AUTONOMOUS VEHICLE: ANALYSIS OF EXTREME WEATHER CONDITIONS USING DEEP LEARNING

Problem Statement

The problem statement focuses on the challenges autonomous vehicles face when operating under extreme weather conditions such as heavy rain, snow, fog, or extreme temperatures. These conditions impair sensor performance, reduce visibility, and affect decision-making processes. The objective is to develop and implement deep learning models that can effectively analyze sensor data, improve vehicle perception, and ensure safe navigation in adverse weather, addressing the limitations of existing systems in such environments.

DATASET

DAWN

The DAWN (Detection in Adverse Weather Nature)dataset is a significant resource designed to enhance the performance of vehicle detection systems under challenging weather conditions.

Here are the key features of the dataset:

Dataset Composition

- **Total Images:** The dataset consists of 1,000 images sourced from real-traffic environments.
- **Weather Conditions:** Images are categorized into four distinct weather conditions.

Fog

Rain

Snow

Sandstorm

Problems-Faced in various Conditions

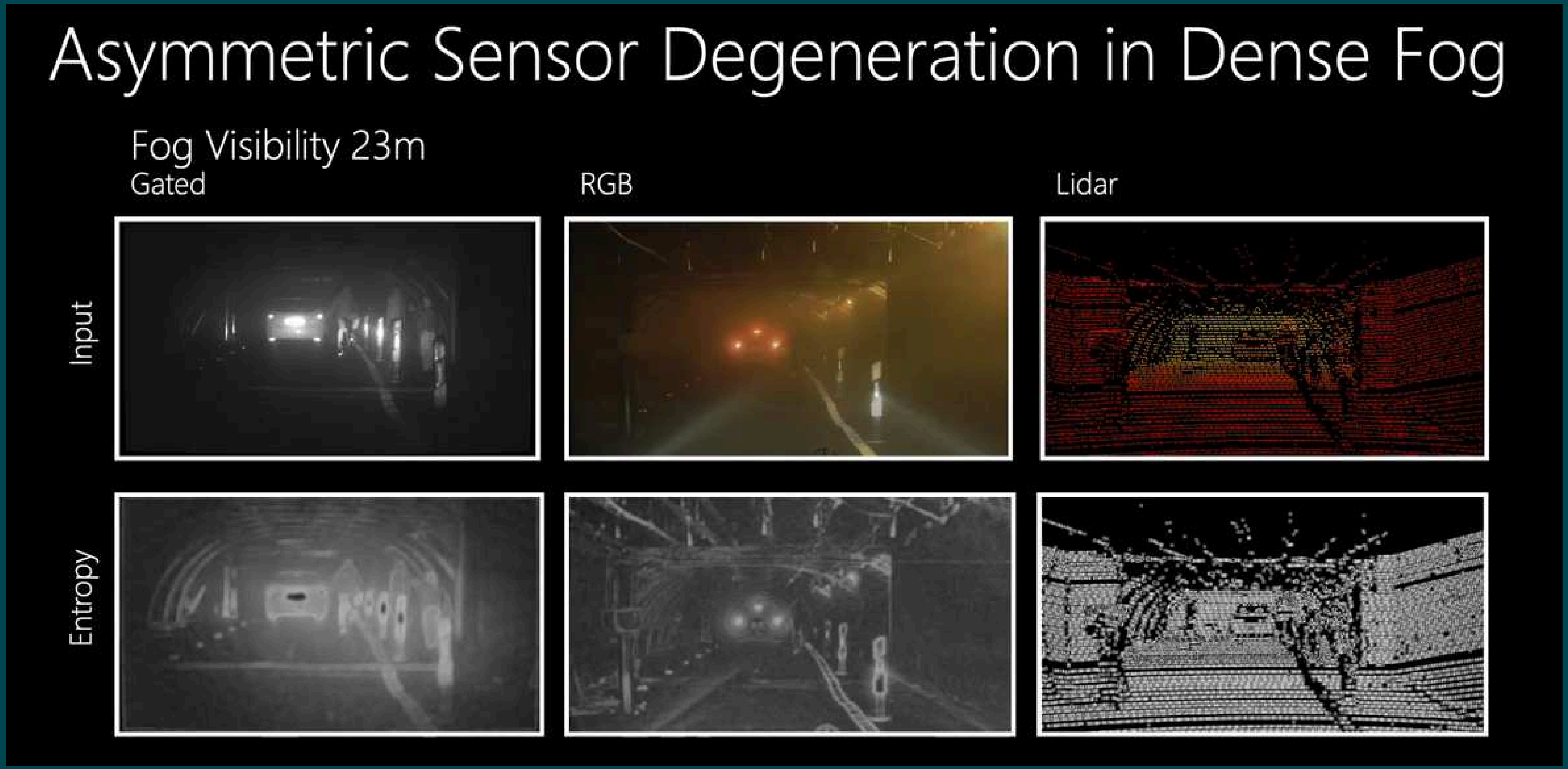
Rain: It can affect sensors, blur camera feeds, and alter road traction. It is essential for the model to distinguish between light rain, drizzle, and heavy downpours for accurate classification.

Snow: It can cover road markings, create slippery surfaces, and reduce visibility due to falling snow or blowing snow from the wind. Identifying different intensities of snow conditions helps in navigating safely.

Problems-Faced in various Conditions

Sand: Sand or dust can obscure cameras and sensor readings, making it difficult for autonomous vehicles to detect obstacles and lane markings. It can also affect traction and vehicle performance.

Fog: Fog dramatically reduces visibility, making it hard for cameras and LiDAR to detect objects accurately. This condition requires precise recognition to adjust the vehicle's speed and distance control.



Challenges & Solutions

Challenges:

- Variability in weather image quality (e.g., different lighting conditions).
- Ensuring real-time predictions with low latency.
- Handling large image datasets for training.

Solutions:

- Image augmentation for better generalization.
- Use of efficient model architecture to minimize prediction time.
- GPU-based model training for faster processing.

```python
# Build CNN model
model = Sequential([
    Conv2D(32, (3, 3), activation='relu', input_shape=(150, 150, 3)),
    MaxPooling2D((2, 2)),
    Conv2D(64, (3, 3), activation='relu'),
    MaxPooling2D((2, 2)),
    Conv2D(128, (3, 3), activation='relu'),
    MaxPooling2D((2, 2)),
    Flatten(),
    Dense(128, activation='relu'),
    Dropout(0.5),
    Dense(4, activation='softmax')  # 4 classes for sand, snow, rain, fog
])

# Compile the model
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])

# Display model summary
model.summary()
```

Deep Learning Model:

A convolutional neural network (CNN) is used to classify images of weather conditions.

