BSCS-S24-003

03-134211-016 KHALID HAMEED
03-134202-089 WAJEEHA QURBAN

AutoMind: Tyre Health Monitoring and Recommendation System

In partial fulfilment of the requirements for the degree of
Bachelor of Science in Computer Science

Supervisor: DAWOOD AKRAM

Department of Computer Sciences
Bahria University, Lahore Campus
January 2025

C e r t i f i c a t e

Bahria University

Lahore Campus

We accept the work contained in the report titled
"Automind: Tyre Health Monitoring and Recommendation System"
written by
KHALID HAMEED
WAJEEHA QURBAN

as a confirmation to the required standard for the partial fulfilment of the degree of Bachelor of Science in Computer Science.

Approved by:

Supervisor: Dawood Akram

______________________________ (Signature)

January 5, 2025

DECLARATION

I hereby declare that this project report is based on my original work except for citations and quotations which have been duly acknowledged. I also declare that it has not been previously and concurrently submitted for any other degree or award at Bahria University or other institutions.

Enrolment	Name	Signature
03-134211-052	KHALID HAMEED	
03-134202-089	WAJEEHA QURBAN	

Date : January 5, 2025

Specially dedicated to
my supervisor, parents, and my late brother
(KHALID HAMEED)
Specially dedicated to
my parents, supervisor and teachers
(WAJEEHA QURBAN)

ACKNOWLEDGEMENTS

We would like to thank everyone who had contributed to the successful completion of this project. We would like to express our gratitude to our research supervisor, Dawood Akram for his invaluable advice, guidance and his enormous patience throughout the development of the research.

In addition, We would also like to express our gratitude to our loving parent and friends who had helped and given us encouragement.

Khalid Hameed
Wajeeha Qurban

AutoMind: Tyre Health Monitoring and Recommendation System

ABSTRACT

Road safety and operational efficiency depends on vehicle tire health. Accidents, increased maintenance cost and poor fuel efficiency are some of the effects of poor tire condition. However, the traditional tire inspection methods are usually time consuming and need expert knowledge. In response to this need, this project develops a Tire Health Monitoring System powered by deep learning techniques. To classify tire health based on images provided by users, the system uses *MobileNetV2*, a lightweight model famous for its computational efficiency.

The system comprises a mobile application for users to upload tire images and enter tire serial numbers. The trained deep learning model analyzes these images to detect the tire condition in one of the four categories, excellent, good, poor, and cracked. A backend system supporting this functionality manages tire data, tracks health trends and issues notifications to users when maintenance or tire replacement is required.

The dataset of tire images was trained using a model, fine tuned and early stopped to limit overfitting and generalize the model for different size tires. The evaluation of the model used in this thesis is presented: evaluation metrics of accuracy, precision, recall, F1 score, and confusion matrix present accuracy of 94%. The results show that the system can supply accurate and reliable tire health assessments.

The Tire Health Monitoring System described in this report provides a proactive approach to tire maintenance, thereby reducing the risk of tire related accidents, minimizing downtime and optimizing vehicle performance. Additionally, the system prolongs tire life and enhances fuel efficiency, demonstrating considerable environmental and cost saving benefits. This project is an example of integrating machine learning with mobile technology which has the potential to change the way of vehicle safety and maintenance in the future, and offers a path for the future development of more advanced solutions in the automotive industry.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS / ABBREVIATIONS

TL: Transfer Learning

TPMS: Tire Pressure Monitoring System

CV: Computer Vision

CNN: Convulation Neural Network

ANN: Artificial Neural Network

GAN: Generative Adversarial Network

ML: Machine Learning

IoT: Internet of Things

PWAs: Progressive Web Apps

OCR: Optical Character Recognition

α: Learning Rate

CHAPTER 1

INTRODUCTION

1.1 Background

The importance of skid resistance cannot be over emphasized, tyres are a very important part of any vehicle. The reason being, tires actually often never get the kind of attention they deserve, putting drivers at the risk of blowouts, bad traction, increased fuel consumption and all. The best way to avoid these problems, to optimize the use of the vehicle to the benefit of both performance and safety, and to comply with regulations, is to regularly monitor and maintain tire health; in recent years, however, advances in technology, computer vision, and machine learning in particular have made it possible assess tire health with new approaches. Nevertheless, existing solutions are either too reliant on manual inspection or necessitate expensive and bespoke equipment that are unavailable to even the average consumers.

To tackle these problems, AutoMind-Tire Health Monitoring System has been made. With two unique features this system allows the vehicle owner to easily monitor his tire's health. The first of those features allows users to enter the tire's serial number either manually or through the submission of an image of the serial number. The second feature provide the Tire Health Monitoring System with the ability to inspect the tires and obtain their condition. Users can simply upload images of their tires, which the system will analyze for evidence of wear or damage — cracks, bald spots, or uneven tread patterns, for example. Users can quickly see if their tires need attention. These features are then integrated into a React Native mobile application as a seamless and user experience. It also has a web portal admin panel that allows the admins to manage products, notifications and other system configuration and result in the platform running smoothly.

1.2 Problem Statements

Tyres are a crucial element of vehicle safety and performance, and too often they're overlooked during routine vehicle maintenance. Problems related to poor tire health can be quite hazardous: blowouts damage rims and threaten others on the road; poor traction endangers the driver and other commuters; greater fuel consumption increases the price tag. Bringing the tires up to shape as well as ensuring efficiency and safety is why regular inspections are essential. The problem is, however, that tracking and inspecting tire health is often inconvenient and time consuming. The goal of this project is to simplify tire health monitoring by providing an efficient, user friendly solution for vehicle owners. In this project, users can either enter a tire's serial number or upload an image of the tire for analysis. Using advanced image processing techniques, the mobile application assesses the condition of the tire and allows users to detect wear, damage and other possible problems in the tire. This project gives users an excellent way to watch tire health and take proactive measures in preventing vehicle safety issues from happening.

1.3 Aims and Objectives:

The main goals of this project include:

- Developing a mobile application for tire health monitoring, providing an intuitive interface for real-time tire condition evaluation
- Allowing users to input or upload tire serial numbers, offering users valuable insights into tire specifications and characteristics.
- Implementing image analysis technology to detect tire wear, damage, and irregularities, providing users with accurate tire health insights.
- providing personalized tire maintenance recommendations based on tire health assessments.
- Suggesting suitable products, such as replacement tires or maintenance services, to improve tire longevity and safety.

- Developing a web portal as an admin panel to manage tire products, notifications, and system configurations.
- Through achieving these goals this project seeks to enhance vehicle safety and performance, enabling users to be proactive in caring for tires health to reduce risk of hazards.

1.4 Scope of Project

The goal of the 'AutoMind' project is to develop a mobile application that forms a complete Tire Health Monitoring System. Users of the mobile app can enter or upload a tire’s serial number to retrieve detailed information and upload images to check tire conditions, identifying wear and damage. Timely intervention, which makes for better safety and performance is then assured. The web portal is also an admin panel from where an administrator can manage tire product listings, edit details, configure notifications. This project provides a seamless way to monitor and handle tire health in combination with these two platforms.

CHAPTER 2

LITERATURE REVIEW

2.1 Tire Health Monitoring Systems: Overview

Tire Health Monitoring Systems (THMS) are intended to continuously monitor tires in vehicles. These systems utilize a variety of sensors and technologies to deliver real time data on everything from tire pressure, temperature, tread wear, etc. to ensure tires are safe, performing, and long lasting. Early detection of problems via these systems reduce accident possibilities and elevate vehicle's efficiency.

2.1.1 Existing Tire Monitoring Technologies

The tire monitoring technology has a close relationship with the vehicle safety and the performance. Tire pressure has a significant influence on the driving safety of road vehicles; therefore, it is mandatory in many countries to equip all new road vehicles with a tire pressure monitoring system (TPMS). There are two types of TPMSs in use: the direct TPMS (dTPMS) and the indirect TPMS (iTPMS), both of which have made significant improvement in the last decade [1].

Besides TPMS, other tyre related sensors (TTS, tire temperature sensors) are often deployed in high performance or commercial vehicles. The most important are these sensors that would detect overheating and blowing out (and hence prevent safety hazard). On the tire side, sensors track tread depth and alert the driver when it's not safe to drive — especially during rainy weather.

Smart tires are another emerging technology, which combines sensors to measure in real time tire pressure, temperature, load and road conditions. This data can be used to optimize tire performance and improve safety.

In commercial vehicles especially, acoustic and vibration monitoring is used to detect internal damage or irregular wear through the use of ultrasonic sensors or accelerometers. Also, use cameras and computer vision inspection visual systems to identify cracks or punctures as another monitoring layer. To get a more complete view of tire health, however, these systems are increasingly integrated into vehicles.

2.1.2 Limitations of Traditional Systems

Currently, there are many limitations of traditional tire monitoring systems such as TPMS. They primarily pay attention to only tire pressure and temperature and miss out on many of other more important things such as tread wear or structural damage. This leads to an incomplete estimation of tire health and could leave trouble not found.

The fault lies with these systems as well, especially since they are neither automatic in nature, nor do they check themselves automatically, rather requiring periodic manual checks or inspections, thereby increasing the time and the error. Continuous monitoring is required (and can be difficult) as users can easily miss critical tire issues which might lead to safety compromises.

Moreover, the existing systems are not portable and are difficult to take along or link with mobile devices. Consequently, users find it difficult to monitor their tire health on the go or receive real time update of their tires, making convenience and safety impossible. Tire pressure monitoring systems usually cost more than traditional tire gauges [2].

Although most traditional systems offer basic alerts, they lack real time visual feedback and require detailed insight into an individual's tire condition, like its wear pattern or cracks. They cannot detect the problems before they end up as failures.

Traditional systems finally suffer from the absence of centralized monitoring, making monitoring of tire health across several vehicles extremely difficult for the fleet operators, which causes inefficiencies. The limitations of these solutions further motivate the development of more advanced, automated, and mobile integrated tire monitoring solutions.

2.1.3 Need for Advanced Solutions

The accurate detection of tire tread wear plays an important role in preventing tire-related accidents[3]. Traditional tire monitoring systems have some limitations, and the need is for a more advanced and a more complete solution. Besides the basic vehicle related solutions such as tire pressure monitoring systems, these solutions should provide a holistic view of tire that covers all tire parameters. A major area of improvement is the integration of real time condition monitoring: sensors and image recognition allows us to assess tire pressure as well as patterns of wear, cracks, and tread depth, which can be utilized to identify problems much earlier than is currently the case.

Another important piece is automation. Traditional systems need to be intervened manually or needed to be kept on check, which can create problems and greater risks. With a highly technological system, it could purely monitor tire health on a continuous loop and in real time alert the driver or vehicle owner in the event abnormality was detected, hence eliminating the component of human error and making the roads much safer.

Growing demand for mobility and convenience in vehicle maintenance also exists. A mobile application enables vehicle owners to monitor tire health easily and to be aware in a more proactive way. Such systems with integration into mobile devices can convey onthe go notifications, images of tires' conditions and even recommends about tire maintenance or replacement.

Last is the use of data analytics and machine learning for tire health monitoring. Predictive analytics can predict tire wear, optimize maintenance schedule and can even

cut costs by identifying problems ahead of time. These are advanced solutions of what will become future solutions for monitoring tire health, to add more safety, convenience and efficiency to personal users and fleet managers.
Several image based tire inspection technologies have been developed.

2.2 Image-Based Tire Inspection Technologies

Tire defect such as tread wear, cracks, bulges are detected by image based tire inspection through visual data and machine learning algorithms.

2.2.1 Role of Image Processing & Computer Vision

Computer vision (CV) and image processing are two closely related fields that utilize techniques from artificial intelligence (AI) and pattern recognition to derive meaningful information from images like tyres [4].These technologies employ high resolution images taken with smartphones or specialized cameras and can show subtle indications of tire wear, damage and abnormalities often neglected on manual inspection or more conventional methods.

Manipulation and enhancement of images by means of techniques such as digital filtering to enhance key areas, e.g. cracks, punctures, uneven tread wear, image segmentation masking out major areas of interest, and image enhancement to improve contrast facilitate automated system analysis of tire health. For example, algorithms can determine wear patterns by evaluating the tread depth and surface condition against a pre defined model determining healthy tires.

However, computer vision goes one further, teaching the system to not only process the images, but also understand and interpret the content within them. Computer vision models are capable of classifying types of damage, evaluating tire quality, or even detecting when something like misalignment or tire pressure problems are beginning.

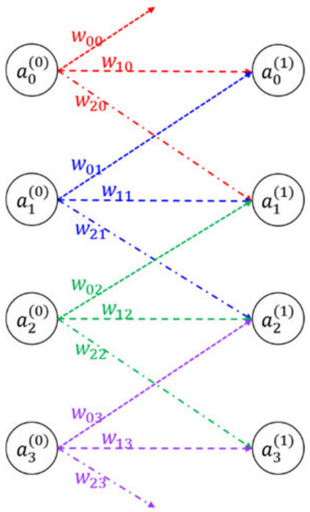
Additionally, these technologies support non invasive monitoring: the user can easily take a picture of their tire, and the system would provide instant feedback. This enables tire maintenance to become more efficient and user friendly to vehicle owners with a reliable, on demand solution for tire health assessment on the vehicle owners' own mobile device.

Using image processing and computer vision, modern tire health monitoring systems can evolve from periodic, manual checks, to continuous, automated monitoring, improving predictive maintenance and improving vehicle safety.

2.2.2 Machine Learning for Tire Damage Detection

Machine learning (ML) has been used as an important component in modern tire health monitoring systems in the detection of tire damage and wear, automatically and efficiently. Tire damage detection using different machine learning models is applied especially on the image based tire inspection system. Consequently, deep learning, in particular the Convolutional Neural Networks (CNNs), has been found useful for defects identification in the form of cracks, punctures, uneven wear, or any other visible type of the damage in the tires.

In recent years, CNNs have shown great success for image processing tasks because of their ability to automatically learn and implicitly extract hierarchical features directly from raw images. CNN, commonly known as ConvNet, is one of the common types of Artificial Neural Network (ANN) that comes under the supervised method category [5]. CNN's can be trained to detect tire damage from images in tire health monitoring, decreasing the need of manual inspection. These can process massive volumes of data and detect the patterns in tire images with surprising accuracy. A CNN works by passing through multiple convolutional layers, which apply filters to input image then pool layers with a goal of reducing dimensionality finally fully connected layers that make the prediction based on learned features. The core formula for a convolutional operation can be expressed as:



$$a_1^{(1)} = \sigma(w_{20}\, a_0^{(0)} + w_{11}\, a_1^{(0)} + w_{02}\, a_2^{(0)} + b_1)$$

$$a_2^{(1)} = \sigma(w_{21}\, a_1^{(0)} + w_{12}\, a_2^{(0)} + w_{03}\, a_3^{(0)} + b_2)$$

$$a_i^{(l)} = \sigma\left(\sum_{m=0}^{N_m-1} w_{m,k-m}\, a_{k-m}^{(l-1)} + b_i\right)$$

$$\begin{cases} k = i + \dfrac{(N_m - 1)}{2} \\ w_{m,k-m} = 0 \; if \; k - m < 0 \; or \; k - m \geq N \end{cases}$$

Figure 2.1: CNN Formula

Generative Adversarial Networks (GANs) are another deep learning technique being researched related to tire damage detection. GANs are built out of two networks; a generator, and a discriminator, trained together to play the adversarial game. The generator is able to generate synthetic images of tire defects, and the discriminator is trying to figure out whether they are real or fake. It aids the improvement of the quality and the diversity of training data, especially if you are limited in your real world data. In recent years, transfer learning has taken the spotlight in cases with limited labeled data. Traditionally, transfer learning can be summarized as applying a pre trained model (trained on large datasets like ImageNet) and fine tuning that model for a certain task (e.g., detecting tire damages). The goal of this project is to adapt models commonly used for image classification, e.g. *VGG16, ResNet, MobileNet, Inceptio*n, for transfer learning, to detect specific tire defects. The process typically involves two main steps:

- **Feature Extraction:** As fixed feature extractors, we use the initial layers of the pre-trained model recognizing general features like edges, textures and patterns. In these layers, we train the frozen layers and run the output through a new classifier layer trained on the tire damage dataset.

