cam and a service truck is used.

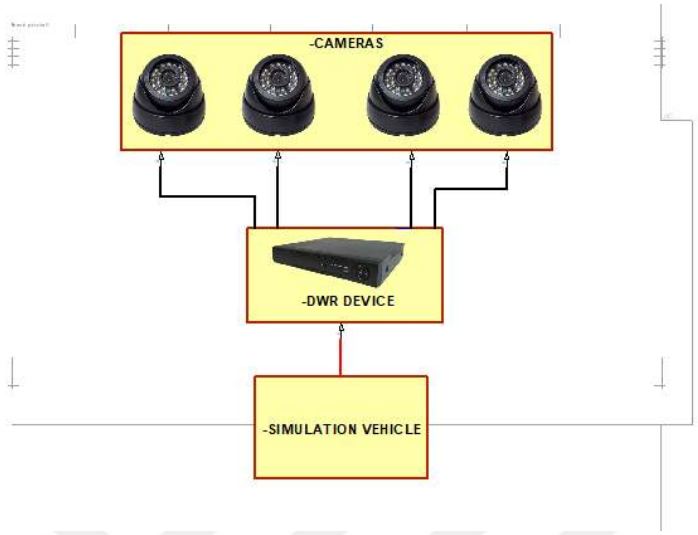


Figure 5.5. Image collection system design

A simulation area has been set and several workers voluntarily participated to project. The data collecting project finished with 5-hour data. The video data is .AVI files (Audio Video Interleave) converted to PNG (Portable Network Graphics) files with third party programs and number of 3099 good, qualified image is taken for model training. The dependent classes for model training is given below:

Table 5.2. Dataset classes distribution

Databases:	Driver Behaviours	
Instances:	3099	
Attributes:	10	
Sum of Weights	286	
Sr. No	Attributes	Type
1	C0_smoking	Nominal
2	C1_talking_passenger	Nominal
3	C2_radio_checking	Nominal
4	C3_reaching_behind	Nominal
5	C4_drinking	Nominal
6	C5_texting	Nominal
7	C6_talking_on_phone	Nominal
7	Class	Nominal

In order to train and quantitatively evaluate the presented method, 3099 image has been selected to use at the models training. The dataset has been split to three parts for better evaluation. Train %70, Validation %20, Test %10 is selected. Dataset median image ratio is 960x1080 and class balance is around equal ratio. The training images used for model training. Validation and training images are used for improving the model accuracy. Dataset used in this article given below.

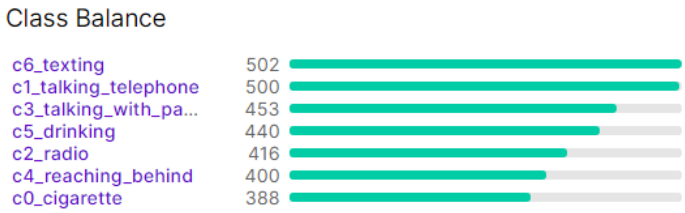


Figure 5.6. Dataset instances distribution

Dataset images are preprocessed for decrease training time and increase performance by applying image transformations to all images in this dataset.

Auto Orient

Resize (Stretch to 416x416)

Grayscale

Auto orient discard EXIF (Exchangeable Image F) rotations format and standardize pixel. Resize is downsizing images for smaller file sizes and faster training. Grayscale merge color channels to make your model faster and insensitive to subject color.

Smokin Texting

Drinking Radio

Figure 5.7. Original dataset samples

5.1.4. Data Augmentation Methods

Data augmentation is covering very important space for deep learning algorithms. Models have bigger and complex datasets confirmed that has a better accuracy predicting the expected values. Due to this problem having better datasets and more instanced always was an important subject for deep learning area. Data augmentation method aim to have more data with using color space augmentations, kernel filters, mixing images, rotation matrixes, cropping images etc. [27].

- Data augmentation is applied to driver behaviour detection dataset.
- Rotation Between -15 ° and +15 °
- Shear ±15° Horizontal, ±15° Vertical
- Saturation Between -25% and +25%
- Brightness Between -25% and +25%

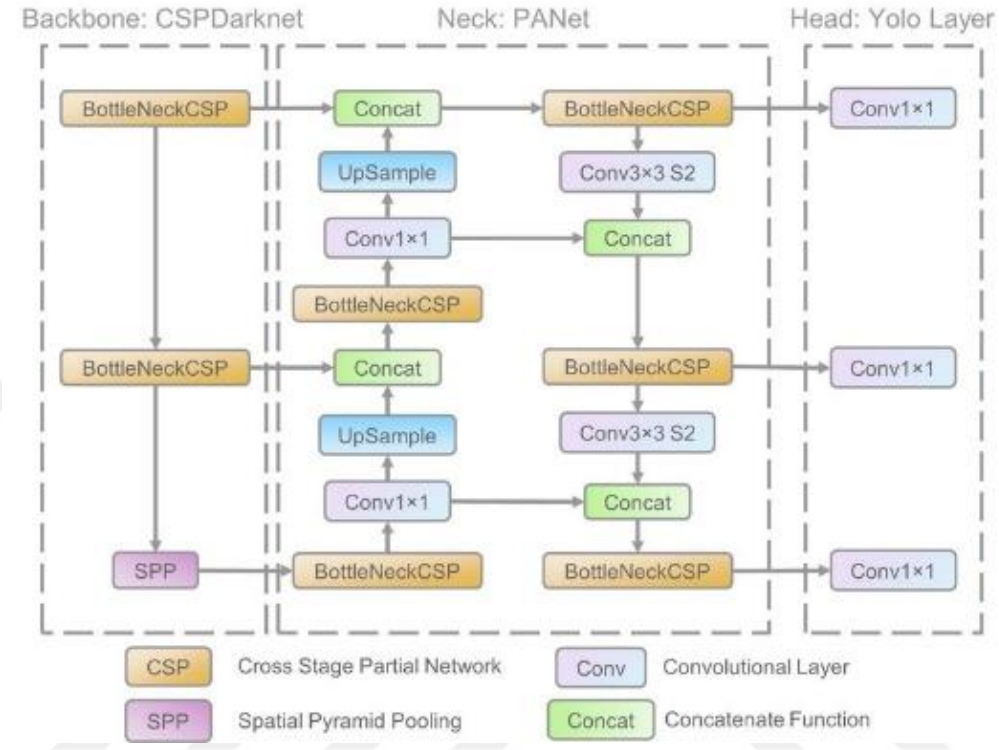
Rotation add variability to rotations to help your model be more resilient to camera roll. Shear add variability to perspective to help your model be more resilient to camera and subject pitch and yaw. Saturation randomly adjusts the vibrancy of the colors in the images. Brightness add variability to image brightness to help your model be more resilient to lighting and camera setting changes.

After data augmentation methods generated maximum version size for images are 7,437 images.

5.1.5 Proposed Method

The proposed method uses one deep learning model called YOLO. The CNN architecture is reviewed and compared with respect to the trained model's accuracy performance on the validation set, and the CNN architecture is trained using the training set. For the purpose of simplicity and computational cost, the input photographs are downsized to 64x64 pixels in the experiments. Table 3 shows the deep learning architecture configuration for YOLO model.

Table 5.3. The Graph of YOLO Model Parameters [18]



During the experiment SGD (Stochastic Gradient Descent) is used. In the convolutional layers, Up Sample and Concat (Concatenate Function) is used. Up sample stride is 2. Batch size taken 32 and 300-epoch training is realized. For the model results tensor board metrics has taken during the time series. Wall time is taken 4.180 hours (4 hours 10 min 48s).

During the experiment SGD (Stochastic Gradient Descent) is used. In the convolutional layer, Up Sample Concat (Concatenate Function) is used. Up sample stride is 2.