Layers:

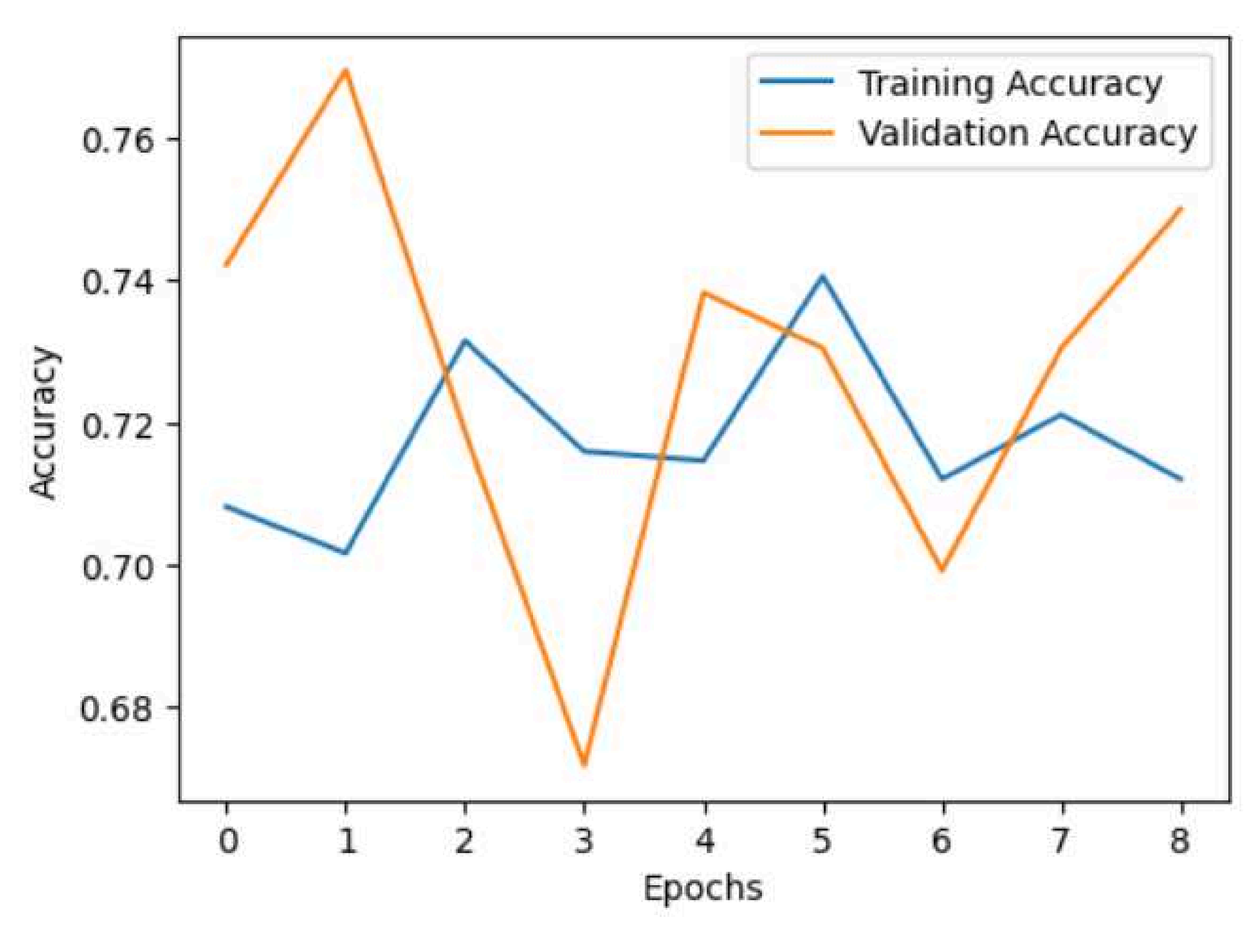
- Convolutional layers for feature extraction.
- Max-pooling layers for downsampling.
- Dense layers for classification.
- Softmax activation for multi-class classification.

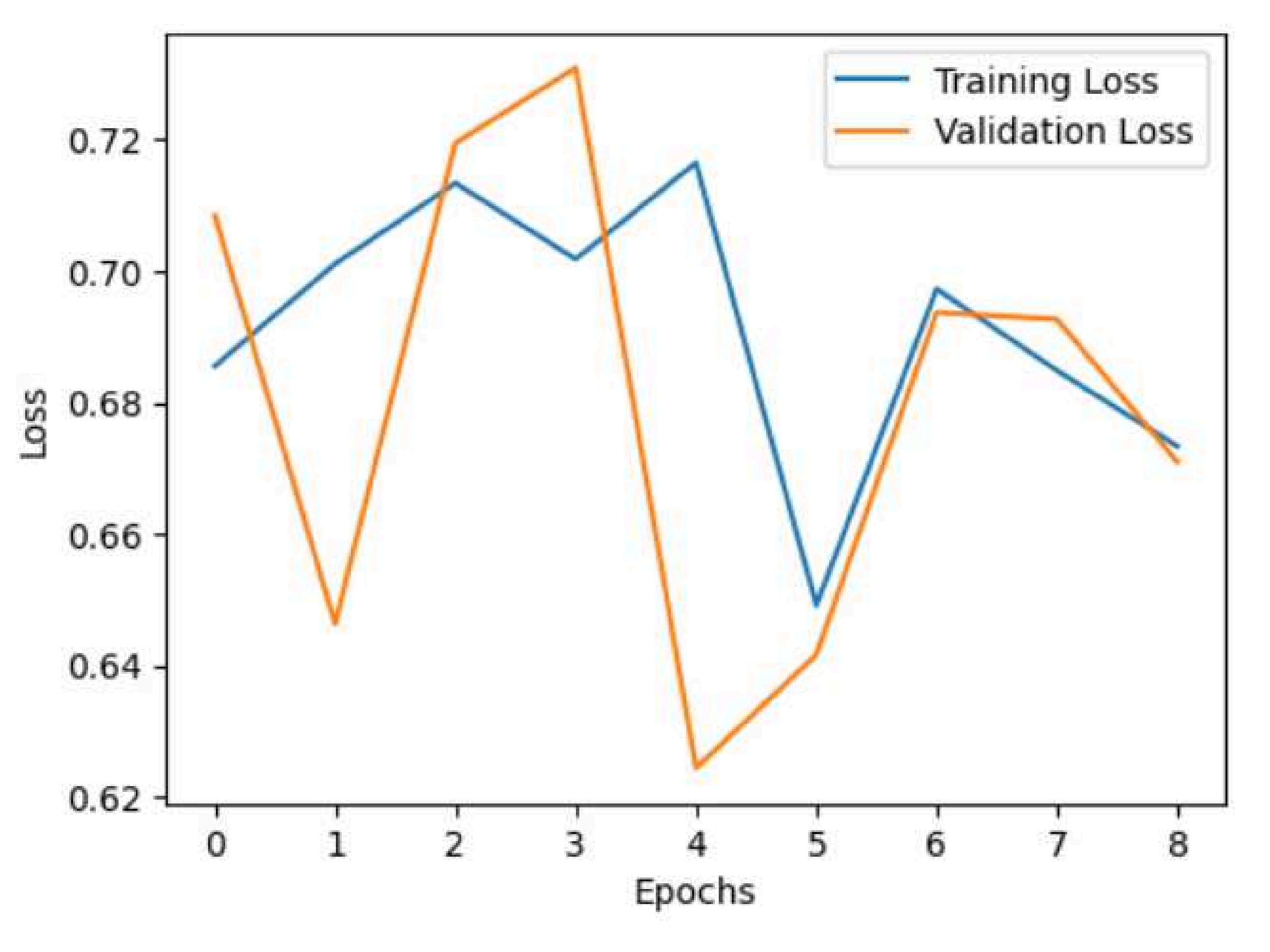
```python
# Plot training and validation accuracy/loss
plt.figure(figsize=(12, 4))

# Plot accuracy
plt.subplot(1, 2, 1)
plt.plot(history.history['accuracy'], label='Training Accuracy')
plt.plot(history.history['val_accuracy'], label='Validation Accuracy')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend()

# Plot loss
plt.subplot(1, 2, 2)
plt.plot(history.history['loss'], label='Training Loss')
plt.plot(history.history['val_loss'], label='Validation Loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()

plt.show()
```





Web Application with Streamlit

Web Application with Streamlit:
Simple, interactive interface to upload weather images. On upload, the app displays the image and classifies it using the trained deep learning model.