- **Fine-Tuning:** Previous layers using pre-trained model are fine tuned on tire specific data after feature extration. It lets the model fix itself and focus on tire specific features like cracks or uneven tire wear.

What we like about using pre trained models is that they have already learned some useful representation from a lot of data that can be transferred to a new task with a lot less data. These models require fewer labeled examples to fine tune them from which the process is faster and more efficient. For example, *ResNet*, with its deep residual learning architecture, can be fine tuned for detecting more complex tire defects such as persistent bead regions of tire valve bumps or mere tire gouges by only modifying deeper layers and leaving the lower layers unchanged. MobileNet, tailored for efficient computation, is also a suitable network for real time tire monitoring needs especially in resource constrained environments.

Table 2.1: CNN,GAN,Transfer Learning Comparison

Feature	CNN	GAN	Transfer Learning
Primary Use	Image classification, detection	Image generation, data augmentation	Fine-tuning pre-trained models for new tasks
Architecture	Convolutional layers, pooling, dense layers	Generator & Discriminator networks	Pre-trained model adapted for new tasks (e.g., MobileNet, ResNet)
Input	Images or grid data	Latent vectors or existing images	Task-specific data (images, text, etc.)
Key Function	Feature extraction for image tasks	Generate new data (e.g., images)	Transfer knowledge from one task to another via fine-tuning
Training	Supervised learning (labeled data)	Unsupervised (adversarial feedback)	Fine-tuning pre-trained models on smaller datasets

Data Requirements	Requires large labeled datasets	Requires paired data (generator & discriminator)	Works well with limited data by using pre-trained weights
Use Cases	Image recognition, medical analysis	Deepfake generation, style transfer	Speed up training on small datasets (e.g., MobileNet, VGG)
Example Models	- AlexNet, VGG, ResNet	- DCGAN, CycleGAN	- MobileNet, InceptionV3, ResNet50, EfficientNet

In conclusion, these machine learning techniques when combined, improve the capability of the existing tire health monitoring systems to detect tire damage with a higher degree of accuracy and lesser need for computational efforts. As our ML models become more advanced, and transfer learning training techniques advance, automated tire inspection is becoming a viable alternative to the traditional manual methods. Using the methods like transfer learning and the availability of pre trained models like *VGG 16, ResNet* and *MobileNet*, the system attains extreme accuracy despite less data thus forming as a powerful tool for real time tire health monitoring.

2.2.3 Limitations of Machine Learning models

However, those ML models each come with its own limitations. Image based tire inspection is one area where convolutional neural networks (CNNs) are particularly effective but they are data hungry. Labeled images of tire defects are hard to acquire in practice and come at significant cost, making the training of CNNs difficult, especially where data availability is limited. Moreover, CNNs are expensive computationally, needing many hardware resources, especially when deal with high resolution images or realtime data. Due to its high sensitivity to variations in image quality including the lighting conditions, image resolutions and image angles, the performance of these networks can be degraded in various environments. Although limited, CNN is still the leading choice in image based defect detection in tire health monitoring.

To solve this problem of limited data, we train on synthetic tire defects generated with Generative Adversarial Networks (GANs). As a result, they are not without their challenges. Training of GANs has turned out to be one of the main limitations. GANs are adversarial which results in the generator and the discriminator acting against each other which means a lot of instability in the training and if not managed properly the model performance is poor. Moreover, GANs are memory and computationally expensive to train, and therefore infeasible in resource constrained deployment environments. GAN generate synthetic data to provide fore the dataset but there is no guaranty that these generated images are the right way to represent the true variability of real world tire defects and this can result in a model that is not robust when is applied with new data.

Even though Decision Trees are simple and interpretable, the approach has issues as well in tire health monitoring. However, their overfitting problems are accentuated in the case of noisy data or high variance. In overfitting, the model is overtraining on noise or anomalies in the training data and thus is unable to generalize to new data. Furthermore, Decision Trees are not suitable for data with complex structure, high dimensionality, e.g., images, for which complex patterns and features are essential for correct prediction.

TL is a popular technique for tire health monitoring which exploits the fact that labeled data may not be available. Pre trained models pretrained on large datasets such as ImageNet are used by using TL and the same is used by fine tuning these models to solve a particular task like tire damage detection. Although TL dramatically cuts the quantity of labeled data required, it is accompanied by its own set of caveats. The structure of the fine tuning process is still highly dependent on the quality and relevance of the pre trained model. The hope is that the transfer does not suffer too much due to the original model not being trained sufficiently enough on similar data to tire images. Furthermore, TL models can be computationally expensive and demand significant resources during their pre training phase—and when fine tuning, not always a possibility for real time applications.

2.3 Serial Number Recognition for Tire Data

Tire images are used to extract serial number recognition to track and manage tire data by recognizing unique identifiers on tires to guarantee accurate inventory and maintenance records.

2.3.1 Importance of Tire Serial Numbers

The tire serial number, in addition to being necessary for tracking the history of a tire (and what might it be made of and if it might have been recalled), is used to identify a specific tire. It can be used to check if the tire has been part of a recall for manufacturing defects or safety issues. For example, the serial number provides an easy way to know what tire batches the manufacturer wants flagged for safety reasons, like the replacement of these tires. It is very important to ensure driver safety and preventing from the accidents triggered by defective tyres.

Additionally, the serial number is a tool used to track tire lifespan, and inventory management. The serial number is recorded and analyzed by vehicle owners, fleet managers and tire retailers to monitor the age of their tires, schedule tire replacements and recognize when a tire reaches the end of it's effective life span. Such a proactive approach means that we reduce the chance of tire incidents, increase the vehicle performance, and improve the resource management. Serial number recognition in tire health monitoring systems helps to ensure that correct tire data is associated with visual inspections and it becomes easier to recommend the correct maintenance actions based on accurate and up to date information.

2.3.2 OCR in Tire Data Extraction

As part of the tire health monitoring system, Optical Character Recognition (OCR) is an important mechanism for extracting tire serial number details. OCR [technology] automatically recognizes and converts printed text or numbers on an image into machine readable data. OCR is mainly used in tire health monitoring to read out the essential tire info like the serial number, manufacturer and specification from the image or photograph of the tire. It means no more data entry by hand thus the improved efficiency on the whole tire management process.

A few typical OCR techniques used to extract data from tire includes its advantages and applications. Traditional OCR technology, like template matching and feature based recognition, match an acquired character with pre defined templates or features in order to recognize a character. In complex or noisy images these methods are less effective and therefore in recent years more advanced approaches have become popular.

Today, modern OCR techniques are based on machine learning and deep learning models (e.g. Convolutional Neural Networks (CNNs)) that can automatically extract and learn features from images as a whole, and hence increase the accuracy and robustn.. We train these models to recognise text in different fonts, orientations, and conditions, and so are suitable for a variety of tyre labelling scenarios. Recurrent neural networks (RNNs) is another advanced technique used for sequence recognition aiding us to understand and extract text from images in which characters are close to each other or skewed.

Several cloud platforms offer OCR service via API to make it easier for developers to integrate the OCR service. Some popular OCR APIs include:

- **Google Cloud Vision API:** Google Cloud's Vision AI suite of tools combines computer vision with other technologies to understand and analyze video and easily integrate vision detection features within applications, including image labeling, face and landmark detection, optical character recognition (OCR), and tagging of explicit content [7].

- **Microsoft Azure Computer Vision API:** Microsoft’s OCR tool is one of many components of Azure Cognitive Services, and it does text extraction quite well. It can detect printed text in images and PDFs, text in different languages, as well as multi page documents. An additional benefit of the API is that it includes handwriting recognition, which is necessary to read tires bearing handwritten labels.

- **Amazon Textract:** Amazon Textract helps you add document text detection and analysis to your applications. Textract doesn't just scrape text from the document via OCR, but understands the structure of the document, extracting not only text but the relationships and key data points between fields. Systems that require extractions of structured data such as tire serial numbers and their related attributes find this useful.

- **Tesseract OCR:** Tesseract is an open source optical character recognition (OCR) platform.Tesseract supports over 100 languages and is very easy to integrate into the applications that extract tire data. It can process printed or handwritten text and as an open source it can be tuned to a specific use case by developers.

- **IBM Watson Visual Recognition:** The IBM Watson Visual Recognition service enables your applications to turn images into actionable data [6]. There is also their OCR which is part of their Visual Recognition API. This is especially useful in dealing with images that have many various text types, or are using nonstandard fonts (which is very common on tire labels).

All these OCR solutions make use of machine learning and deep learning to reach high accuracy text extraction. The choice of an OCR tool to use is based on many different parameters like the image quality, the processing speed required, language support and the easiness of integration with the tire health monitoring system. Integrating these OCR solutions into the monitoring system allows users to quickly extract vital tire data eliminating the need to do it manually and improving the accuracy of tire tracking and maintenance.

2.4 Mobile Applications in Vehicle Maintenance

Technology has advanced and mobile applications are used for real time monitoring, diagnosis and status of the vehicle to enhance efficiency and safety during vehicle maintenance.

2.4.1 User Experience & Usability

For mobile application for tire health monitoring development, developers can create it as a native, cross platform or a web app. Native apps are native to iOS and Android, known as iOS native apps and Android native apps respectively, and are separate apps built for each platform. These perform well, and they perform well on their respective phones, but and have features specific to each platform. But they're costly to develop, and to maintain, because there's separate codebases to develop for iOS and Android. React Native and Flutter are cross platform frameworks that enable you to build your app with one codebase that the app is supported across both platforms, cutting down on development time and resources. Written in JavaScript, React Native offers near native performance while Flutter, written in Dart gives you smooth animations and great performance. The cost and time savings when compared to native development are huge for both options. In contrast, Progressive Web Apps (PWAs) are lightweight, though less featureful and less performant counterparts. React Native is a cross platform solution that is kind of middle point between cost, development time and performance which enables the monitoring of tire health to a much wider set of user base.

2.4.2 Integration with DL models

While integrating DL models with mobile platforms like React Native usually means offloading model inference to cloud platforms, such as AWS, Google Cloud or Microsoft Azure, it may be possible to run the models locally on react native. These platforms have the computational power needed to run complex models and shouldn't tax mobile devices. Frameworks like Flask or FastAPI can be used to serve models via

APIs (RESTful APIs whereby the React Native app sends data i.e. tire images to the server and then the server sends back predictions).

Cloud based solutions ease the load on mobile devices, but face challenges related to network latency which can introduce latency in predictions. In order to run, a stable internet connection is required and bad connectivity can affect the real time performance in a negative way. Plus, it can be complicated to manage cloud infrastructure, especially if you don't understand how it should be scaled to manage fluctuations in demand from your users.

Deploying model based upon cloud platforms such as Flask or FastAPI is efficient; however, latency, connectivity, and infrastructure concerns are responsible to fantastic performance.

2.5 Challenges in Implementing Tire Health Systems

There are various challenges in implementing Tire Health Systems In real time, following are some of the challenges:

2.5.1 Technical Challenges in Image-Based Monitoring

One of the main problems in developing an image based tire health monitoring system is in the selection of the dataset. However, such training and evaluation of the machine learning models within all domains can only be achieved if large, diverse and, well labeled datasets exist. Yet, acquiring high quality datasets spanning various tire conditions (e.g. different wear patterns, damage types and tire brands) is difficult to do. However due to the small volume of many datasets or not enough diversity, we are still not able to train models that are practical for real scenarios.

The second challenge is how to determine the right machine learning model needed. However, due to the existence of a large variety of models, like CNNs, GANs and transfer learning based models, it becomes difficult to choose the right model to fulfill the specific demands of tire health monitoring. Others are factors such as model complexity, computational resources, and the nature of their tire data (such as what the image resolution is, what types of damage occur, etc.). Even after having chosed a model, one is continaously fine tuning it so that it becomes optimal and does not over fit.

Limits of Resource, especially when we consider computational power, present even greater challenges. Deep learning models are trained by using a large amount GPUs and high performance servers, which are not always available. When experimenting with different models and hyperparameters, longer training times and increased costs can result from resource issues.

There's yet another hurdle, model integration. However, integrating the best model into a seamless workflow with mobile applications and cloud based platforms can be technically complex. This means that the model has to work effectively on real time data from the app; including processing images from users, providing feedback, etc, all without any delays; and all of this has to be ensured while fitting in APIs, cloud services and interfaces from the front end.

Last but not least, keeping the accuracy of the system up to date is very hard. However, the model constantly needs to be monitored and retrained since tire wear pattern and damage type vary over time and across regions. Moreover, this process becomes complicated by the fact that data sources are being incorporated, such as user uploaded images of different tire conditions or real world environments. Regular updates to the model are necessary to ensure that the model does not lose the capability to detect emerging patterns; otherwise, it will impact the entire tire health monitoring system reliability.

2.6 Gaps in Research and Technology

There are still gaps in the existing techonogies regarding the tire Health monitoring systems that needs to be addressed:

2.6.1 Gaps in Existing Tire Health Systems

There are several gaps in tire health monitoring systems that exist currently in the landscape of such systems and where they fall short of their full potential. One of the most glaring gaps is the absence of complete datasets that adequately capture the numerous kinds of tire defects spanning across variations in brands, models and environmental conditions. However, many existing datasets are of narrow scope and low diversity, leading to models that may not generalize outside of conservative lab based settings. Besides, tire damage types often lack representation in these datasets, rendering it extremely hard to train machine learning models for detecting various damage types.

Integration of multi modal data sources into tire health monitoring is another gap. Though image–based monitoring systems have progressed, integrating image data with sensor data (e.g. tire pressure, temperature and wear rate) continues to be an issue. Correlating visual tire defects with real time sensor data could enable improved accuracy and prediction capabilities of tire health systems, however integrating these sensors types into most systems is not yet fully realized.

Moreover, tire health is still an underdeveloped application of predictive analytics. Currently, most tire health systems address existing damage or wear but do not predict future failures based on past data and tire condition trends. The demand for predictive models that can predict tires failures before they happen, with proactive maintenance recommendations and possibly, avoid accidents, is increasing.

Secondly, we found that there are currently no systems that employ advanced machine learning techniques, specifically transfer learning and deep reinforcement learning, for tire health systems. Although there are many advanced models in practice for defect detection when the input data are images (e.g. convolutional neural networks (CNNs)), exploration of such models is relatively little. For example, transfer learning can greatly reduce the number of labeled data needed for training, but has seen little adoption for use in tire health.

Lastly, we need an improvement on tire health monitoring apps and systems user interface and user experience (UI/UX). Although mobile apps have made tire monitoring more convenient, many do not provide seamless integration with vehicle data, a rich set of features, or offer unnecessarily convoluted interfaces from which the average user finds himself lost. Future work is still needed to simplify and enhance these systems, while maintaining compatibility with different devices and operating systems.

Focusing on closing these gaps enables innovation and improvement in the field, as doing so would result in more accurate, predictive, and user friendly tire health monitoring system.

2.6.2 Improvements in Image Recognition

For improving image recognition for tire health monitoring, efforts are put on improving model accuracy and robustness are made. However, current systems suffer from such problems as moving lighting conditions, image quality, and the surrounding environment that can obscure defects. Addressing these will require better training of models on more diverse datasets, to ensure detection of tire conditions in different environments and on different tire types.

One key advancement is transfer learning — using pre trained models (i.e, ImageNet), to fine tune them for specific tire damage detection task. This means we require less data to train a model and the model performs better with less data. Furthermore, we also applied data augmentation and Generative Adversarial Networks (GANs) to create synthetic data which allows us to help the train model be better able to handle a larger variety of tire images and scenarios.

Micro level tire wear detection is also critical and image recognition systems should be enhanced to detect subtle defects. To do so, then, we need to develop more advanced models that can detect some of the finer nuances that some traditional methods (mistakenly) may ignore. As these improvements gains momentum, the accuracy and reliability of image recognition will continue to increase and provide us with better tyre health assessment and hence better proactive maintenance.

CHAPTER 3

DESIGN AND METHODOLOGY

3.1 System Design and Architecture

A deep learning model for tire health assessment is integrated with a mobile application that collects, analyzes, and provides user feedback on real time data.