Proposed methods steps are given below.

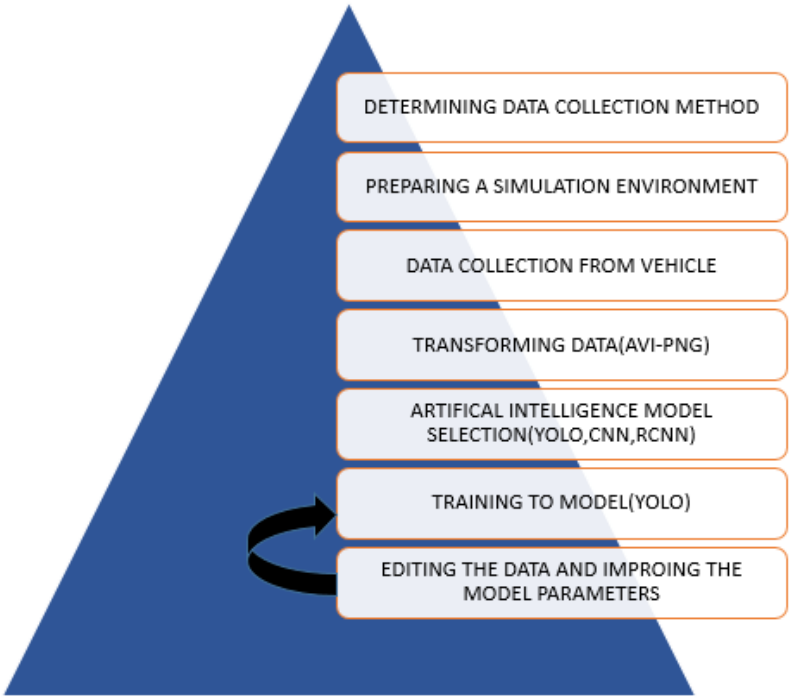


Figure 5.8. Steps for proposed method

5.1.6. Experimental Study And Discussion

The experiment is conducted on a platform called Google Colab which technical information's for developing a model in this platform are n1-highmem-2 instance, 2vCPU @ 2.2 GHz, 12 GB RAM, 100GB Free Space, idle cut-off 90 minutes, maximum 12 hours.

The test results of the proposed method illustrate the classification performance. The proposed approach' mAP_ 0.5(Mean Average Precision) value is 0.3264 and precision value is around 0.6342.

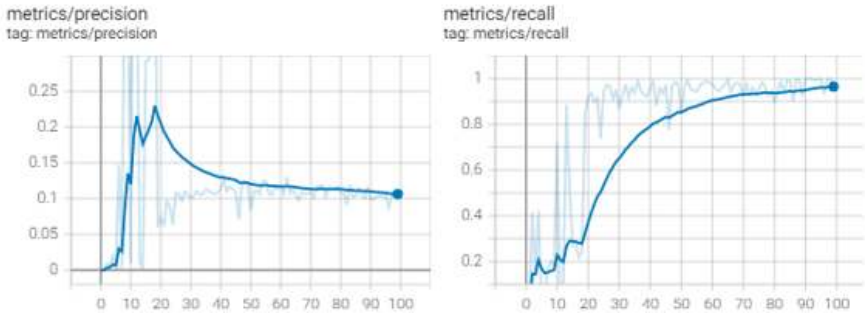


Figure 5.9. metrics/precision and metrics/recall value (100 epochs)

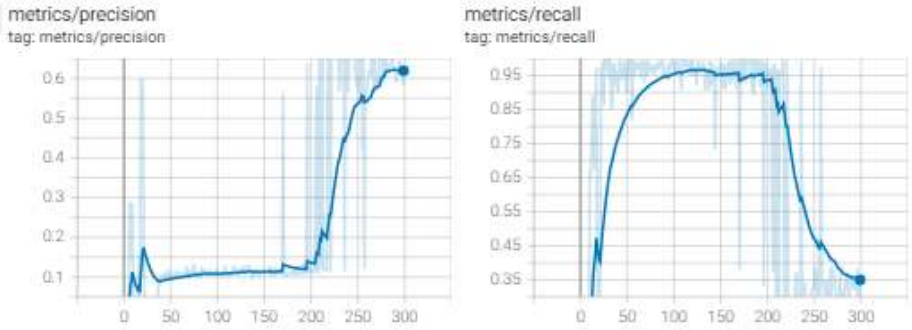


Figure 5.10. metrics/precision and metrics/recall value (300 epochs)

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives}$$

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

$$mAP = \frac{1}{11}\sum_{Recall_i} Precision(Recall_i)$$

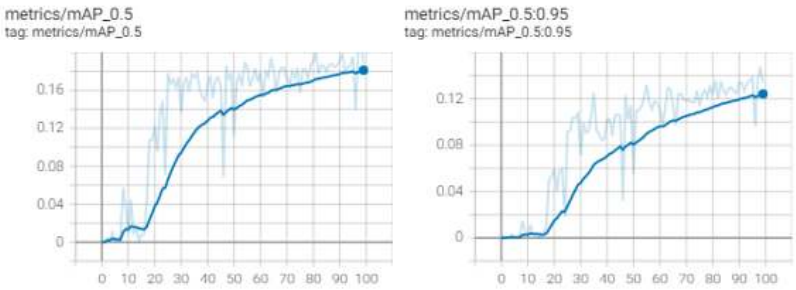


Figure 5.11. metrics/mAP_0.5 and mAP_0.95 value (100 epochs)

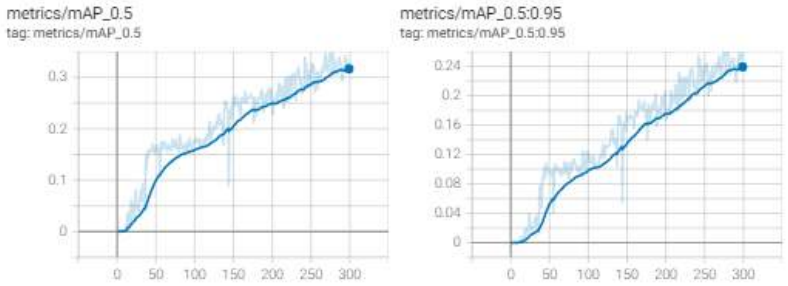


Figure 5.12. metrics/mAP_0.5 and mAP_0.95 value (300 epochs)

Radio checking Talking on phone

Texting Talking passenger

Figure 5.13. Positive predicted dataset samples

Regarding these results, the proposed method is not assuring acceptable performance metrics toward driver activity recognition task. But also, it is giving hope with the results with evaluations. During the evaluation after the 200-epoch precision value is increasing while recall value are decreasing. Generally, for models better than random precision and recall have an inverse relationship. Epochs show that after 200 epochs model gets better at making decision. Its determining that improving the model could give outperforming results for other models.

In this study, we generate the training, validation, and test datasets in KOE industry. Due to lack of resource about driver activity prediction articles making comparison between results is not fair, but it can be applied to have a general intuition about the quality of proposed model.

YOLO is a data dependent deep learning model with knowing this information, having more data and more different people types for having datasets could increase the accuracy of the model. Also increasing the epoch and batch size

in model training will affect the predictions of model. Another approach for increasing the model accuracy is combining different models with YOLO model. As an open problem, complex or black box approaches like deep learning can have significant drawbacks when used for decision making (such as in this work) which requires interpretability and accountability [28]. When a decision is made incorrectly, this problem gets more serious, therefore integrating two models to correct one model's flaw could also be a good idea.