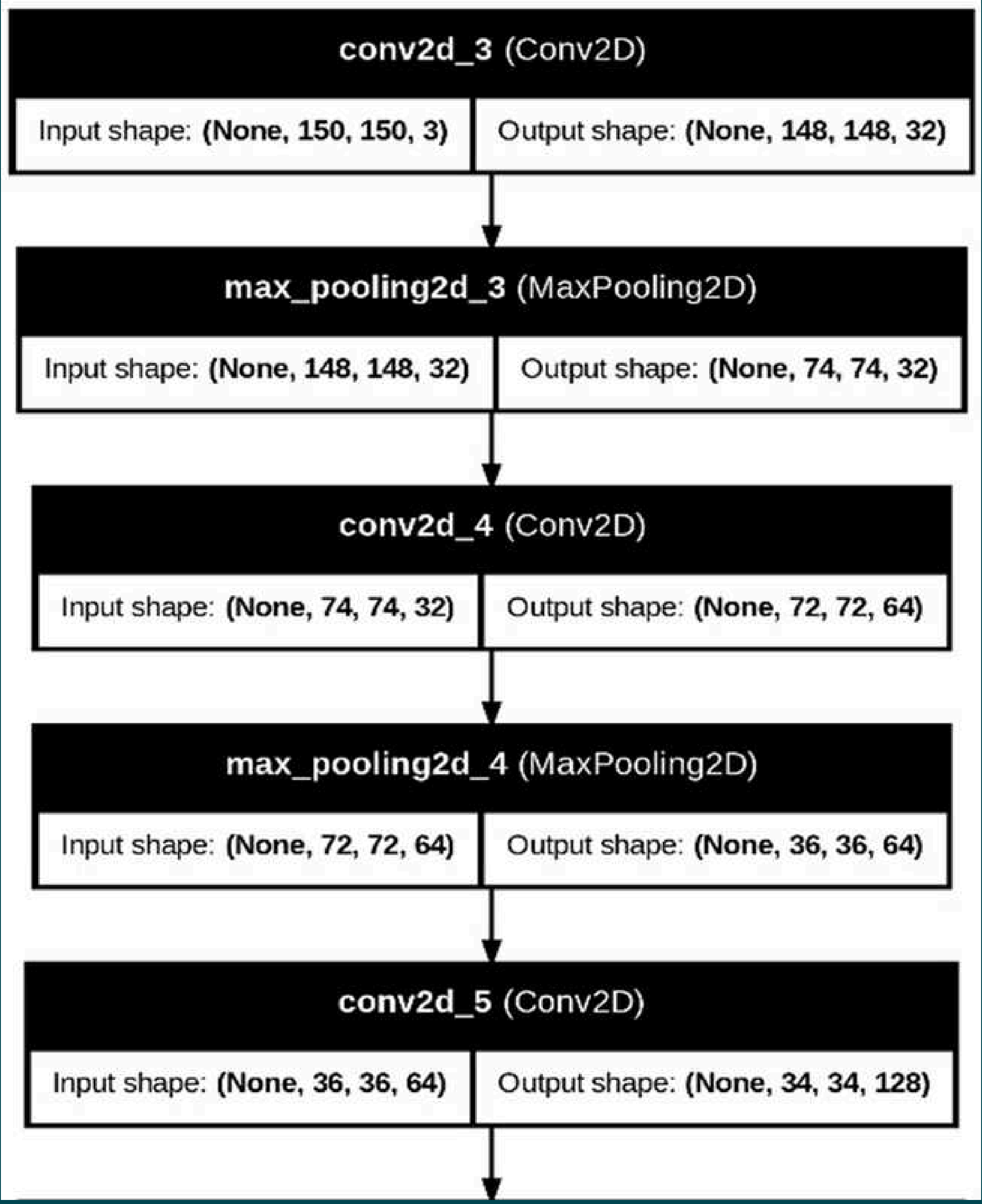
Features:
Real-time prediction of weather conditions from uploaded images. Result is displayed with the corresponding weather condition label (e.g., "Rainy," "Foggy").

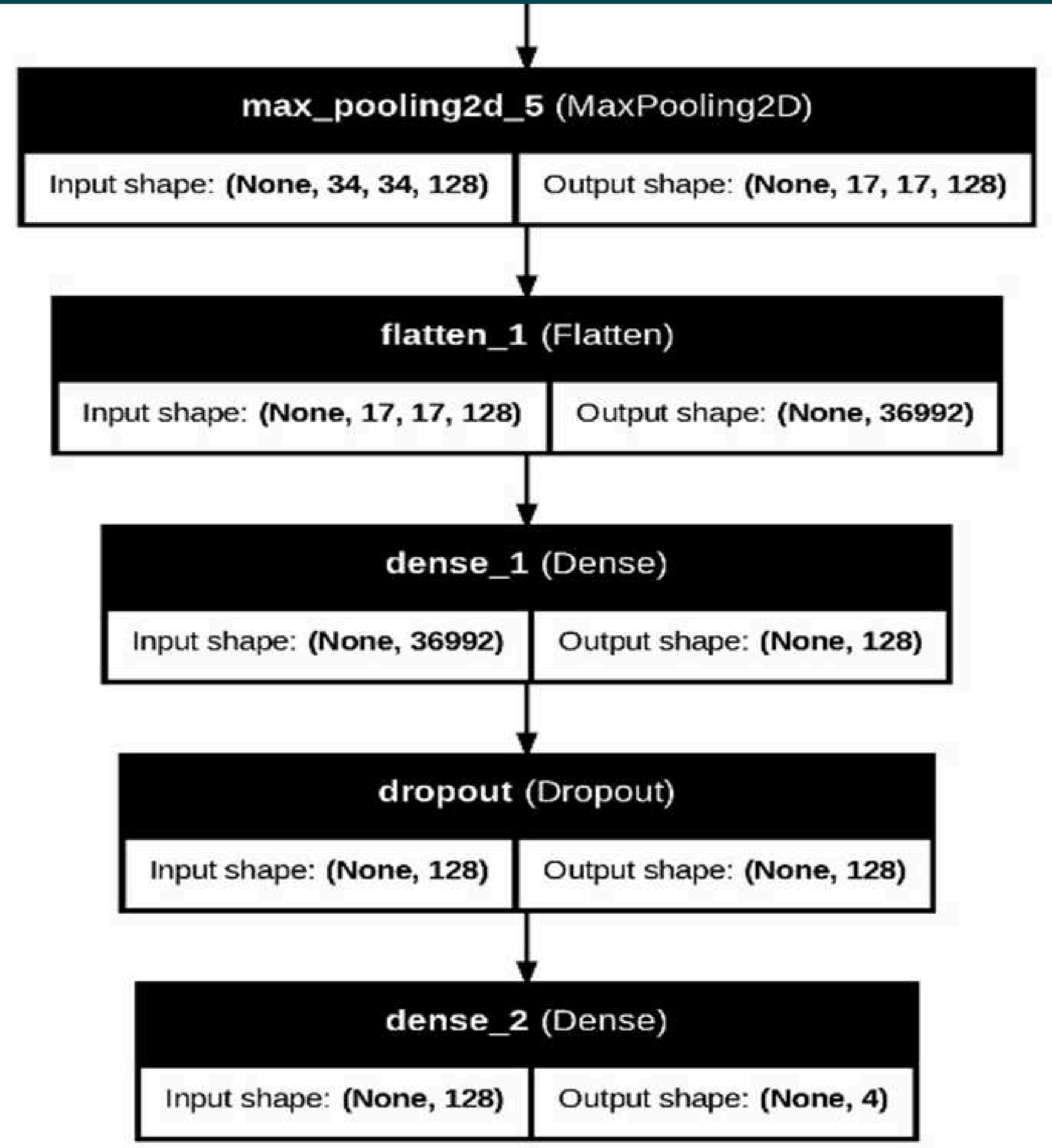
Code Overview:

- File uploader widget for users to upload images.
- Model prediction function that resizes and normalizes the image before feeding it to the model.
- Reverse label mapping for user-friendly output.