3.1.1 Overview of system components

The system design for the Tire Health Monitoring System is based on a modular architecture that integrates three key components: to continue to the Mobile Application, the Admin Web portal and the Database. Each components has its own function, yet they work together seamlessly to be efficient and to give the user a good experience.

3.1.1.1 Mobile Application:

The system's user interface is the mobile application which is mainly available for both Android (source code) and iOS (Ionic) platforms using React Native. Users can monitor the health of your tires using the app, which enables you to upload tire images for condition analysis, enter tire serial numbers and view detailed reports on tire health. For its part, it also offers real time feedback, like how it will alert you to any detected tire damage, wear or problems. Additionally, the app connects with machine learning models for performing image based tire inspections and Optical Character Recognition (OCR) for extracting tire serial numbers.

3.1.1.2 Admin Web Portal:

The web portal, built using React, CSS and HTML provides an administrative interface for managing the system. The functionalities of adding, editing, and deleting tire product in the database, configuring system notification for admins can help administrators efficiently to manage data flow and user interactions. Built as responsive web application, works in any modern web browser.

3.1.1.3 Database

All system data, including user data, products and nontifications, is stored in database as a central repository. This data is stored in and managed by Firebase, a cloud solution that organises and syncs all the data in a real time basis, such as the mobile app and the web portal. Firebase is scalable, therefore it can support huge data and built in security features protects the data. This database provides an efficient platform for storing and retrieving the data related to the tire health assessment for easy compatibility with the mobile application, and the web portal.

3.1.2 Interaction between system components

The Tire Health Monitoring System integrates three components: It has the mobile app, an admin web portal, and a Firebase database. The built app is made with React Native enabling the user to upload tire images and enter serial numbers through OCR. The data is stored in Firebase for real time synchronization, and ML models process the data. React was used to create the admin web portal for product, notification, and backend operation management.

The server side components run on Azure, performing model inference and data processing, and the OCR feature for serial number is powered by cloud based APIs. User data, products, and notifications are managed by Firebase to scale and secure. Seamless data flow and real time updates over all these components are made possible with this architecture.

3.2 Data Collection

Following are sources for Data Collection:

3.2.1 Tire Image Data

Two models were used in the development therefore 2 datasets were needed for the development:

3.2.1.1 Tyre health prediction:

A combination of internet sourced data and images obtained from tire markets were used for the tire health prediction model. Although a (small) dataset of 2 classes was given from Kaggle [8], the rest of the data was gathered by going out, visiting tire markets and collecting images of varying quality from a set of 4 classes. We gathered 1200 images in total which should be augmented to about 6000 for model training. Such augmentation was necessary to boost the size and diversity of the dataset and hence, this allowed for better prediction of tire conditions.
tire and non-tire images:

Besides the tire health images, we needed a different dataset to train the model to recognize tire images and non–tire images. Pre existing tire datasets were used for tire related data. For non tire images we used many sources, ie. Kaggle and other sites, mainly, however, COCO dataset was used as non tire source of images [9]. Carefully curated and augmented data combined with a rich enough model enabled recognition of a wide range of tire conditions and successful classification of tire vs. non tire imagery.

3.2.1.2 Tire Serial Number Data

Users have two input methods for entering tire serial number data. The first action you need to take is to upload the tire image which should show the serial number. Using Optical Character Recognition (OCR) technology the system extracts the serial number from the image. The second one enables user to manually enter the tire serial number. In this regard, users are requested to provide the serial number in a particular format for example 255/55 R16 92W to make the data consistent and accurate. The first method allows the system to retrieve and store tire information efficiently, and OCR provides automation in the case where the input is image based.

3.3 Image-Based Tire Inspection

Following steps were taken during the preparation and training of model:

3.3.1 Preprocessing of Tire Images:

- **Rescaling:** The image pixel values are rescaled by dividing it with 255 (rescale=1./255). This step normalizes pixel values such that pixel values are converted from a range of 0 to 255, to a range of 0 to 1, so this helps the model converge faster in the training process.

- **Resizing:** We resize the images to a uniform size of 128x128 pixels (target_size=(img_height, img_width)). This means that all images have the same dimensions before passing into the model. Reduces computational complexity by making the images smaller meaning shorter training time.

- **Augmentation:** 6 approaches through augmentation techniques are used:

Table 3.1: Augmentation Techniques

Augmentation techniques	**Values**
shear_range	**0.2**
vertical_flip	**True**
brightness_range	[0.8, 1.4]
zoom_range	**0.1**
channel_shift_range	**50**
preprocessing_function	add_gaussian_noise

3.3.2 Model Selection and Training:

For the Tire Health Monitoring System, we initially tested three pre-trained models: Novel Convolution Neural Networks (CNN) Architectures: *MobileNetV2*, *ResNet50*, and *EfficientB0* as per their optimal performance with respect to accuracy and Computational Efficiency. Having evaluated these models, *MobileNetV2* was the chosen model, offering a high performance (94%) yet manageable size that is suited to real time mobile application deployment. As such, the model selection was driven by both high performance and resource optimization.

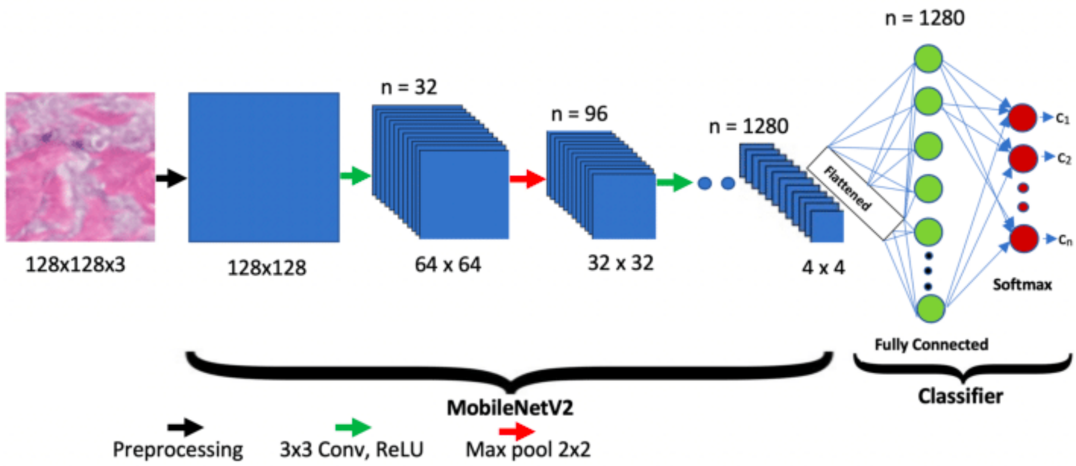


Figure 3.1: Mobilenet model architecture

3.3.3 Model Training and Evaluation:

A dataset of pre processed tire images was split into training, validation, and testing sets for the training process. We fine tune *MobileNetV2* with custom layers for multi class classification. The training was performed using the Adam optimizer, for a learning rate of 0.00001. Performance of the model was evaluated by means of evaluation metrics like accuracy, precision, recall and F1 score. To prevent overfitting and to improve generalization of the model, we applied a variety of regularization techniques — including L2 regularization and Dropout — to the model.

3.3.4 Model Training and Evaluation Approach

With transfer learning, the model was trained and performance was evaluated based on accuracy and loss.

3.3.4.1 Model Training Overview:

In this step we performed the training process with the pre selected model. The test accuracy was the most important metric determining whether the model would actually be useful in the real world, and I used that to evaluate how effective the model was.

3.3.4.2 Performance Analysis Strategy:

For analyzing the performance of model, we used confusion matrix and other related measures i.e. precision, recall and F1 score. Identifying misclassifications and understanding model behavior during training were helped by this.

3.3.4.3 Model Validation:

During the training process, we periodically use a validation set to check that the model is not overfitting to the training data, and instead does generalize to the real task at hand.

3.3.4.4 Optimization Strategy:

Additionally, we used strategies for optimizing our different design decisions, specifically fine tuning and early stopping. To adjust model baselines after training, fine tuning was used; early stopping was also used to prevent model over fitting and improve the model generalization.

3.3.4.5 Model Deployment Preparation:

The best performing model was optimized and the resulting model was saved for future use for tire health prediction tasks. This model was ready for real time deployment.

3.4 Tire Serial Number Recognition

OCR is used by the system to extract tire serial numbers from images to ensure accurate identification for further analysis.

3.4.1 OCR Technique

Optical Character Recognition (OCR) is used in the Tire Health Monitoring System to extract the tire serial number from user uploaded pictures of tires. This is necessary so the system can gather the specific tire details and select, without manual effort, automatic selection of the tire size and specifications. The OCR process was powered with Microsoft Azure's Computer Vision API which provides a stable and highly accurate text extraction mechanism even documents with inconsistencies on the fonts or orientations.

Azure OCR API uses advanced machine learning models trained on millions of fonts, languages, and complex layout, powering it to accurately identify text in images of tires (backgrounds are often confusing, surfaces are curving or lights are inconsistent). The system applies basic preprocessing, e.g. resizing and adjusting contrast to the image before sending it to Azure to make the serial number more visible and improve the ability of the API to find the text.

After Azure processes the image, the detected text is returned from the Azure API, parsed from which the tire serial number is extracted into a structured format. In cases where the OCR is unable to pull the serial number (poor image quality, corruption, warping, etc), the system notifies the user to re-upload the image with better quality. Azure OCR is seamlessly integrated into the mobile application and provides instant and accurate retrieval of tire serial numbers which enable users to find all the details about a particular tire: size, load index, speed rating, etc. It makes OCR a top capability to have that greatly improves the overall user experience with something that would be mostly a manual task.

```
import requests

endpoint = "https://<yourregion>.api.cognitive.microsoft.com/vision/v3.2/ocr"
subscription_key = "<your-subscription-key>"
image_path = "path/to/tire_image.jpg"
image_data = open(image_path, "rb").read()
headers = {
    "Ocp-Apim-Subscription-Key": subscription_key,
    "Content-Type": "application/octet-stream"
}
response = requests.post(endpoint, headers=headers, data=image_data)
ocr_result = response.json()
```

3.4.2 Integration of OCR

The Optical Character Recognition (OCR) capability is embedded within the mobile application so users can simply pull up tire serial numbers. If a user uploads an image of a tire, the mobile app passes this image to Microsoft Azure's Computer Vision API which processes the image and provides the extracted serial number.

The integration involves several key steps: Next, the user is asked to either upload an image of the tire, or manually input the tire's serial number in the format of (e.g. 255/55 R16 92W). The app will send the image to the Azure API, but before that the image is properly oriented and is clear (for the best result) for image uploads. After receiving the OCR output, the app parses the detected text to identify the serial number, and automatically fills in the corresponding fields for further tires analysis.

3.4.3 Challenges in OCR:

Extraction of tire serial number using OCR is not a simple task. Factors such as poor image quality (lack of contrast on some parts of the used car or the background), distorted or incomplete serial numbers on use cars or object in photos and environmental factors (lighting, angles) also affect the quality of generated data. On top of this, text recognition is complicated by inconsistent font styles and tire surface textures. Since any delays in extracting serial numbers can frustrate users, we need real time performance to ensure a good user experience. Operating on such applications demands large system resources while achieving high accuracy in a timely manner on very large volumes of data. Another problem area is image quality; unclear or blurry images, due to poor lighting or camera positioning, for example, can lead to inaccurate, or even failed, OCR processing. To eliminate these problems, some preprocessing steps apply, i.e. image enhancement, and guaranteeing best image capture.

3.5 Mobile Application Development

A mobile application was developed and integrated with the trained model.

3.5.1 Platform Choice

The development of the Tire Health Monitoring System mobile application was implement through the use of the React Native framework because it posess the characteristics of creating the same application for both iOS and Android environment through the use of a single code base. This in turn shortens development time and is cost effective while still maintaining quality. Another feature of the application is that React Native works well with native components; therefore it is ideal for real-time data handling, like images and API. Its environment and instruments allow for connecting machine learning models and OCR for serial numbers' recognition. Secondly, React Native has OTA updates, which make update and maintenance easy thus allows fast feature update.

3.5.2 UI/UX Design: Design and flow of user interactions

Initially, the UI/UX design of the mobile application under consideration was developed with Figma, which offers a convenient platform for creating the design and a prototype of a user interface. Figma provided the utility of making wireframes and clickable prototypes; however, final UI design was made and matched with the user flow. It is little complex, but this is actually very convenient for a user since all the options are divided into categories and can be easily reached by a mouse click or a finger swipe. They include tire image upload, serial number entry and the tire health analysis among others. The mobile app's interface follows efficient design of the work flow so that the number of steps required to perform a certain work such as uploading an image or entering the tire serial number is reduced so as to make the operation of the mobile app efficient..

3.5.3 Integration with Models

The machine learning models were loaded using Flask since it is a lightweight server that could be used to serve the models. For the efficient roll out, the Flask app was then dockerized this made it very easy to port and scale. After dockerization the application was uploaded to Azure server and the application was linked to the mobile application through the domain provided. This approach made it easy to interface the models with the app such that a free flow of data between the mobile app and cloud-based model inference is achieved.

3.5.4 Real-Time Data Processing

Analysis of the tire health in real-time is made possible through the mobile application and cloud based technologies. For instance, suppose the user uploads an image of a tire; in this case, the mobile application transmits it to the server where Machine Learning models parse it to arrive at the tire health prediction. The image data is processed in real time by Flask on Azure for backend services and by interfacing with the pre-trained machine learning models that identify the condition of the tire.

For real-time performance, the system has incorporated the efficient image preprocessing and the model inference. Properties like resizing and normalization are performed on the image prior to transporting the image to the server. It processes the image and the results are fed back to the mobile app in an almost instant way, so response time is kept to a minimum. Such configuration makes it possible to find tire health prediction within a matter of minutes after uploading an image, which makes it easy to get a real-time chance to analyze the condition of tires.

3.6 Web Portal (Admin Panel)

For the management of database, a web portal was developed for the admin control.

3.6.1 Admin Panel Design

This panel is meant to act as an admin panel to efficiently manage tire related products and notifications. On the Product Info page, administrators can easily add, modify, or even erase product attributes such as the name of the product, the category, and links to other stores. This help to manage product catalog in the system effectively. Also, to this panel, it is possible to create and track those that will be displayed to the users, for instance, product updates and alerts, promotions, etc.

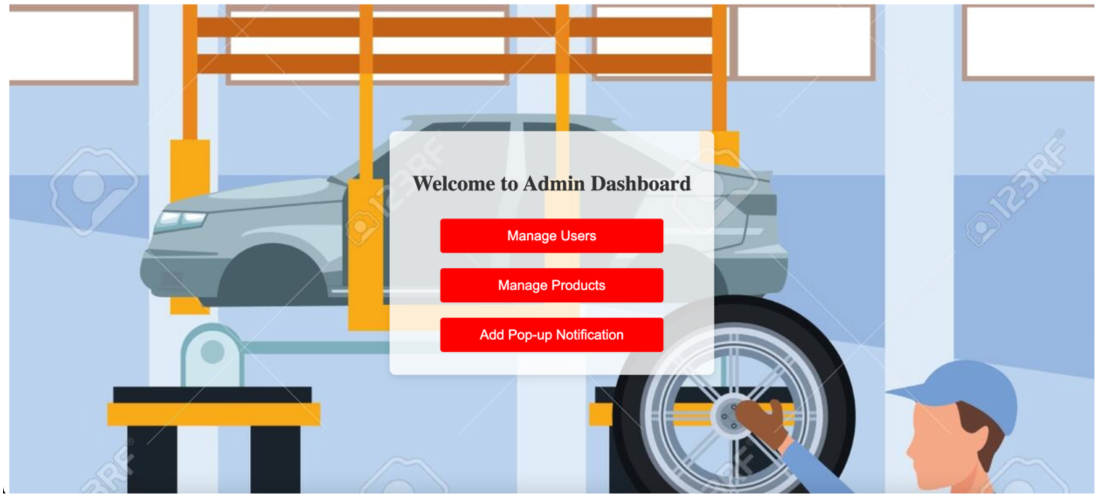


Figure 3.2: Web Portal

3.6.2 Integration with Mobile Application

This integration with Firebase guarantees one way synchronization between the admin panel and the mobile application. The changes made through the admin interface are automatically displayed in the mobile app, which increases the relevance of information. The overall design of the panel is pleasingly intuitive, which means a non-technical administrator is capable of performing most back-end tasks without feeling overwhelmed.