5.2 360 Degree Surrond View Camera System

5.2.1 Introduction

Advanced Driver Assistance Systems (ADASs) are one of the most known technology at automotive industry. Several important studies like Butakov and Ionnou, Ziebinskie described that automotive industry market intention is increasing for ADASs systems because ADASs increase the control of the vehicles for driver's by providing real-time enhanced visualization. The target is to increase the awareness of the vehicle's driver with developing new technologies.

One of the the real-time ADASs system is Bird's-eye Surround View (SV) which integrates several cameras to produce a virtual bird's-eye view in real time. Bird's-eye surround view is an important specification for Advanced Driver Assistance System (ADASs) that provides a real time surrounding view of the vehicles to the drivers.

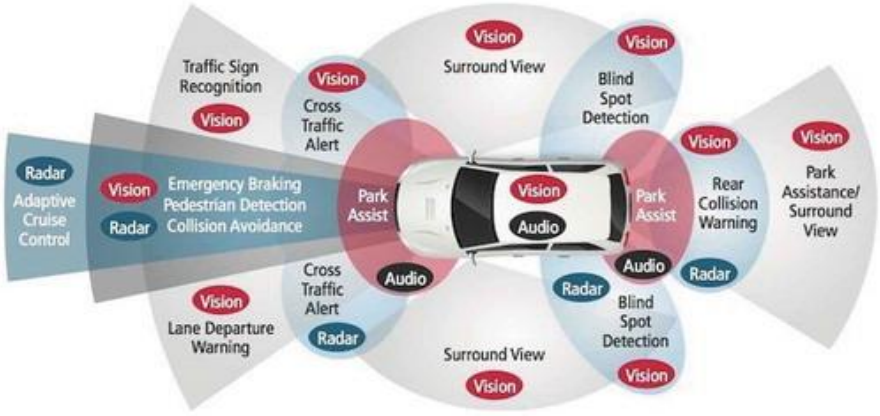


Figure 5.14. ADAS System Mechanism

SV is a sensor fusion problem in which many sensors, in this case cameras, are merged and processed together to create a complete image of the world surrounding a vehicle and to assist a machine or human driver in making quick choices. According to Ananthanarayanan [29] et al., processing many sensors in real time can be a difficult computing task, especially for high bandwidth applications incorporating video or radar inputs (2017).

The study system combines fish-eye cameras located 4 different side of a vehicle, 4 different camera frames are taken into recording but due to fish-eye lenses are wide range of visions the images are need to be calibrated if it will be used for top view vision.

There are related works which actively is on works like Brown and Lowe [30], Image stitching also defined problem with different approaches in the literature described in Kwatra [31] and Szeliski [32]. While we are dealing with stiching images there is an overlap area between two different images. The problem is the overlapped area can provide two sources of pixels and need to be determined for the boundary lines. Another problem is, the problem has more than two frames and has more dimensions than the typical stitching process. Also these frames are produced at real-time thus it's requires a good process unit. Second the generated frames need to be fixed due to camera height could be different and the distance for the ground

need to be equalized. The algorithm first correct the lens distortion for every camera with using calibration and fixes the perspective of camera visions from birds view as looking from a downwards, away from the car to top down view. After step the fixed frames from cameras are stitched with together into one frame. When a calibration is fixed for frames a data file with. Yaml extension camera frames will be fixed whenever the calibration is used but now it will not do calibration from zero it only gives the important parameters to function then it will be calibrated directly. This will minimize the cost of processing cycles and minimize the partitions system memory on the processor.

The study is organized as follows; Firstly the related work is explained, after lens correction calibration methods and top down view projection are presented. Next image stitching with masking and graph-cut approach is used. Lastly car image integration and blending is used and final surround view is created via different methods fusion.

5.2.2. Lens Correction And Top Down View Projection

The main reason we can perceive depth is because of our two perfectly aligned eyes. If you have noticed, if we look at close objects with one eye, the object location that we perceive with our right and left eyes will be different. But if you look at distant objects, you will see almost no difference. This relationship between our right and left eyes causes us to perceive depth autonomously. Animals with side-eyes like ducks, on the other hand, constantly move their heads or run in order to create a sense of depth, since their eyes do not see a common place. This is called structure from motion. Let's leave this method aside for now and consider a system that looks like our own eyes.

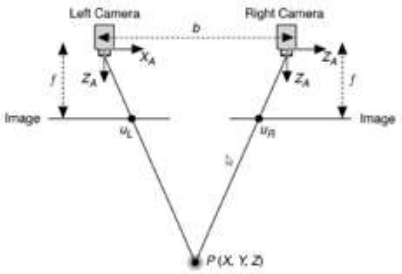


Figure 5.15. Lens Correction Coordinate Show off

If the two cameras are different in the vertical plane but aligned on the horizontal plane (like the human eye), as you can imagine, a point of the object we see will fall on the same horizontal axis in both cameras. We could have information about the depth from the proximity and distance of the objects. Since it is a big burden to search where the same point coincides in two images, when we match the horizontal axes, we get rid of a great burden by searching only on the horizontal axis. This is called stereo calibration and correction. Let there be such a calibration that the images we get from the cameras can be free from noise and their axes are the same. As seen in the figure above, in this situation, we may locate the equivalent of a P point in photos with only a modest amount of processing resources. The next step is to simply create a new image using the different pixel values. While discussing the code portion, I'll go into further detail. The outcome can be visually summarized as follows:

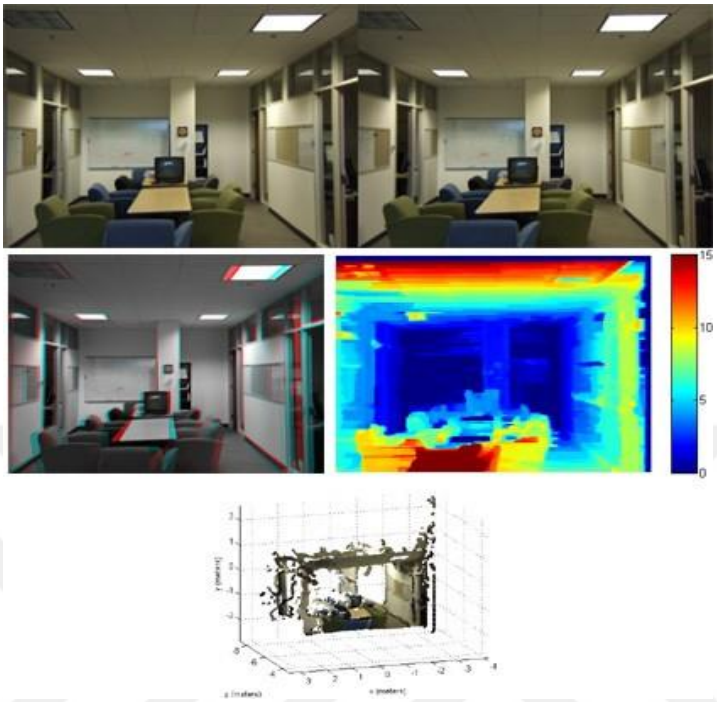


Figure 5.16. Image Perspective Depths with Colors

The cameras must first be securely and correctly mounted on the horizontal axes. For this, you can stick two cameras on a wooden plane. You can also get cameras specially fixed to the horizontal axis with small margins of error, such as Intel realsense, Zed. The next step is to calibrate both cameras one by one. You can read my calibration post for this, I will have references here. For stereo calibration, we apply a similar method to single camera calibration. We will take images of our chessboard with both cameras:

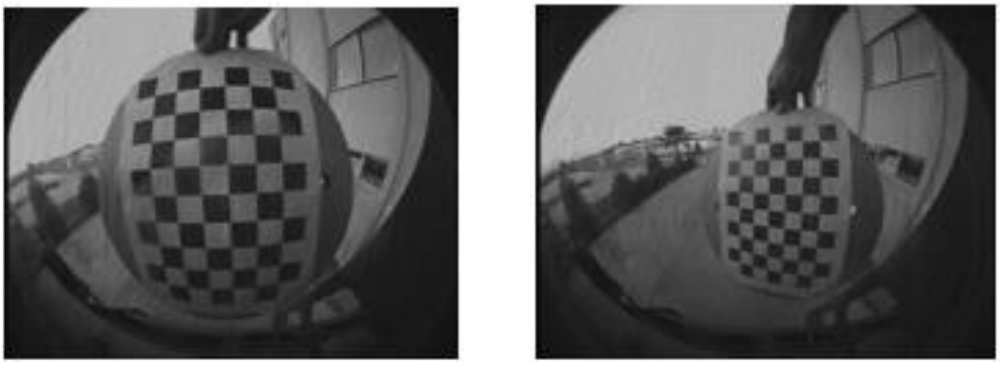


Figure 5.17. Calibration Examples with Chessboard

These images will give us the relationship between the two cameras. You must stand still while taking the images, the left and right images should be synchronized. For this, you can try to get images in as close synchronization as possible by using opencv and grab&retreive methods. You can use the code I prepared to get the images. If you believe that you have enough images, you can perform the calibration with the code below. Two tasks this code does:

OpenCV stereo calibration function:

ret, K1, D1, K2, D2, R, T, E, F = cv2.stereoCalibrate(objp, leftp, rightp, K1, D1, K2, D2, image_size, criteria, flag)

The thing we need to pay attention to in the stereoCalibrate function is the flag part. Let's see the flag values:

CV_CALIB_FIX_INTRINSIC: Does not change the camera matrix and noise coefficients referred to as K and D. It is the default value. This value will be important for you when you have calibrated the camera well before. It's enough if you just take the matrices that give the relationship.

CV_CALIB_USE_INTRINSIC_GUESS: Optimizes K and D matrices. The initial values need to be logical, calculated so that it can be optimized.

CV_CALIB_FIX_PRINCIPAL_POINT: Fixes the reference point located in the K matrix.

CV_CALIB_FIX_FOCAL_LENGTH: Fixes the focal length in the K matrix.

CV_CALIB_FIX_ASPECT_RATIO: Fixes the ratio between focal lengths.

CV_CALIB_SAME_FOCAL_LENGTH: Optimizes focal lengths and fx, fy focal lengths are equal. I think that if you use cameras of different models or situations here, it may cause a margin of error.

CV_CALIB_ZERO_TANGENT_DIST: Fixes the noise coefficients by resetting them.

CV_CALIB_FIX_K1,…,CV_CALIB_FIX_K6: Used to fix noise coefficients from first to last. It is one of the most important flag values. Even though I can't interpret it mathematically, I am saying it with the questions I asked in the opencv forum and the experience I gained in the countless (really countless) calibration attempts I made.

As a result, the R, T, E, F coefficients we obtained give the relationship between the two cameras. The next step is correction:

R1, R2, P1, P2, Q, roi_left, roi_right = cv2.stereoRectify(K1, D1, K2, D2, image_size, R, T, flags=cv2.CALIB_ZERO_DISPARITY, alpha=0.9)

You may have noticed that we transmit the R and T matrices, camera and noise matrices from the stereoCalibrate function with image resolution. There is only one flag here, CALIB_ZERO_DISPARITY. As you can guess, this flag should be active, we use it to map horizontal axes. The Alpha value, on the other hand, helps us adjust the black parts that will occur as a result of the transformation to be applied when mapping the axis of the images. I adjusted the value according to my own camera. The settings are as follows:

R and P matrices from our returned values are referred to as rotation and projection matrices. R1 and P1 give the rotation and position of the first camera (left) relative to the second camera (right). R2 and P2 are the opposite. The Q matrix allows us to switch from the disparity image to the depth map. It contains values such as the distance between the camera and the focal length I want to remind you that depending on which metric you used to measure the chessboard, you will receive these depth values when you move to the depth map. Then don't say I'm calculating 5 meters, it says 7.8 meters. Let's not forget that there are too many parameters and the problem must be well diagnosed.

We do the correction along with the noise canceling:

LeftMapX, leftMapY = cv2.initUndistortRectifyMap(K1, D1, R1, P1, (width, height), cv2.CV_32FC1) left_rectified = cv2.remap(leftFrame, leftMapX, leftMapY, cv2.INTER_LINEAR, cv2.BORDER_CONSTANT) rightMapX, rightMapY = cv2.initUndistortRectifyMap(K2, D2, R2, P2, (width, height), cv2.CV_32FC1) right_rectified = cv2.remap(rightFrame, rightMapX, rightMapY, cv2.INTER_LINEAR, cv2.BORDER_CONSTANT)

With the initUndistortRectifyMap function, we both remove the noise and map the images on the horizontal axis. For our left camera, we perform noise removal with K1 (camera matrix) and D1 (noise matrix), while performing axis alignment with R1 (left to right rotation transformation matrix) and P1 (left to right

parallel reflection matrix). Of course, we also include the resolution and data type of the result image. After obtaining the transformation matrices and passing them to the remap function, our image is corrected. By doing the same for the right image from the right camera to the left camera, we obtain our corrected images.

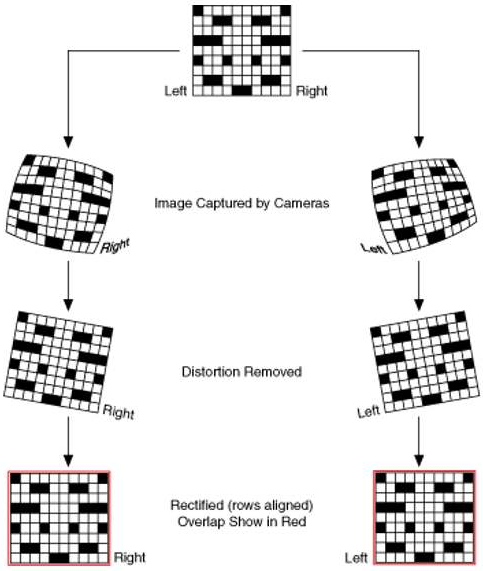


Figure 5.18. Image Calibration Steps

5.2.3. Image Stitching Methods

After cameras are calibrated the video records need to be taken to stitch from the camera frames. There are 4 different fish-eye camera we had used in this study.

The camera frames are taken from operation system which is Linux/Ubuntu. There is a capture card requires for taken the camera frames. The capture card has 16 video channels and the cables are used to transmit the video signals.

Ubuntu has serial channels in a device directory "/dev/video*" all video channels are displayed here. Before use of this directory the video capture card need to be used. Linux give the user permission the deploy its mods for spesified cards. Capture card has its own mods and functions that stored in its libraries. First capture

card libraries are installed into the PC. With using the make file the library files are compiled directly to “usr/bin” after compiled the library the library functions can be used via writing to terminal “sudo modprobe tw68”. Modding the capture give the user permission to acces video signals. Using the command of “cat /dev/video0” will give video signals by terminal. Or it can be accesed by 3rd party programming like opencv functions via python. The usage of the access is depended to user. There is only one problem need to be fixed. That is video signals coming but it is grayed out. This problem happens to video capture types time to times

Video capture cards has different types to emit video types. It could be PAL or NTSC. What is the difference between PAL and NTSC?

The main differences between Pal and Ntsc are; Both are terrestrial broadcast signals. Countries located on the European continent generally use the pal broadcast. Countries in America and Asia generally use Ntsc. Turkey Pal uses terrestrial broadcasting. The image quality of both is the same. Pal is the world's most used color coding system. PAL uses a 4.43MHz color signal, while NTSC video format uses a 3.58MHz color signal.