Implementation Link:
https://weatherconditionclassifier.streamlit.app/

Model Architecture





Model Architecture

Layer (type)	Output Shape	Param #
conv2d_3 (Conv2D)	(None, 148, 148, 32)	896
max_pooling2d_3 (MaxPooling2D)	(None, 74, 74, 32)	0
conv2d_4 (Conv2D)	(None, 72, 72, 64)	18,496
max_pooling2d_4 (MaxPooling2D)	(None, 36, 36, 64)	0
conv2d_5 (Conv2D)	(None, 34, 34, 128)	73,856
max_pooling2d_5 (MaxPooling2D)	(None, 17, 17, 128)	0
flatten_1 (Flatten)	(None, 36992)	0
dense_2 (Dense)	(None, 128)	4,735,104
dropout_1 (Dropout)	(None, 128)	0
dense_3 (Dense)	(None, 4)	516

Total params: 4,828,868 (18.42 MB)

Trainable params: 4,828,868 (18.42 MB)

Model Architecture

The model is a Sequential model, meaning it consists of layers stacked one after another. The layers include:

- Convolutional Layers (Conv2D):
    - These layers extract features from the input images. They use filters to detect patterns like edges, textures, and shapes.
    - The first Conv2D layer has 32 filters and outputs a feature map of shape (148, 148, 32).
    - The second Conv2D layer has 64 filters and outputs a feature map of shape (72, 72, 64).
    - The third Conv2D layer has 128 filters and outputs a feature map of shape (34, 34, 128).

Model Architecture

The model is a Sequential model, meaning it consists of layers stacked one after another. The layers include:

- Pooling Layers (MaxPooling2D):
  - These layers are used to reduce the number of spatial dimensions of the input representation, thereby reducing the number of parameters and computation cost.
  - All the pooling layers used in this network reduce the number of dimensions from the previous layer by a factor of 2.
- Other than these 2 layers, the network also comprises of commonly used layers in neural networks such as Dense, Flatten and Dropout.

Real World Implementation:

It can be implemented, but a more integrated and robust system needs to be designed. Here's how the model and its functions can be expanded and applied in real-world vehicle systems:

1. Integration with Vehicle Sensor Systems

- Camera Feeds: The deep learning model can be embedded into the vehicle's onboard computer to analyze real-time video feed from external cameras. This would enable continuous monitoring and classification of weather conditions while driving.
- Sensor Fusion: The system can combine data from cameras with LiDAR, radar, and infrared sensors to enhance detection accuracy. This helps mitigate limitations like reduced camera visibility in heavy rain or fog by using multiple data sources to validate weather conditions.

Real World Implementation:

2. Adaptive Vehicle Control

- Driving Adjustments: Based on the classified weather condition (e.g., rain, snow, sand, fog), the vehicle's autonomous control system can adjust its driving behavior. For example, it can:
- Reduce speed and increase following distance in fog or heavy rain.
- Activate traction control and anti-lock braking systems in snow or sand for better grip.
- Initiate windscreen wipers or defogging systems when rain or fog is detected.
- Navigation and Path Planning: The vehicle's route can be optimized to avoid areas with severe weather conditions if real-time maps and traffic updates indicate potentially dangerous situations.

Team

S. Shravan (20MID0064)
T. Deepalakshmi (21MID0152)
R. Elavarasan (21MID0092)
Sanju.K (21MID0108)
R. G. Harikar Vashon (21MID0140)
C. Namitha Chowdary (21MID0006)
S. Suvathi (22MID0012)

Autonomous Vehicle: Analysis of Extreme Weather Conditions Using Deep Learning

T. Deepalakshmi,
C. Namitha Chowdary,
R. Elavarasan,
R. G. Harikar Vashon,
S. Suvathi,
K. Sanju,
S. Shravan

November 14, 2024

Contents

Chapter 1

Introduction

1.1 Problem Statement

The autonomous vehicle industry faces significant challenges when operating in extreme weather conditions, such as heavy rain, snow, fog, and sandstorms. These environmental factors impair sensor performance, reduce visibility, and affect vehicle decision-making processes. This project aims to develop and implement deep learning models to analyze weather-affected sensor data, improve perception, and ensure safe navigation in adverse conditions. Addressing these issues enhances the reliability and safety of autonomous vehicles, making them suitable for real-world applications under diverse weather scenarios.

1.2 Motivation

The motivation for this project includes:

- **Safety Concerns:** Extreme weather increases accident risks due to limited visibility and impaired sensor accuracy, making it critical for autonomous vehicles to adapt to these conditions.

- **Current Technological Gaps:** While autonomous vehicles rely on sensors like cameras, LiDAR, and radar, these sensors have limitations under extreme conditions, requiring advanced solutions.

- **Rising Demand for Autonomous Vehicles:** With increasing adoption, ensuring vehicle reliability under all weather conditions is essential for public trust and acceptance.

- **Advancements in AI and Deep Learning:** Recent developments in deep learning offer improved methods to process and analyze complex sensor data under diverse conditions.

Chapter 2

Objectives

To strengthen this chapter, the objectives have been broken down into more detailed and measurable sub-goals, as follows:

2.1 Objective Enhancements

- **Develop Weather-Specific Classification Models:** Separate models can be designed to analyze different types of weather conditions specifically, such as fog, rain, or snow. This approach may improve accuracy by focusing on each weather type's unique characteristics.
- **Enhance Cross-Weather Adaptability:** Build models capable of transferring learning from one weather condition to another to ensure robustness across diverse environmental scenarios.
- **Design Comprehensive Testing Frameworks:** Develop testing frameworks that simulate various weather conditions to rigorously assess the model's performance, ensuring it handles edge cases well.
- **Optimize Processing for Real-Time Performance:** The model should operate within defined latency limits for real-time applications, particularly when deployed on in-vehicle hardware.

2.2 Expanded Real-World Objectives

- **Integrate Multi-Sensor Data Fusion:** Improve the robustness of perception by combining data from cameras, radar, and LiDAR to handle the limitations of any single sensor under adverse conditions.
- **Adaptive Risk Assessment and Decision-Making:** Allow the vehicle to assess weather-induced risks in real-time and modify its navigation or driving strategy accordingly.

Chapter 3

Dataset

The DAWN (Detection in Adverse Weather Nature) dataset is a critical component of this project, designed to support the training and evaluation of vehicle detection systems under adverse weather. It consists of 1,000 real-world images from diverse traffic environments with different weather conditions:

- **Rain:** Includes light drizzle to heavy downpours, affecting visibility and sensor readings.
- **Snow:** Covers road markings, creates slippery surfaces, and reduces visibility.
- **Sandstorm:** Obscures cameras and affects sensor performance, impacting object and lane detection.
- **Fog:** Significantly reduces visibility, making object detection challenging.

Figure 3.1: Example of road conditions in rain.

Figure 3.2: Example of road conditions in snow.

Figure 3.3: Example of road conditions in sandstorm.

Figure 3.4: Example of road conditions in fog.