3.6.3 Backend Infrastructure: Using Firebase

The backend framework of the Tire Health Monitoring System, mainly focuses on Firebase for data storage and organization. Firebase acts as a backend storage keeping all the user data and the app synchronized with real-time notifications and an admin dashboard. Its characteristics are: it is cloud-based thus can be updated easily; data can be accessed easily; it has the capacity to contain large information which serves the variability of the system.

3.7 Challenges in Model Integration

Following challenges were faced during the integration of trained model with the mobile application.

3.7.1 Model Deployment and Cloud Integration

Having integrated machine learning models into the Tire Health Monitoring System, some problems arise when using models from development to production. One of the challenges is weighing the options that are local deployment of the models or on the cloud. There are some disadvantages of local deployment, such as limited response time and scalability and the ability to be burdened by the physical complexities of the hardware. On the other hand, cloud deployment provides more scalability as well as flexibility for applications, however these applications need a stable internet connection and can also have latency due to the transfer of data over the web.

For the system, cloud integration using platforms like Azure is a more feasible option when it comes to hosting of the machine learning models. Currently, Azure makes it easy to deploy models because the cloud infrastructure enables real-time sharing of the model with both the mobile application and the admin panel. This makes it possible for model inference to happen efficiently and at scale with out the need to develop and maintain complicated local hardware systems. Furthermore, utilization of the cloud platforms enables the easier integration with other services like large data storage and image processing as APIs, guaranteeing the proper functioning of all firm's components of the system. These shortcomings notwithstanding, control of cloud resources and guarantee of low latency during model inference can remain daunting tasks in the future.

CHAPTER 4

DATA AND EXPERIMENTS

4.1 Model Selection and Training

Three pre trained models were tested for tire health classification and MobileNetV2 was chosen as the best model as it has a good balance of accuracy and efficiency.

4.1.1 Model Comparison:

Three pre trained models (*MobileNetV2, ResNet50, EfficienetB0*) were tested on tire health classification. Accuracy was used for evaluation and *MobileNetV2* was the best model for the task as it is the best balance of accuracy and computational efficiency.

4.1.2 Accuracy Comparison:

A table comparing the accuracy of each model:

Table 4.1: TL Model Comparison

Model	**Accuracy**	**Comments**
MobileNetV2	94%	Best balance of accuracy and efficiency
ResNet50	92%	Higher accuracy but computationally expensive
EfficientNetB0	26%	Low accuracy and more resource-intensive

4.1.3 Model Selection:

MobileNetV2 was chosen for its superior performance and resource efficiency, so that it could be easily integrated into the tire health monitoring system.

4.2 Model Training Process

4.2.1 Model Architecture and Pre-training:

As the tire health prediction model used *MobileNetV2* which is a lightweight pre-trained model that is very efficient. As the tradeoff between accuracy and computational efficiency is important for real time applications such as the Tire Health Monitoring System, *MobileNetV2* was chosen. The first step was to use the pre trained *MobileNetV2* model as a feature extractor, meaning we used the trained convolutional layers for feature extraction and removed the top layers of the model to adapt the model to our specific multi class classification task.

4.2.2 Pre-Fine-Tuning Phase:

In the first training phase (before fine tuning), we made the weights of all the layers except the last classification layers frozen to avoid overfitting and to accelerate the training. This made possible to use the pre learned features from ImageNet dataset, and learn task specific features relevant for tire health.

First, *MobileNetV2* model was trained with only last few layers (Dense layers) trainable.

Since edges and textures are general features that are known to be relevant for identifying tire conditions, we kept the initial convolutional layers frozen.

4.2.3 Training Configuration:

- **Batch Size:** This size of batch (32 images per batch) is commonly used in deep learning because training time is faster and the performance of the model typically in higher.

- **Image Size:** To make processing faster and to reduce computational load, the input images were resized to 128x128 pixels, losing little in terms of feature quality.

- **Training/Testing Ratio:** To ensure that the model was well trained, but also had enough data to provide unbiased evaluation, the data was split into an 80% training set and a 20% test set.

- **Validation Split:** We used 10% of the training data as the validation set, which was useful in monitoring the model performance during training, and to catch overfitting.

- **Learning Rate α**: The learning rate was set to 0.00001 so that during the training the weights can be updated gradually, instead of overshooting on the optimal solution. Especially for the fine tuning phase, this small learning rate was crucially important, as only very subtle adjustments are needed to improve performance without forgetting the pre trained knowledge Preventing overfitting.

- **Regularization:** L2 Regularization and Dropout were applied to prevent overfitting. We applied L2 regularization to the Dense layers, and added Dropout of 50% for model robustness.

- **Loss Function:** Because the classification problem is multi class, Categorical Crossentropy was used as the loss function.

- **Optimizer:** Adam optimizer was used to assure the smooth convergence

4.2.4 Evaluation Metrics:

The model was evaluated using accuracy, precision, recall, and F1 score.
Summary of Evaluation Metrics:

Table 4.2: Evaluation Metrics

Evaluation Metrics	**Scores**
Accuracy	94%
Precision	94%
Recall	94%
F1 Score	94%

4.3 Model Evaluation

The model's performance was assessed using the test dataset, where accuracy was the primary metric.

4.3.1 Confusion Matrix:

A confusion matrix was used to identify misclassifications and the model's performance across all classes.

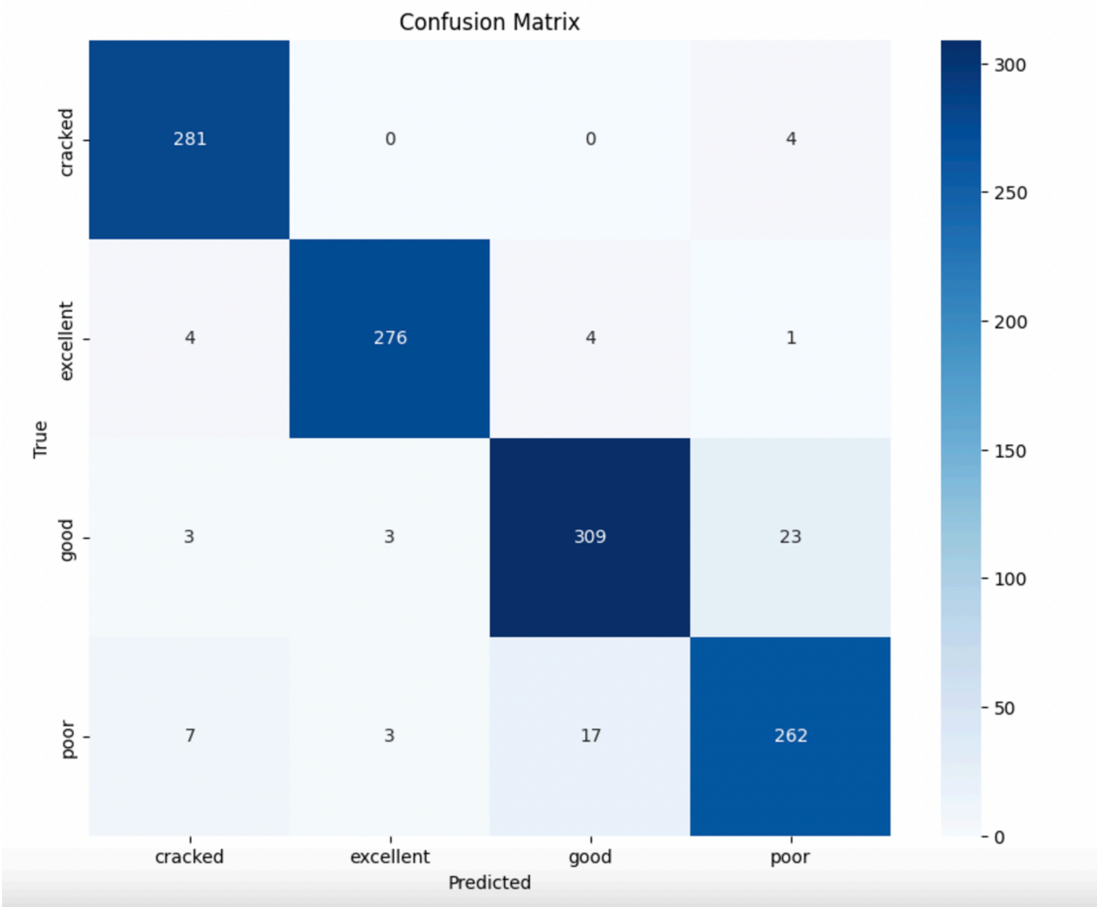


Figure 4.1: Confusion matrix

4.3.2 Classification Report:

A classification report was generated, providing insights into precision, recall, F1 score, and support for each class. This report is essential to assess the model's reliability for each tire condition class.

Classification Report:

	precision	recall	f1-score	support
cracked	0.9819	0.9509	0.9661	285
excellent	0.9305	0.9860	0.9574	285
good	0.9457	0.8757	0.9094	338
poor	0.8824	0.9343	0.9076	289
accuracy			0.9340	1197
macro avg	0.9351	0.9367	0.9351	1197
weighted avg	0.9354	0.9340	0.9339	1197

Figure 4.2: Classification report

4.4 Optimization Techniques

Following optimization techniques were used:

4.4.1 Fine-tuning:

Fine-tuning refers to taking a pre-trained model and further training it on a new dataset. Fine-tuning involves training the entire model, including the initial layers. To achieve better and more specific patterns for this task, we applied fine tuning to the *MobileNetV2* model. Firstly, the lower layers of the *MobileNetV2* model pre trained on a large dataset (ImageNet) were frozen to keep the overall feature extraction ability. Next, the top layers were unfrozen, such that the model could change these layers in order to capture more task specific features relevant to tire images. This helped the model to catch up with some of the tiny details about the tire which are exclusive to the tire conditions like cracks, tread wear, and other health indicators which ultimately resulted into overall improvement in the prediction accuracy.

4.4.2 Early stopping:

Early Stopping is a technique used while training neural networks to prevent the model from overfitting.Early stopping stops a training process after a specified number of epochs to see if validation loss is not better than before (patience), where specified number of epochs is pre-defined. By stopping the model from learning further, it prevents it from over learning and fitting to the noise in the training data which otherwise decrease its performance over real world tire images. Instead of training to a fixed number of epochs, training to an optimal point allows us to stop training when the model has sufficient ability to generalize, prediction with more certainty out in the real world.I saved the optimized model for future use and deployment.sential technique used to refine the performance of a pre-trained model. After the initial training phase, the model's performance can be enhanced by making small adjustments to its parameters. In this case, fine-tuning was applied to the *MobileNetV2* model to help it learn more specific patterns relevant to tire health. Initially, the lower layers of the *MobileNetV2* model, which were pre-trained on a large dataset (ImageNet), were frozen to retain the general feature extraction capabilities. Subsequently, the top layers were unfrozen, allowing the model to adjust these layers for more task-specific features related to tire images. This allowed the model to better capture subtle details specific to tire conditions such as cracks, tread wear, and other health indicators, improving overall prediction accuracy.

4.5 Model Saving:

The optimized model was saved for future use and deployment.

```
model.save("/Users/khalidhameed/Documents/Khalid/FYP/models/mobilenet_model.keras")
```

4.6 Training Results

Multiple epochs were trained on the model to get an optimized level of classification of tire conditions. The following key metrics were observed during the training process:

4.6.1 Training Accuracy:

To investigate it, I created a chart which plotted the training accuracy during the model training, in terms of epochs, and found that, as the model progressed through each epoch, the training accuracy improved steadily, showing that the model was actually learning from the data. Early stopping is used to prevent the model overfitting and thus the model was trained for a total of 20 epochs. However, by the end of the training, the model's training accuracy was 94%.

4.6.2 Validation Accuracy:

The training accuracy and validation accuracy were monitored at every epoch and the validation accuracy was always improving with the training accuracy, which means that our model learned to generalize to unseen data. In the final epoch, the validation accuracy got to 93% which confirms that model is not overfitting on training data.

4.6.3 Loss Function:

When training continued through the epochs, the model's loss (evaluated using categorical crossentropy) dropped evenly as the predictions became more accurate by each iteration.The convergence was good as the final loss value was around 0.8.

4.6.4 Effect of Fine-tuning:

We find that fine tuning helped our model. Upon fine tuning the model accuracy improved both on training accuracy and validation accuracy which showed that fine tuning the final layers to learn tire specific feature did improve the model’s predictive ability.

CHAPTER 5

USER MANUAL AND DISCUSSION

5.1 Splash Screen:

After launching the app, the user will see a splash screen with app's logo.

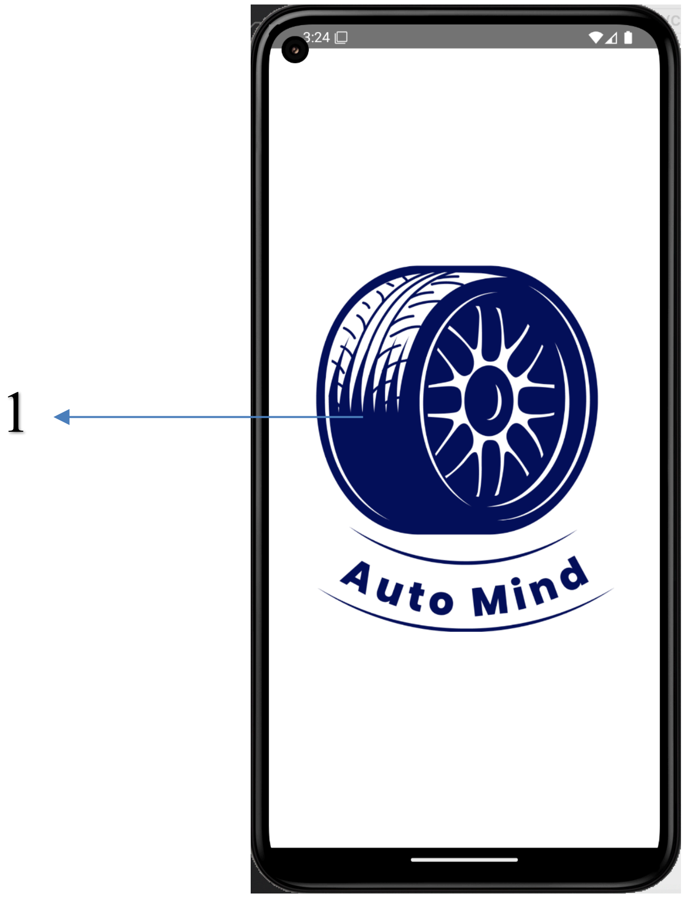


Figure 5.1: Splash Screen

1. Splash Screen Logo

5.2 Login Screen:

After the application is launched, a login page will appear, the user is supposed to enter his email and password if his profile has already made.

