AI Course

Team Project Final Report

For students (instructor review required)

ⓒ2019 SAMSUNG. All rights reserved.

Samsung Electronics Corporate Citizenship Office holds the copyright of this document.

This document is a literary property protected by copyright law so reprint and reproduction without permission are prohibited.

To use this document other than the curriculum of Samsung innovation Campus, you must receive written consent from copyright holder.

|  |
| --- |
| Distracted Driver Detection |

<14/8/2023>

InnovateX

<Meshal Aldajani>

<Ruba Almohya>

<Abdullah Alsaab>

<Sara Alahmari>

<Murtadah Albaik>

Content

1. Introduction

1.1. Background Information

1.2. Motivation and Objective

1.3. Members and Role Assignments

1.4. Schedule and Milestones

2. Project Execution

2.1. IoT Service Model

2.2. Data Processing

2.3. Service Implementation

2.4. System Diagram

3. Results

3.1. Data Acquisition (Sensor, Actuator, Controller)

3.2. Network & Communication

3.3. Hardware Implementation

3.4. Data Visualization

3.5. Testing and Improvements

4. Projected Impact

4.1. Accomplishments and Benefits

4.2. Future Improvements

5. Team Member Review and Comment

6. Instructor Review and Comment

1. Introduction

1.1. Background Information

Distracted driving is a major problem that contributes to millions of car accidents and injuries every year. In the United States, one in five car accidents is caused by distracted driving. This translates to 425,000 people injured and 3,000 people killed by distracted driving every year. The first DDDR projects using AI were developed in the early 2000s. These projects were relatively simple and could only detect a few types of distractions. However, recent advances in AI have led to the development of more sophisticated DDDR projects that can detect a wider range of distractions. Overall, AI has the potential to make a significant contribution to the development of distracted driver detection systems. As the technology continues to improve, AI-based systems will become more accurate, scalable, and cost-effective. This will make them more widely available and effective in preventing accidents.

1.2 Motivation and Objective

The motivation behind the distracted driver detection project using AI is to enhance road safety and reduce the number of accidents caused by drivers who are distracted while driving. Distracted driving is a significant problem worldwide, and it includes activities such as texting, talking on the phone, eating, or using in-car technologies while operating a vehicle. These distractions can significantly impair a driver's ability to react to hazards and increase the risk of accidents.

The objective of the distracted driver detection project is to develop an AI-based system that can accurately identify and classify instances of distracted driving in real-time. By analyzing various data sources, such as video feeds from in-car cameras or sensors, the AI model can detect signs of distraction, including head movements, eye gaze patterns, hand gestures, and other relevant cues.

1.3 Members and Role Assignments

|  |  |
| --- | --- |
| **Team Member** | **Role** |
| Meshal Aldajani | Data Cleaning | Data Labeling | Model Training |
| Abdullah Alsaab | Data Augmentation | Model Training |
| Martada Albaik | Data Splitting | Model Testing | Model Deployment |
| Ghassan Alward | Data Normalization | Model Testing |
| Sara Alahmari | Data Preprocessing | Model Testing |
| Ruba Almohya | Data Collection | Model Training |

1.4 Schedule and Milestones

|  |  |  |
| --- | --- | --- |
| **Date** | **Task** | **Milestones** |
| 31/7/2023 | Data Collection |  |
| 1/8/2023 – 5/8/2023 | Data Cleaning |  |
| Data Augmentation |  |
| Data Splitting |  |
| 6/8/2023 – 9/8/2023 | Data Normalization |  |
| Data Pre-processing |  |
| Data Labeling |  |
| 10/8/2023 – 16/8/2023 | Model Training and Testing | ✓ |
| 14/8/2023 – 16/8/2023 | Model Deployment | ✓ |

2. Project Execution

2.1 Data Acquisition

The data was taken from Kaggle <https://www.kaggle.com/c/state-farm-distracted-driver-detection/data>.

It can be categorized by 10 classes:

* c0: safe driving
* c1: texting - right
* c2: talking on the phone - right
* c3: texting - left
* c4: talking on the phone - left
* c5: operating the radio
* c6: drinking
* c7: reaching behind
* c8: hair and makeup
* c9: talking to passenger

2.2 Training Methodology

1. Collect a dataset of images or videos of drivers in both distracted and non-distracted states. This dataset should be as large and diverse as possible, to ensure that the AI model can learn to recognize a wide range of distracted behaviors.
2. Label each image or video in the dataset with the driver's state. This can be done manually or automatically.
3. Use a deep learning algorithm to train a model on the labeled dataset. The most common deep learning algorithms for distracted driver detection are convolutional neural networks (CNNs).
4. Evaluate the performance of the trained model on a held-out test dataset. This will help to ensure that the model is not overfitting to the training data.
5. Deploy the trained model to a real-world system for detecting distracted drivers.