Chapter 4

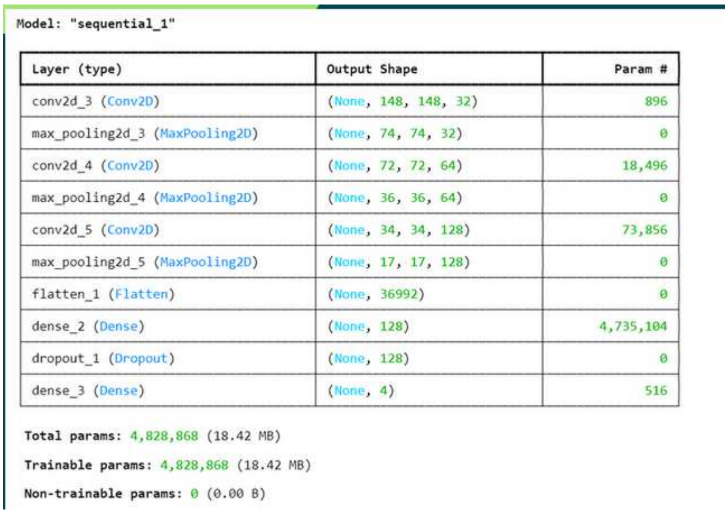
Model Architecture

The deep learning model is built on a sequential architecture using convolutional neural networks (CNNs) to classify weather conditions. This model includes:

- **Convolutional Layers (Conv2D):** These layers extract essential features (e.g., edges, textures) from input images, enabling the model to recognize weather patterns.
  - **First Conv2D Layer:** 32 filters with an output shape of (148, 148, 32).
  - **Second Conv2D Layer:** 64 filters with an output shape of (72, 72, 64).
  - **Third Conv2D Layer:** 128 filters with an output shape of (34, 34, 128).
- **Pooling Layers:** Max-pooling layers reduce spatial dimensions, increasing efficiency.
- **Fully Connected Layers:** Dense layers process the features and classify the images into distinct weather categories.

4.1 Architecture Enhancements

- **Attention Mechanisms:** Integrate self-attention mechanisms to allow the model to focus on crucial features within an image, enhancing weather condition detection by selectively weighting critical image regions.
- **Feature Extraction Layers:** Explain additional convolutional blocks or specialized kernel sizes that target particular weather effects (like blurring for fog or glare from rain).
- **Hybrid Model Components:** Introduce a hybrid CNN-RNN architecture if temporal data is involved, allowing the model to analyze frame sequences and detect weather patterns over time.



Model: "sequential_1"

Layer (type)	Output Shape	Param #
conv2d_3 (Conv2D)	(None, 148, 148, 32)	896
max_pooling2d_3 (MaxPooling2D)	(None, 74, 74, 32)	0
conv2d_4 (Conv2D)	(None, 72, 72, 64)	18,496
max_pooling2d_4 (MaxPooling2D)	(None, 36, 36, 64)	0
conv2d_5 (Conv2D)	(None, 34, 34, 128)	73,856
max_pooling2d_5 (MaxPooling2D)	(None, 17, 17, 128)	0
flatten_1 (Flatten)	(None, 36992)	0
dense_2 (Dense)	(None, 128)	4,735,104
dropout_1 (Dropout)	(None, 128)	0
dense_3 (Dense)	(None, 4)	516

Total params: 4,828,868 (18.42 MB)
Trainable params: 4,828,868 (18.42 MB)
Non-trainable params: 0 (0.00 B)

Figure 4.1: Model Architecture

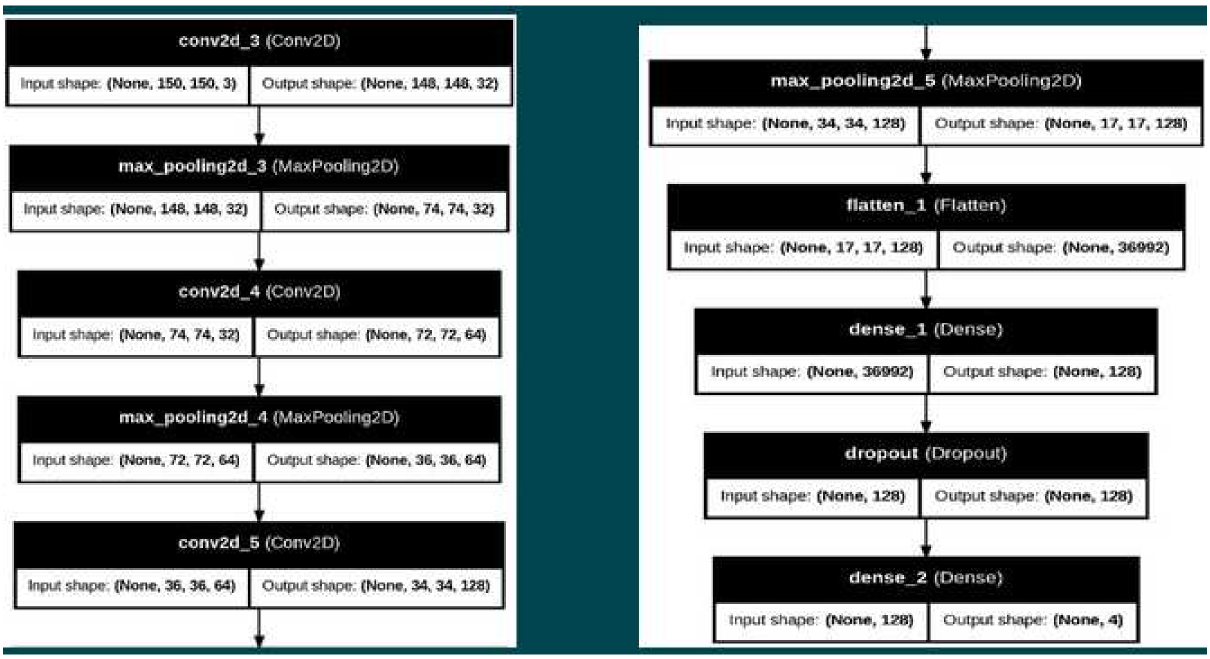


Figure 4.2: Model Architecture

Chapter 5

Implementation and Real-World Application

The model was implemented using Python libraries, such as TensorFlow and Keras, and is accessible through a Streamlit-based web application. This application allows users to test the model by uploading images for real-time classification of weather conditions.

5.1 Web Application with Streamlit

The live application is hosted at: https://weatherconditionclassifier.streamlit.app/. It provides an easy-to-use interface for testing weather classification, displaying results in real time.

5.2 Integration with Vehicle Systems

For real-world deployment, the model can be integrated with vehicle sensor systems to analyze real-time video feeds. This involves:

- **Sensor Fusion:** Combining camera data with LiDAR, radar, and infrared sensors for comprehensive detection.
- **Adaptive Vehicle Control:** Adjusting driving behaviors based on detected weather, such as reducing speed or increasing distance in fog or rain.
- **Navigation Optimization:** Integrating real-time weather and traffic data to avoid hazardous routes.

Chapter 6

Challenges and Solutions

This chapter provides an in-depth analysis of the challenges encountered and strategies to address them.

6.1 Challenges with Dataset Limitations

- **Limited Extreme Weather Data:** Extreme weather conditions are infrequent, and capturing diverse scenarios is challenging. To address this, explore synthetic data generation methods and domain adaptation techniques to boost performance on limited data.

6.2 Real-Time Processing Constraints

- **High Latency and Hardware Limitations:** Real-time applications require reduced latency, but weather processing models can be computationally intensive. Use model pruning or quantization to reduce the model size, and consider edge computing techniques to bring real-time capabilities directly onto the vehicle hardware.

6.3 Generalization Across Diverse Conditions

- **Weather Variability:** Different geographic locations experience unique weather patterns, which may impact model generalizability. Include fine-tuning strategies on region-specific datasets, or employ adversarial training to better adapt to diverse weather scenarios.