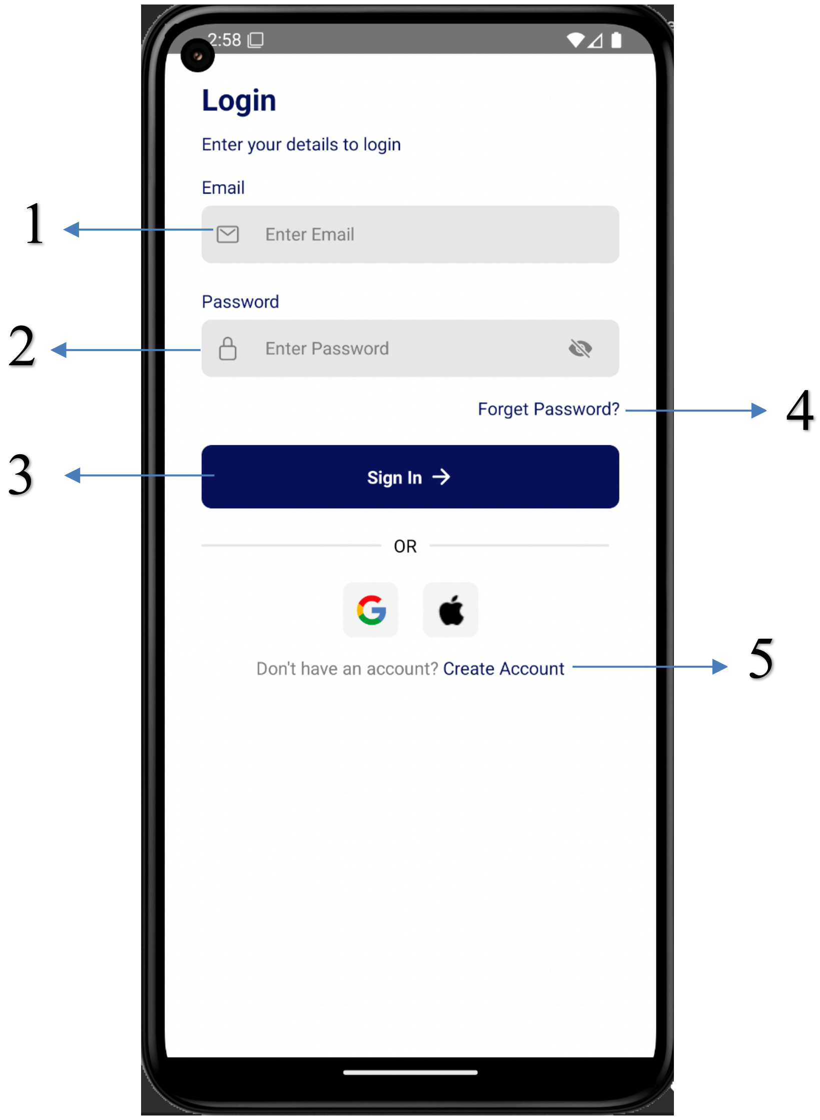


Figure 5.2: Login Screen

1. Enter Email
2. Enter password
3. Sign-in button to login
4. Click if you forgot password
5. Click if you are a new user

5.3 Signup Page:

To Signup into the application, users must enter their Name, email and passwords.

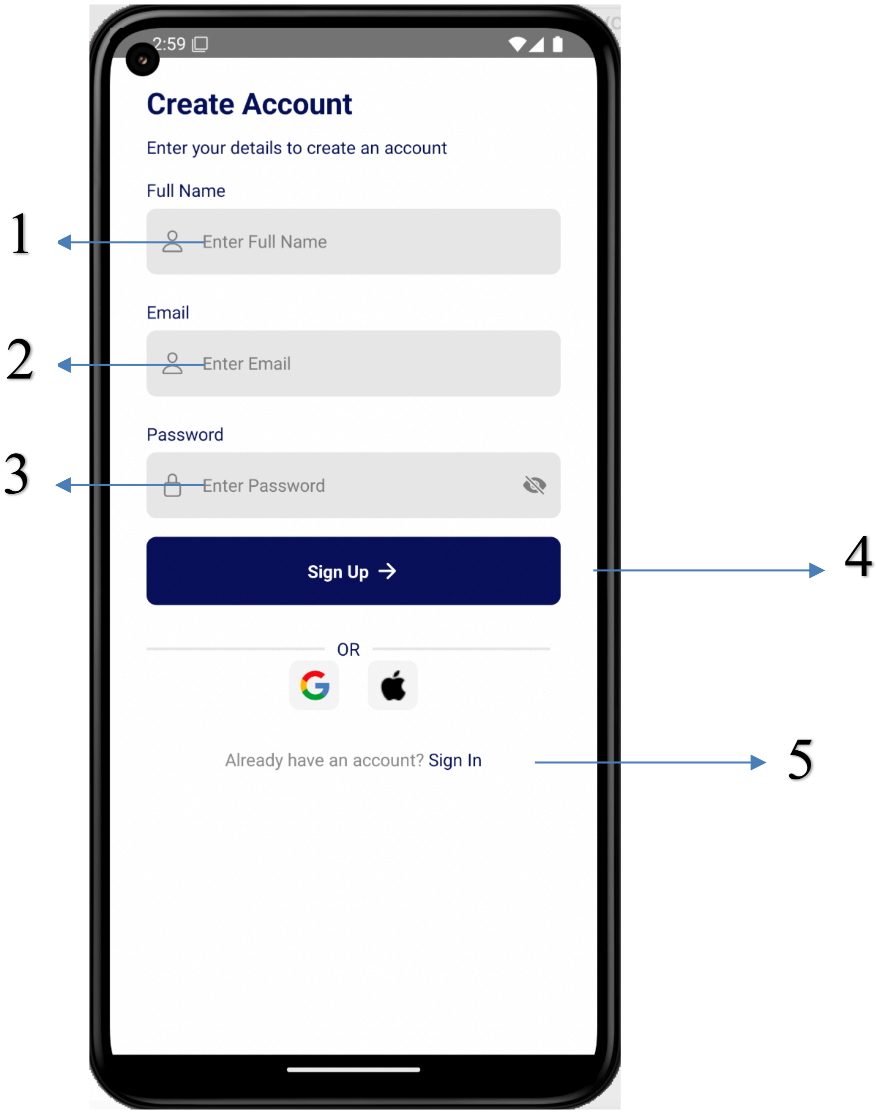


Figure 5.3: Signup screen

1. Enter your full name
2. Enter you valid Email on which authentication code will be sent
3. Enter a password
4. Click to signup
5. Click to Signin if account exists

5.4 Forget Password Page:

Forget password is designed for users who forgot their password, the users will now need to authenticate with their email upon which a verification mail will be sent.

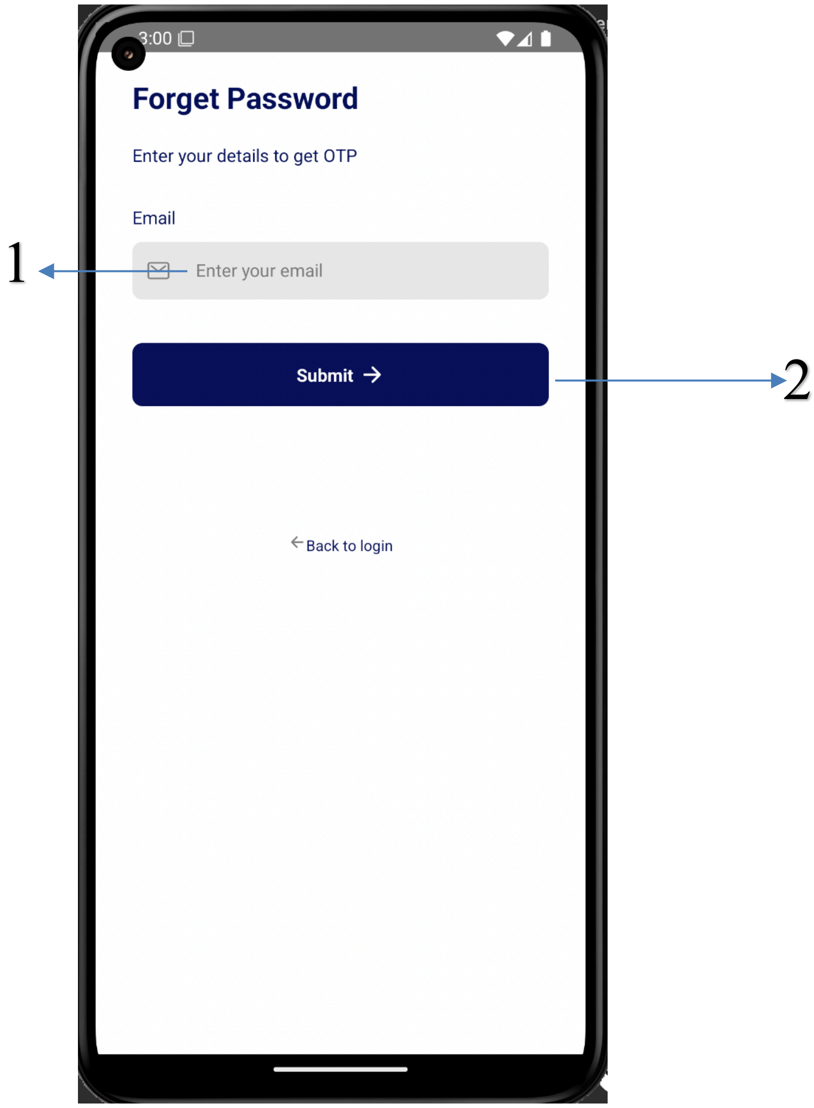


Figure 5.4: Forget Password Screen

1. Enter your Email
2. Submit button

5.5 Home Page:

After logging in, the user will be directed to the home page where he will be accessed to the application's features.

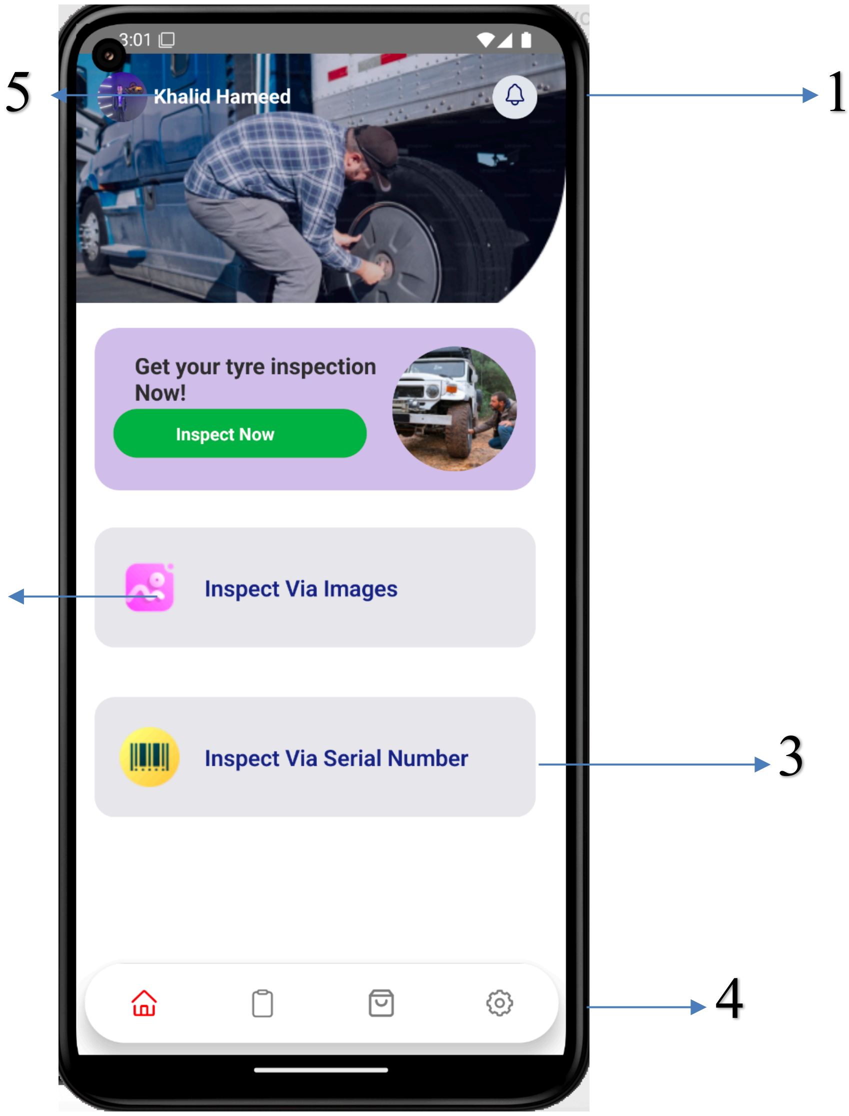


Figure 5.5: Home Page

1. Your Profile Picture and Username
2. Weather nofication icon
3. Inspection Via Images Screen (Tyre Condition)
4. Inspection Via Serial Number
5. Navigation Bar

5.6 Nofication Bar:

A notification bar will appear on the top of the home page screen which will display necessary information regarding vehicile's safety.

1

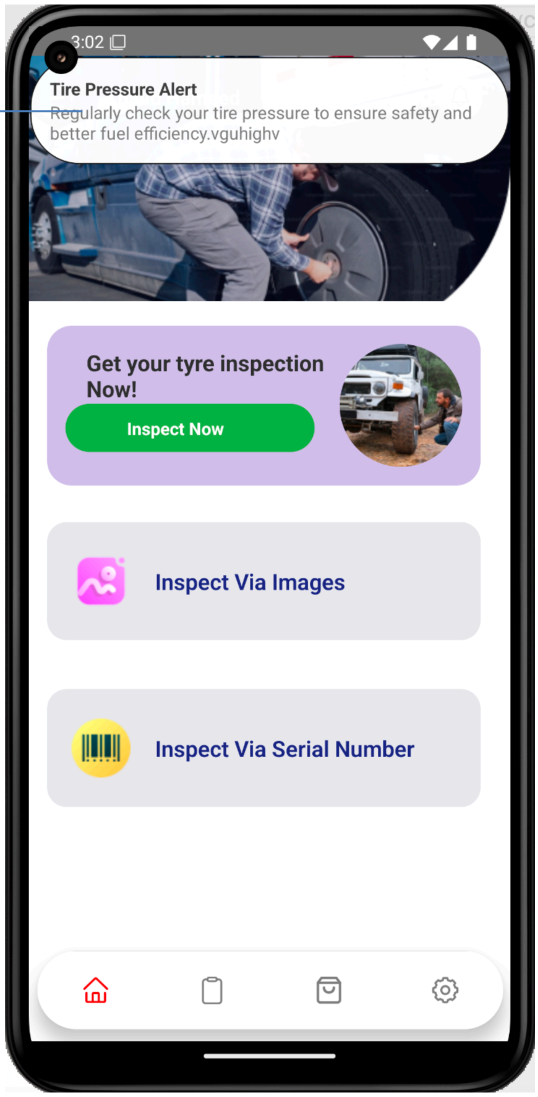


Figure 5.6: Notification Screen

1. Notifcations

5.7 Weather Nofication Page:

User can access the weather nofication page where he will be provided with updated weather updates.

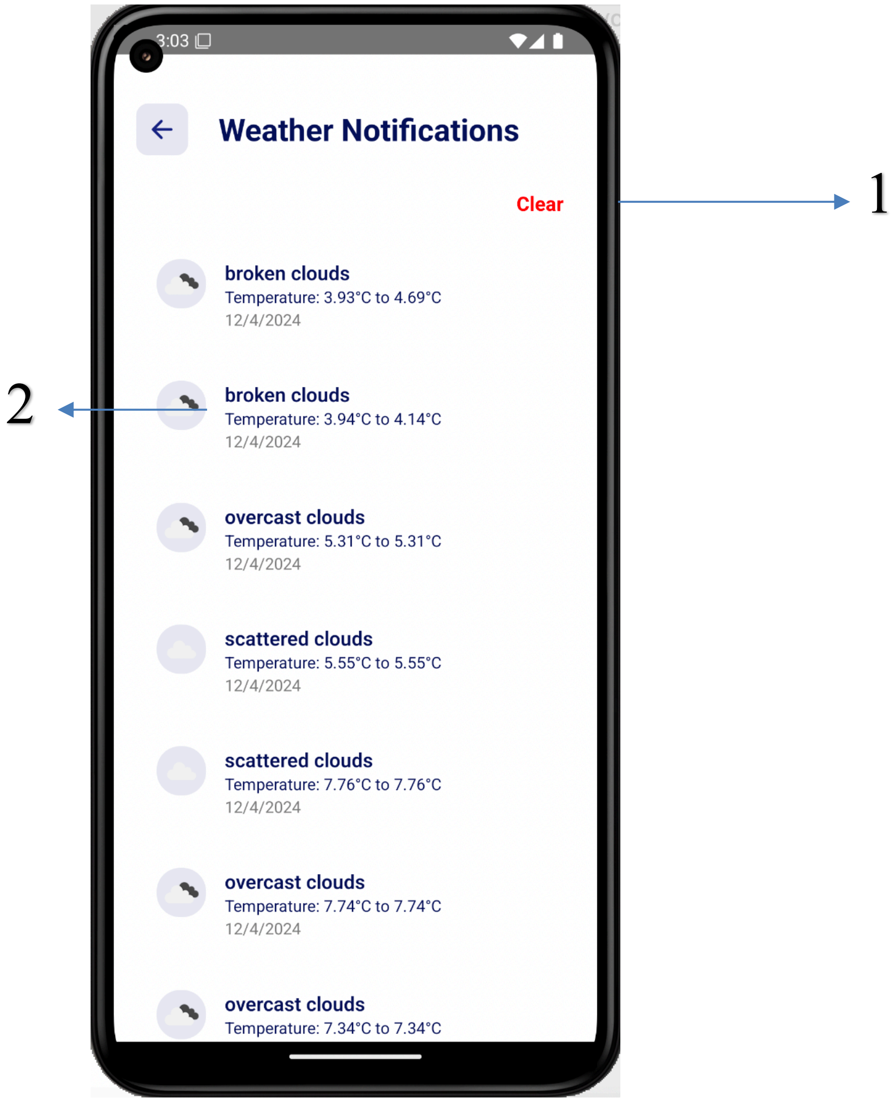


Figure 5.7: Weather Notification Screen

1. Clear Notifications
2. Weather Notifications

5.8 History Page:

Users can access their all previous Inspection Records on the history page. They use this feature via navigation bar shown at the botton of the screen.