2.3 Workflow

1. Data collection. The first step is to collect a dataset of images or videos of drivers engaged in different activities, including safe driving, texting, talking on the phone, drinking, etc. This dataset can be collected from public sources, such as Kaggle, or it can be collected by using a dash cam or other recording device.
2. Feature extraction. Once a dataset of images or videos has been collected, the next step is to extract features from each frame. These features can be used to represent the driver's facial expressions, head position, eye gaze, and other visual cues that can be used to identify distracted driving.
3. Model training. The extracted features are then used to train a machine learning model. This model can be a traditional machine learning algorithm, such as a support vector machine or a random forest, or it can be a deep learning algorithm, such as a convolutional neural network.
4. Model evaluation. Once the model has been trained, it is important to evaluate its performance on a held-out test set. This will help to ensure that the model is not overfitting to the training data and that it can generalize to new data.
5. Model deployment. Once the model has been evaluated and found to be effective, it can be deployed in a production environment. This could involve using the model to detect distracted drivers in real time and issuing warnings to the driver or to law enforcement.

2.4 System Diagram

The system consists of the following components:

* Camera: This captures images or video of the driver's face and surroundings.
* Image processing: This extracts features from the images or video, such as the driver's facial expressions, head pose, and eye gaze.
* Machine learning model: This uses the extracted features to classify the driver's behavior as distracted or not distracted.

The system can be implemented in a variety of ways, depending on the specific hardware and software that is available. For example, the camera could be a built-in camera on a car or a separate camera that is mounted in the vehicle. The image processing and machine learning models could be implemented on a cloud server or on a dedicated device in the vehicle.

The system can be used to detect a variety of distracted driving behaviors, such as:

* Talking on a cell phone: The driver's facial expressions and head pose can be used to identify if they are talking on a cell phone.
* Texting: The driver's eye gaze can be used to identify if they are looking at their phone.
* Eating or drinking: The driver's facial expressions and hand movements can be used to identify if they are eating or drinking.
* Adjusting the radio or navigation system: The driver's hand movements can be used to identify if they are adjusting the radio or navigation system.

3. Results

3.1. Data Preprocessing

* Resize images: This is important because different images may have different dimensions, which can make it difficult for the model to learn. A common size for images is 224x224 pixels.
* Augment images: This involves creating new images by applying transformations to the original images. This can help to improve the robustness of the model to variations in the input data.
* Normalize images: This involves normalizing the images by dividing it by 255.

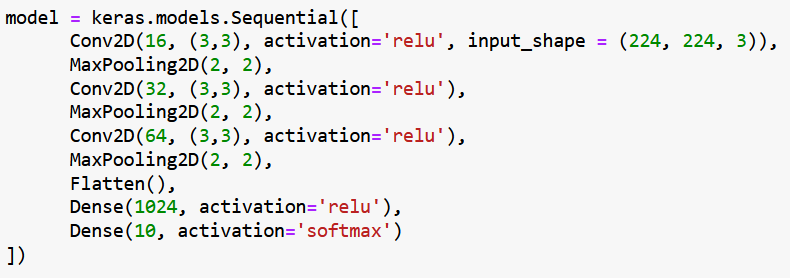
3.2 Exploratory Data Analysis (EDA)

* The distribution of classes: the number of images in each class is: c0: 2459, c1: 2237, c2: 2287, c3: 2316, c4: 2296, c5: 2282, c6: 2295, c7: 1972, c8: 1881, c9: 2099. The number of images in the test data is 79696.
* The distribution of image sizes: All images have been converted to 224x224.
* The distribution of image quality: All images are in a good quality and have been converted to the 3 main colors.
* The presence of noise and artifacts: All images are in the same condition and don’t have noise or artifacts.
* The correlation between features: Based on the accuracy of the model, it seems that all features are highly correlated to the output.

3.3 Modeling

Our training methodology for developing an AI-based system to detect distracted drivers includes the following steps:

* Collect a large and diverse dataset of images.
* Train a model on the labeled dataset using a deep learning algorithm, such as a convolutional neural network (CNN).
* Evaluate the performance of the trained model on a held-out test dataset to ensure it is not overfitting.