Ubuntu terminals helps to convert image type with v4l2-ctl tool used to control video4linux devices, either radio or video both input and output.

“v4l2-ctl -d /dev/video0” Video1 input
“v4l2-ctl -d /dev/video1” Video2 input
“v4l2-ctl -d /dev/video2” Video3 input
“v4l2-ctl -d /dev/video3” Video4 input

After all the controls are set the videos are recorded for image stitch test.

5.2.4 Frame Capturing

Code 1.1 Example code to perform a simple capture

```
   cv::Mat frame;
   VideoCapture cap("filepath");
   Size size = Size(640, 480);
   int codec =VideoWriter::fourcc('codec');
   VideoWriter writer;
   writer.open("filepath.extension", codec, 15.0, size2, true);
   if(!cap.isOpened()){  // check if we succeeded
      cout<<"Cap1 is not opened";}
   while(1){
      cap>>frame;
      writer.write(frame);
      if(waitKey(30) == 27) // Wait for 'esc' key press to exit
      {
         break;
      }
}
   cap1.release();
```

Code 2.1 Example code to perform a simple calibration

```
   FileNode K = fs["camera_matrix"];              // Read string sequence - Get node
   //cout<<K.mat()<<endl;
   cv::Mat Ktest=K.mat();
   FileNode k= fs["distortion_coefficients"];
   //cout<<k.mat()<<endl;
   cv::Mat ktest=k.mat();
   //Rectification Matrix (R)
   float rectification[9]={1,0,0,0,1,0,0,0,1};
   cv::Mat R(3,3,CV_32FC1,rectification);
   //cout<<R<<endl;

cv::fisheye::estimateNewCameraMatrixForUndistortRectify(K.mat(),k.mat(),frame
Size,R,newCameraMatrix,1);

cv::fisheye::initUndistortRectifyMap(K.mat(),k.mat(),R,newCameraMatrix,frameS
ize,CV_32FC1,mapX,mapY);
   cv::Mat imgUndistorted;
   cv::remap(src,imgUndistorted,mapX,mapY,cv::INTER_LINEAR);
```

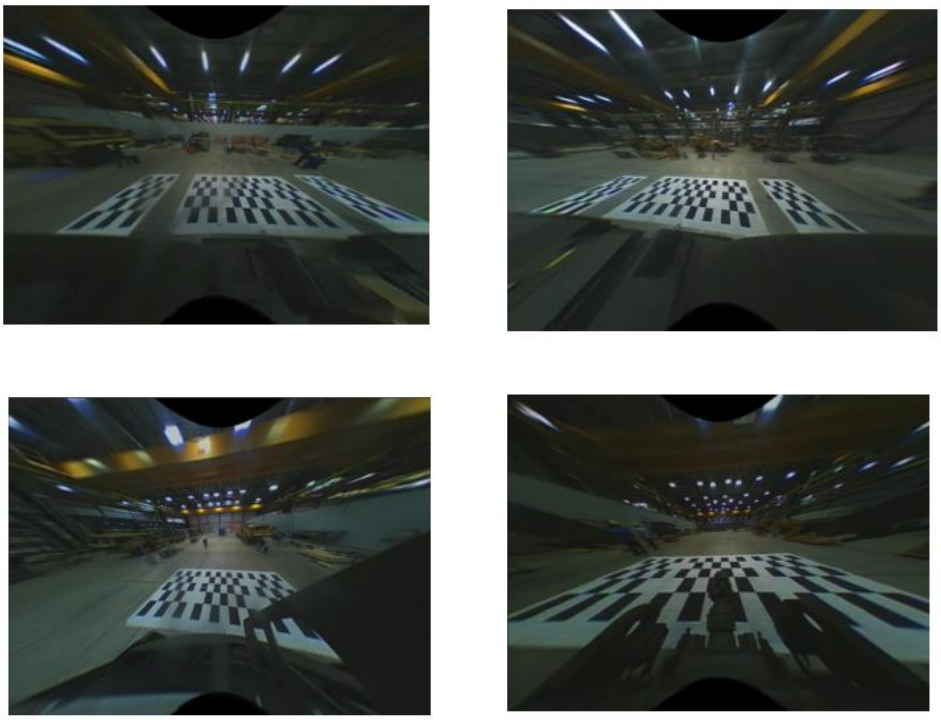


Figure 5.19. Calibrated Image Examples

5.2.5. Pixel Manipulation And Image Warping

The next sections is take actions with different procedures used to warp the image. Below procedured done to the image to produce a surround view image system.

Input Images → Undistortion → Calibration → Warping → Stitching → Output Image

After calibration of the image, fish-eye lens frames are fixed and give out put of calibrated image frame. Calibrated images perspective visions are selected and warped due to surround view requires 4 different image and stitching. Warping with the image needs the 4 points due to take the frame and warp to destination points.

Code 3.1 Example code to perform a simple warping

```
cv::Point2f srcPointsF[] = {Point(x1,y1),Point(x2,y2),Point(x3,y3),Point(x4,y4)};

cv::Point2f dstPointsF[] = {Point(z1,x1),Point(z2,x2),Point(z3,x3),Point(z4,x4)};

Mat F = getPerspectiveTransform(srcPointsF, dstPointsF);
```

getPerspectiveTransform calculates a perspective transform from four pairs of the corresponding points. The function calculates the 3 x 3 matrix of a perspective transform so that:

$$\begin{bmatrix} t_i x'_i \\ t_i y'_i \\ t_i \end{bmatrix} = \texttt{map\_matrix} \cdot \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}$$

$$dst(i) = (x'_i, y'_i), src(i) = (x_i, y_i), i = 0, 1, 2, 3$$

Parameters

src	Coordinates of quadrangle vertices in the source image.
dst	Coordinates of the corresponding quadrangle vertices in the destination image.
solveMethod	method passed to cv::solve (DecompTypes)

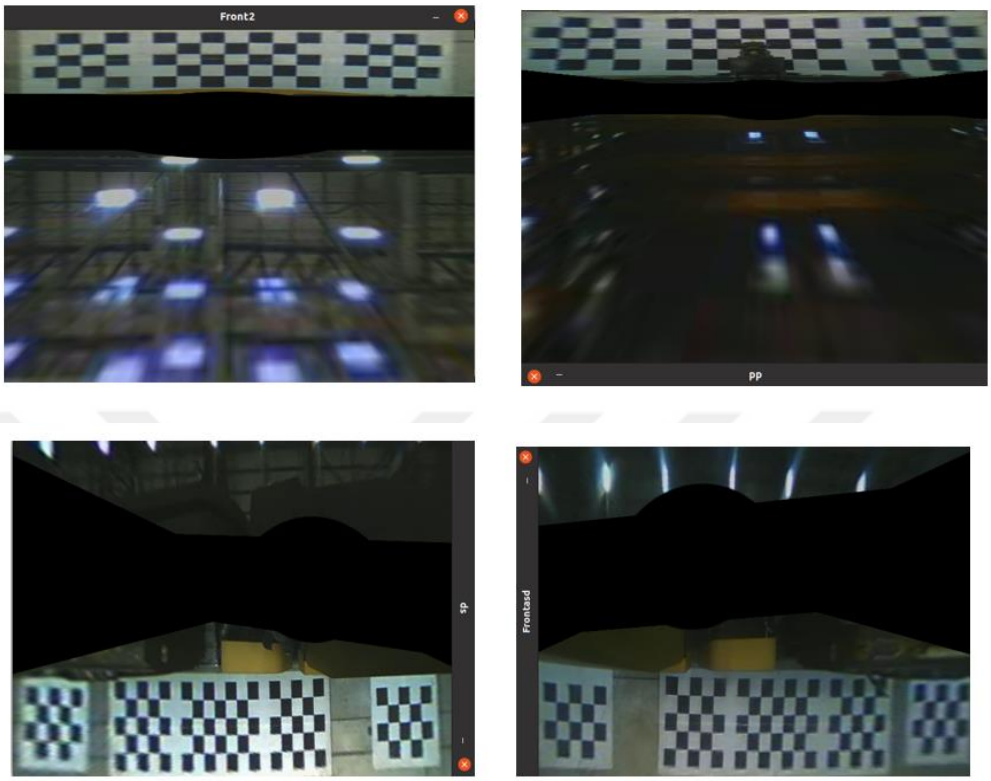


Figure 5.20. Warp Perspective Image Examples

Next step creating a black mask with same size of the images that recorded. And use a bitwise operator to stitch the image together.