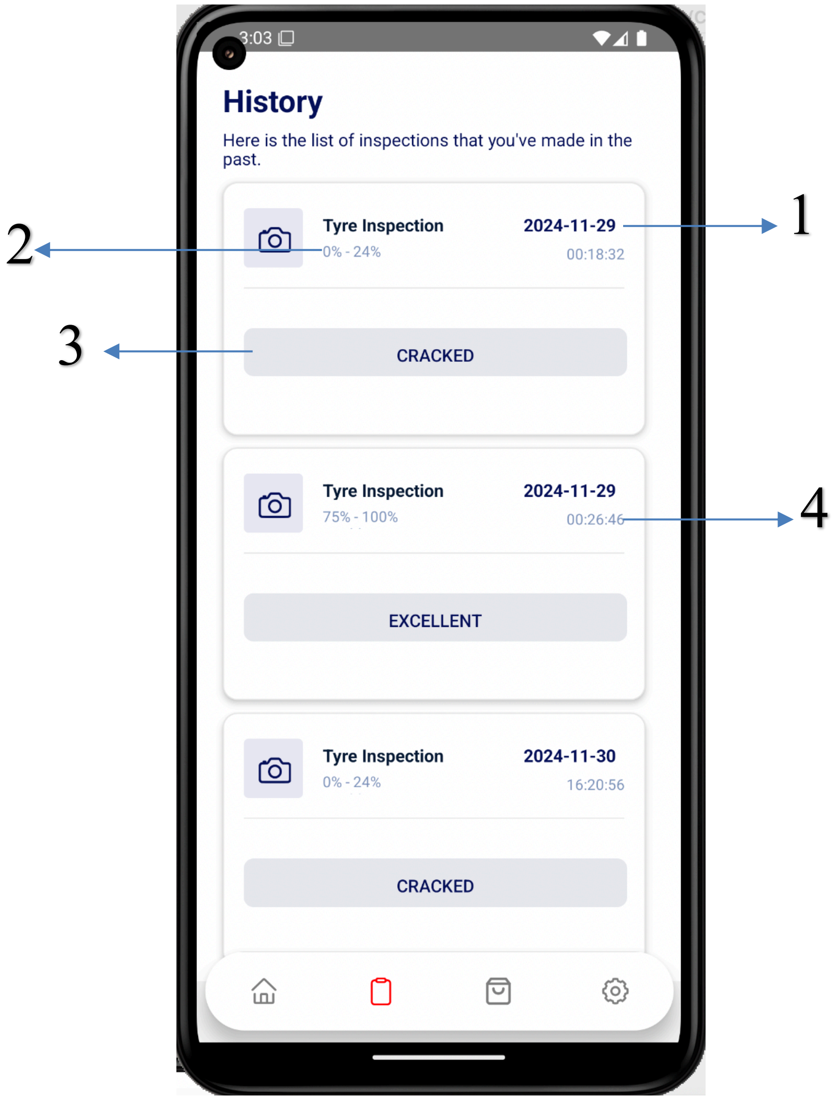


Figure 5.8: History Screen

1. Inspection Date
2. Tyre Health Percentage
3. Tyre Condition
4. Inspection Time

5.9 Products page:

Users can shop their favorite product from the store via application, upon clicking on a product, the user will reach to the store where the actual product is placed from where they can shop their product.

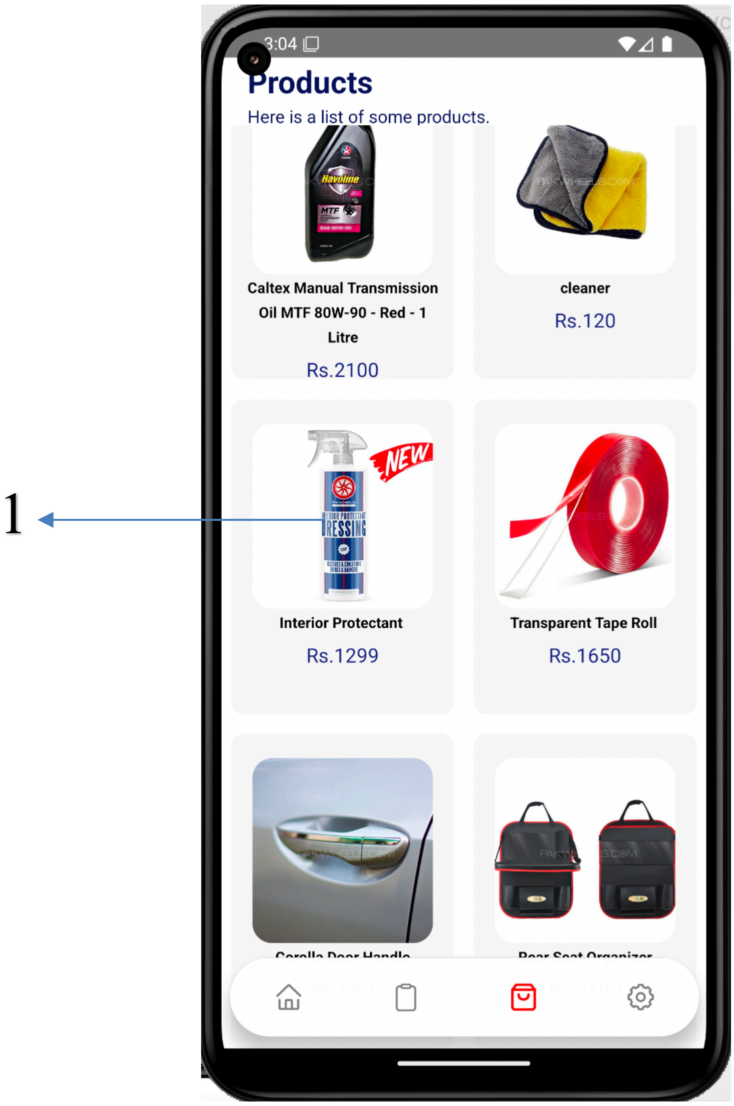


Figure 5.9: Products Page

1. Products

5.10 Inspect Via Images Page:

This page will allow the users to access one of the main feature of the application, users can upload the images of their tyres and get their inspection done, the model will access the tyre's condition and provide the user will respective response. Users can upload upto 4 images.

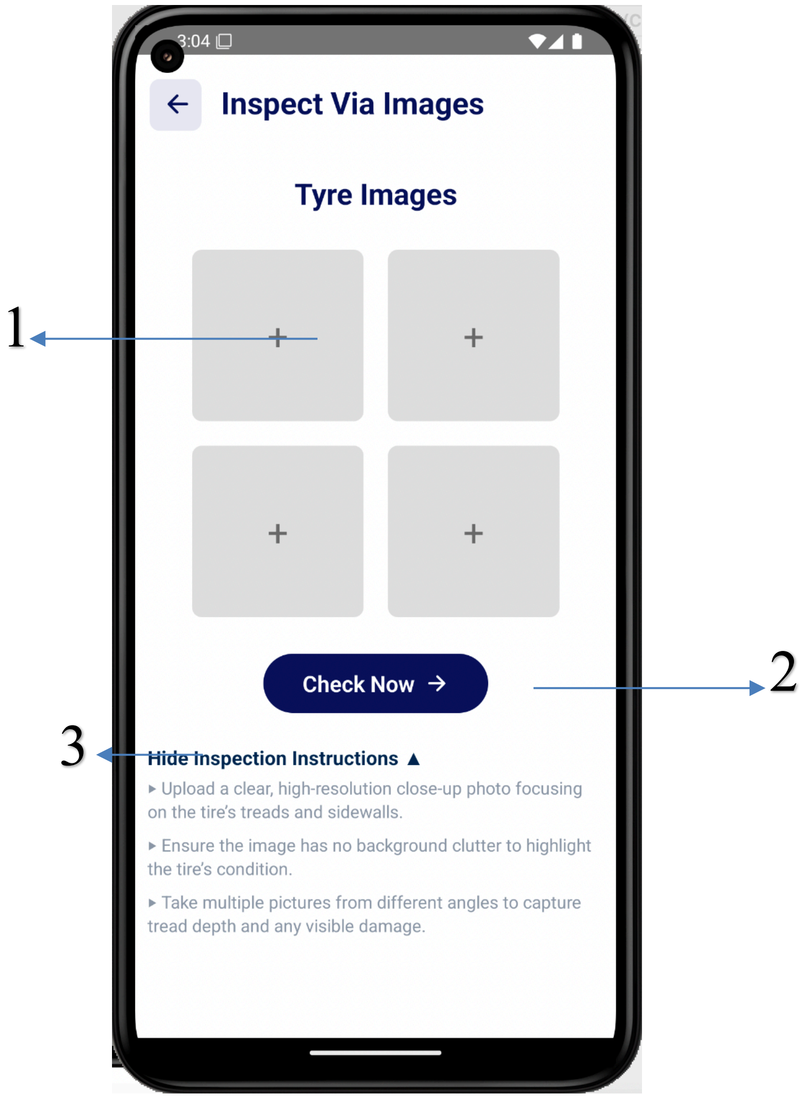


Figure 5.10: InspectViaImage Screen

1. Add Valid Tyre images
2. Submit
3. Inspection Instructions

5.11 Error Page:

An error will be shown if the user doesn't provide a valid tyre image.

1

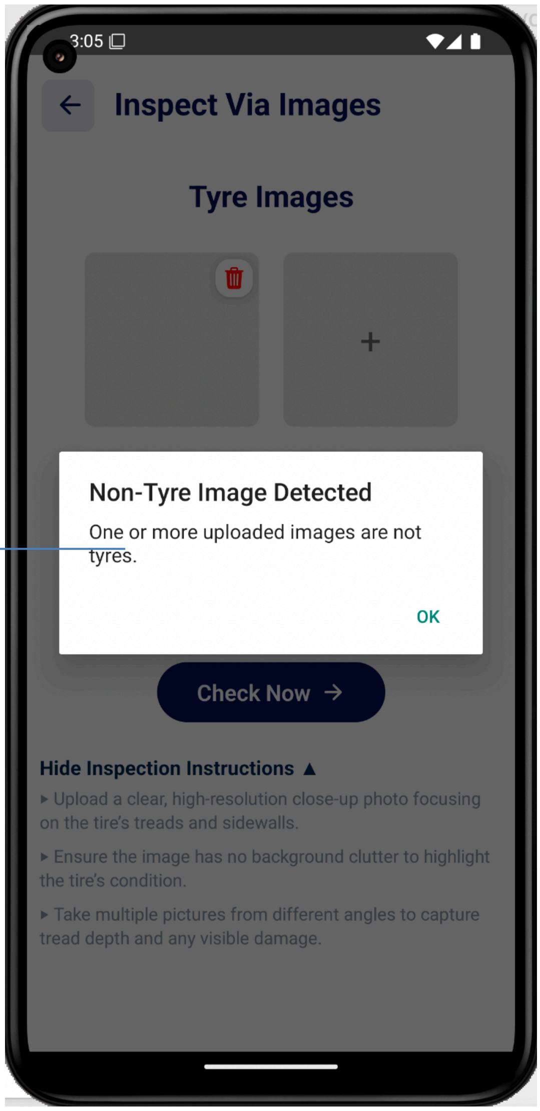


Figure 5.11: Error Screen

1. Error displayed on submitting invalid image

5.12 Tyre Health Result Page:

After submitting the images of tyres, the users will be navigated to the result's page where the results of their inspection will be provided to them with personalized recommendation and Products relevent to their tyre's health.

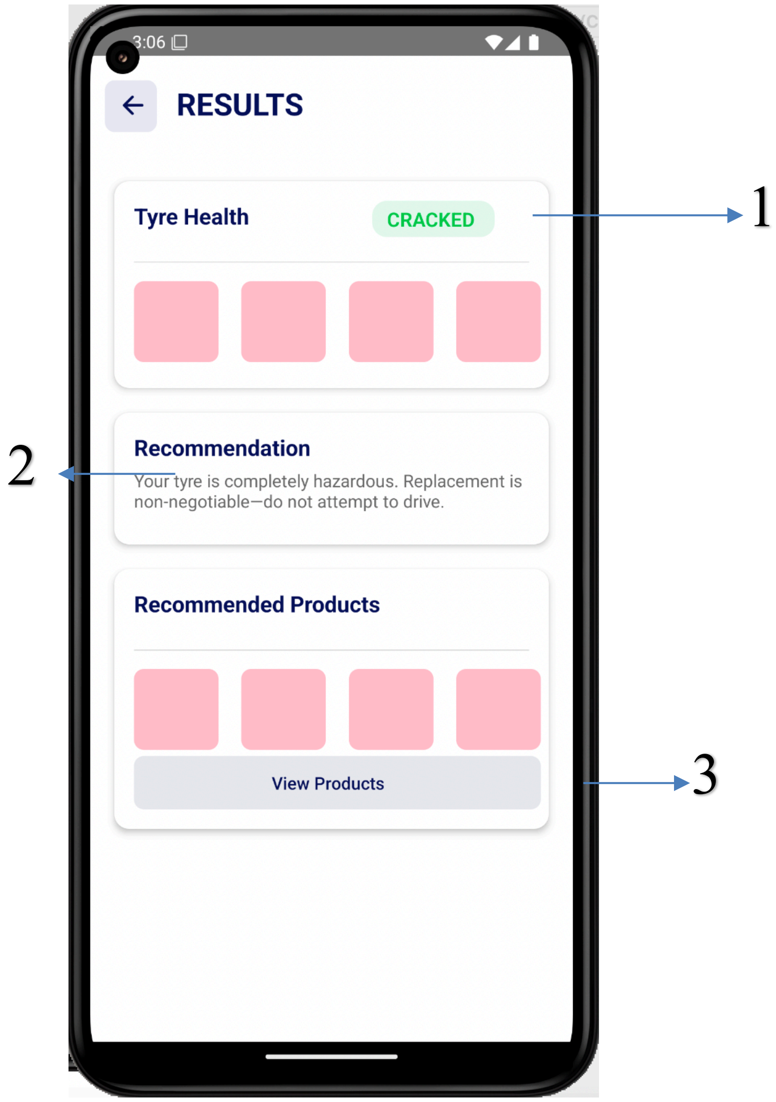


Figure 5.12: Tyre Health Result Screen

1. Tyre Health
2. Recommendation based on tyre health
3. View Recommened products based on tyre health

5.13 Recommeded Products Page:

This page will assist the user in finding the relevent product's related to their tyre's health. User will be navigated to the product's original store on clicking the product.