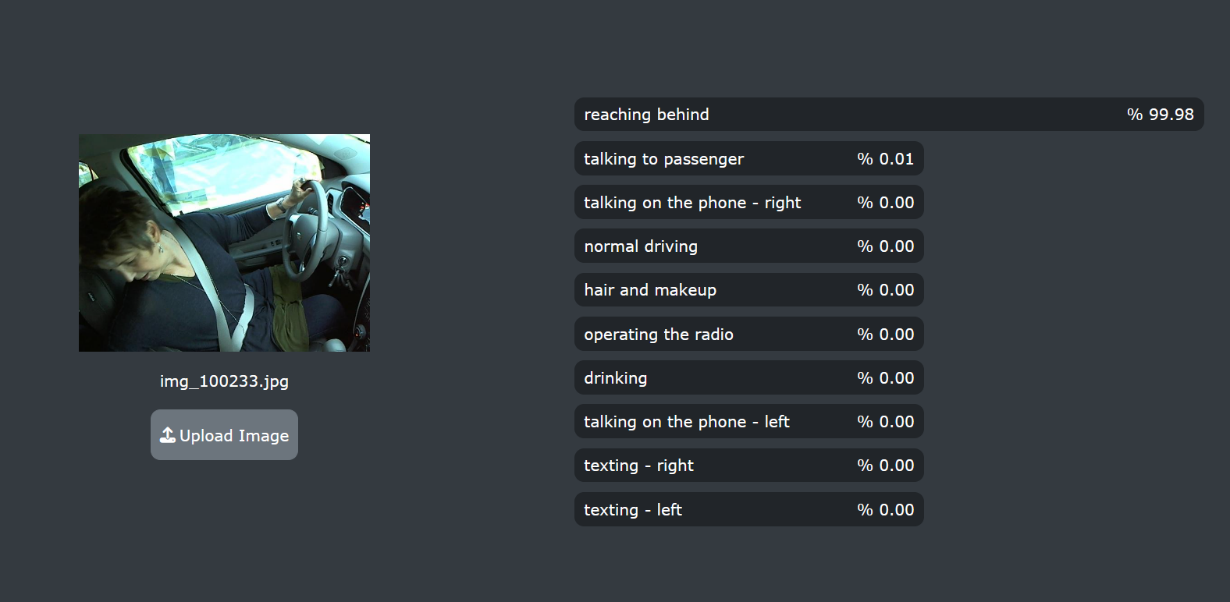


3.4 User Interface (Interface).

To implement a TensorFlow model, we utilize various tools and technologies. The process comprises the following steps:

* Train and save the model using TensorFlow in Python.
* Convert the saved model to JSON format utilizing the tensorflowjs\_converter tool.
* Construct a web application employing HTML, CSS, and JavaScript to load and utilize the converted model.
* Deploy the web application via GitHub Pages.

The web page allows the user to upload a picture from his device, then the application shows how much, in percentage, does the picture match in each class.



3.5. Testing and Improvements.

The testing phase was included in developing the web application, so we concluded from the results it gave us that the model is accurate enough to give correct predictions.

4. Projected Impact

4.1. Accomplishments and Benefits

**Accomplishments:**

* Trained a model to predict the pose of distracted drivers.
* High accuracy for each of the training, validation, and testing phase.
* Develop a web application to show the predictions given the pictures.

**Benefits:**

* Avoid distractions while driving.
* Decrease the traffic accidents rate.
* Promote the digital transformation in Saudi Arabia.

4.2 Future Improvements

To improve our distraction detection project, we can use techniques to make the model compact and efficient for real-time detection. Potential improvements include:

* Grayscale mode: Converting color images to grayscale to reduce the amount of data that needs to be processed.
* Network pruning: Removing specific weights and their respective connections in a neural network to compress its size.

5. Team Member Review and Comment

|  |
| --- |
| C:\Users\T-maldajani\Downloads\is_your_idea_innovative.jpeg |

|  |  |
| --- | --- |
| NAME | REVIEW and COMMENT |
| Meshal Aldajani | No Comment |
| Murtadah Albaik | No Comment |
| Ruba Almohya | No Comment |
| Abdullah Alsaab | No Comment |
| Sara Alahmari | No Comment |
| Ghassan Alward | No Comment |

6. Instructor Review and Comment

|  |  |  |
| --- | --- | --- |
| CATEGORY | SCORE | REVIEW and COMMENT |
| IDEA | \_\_/10 |  |
| APPLICATION | \_\_/30 |  |
| RESULT | \_\_/30 |  |
| PROJECT MANAGEMENT | \_\_/10 |  |
| PRESENTATION & REPORT | \_\_/20 |  |
| TOTAL | \_\_/100 |  |