Code 4.1 Example code to perform a simple masking

```
cv::Mat maskR = cv::Mat::zeros(cv::Size(width, height), CV_8U);
    vector<Point> pts =
{Point(x1,y1),Point(x2,y2),Point(x3,y3),Point(x4,y4)};
    fillPoly(maskR,pts,Scalar(255));

cvtColor(maskR, maskR, COLOR_GRAY2BGR);

cv::Mat resR;

bitwise_and(frame,maskR,resR);
```

5.2.6. Experimental Study With Result

To obtain a better result in a uncertaing environment there are several extension methods to apply to frames in order to get better surrounding view result. Another good approach is to use homograph matrix. Unfortunately the results were not good as manual selection . Therefore homography merging could be a better approach to handle surround view.

The modification presented before homography merging is not a ideal stitching function, basically it consists the image in regions with 4 control points for each region. Selected regions error between two pixels increase with distance and the selected regions will have lesser errors between two points.

Next step is mask the image with black frames that created and integrated with the homograph matrix via bitwise_and operator. If a point has a value in frame

it keeps the value but if the frame point has no value it will be cropped from the image.

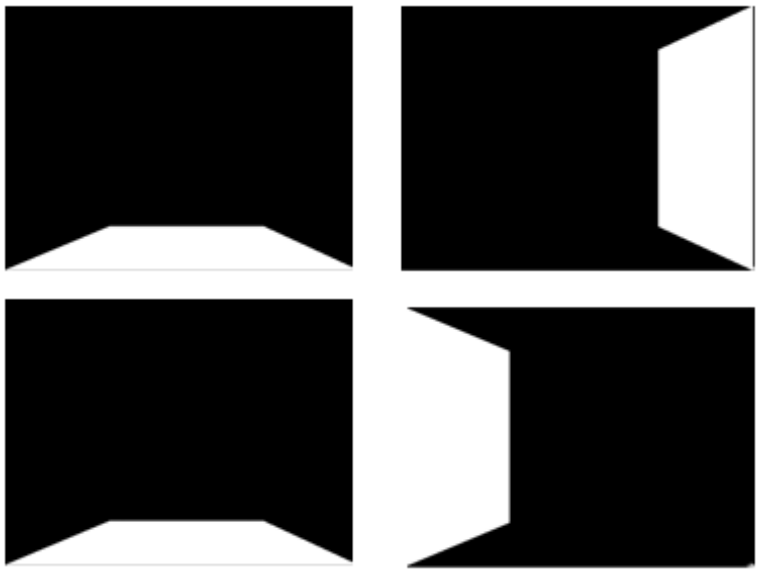


Figure 5.21. Black Mask Image Examples

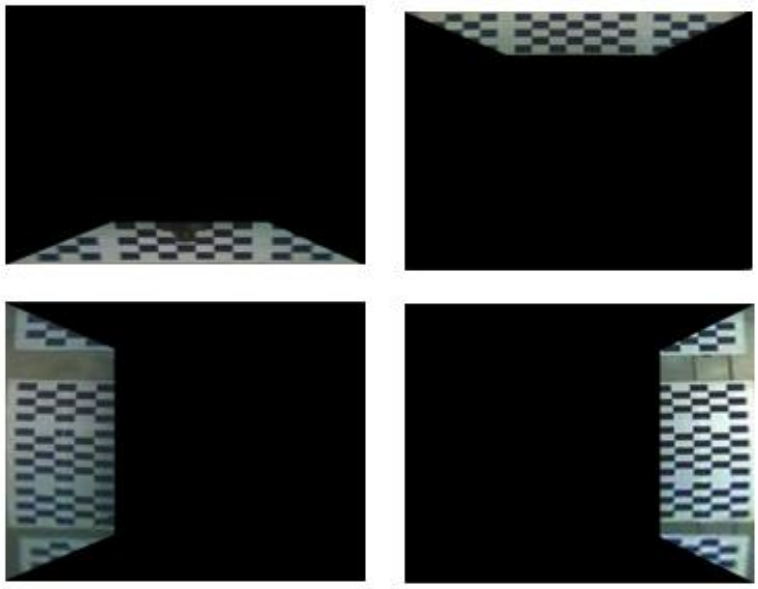


Figure 5.22. Bitwise Operator Iml Image Examples

Last step is again sum up all the images to one total cv: map and resize the image due to car size in real life therefore we get;

Figure 5.23. CV Mat sum up Image Examples

As well as shown up side 360 degree surround view is acquired but the cross sections not so well. There fore warping image pixels are changed due to extensions and depth are changed with source points has changed.

Lastly we get the final mat where all of the 360 degree surround view is achieved.

Figure 5.24. Pixel Manipulated cv mat Images

Figure 5.25. Final cv mat surround view

5.2.7. Blind/Referenceless Image Spatial Quality Evaluator(BRISQUE)

Original Surround View is blured and pixels are manipulated due to better visuality. Gaus blur and noise added to final frame on purpose. The corner cuts are highly observable at first frame without any modification and this is called as an distraction opposite of the purpose this study is concerning. Added blur giving a better visual effect for a complete one full frame. The differences between original frame and modifications are calculated with brisque technique.

Brisque is a No-Reference IQA Metrics called Blind/Referenceless Image Spatial Quality Evaluator(BRISQUE). Manual pixel calibration method is a new method for surrond view capture therefore there aren't a comparison metrics for base frame. The evaluation for quality requires a no-reference quality metrics. [33]

Brisque is very effective between no-reference metrics. It detects the distortions(blur,noise,watermarking,color transformations, geometrics transformations...) between natural image and distorted image which an natural image has no post processing.

Quality is a subjective matter and there are algorithms about bad and good quality. IQA depends on several humans opinions and assign the image a mean score between 0(good) to 100(bad). This is called Mean Quality Score in academic literature.[34]

Figure 5.26. TID2008 Image Quality Score Scaling (0 to 100) : lesser the score, better the subjective quality.

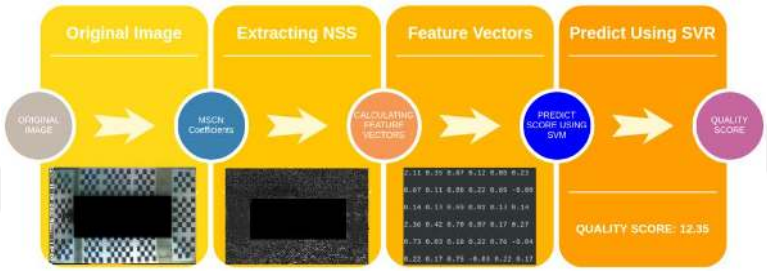


Figure 5.27. Brisque calculation Steps

Step 1: Extract Natural Scene Statistics (NSS)

Natural images have a different pixel intensity distribution than deformed images. When we calculate the distribution over the pixel intensities after normalizing them, the difference between the distributions becomes much more obvious. Particularly, pixel intensities of natural images after normalization adhere to a Gaussian Distribution (Bell Curve), whereas those of artificial or distorted images do not. The amount of distortion in the image is consequently measured by the distribution's departure from the ideal bell curve.