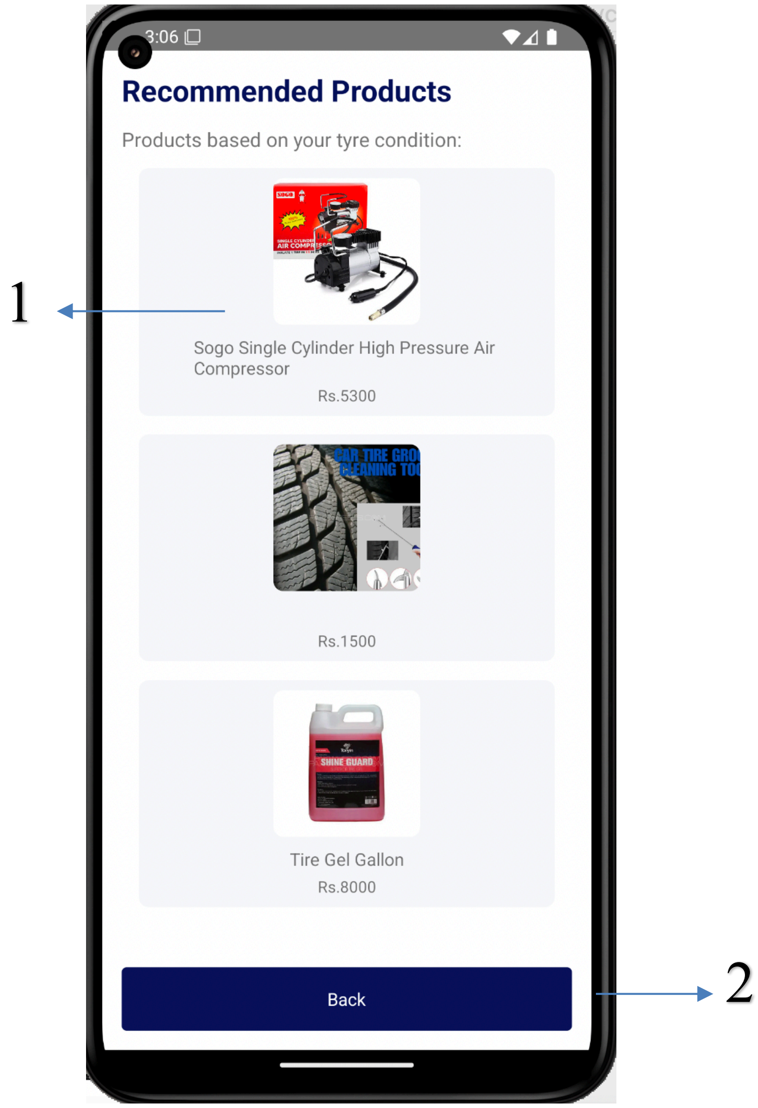


Figure 5.13: Recommened Products Screen

1. Recommended Product
2. Back Button

5.14 Inspect Via Serial No Page:

This page will allow the users to access one of the main feature of the application, users can either upload the images of their tyre's serial number or manually enter the serial numbers to get details about the tyre's specifications.

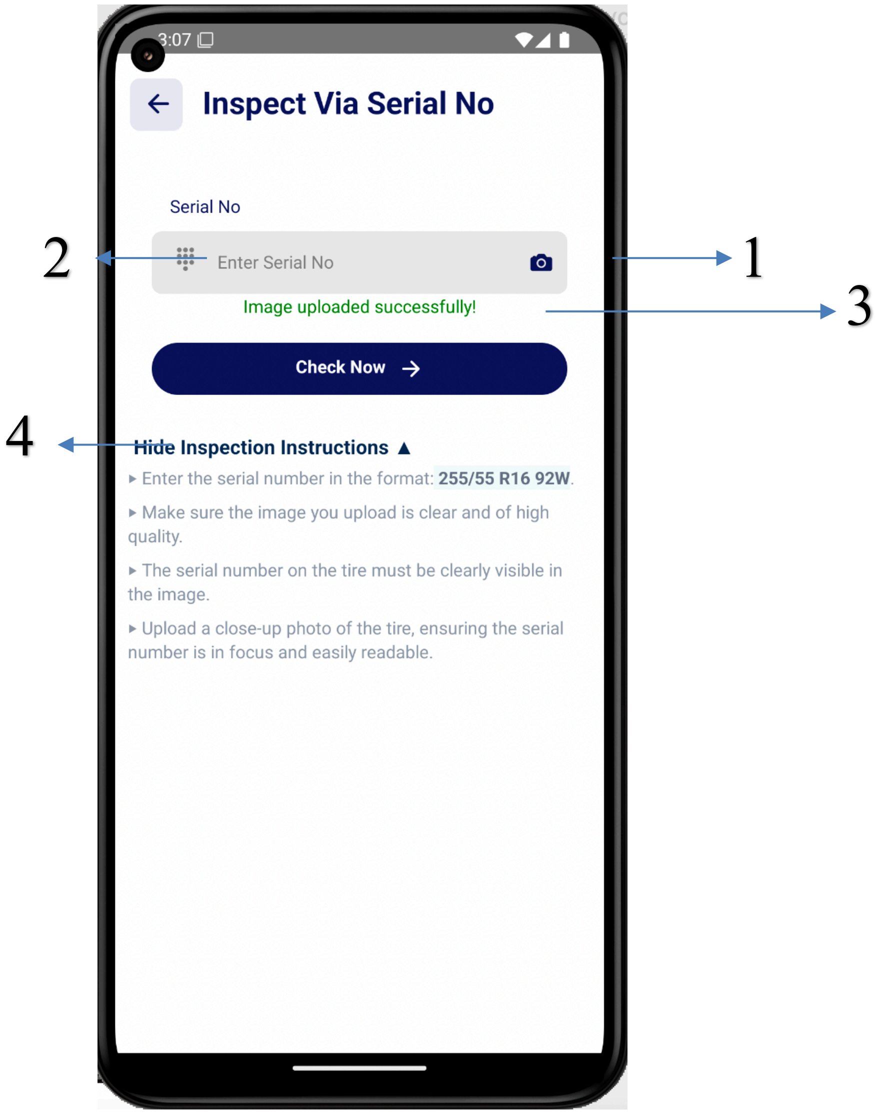


Figure 5.14: InspectViaSerialNo Screen

1. Upload Tyre image to scan serial number
2. Can manually enter tyre serial number
3. Image successfully uploaded
4. Inspection Instructions

5.15 Error Screen:

An error will be shown if the user doesn't enter a valid serial number.

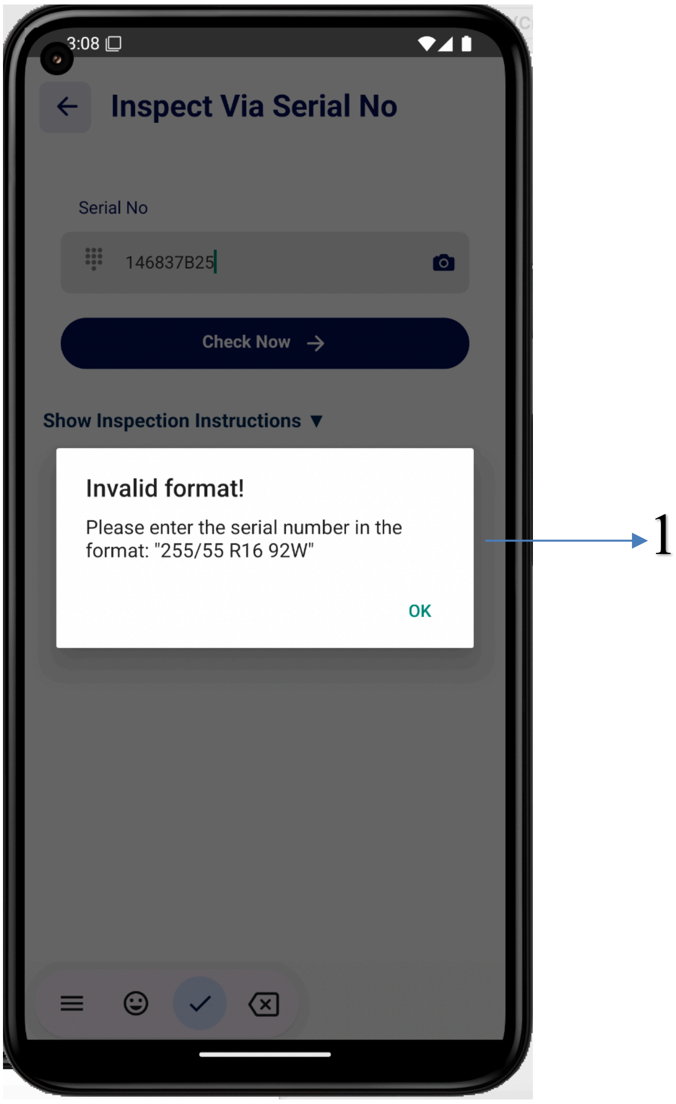


Figure 5.15: Serial No Error Screen

1. Error being shown on entering wrong format Serial Number

5.16 Inspect Via Serial No Results Page:

Upon entering a valid serial number, the users will be directed to the results page where all the information related to tyre will be displayed to the user like width,aspect ratio, Rim Size, Load Index and Speed Rating. Other Markings will display either the tyre is tubeless or not.

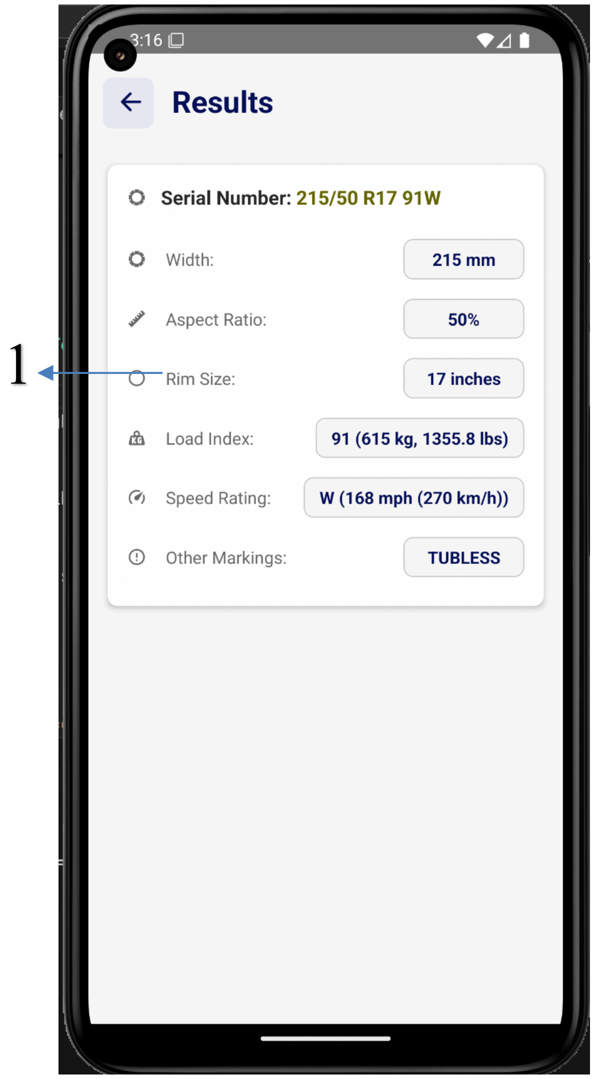


Figure 5.16: Serial No Results Screen

1. Tyre Serial Number Specifications

5.17 Settings Page:

Settings Page assist the user in maintaining the application's as well as user's personal detials. Users can switch between Dark and Light mode from this page also can logout.

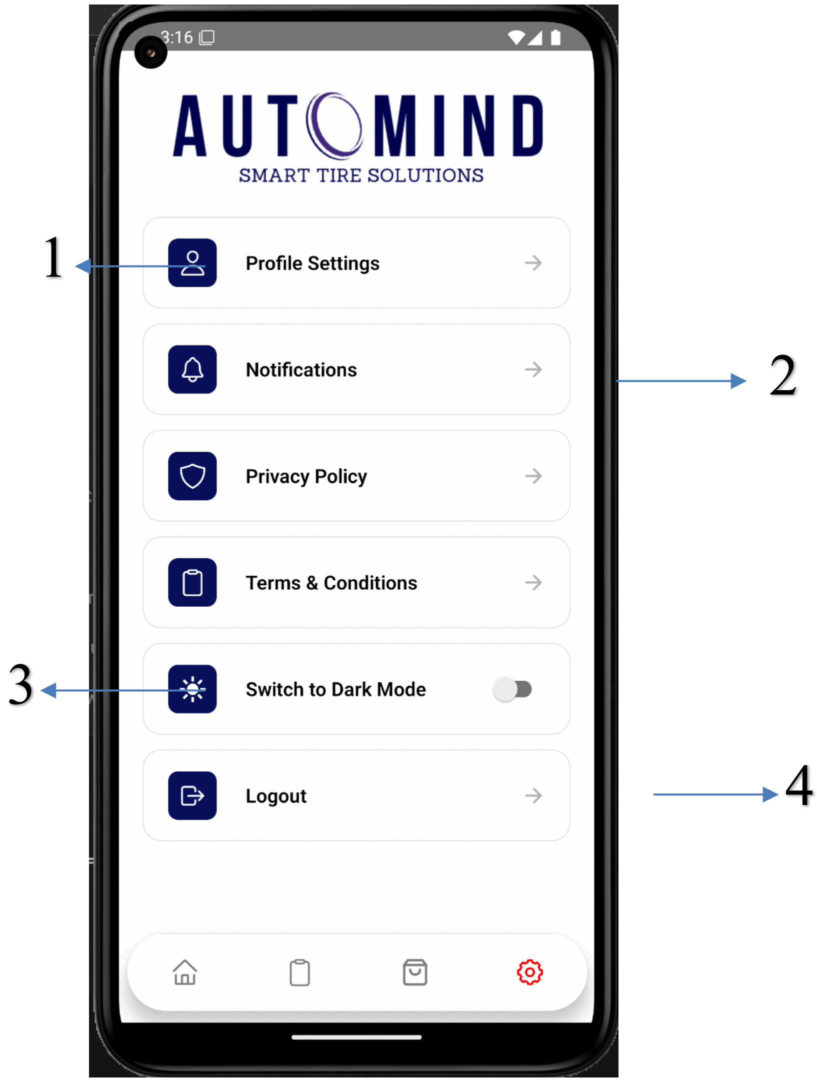
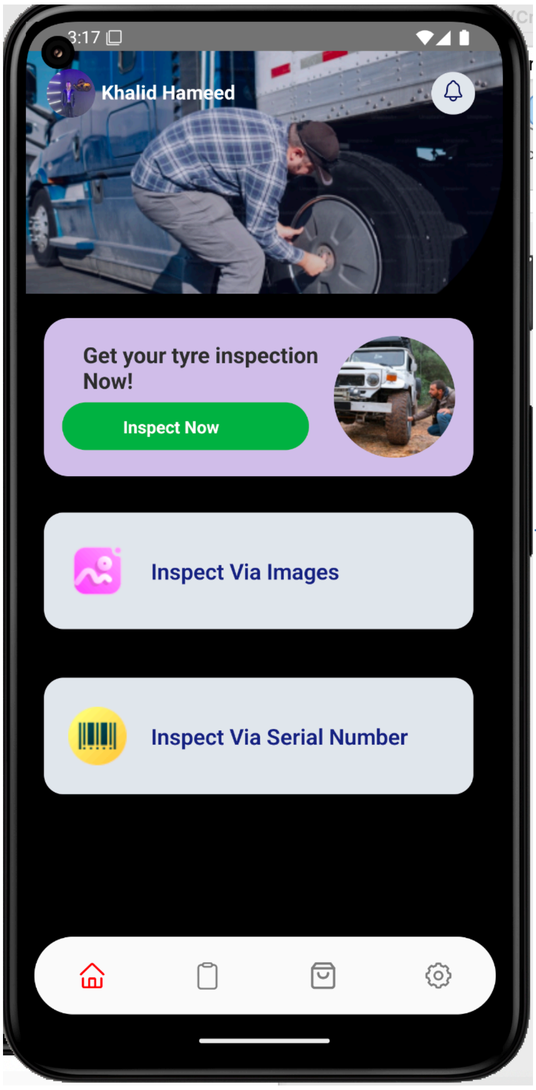


Figure 5.17: Settings Screen

1. Update User's Information
2. Can Turn On/Off Notifications
3. Switch to Dark/Light Mode
4. Logout Button

5.18 Dark Mode:

Upon Switching the toggle on, Dark mode will be activated.