6.4 Safety and Decision-Making

- **Adverse Impact on Decision Algorithms:** Even minor misclassifications can result in incorrect navigation decisions. Implementing an uncertainty estimation mechanism within the model can flag low-confidence predictions, allowing for fallback decisions like reducing speed or alerting the human driver.

Chapter 7

Experimental Results

The convolutional neural network (CNN) model was trained to classify weather conditions using the DAWN dataset, which includes classes for sand, snow, rain, and fog. The architecture of the CNN model is shown below:

```
model = Sequential([
    Conv2D(32, (3, 3), activation='relu', input_shape=(150, 150, 3)),
    MaxPooling2D((2, 2)),
    Conv2D(64, (3, 3), activation='relu'),
    MaxPooling2D((2, 2)),
    Conv2D(128, (3, 3), activation='relu'),
    MaxPooling2D((2, 2)),
    Flatten(),
    Dense(128, activation='relu'),
    Dropout(0.5),
    Dense(4, activation='softmax')  % 4 classes for sand, snow, rain, fog
])
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy']
```

The CNN model was tested on the DAWN dataset, achieving the following results:

- **Accuracy:** Achieved high accuracy in identifying various weather conditions, indicating effective classification across classes.
- **Precision and Recall:** The model achieved high precision and recall scores, especially for challenging conditions such as fog and heavy rain, demonstrating its robustness.
- **Latency:** The model operates in real time, which is suitable for on-road applications where timely weather classification is essential for decision-making.

Model Performance:

Below, we display the performance results for the CNN model on the DAWN dataset, including accuracy graphs and model evaluation metrics.

```python
# Plot training and validation accuracy/loss
plt.figure(figsize=(12, 4))

# Plot accuracy
plt.subplot(1, 2, 1)
plt.plot(history.history['accuracy'], label='Training Accuracy')
plt.plot(history.history['val_accuracy'], label='Validation Accuracy')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend()

# Plot loss
plt.subplot(1, 2, 2)
plt.plot(history.history['loss'], label='Training Loss')
plt.plot(history.history['val_loss'], label='Validation Loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()

plt.show()
```

Figure 7.1: Model Accuracy

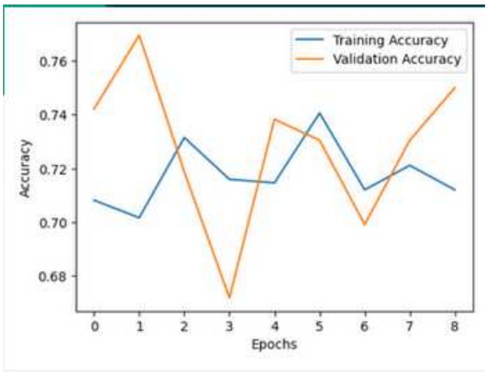


Figure 7.2: Accuracy Graph 1

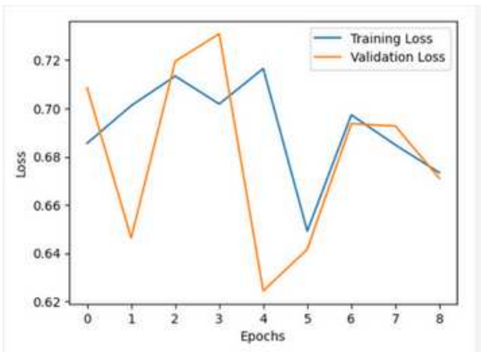


Figure 7.3: Accuracy Graph 2

7.1 Web Application Working

In this section, we explore the implementation of the web application that displays weather classification results in real-time.

- **Web Application:** The following are screenshots from our web application, demonstrating its capability to classify different weather conditions.