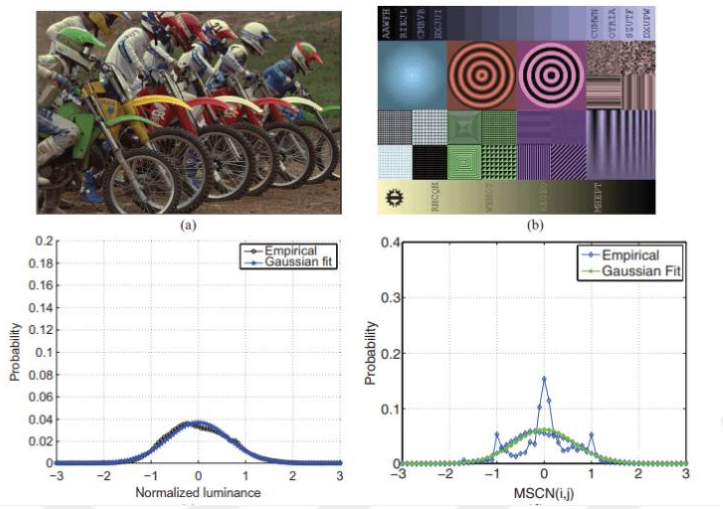


Figure 5.28. On left: shows a natural image with no artificial effects added, fits gaussian distribution. On right: An artificial image, doesn't fit the same distribution well.

Mean Substracted Contrast Normalization (MSCN)

There are several methods for normalizing an image. Mean Subtracted Contrast Normalization is one such normalization (MSCN). The calculation of MSCN coefficients is shown in the image below.

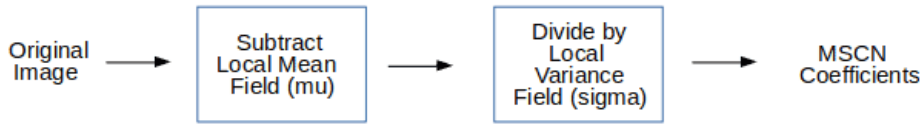


Figure 5.29. Steps to calculate MSCN Coefficients

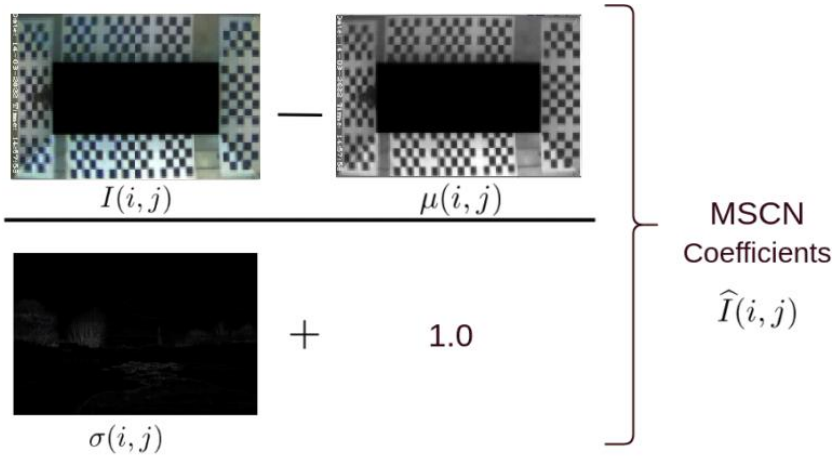


Figure 5. 30. Calculation of MSCN Coefficients

The picture intensity I(i,j) at pixel (i,j) is converted to the luminance I to calculate the MSCN Coefficients (i, j).

$$\{I\}(i,j) = \frac{I(i,j) - \mu(i,j)}{\sigma(i,j) + C}$$

i € 1,2, ...M, j € 1,2, ...N (M and N are height and width respectively). Functions υ (i, j) and σ (i, j) are local mean field and local variance field. Local Mean Field is Gaussian Blur of the original image and Local Variance Field is the Gaussian Blur of the the difference of original image and Local Mean Field.

$$\mu = W * I$$
$$\sigma = \sqrt{W * (I - \mu)^2}$$

W is the Gaussian Blur Window function.

The pairwise product for the MSCN coefficients is determined using the following four orientations: horizontal (H), vertical (V), left-diagonal (D1), and right-diagonal (RD) (D2).

$$H(i,j) = \{I\}(i,j)\,\{I\}(i,j+1)$$
$$V(i,j) = \{I\}(i,j)\,\{I\}(i+1,j)$$
$$D1(i,j) = \{I\}(i,j)\,\{I\}(i+1,j+1)$$
$$D2(i,j) = I(i,j)I(i+1,j-1)$$

Step 2 : Calculate Feature Vectors

The first two elements of the 361 feature vector are calculated by fitting the MSCN image to a Generalized Gaussian Distribution (GGD). Two parameters make up a GGD: one for shape and one for variance.

Next, each of the four paired product images is fitted with an asymmetric generalized distribution (AGGD). Asymmetric Generalized Gaussian Fitting is known as AGGD (GGD). Shape, mean, left variance, and right variance are its four parameters. We get 16 values because there are 4 paired product images.

The feature vector so has 18 components in total. The image is reduced to half its original size before repeating the process to create 18 more numerals, bringing the total to 36.

Table 5.4. Feature Vectors

Feature Range	Feature Description	Procedure
1-2	Shape and Variance	GGD fit to MSCN coefficients
3-6	Shape, Mean, Left Variance, Right Variance	AGGD fit to Horizontal Pairwise Products
7-10	Shape, Mean, Left Variance, Right Variance	AGGD fit to Vertical Pairwise Products
11-14	Shape, Mean, Left Variance, Right Variance	AGGD fit to Diagonal (left) Pairwise Products
15-18	Shape, Mean, Left Variance, Right Variance	AGGD fit to Diagonal (Right) Pairwise Products

Step 3: Prediction of Image Quality Score

A feature vector is created first from a picture. Then, a learning algorithm like Support Vector Machine is fed the feature vectors and outputs (in this example, the quality score) of all the photos in the training dataset (SVM).

By loading the trained model first and estimating the probability using the support vectors generated by the model, LIBSVM can forecast the eventual quality score.

Table 5.5. Brisque Scores of Original and Modified Surrond View

Original Surrond View	Modified Surrond View
24.827	56.108
28.592	46.661
18.911	48.012
25.743	53.881

6. RESULTS

In this chapter the results of smart trailer system with driver behaviour recognition and 360-degree surround view camera explained briefly. A new method fors mart trailer system and a better fix for 360 degree is taken in results. Both method is open to development. In this chapter the reasons for development and the results of the developments is given with how can such a persone achive and might improve the solutions.

6.1 Smart Trailer System With Driver Behaviour Recognitıon

A new driver activity recognition method is provided in this study, and its efficiancy is demonstrated experimentally. This strategy gives an insight by using deep learning-based image classification with YOLO model.

The multi-class driver activity recognition system can be supported with ADAS system's analyzing lane borders and drivers control authority on the car. If driver is unable to focus on the driving task, driver assistance can issue a warning to stop non-driving activities by encouraging the driver to focus on his or her primary task.

YOLO model predictions were not assuring outperforming accuracy in this study but also it needs to be used due to having faster prediction and fps values. It is considered to have fast model with good accuracy, so it is proposed to increase the model performance by time. Toward the integrity of the system, a more advanced model could be achieved by having more data and more complex model architectures could have better results. Another approach is for proposing method is having another model and combining two models for better metric results. Also, an object detection system to detect a cell phone, bottle, cigarette etc. Or face position where driver's head position looking for could be added to the system for better accuracy in the model. A comprehensive classification system requires combining more models with this system.