1

Figure 5.18: Dark Mode

1. Dark Mode Activated

5.19 Account Details Page:

This screen allows the users to save their personal information if their using the application for the first time, otherwise the users can view and edit their profile from this screen.

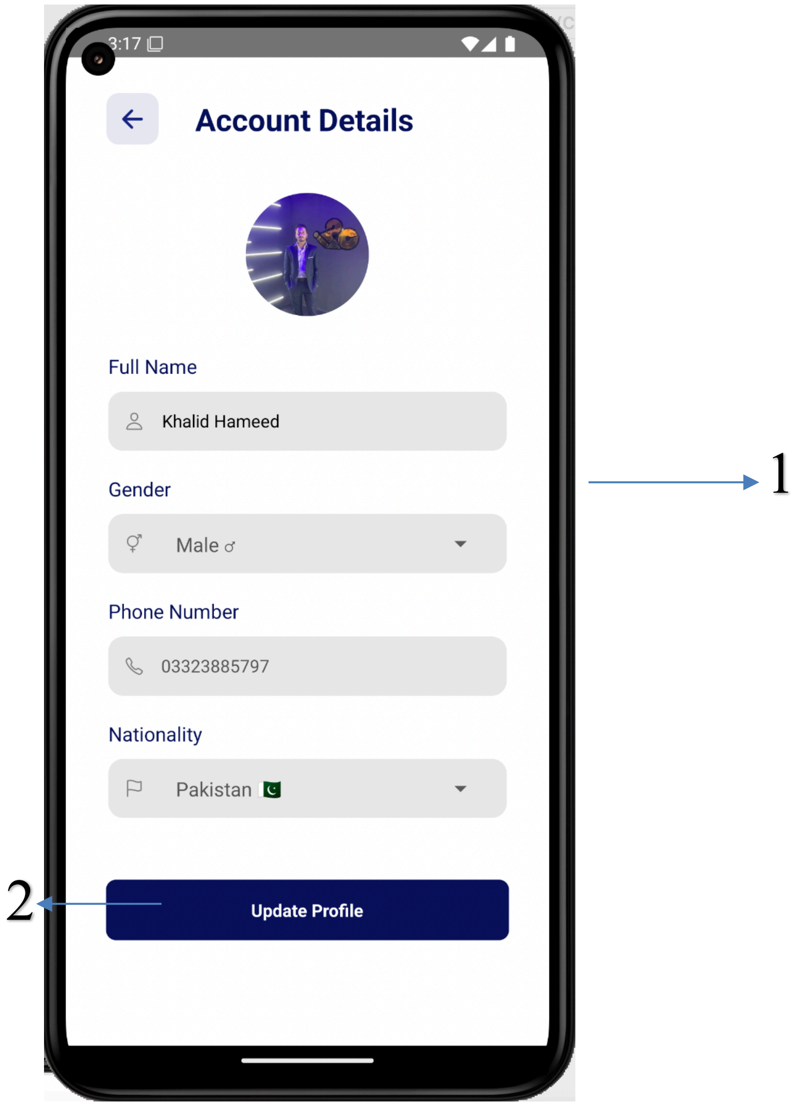


Figure 5.19: Account Details Screen

1. Enter your User details
2. Submit your data to the database

5.20 Notification Toggle Page:

This Page allows the users to use the toggle to turn Off/On the notifications that are being shown on the home page.

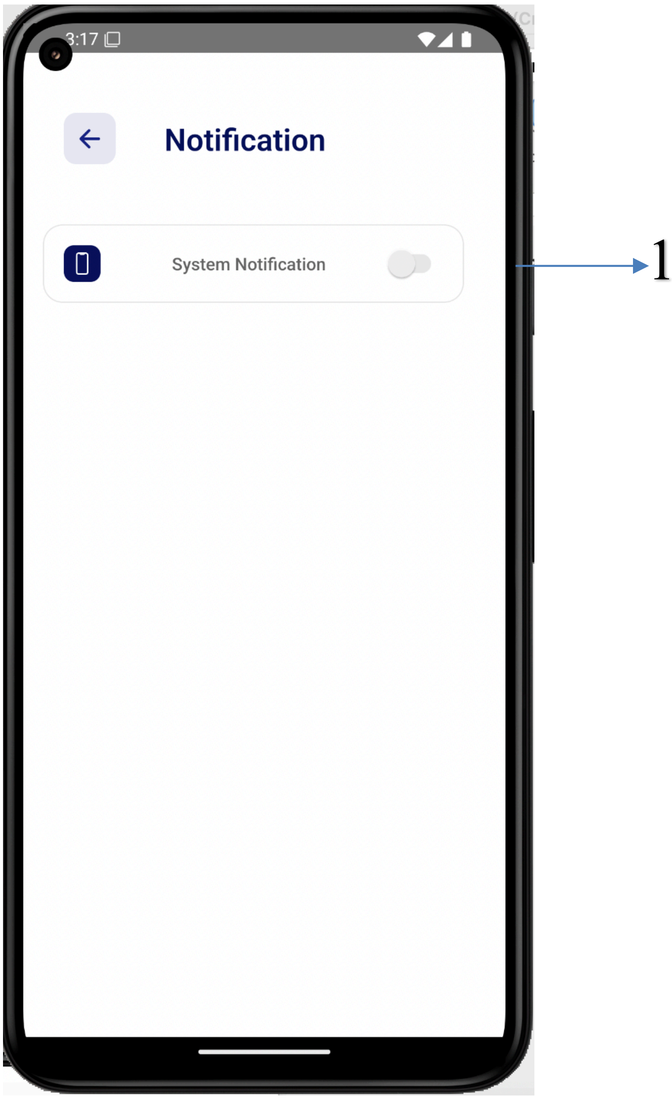
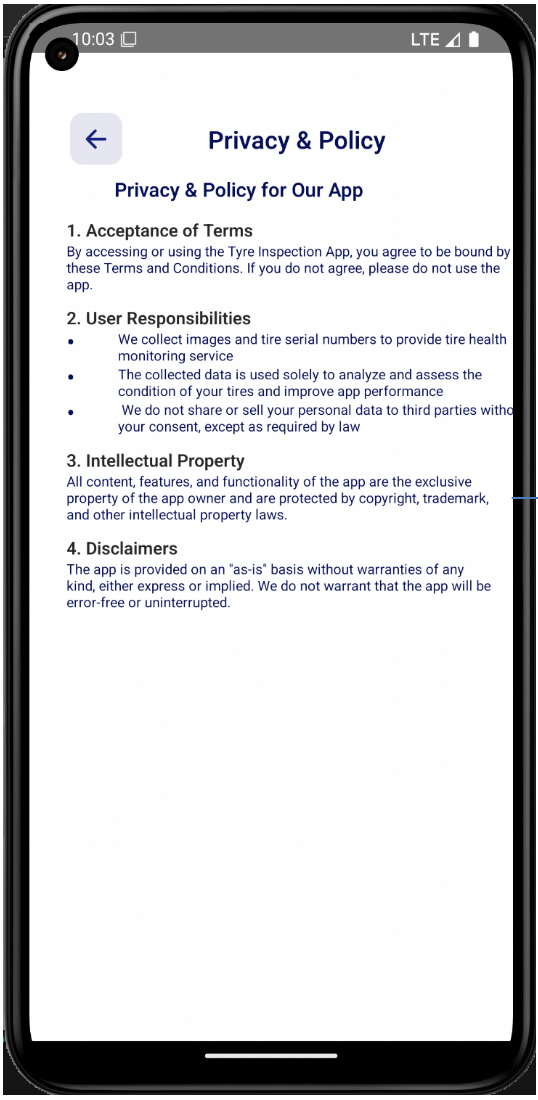


Figure 5.20: Notification Toggle Screen

1. Toggle switch to turn on/off notifications on the home screen

5.21 Privacy policy Page:

The privacy policy regarding the application is written on the privacy and policy page.



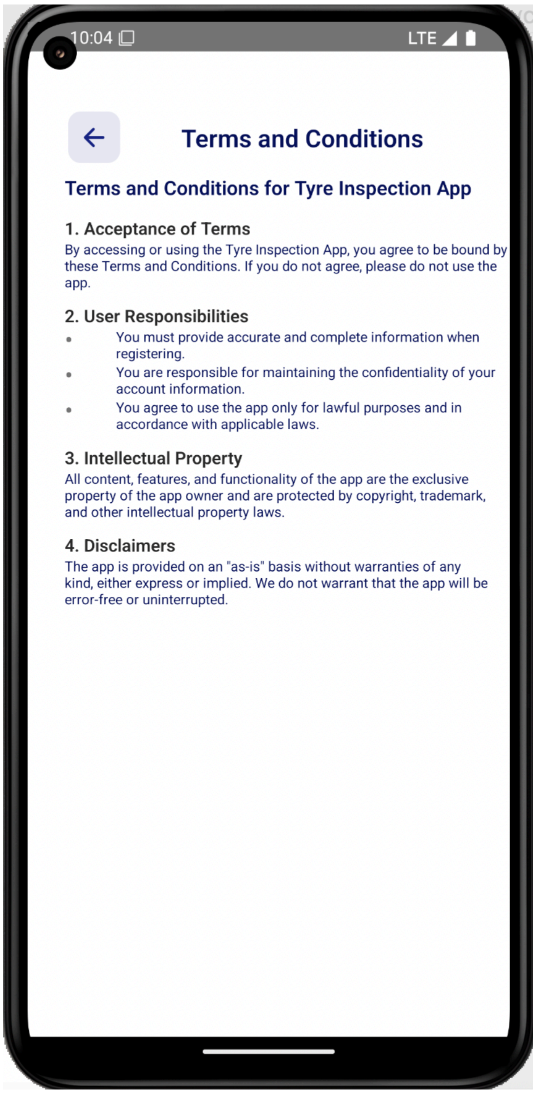
1

Figure 5.21: Privacy Policy Screen

1. Application’s privacy policy

5.22 Terms and condition Page:

The Application's terms and condtions which the user is supposed to agree is shown on the designated page.



1

Figure 5.22: Terms & Conditions Screen

1. Application's Terms and conditions

CHAPTER 6

CONCLUSION AND RECOMMENDATIONS

6.1 Conclusion:

The deep learning application developed in this project for tire health monitoring is successfully used in real world practice to classify tire conditions with high accuracy for vehicle maintenance and safety. Based on this system, our model, which is based on *MobileNetV2* model, achieved 94% accuracy, making it a reliable and efficient tool for tire health assessment. Fine tuning and early stopping techniques, which model optimization techniques, were used to improve performance and to ensure the model is generalizing well to unseen data. Finally, the model was validated through integrating a confusion matrix and classification report to distinguish between different tire conditions.

Significantly improving tire maintenance practices, the system is integrated into mobile applications and fleet management platforms. The system offers real time tire health assessment, reducing tire related accidents and costly downtime, and optimizing vehicle performance. Besides, the proactive monitoring feature cuts down on emissions, reduces expenses and tires wear and tear and enhances overall safety and environmental features. For instance, the app includes an input feature for the serial number which helps users to simply track and manage the tire information, hence adding the functionality to the system.

At the same time, this project proves the possibility of applying machine learning in vehicle maintenance, and reveals the necessity of continuous innovation in this area. The system combines real time data, machine learning and mobile technology to revolutionize tire health monitoring, making monitoring not only more accessible, more reliable and more accessible. This system provides a foundation for future improvements and larger applications in the automotive field, opening the door for AI driven solution to vehicle safety and maintenance.

6.2 Recommendations

6.2.1 Integration with IoT Sensors:

For the system to be more accurate and real time, IoT (Internet of Things) sensors should be integrated into the system. IoT is a network of interrelated devices that connect and exchange data with other IoT devices and the cloud. If these sensors were to be used they would be able to monitor tire pressure, temperature and wear in real time, giving a more complete view of the tire's health. The deep learning model could also improve system accuracy and reliability by combining its analysis with sensor data.

6.2.2 Expansion of Tire Condition Categories:

The current model classifies tires into four conditions, but extending their classification categories to include several more detailed tire health states, like different levels of tread wear or indications of tire damage would make their predictions more accurate. It would result in more actionable insights for the users to take decisions on tires maintenance and replacement.

6.2.3 Multi-Language Support in the Mobile

App: Because of this, support of multi-language in mobile application promises to make mobile app more accessible and usable for various user groups. This would guarantee that users with different regions and different linguistic backgrounds can deal with the system simply and enjoy tire health monitoring.

6.2.4 Data Enrichment and Continuous Learning:

Over time the model will be updated with new tire images and conditions from different manufacturers and environments to stay current and make the accuracy of the model better. Additionally, periodic model updates with additional data would allow the system to keep updated with changing tire technologies and conditions.

6.2.5 Collaboration with Tire Manufacturers:

Working together with tire manufacturers would allow the creation of a more tailored model, incorporating brand and model specific data and features about the tire from the various tire brands. First, this would help the model to detect tire specific issues and recommend personalized maintenance.

REFERENCES

[1] Z. Márton, I. Szalay, and D. Fodor, "Convolutional Neural Network-Based Tire Pressure Monitoring System," IEEE Access, vol. 11, pp. 12345-12353, 2023. [DOI: if available]

[2] "Tire Pressure Monitoring System: Pros & Cons," CarFromJapan, 2023. [Online]. Available: https://carfromjapan.com/article/car-maintenance/tire-pressure-monitoring-system-pros-cons/. [Accessed: Dec. 5, 2024].

[3] J. Y. Han, J. H. Kwon, S. Lee, K. C. Lee, and H. J. Kim, "Experimental evaluation of tire tread wear detection using machine learning in real-road driving conditions," IEEE Access, vol. 11, pp. 32996-33004, 2023. [DOI: if available]

[4] "Computer Vision and Image Processing," Northwestern University, [Online]. Available: https://www.mccormick.northwestern.edu/electrical-computer/academics/graduate/masters/specializations/computer-vision-and-image-processing.html#:~:text=Image%20processing%20algorithms%20are%20used,recognize%20and%20categorize%20image%20data. [Accessed: Dec. 5, 2024].

[5] W. Salehi, S. Khan, G. Gupta, B. I. Alabduallah, A. Almjally, H. Alsolai, T. Siddiqui, and A. Mellit, "A study of CNN and transfer learning in medical imaging: Advantages, challenges, future scope," Sustainability, vol. 15, no. 7, p. 5930, 2023. [DOI: if available]{

[6] "Sharpen Watson Visual Recognition Results," IBM Blog, [Online]. Available: https://www.ibm.com/blog/sharpen-watson-visual-recognition-results/. [Accessed: Dec. 5, 2024].

[7] "Cloud Vision API," Google Cloud, [Online]. Available: https://cloud.google.com/vision?hl=en. [Accessed: Dec. 5, 2024].
[8] Warcoder, "Tyre Quality Classification," Kaggle, 2021. [Online]. Available: https://www.kaggle.com/datasets/warcoder/tyre-quality-classification. [Accessed: 05-Dec-2024].
[9] Aghapoor, A., "COCO Dataset Tutorial: Image Segmentation," Kaggle, 2021. [Online]. Available: https://www.kaggle.com/code/armanasgharpoor1993/coco-dataset-tutorial-image-segmentation. [Accessed: 05-Dec-2024].

APPENDICES

APPENDIX A:. Dataset Validation by Tire Experts

Expert validation was done to ensure the quality and accuracy of the dataset. Images with tires with punctures, tread wear, and cracks were correctly annotated by the images, and the correctness of the damage labels was verified by experts. The categorization of damage severity was also cross verified by experts, to ensure that the level of tire damage (e.g. minor, moderate or severe) was correctly labelled:

1. **Label Verification:**

- Experts reviewed the images to verify the correctness of the damage labels, ensuring that the images of tires with punctures, tread wear, and cracks were correctly annotated.
- Experts also cross-verified the categorization of damage severity, ensuring that the level of tire damage (e.g., minor, moderate, severe) was accurately labeled.

2. **Damage Classification Review**:

- Tire experts also provided suggestions to improve the criteria for classifying the types of tires.. For instance, the following types of tire damage may not be clearly noticeable without professional help; slight surface abrasion. This validation process was helpful in establishing that the data set was diverse in terms of various classes of damages in real-world application.