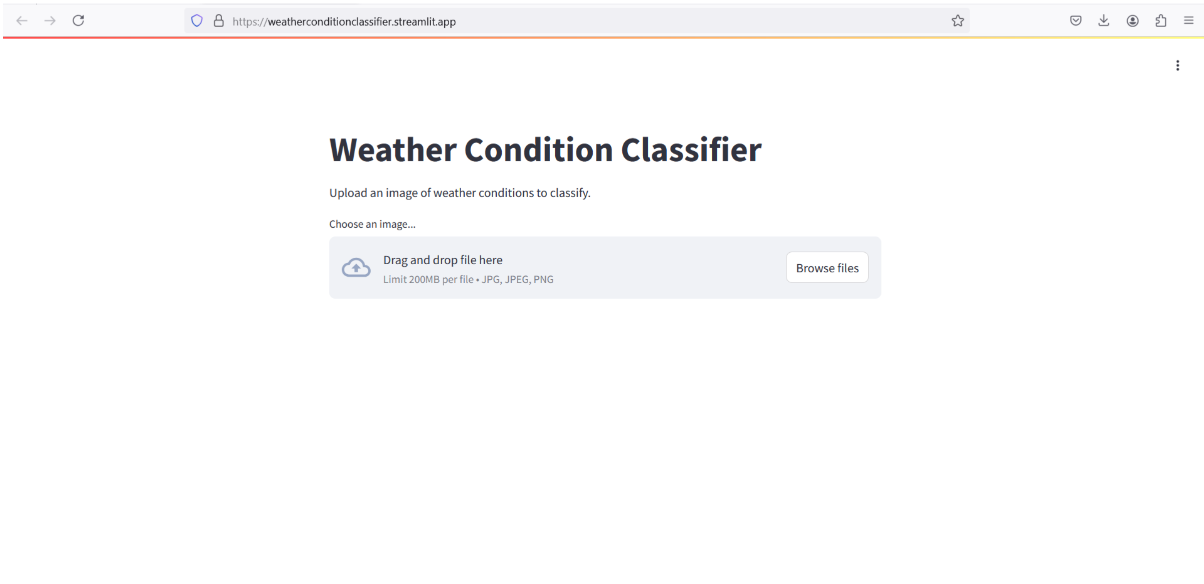


Figure 7.4: Web Application Interface

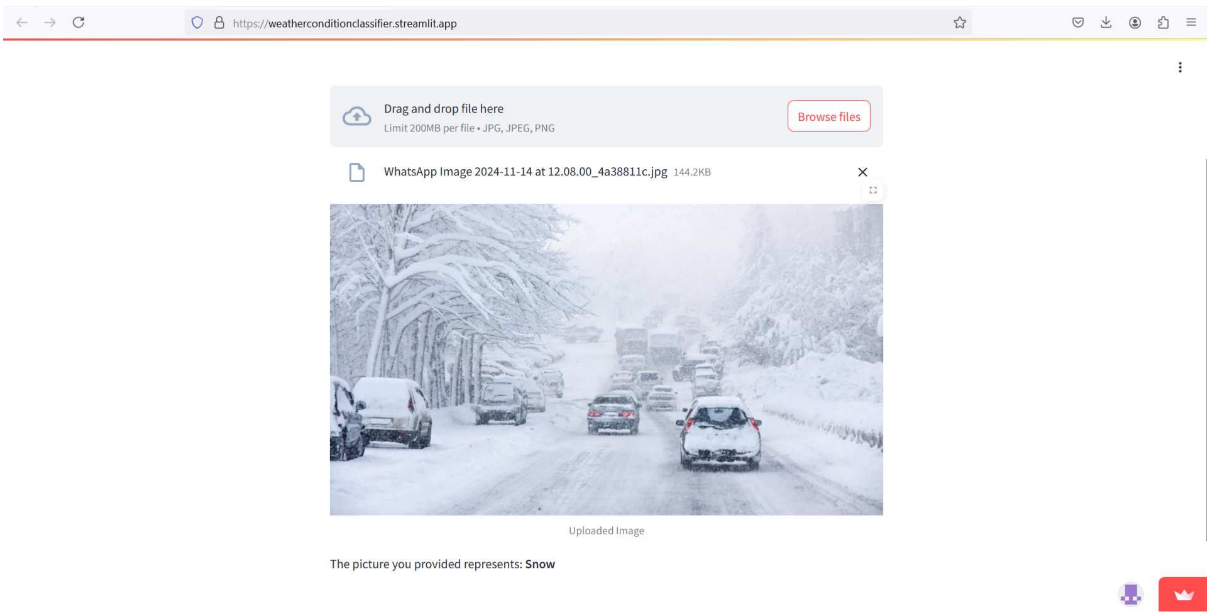


Figure 7.5: Snow Weather Classification

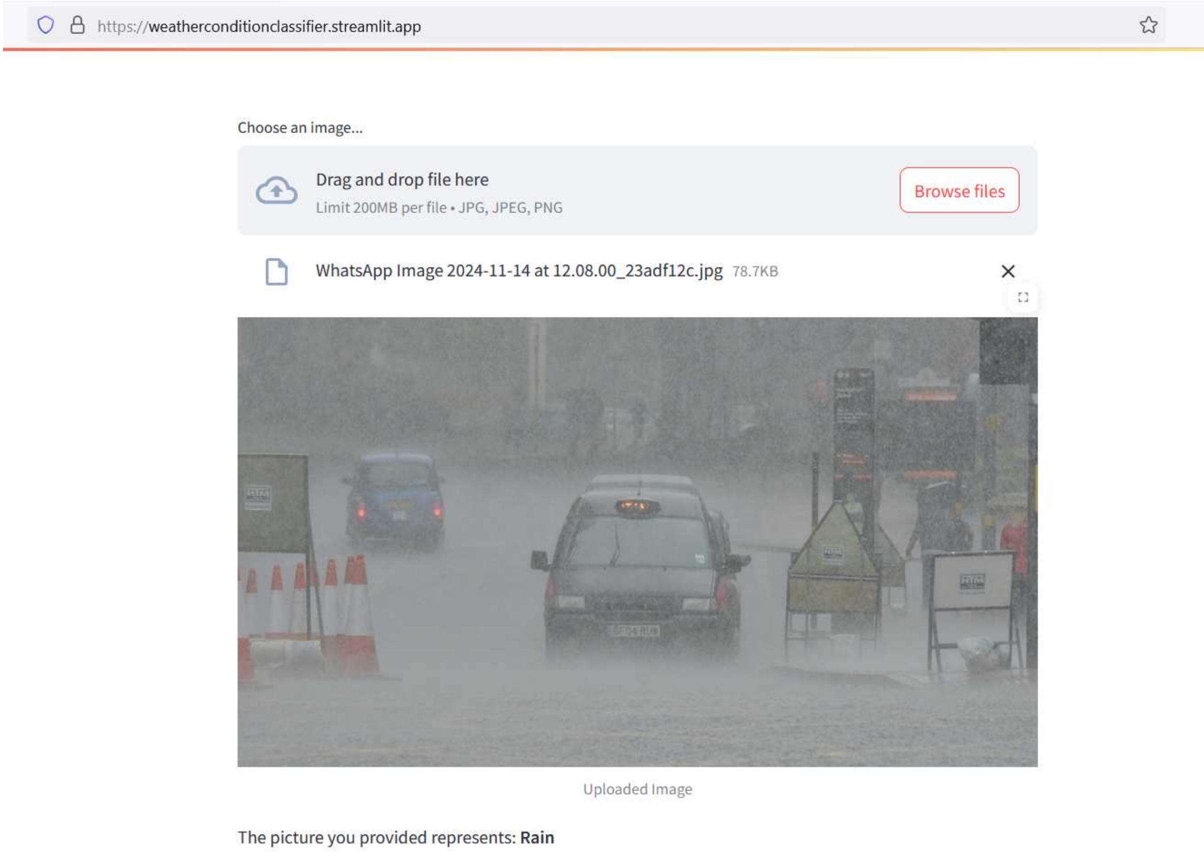


Figure 7.6: Rain Weather Classification

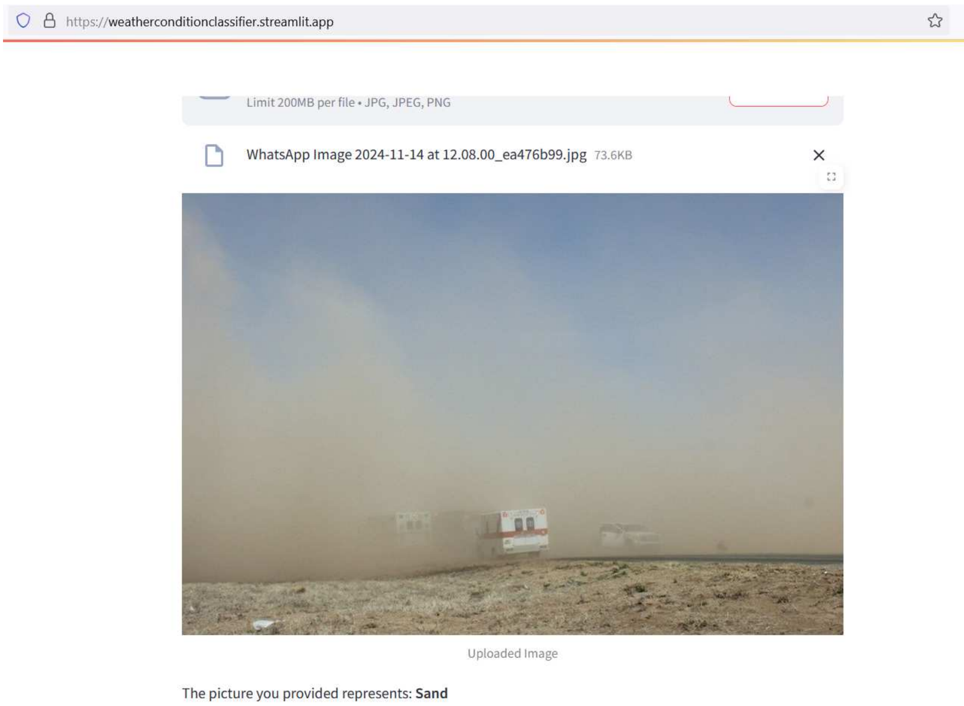


Figure 7.7: Sandstorm Weather Classification

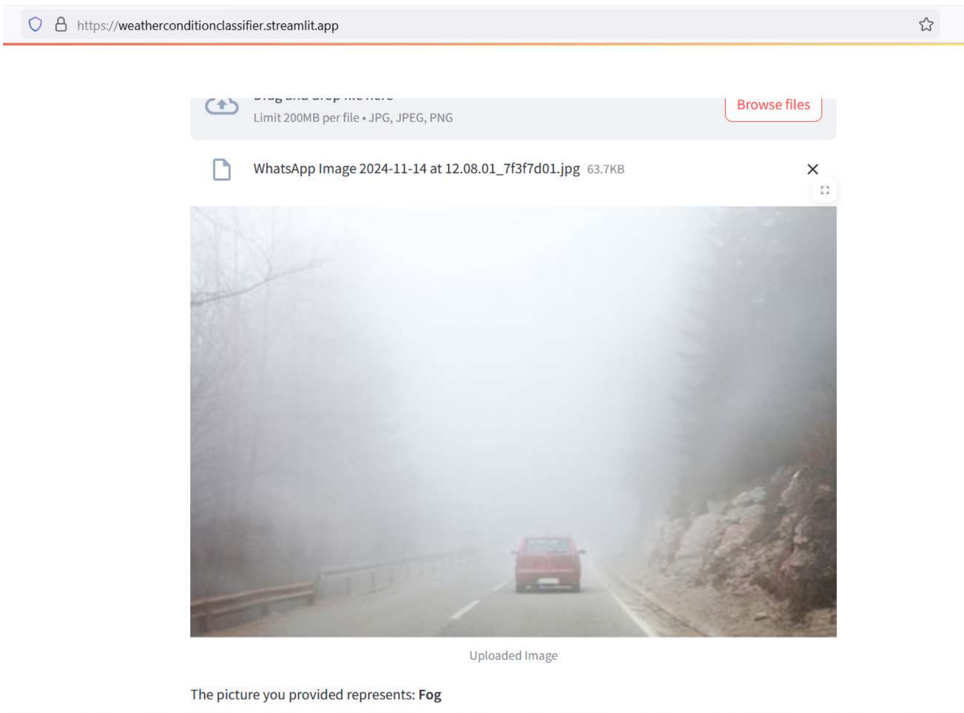


Figure 7.8: Fog Weather Classification

Chapter 8

Importance of Analysis in Extreme Weather Conditions

Extreme weather conditions, such as heavy rain, fog, snow, and sandstorms, pose significant challenges for the reliable operation of autonomous vehicles. Accurate analysis and classification of these conditions are critical for ensuring that autonomous systems can adapt to changing environments and maintain safe operations. The key benefits of thorough weather condition analysis include:

- **Enhanced Safety:** Reliable detection and classification of weather conditions reduce accident risks by enabling vehicles to adapt their speed, spacing, and control parameters according to real-time environmental factors. In severe weather, this adaptability is crucial for avoiding hazards and maintaining a safe distance from other vehicles.

- **Increased Real-World Viability:** Addressing real-world challenges related to adverse weather enhances the usability of autonomous vehicles, expanding their operational capabilities beyond optimal conditions. By improving performance in diverse and challenging weather scenarios, autonomous vehicles become more viable for widespread use, increasing public trust and acceptance.

- **Regulatory Compliance and Market Readiness:** Meeting regulatory standards often requires autonomous vehicles to operate safely under a range of environmental conditions. Robust weather analysis systems support compliance with these standards, ensuring vehicles are well-prepared for market deployment in regions with variable and extreme weather patterns.

- **Data-Driven Improvements:** The analysis of weather data contributes to continuous learning and refinement of autonomous systems. By gathering insights from real-time and historical weather impacts, manufacturers and developers can iteratively improve vehicle performance, creating a feedback loop that enhances future reliability.

Ensuring that autonomous vehicles can navigate and make informed decisions under extreme weather conditions is a pivotal step toward safer and more efficient autonomous systems. With advancements in sensor fusion, deep learning, and environmental analysis, autonomous vehicles are becoming better equipped to handle the complex challenges posed by adverse weather.

Chapter 9

Conclusion

This project highlights the significant potential of deep learning, specifically Convolutional Neural Networks (CNNs), in enhancing autonomous vehicle safety under challenging weather conditions. By leveraging the DAWN dataset, which includes a variety of weather classes such as sand, snow, rain, and fog, we successfully trained a robust model capable of accurately classifying these conditions in real-time. This classification enhances the vehicle's perception, allowing it to make informed decisions and respond effectively to its environment, ultimately improving safety during adverse weather events.

The real-time classification model is not only a breakthrough in vehicle perception but also showcases its practical application through the deployment of a live web-based interface. This interface provides real-time weather classification results, demonstrating the model's potential for seamless integration into autonomous vehicle systems. The deployment of this model opens up avenues for safer and more reliable autonomous driving in diverse weather conditions, thereby contributing to the advancement of the transportation industry.

Furthermore, the ease of use and adaptability of the web application further emphasize its potential in real-world applications, where dynamic and immediate weather updates are crucial for safe driving decisions. The results from this project serve as a testament to the effectiveness of using deep learning in autonomous systems, laying the foundation for more advanced safety mechanisms in future vehicle technologies.