6.2. 360-Degree Surround View Camera

A new 360 degree surrond view method is provided in this study and method is implied to real world application. Showcase of real world problems are integrated with solutions in this study.

The 360 degree surround view method system can be supported with ADAS system's to show blind spots of the car and give driver much more visible awareness with preventing unexpected accidents.

OpenCV libraries is very good to assure 360 degree surround view due to it's expanded library functions. Various functions are required to provide a surround view such as pixel manipulation, video recording and perspective change etc… OpenCV provide all the functions required for the system. Last results shown with manual pixel manipulation it is granting a good visible 360 surrond view and that was the results we wanted for our surround system. So we can expect the provided method is a real world problem solver.

REFERENCESS

A. Behera, Z. Wharton, A. Keidel and B. Debnath, “Deep CNN, Body Pose and Body-Object Interaction Features for Drivers' Activity Monitoring,” in IEEE Transactions on Intelligent Transportation Systems.

Ananthanarayanan, G., P. Bahl, P. Bodk, K. Chintalapudi, M. Philipose, L. Ravindr anath, and S. Sinha. 2017. Real-time video analytics: The killer app for edge computing. computer 50:58–67. IEEE. doi:10.1109/MC.2017.3641638.

Anish Mittal, Anush Krishna Moorthy, and Alan Conrad Bovik, Fellow, IEEE. No-Reference Image Quality Assesment in the Satial Domain. IEEE Transaction on image processing, Vol. 21. No. 12, December 2012.

Author: Priyadharshini, “What is Machine Learning and How Does It Work?”, *Simplilearn*, September 2021.

C. Rudin, “Stop Explaining Black Box Machine Learning Models for High Stakes Decisions and Use Interpretable Models Instead,” arXiv: 1811.10154v3, 2019.

C. Zhang, R. Li, W. Kim, D. Yoon and P. Patras, “Driver Behavior Recognition via Interwoven Deep Convolutional Neural Nets with Multi-Stream Inputs,” in IEEE Access, vol. 8, pp. 191138-191151, 2020.

Chaoyun Zhang, Ruı lı, Woojın Kım, Daesub Yoon and Paul Patras., “Driver Behaviour Recognition via Interwoven Deep Convolutional Neural Nets with Multi-stream Inputs” 2021

F. Nel and M. Ngxande, “Driver Activity Recognition through Deep Learning,” 2021 Southern African Universities Power Engineering Conference/Robotics and Mechatronics/Pattern Recognition Association of South Africa (SAUPEC/RobMech/PRASA), 2021, pp. 1-6.

H. Bandyopadhyay, “YOLO: Real-Time Object Detection Explained”, *V7,2020*

H. M. Eraqi, Y. Abouelnaga, M. H. Saad, M. N. Moustafa, “Driver Distraction Identification with an Ensemble of Convolutional Neural Networks,” in

Journal of Advanced Transportation, vol. 2019, Article ID 4125865, 12 pages, 2019.

J. C. Stutts, D. W. Reinfurt, L. Staplin, E. A. Rodgman et al., "The role of driver distraction in traffic crashes," 2001.

J. Redmon, S. Divvala, R., R. Girshick, A. Fardhadi, "You Only Look Once: Unified, Real-Time Object Detection" *The Computer Vision Foundation (CVPR), 2016.*

J. Yan, Z. Lei, L. Wen, and S. Z. Li. The fastest deformable part model for object detection. In Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, pages 2497–2504. IEEE, 2014.

K. Lenc and A. Vedaldi. R-cnn minus r. arXiv preprint arXiv:1506.06981, 2015.

Kushashwa Ravi Shrimali 2018. Image Quality Assesment : BRISQUE. LearnOpenCV. Avaible: Image Quality Assessment : BRISQUE | LearnOpenCV

Kwatra, V., A. Schödl, I. Essa, G. Turk, and A. Bobick. 2003. Graphcut textures: Image and video synthesis using graph cuts. ACM Transactions on Graphics (Tog) 7:277–86. doi:10.1145/882262.882264.

L. Calderone "Top Articles from 2019 What is Deep Learning", *Robotics Tomorrow*, February 2020.

L. Yang, K. Dong, Y. Ding, J. Brighton, Z. Zhan and Yifan Zhao, "Recognition of Visual-Related Non-Driving Activities Using a Dual-Camera Monitoring System," Pattern Recognition, vol. 116, 2021, 107955.

L. Zhao, F. Yang, L. Bu, S. Han, G. Zhang and Y. Luo, "Driver Behavior Detection via Adaptive Spatial Attention Mechanism," Advanced Engineering Informatics, vol. 48, 2021, 101280.

M. Kutila, M. Jokela, G. Markkula, and M. R. Rué, "Driver distraction detection with a camera vision system," in Proc. IEEE International Conference on Image Processing, vol. 6, pp. VI–201

M. Leekha, M. Goswami, R. R. Shah, Y. Yin, and R. Zimmermann, "Are you paying attention? detecting distracted driving in real-time," in IEEE International Conference on Multimedia Big Data (BigMM), 2019.

M. Martin, M. Voit and R. Stiefelhagen, "Dynamic Interaction Graphs for Driver Activity Recognition," 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC), 2020, pp. 1-7.

M. Mishra, "Convolutional Neural Networks, Explained", *Towards Data Science,* August2020, Avaible:https://towardsdatascience.com/convolutional-neural-networks-explained-9cc5188c4939

Mo'taz Al-Hami, Raul Casas, Subhieh El Salhi, Sari Awwad & Fairouz Hussein (2021) Real-Time Bird's Eye Surround View System: An Embedded Perspective, Applied Artificial Intelligence, 35:10, 765-781, DOI: 10.1080/08839514.2021.1935587

NSC Injury Facts, "Historical Fatality Trends*", National Safety Council,*2021. Available: https://injuryfacts.nsc.org/motor-vehicle/historical-fatality-trends/deaths-and-rates/

O. Halabi, S. Fawal, E. Almughani and L. Al-Homsi, "Driver Activity Recognition in Virtual Reality Driving Simulation," 2017 8th International Conference on Information and Communication Systems (ICICS), 2017, pp. 111-115.

R. B. Girshick. Fast R-CNN. CoRR, abs/1504.08083, 2015.

R. Kahalaf, Y. Akbulut, "Smart Arms Detection System Using YOLO Algorithm and OpenCV Libraries.", *Turkish Journal of Science & Technology, 2021.*

R. Suttle "The Importance of Safe Driving", ItStillRuns, 2021.

S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. arXiv preprint arXiv:1506.01497, 2015

Shorten, C., Khosgoftaar, T.M. A survey on Image Data Augmentation for Deep Learning. J Big Data 6, 60(2019). Avaible: https://doi.org/10.1186/s40537-019-0197-0

Szeliski, R. 2006. Image alignment and stitching: A tutorial. Foundations and Trends in Computer Graphics and Vision 2:1–104. Now Publishers Inc. doi:10.1561/0600000009.

X. Renjie, L. Haifeng, L. Kangjie, C. Lin, L. Yunfei, “A Forest Fire Detection System Based on Ensemble Learning”, 2021. DO -10.3390/f12020217

Y. Xing, C. Lv, H. Wang, D. Cao, E. Velenis and F. Wang, “Driver Activity Recognition for Intelligent Vehicles: A Deep Learning Approach,” in IEEE Transactions on Vehicular Technology, vol. 68, no. 6, pp. 5379-5390, June 2019.