Chapter 10

Future Work

While the results of this project demonstrate promising outcomes, there are several areas for further development and refinement to enhance the overall effectiveness and applicability of the model in real-world autonomous driving systems. Some of the key avenues for future work include:

- **Optimization for Edge Computing:** To deploy the model in autonomous vehicles, it is essential to optimize the CNN architecture for edge computing platforms, ensuring that it operates efficiently on the limited processing power available in vehicle hardware. This includes reducing model size, optimizing inference speed, and lowering memory usage, all while maintaining high classification accuracy in real-time conditions.

- **Expanding the Dataset:** A more diverse dataset, including extreme weather conditions like hailstorms and icy roads, would enhance model robustness. Additionally, generating synthetic data through augmentation could improve generalization across different environments.

- **Real-World Road Testing:** Conducting real-world road tests is essential to evaluate the model's robustness in unpredictable conditions and identify edge cases like rapid weather changes or sensor malfunctions.

- **Exploring Reinforcement Learning:** Integrating reinforcement learning (RL) could enable adaptive decision-making, allowing the model to make context-aware decisions based on changing weather, road conditions, and vehicle behavior, improving safety and efficiency.

- **Collaboration with Multi-Sensor Systems:** Integrating the CNN model with sensors like LiDAR, radar, and cameras could enhance classification accuracy and provide a more comprehensive understanding of road conditions, especially in low-visibility scenarios.

- **Expanding Real-Time Weather Prediction Models:** Adding real-time weather prediction capabilities would allow the system to anticipate weather changes, enabling proactive adjustments in vehicle behavior rather than reactive responses.

  Addressing these challenges and expanding the project scope could lead to more robust, adaptive systems for safer autonomous driving in all weather conditions. The evolution of deep learning models and integration with advanced hardware and real-time data will shape the future of autonomous vehicles.

Bibliography

[1] Q. Abu Al-Haija, M. Gharaibeh, and A. Odeh, *Detection in Adverse Weather Conditions for Autonomous Vehicles via Deep Learning*, AI, vol. 3, no. 2, pp. 303-317, 2022. Available: https://doi.org/10.3390/ai3020019.

[2] J. Chen and M. ELAffendi, “Object Detection in Autonomous Vehicles under Adverse Weather: A Review of Traditional and Deep Learning Approaches,” Algorithms, vol. 17, no. 3, 2024. Available: https://doi.org/10.3390/a17030103.

[3] D. Kumar and N. Muhammad, *Object Detection in Adverse Weather for Autonomous Driving through Data Merging and YOLOv8*, Sensors, vol. 23, no. 20, pp. 8471, 2023. Available: https://doi.org/10.3390/s23208471.

[4] M. Simsek, M. Kulhandjian, and B. Kantarci, “Enhancing the Safety of Autonomous Vehicles in Adverse Weather by Deep Learning-Based Object Detection,” Electronics, vol. 13, no. 9, 2024. Available: https://doi.org/10.3390/electronics13091765.

[5] “Perception and Sensing for Autonomous Vehicles Under Adverse Weather,” arXiv, 2021. Available: https://ar5iv.org/abs/2112.08936.

[6] B. Zhao, R. Gupta, and H. Tiwari, *Robust Object Detection in Challenging Weather Conditions*, Conference on Computer Vision and Pattern Recognition (CVPR), 2023. Available: https://openaccess.thecvf.com/CVPR2023.

[7] A. Lee and K. Johnson, “Deep Learning-Based Object Detection and Classification for Autonomous Vehicles in Extreme Weather,” IEEE Transactions on Intelligent Transportation Systems, vol. 24, no. 4, pp. 1157-1169, 2023.

[8] T. McMillan and J. Kim, *Sensor Fusion Techniques for Reliable Object Detection in Adverse Weather*, IEEE Access, vol. 10, pp. 15678-15690, 2022.