BABEŞ BOLYAI UNIVERSITY, CLUJ NAPOCA, ROMÂNIA
FACULTY OF MATHEMATICS AND COMPUTER SCIENCE

Recognition of Road Signs Using Federated Learning

– Lab 2 report –

Team members
Ebanca Stefan
Gabor Raul
Cadar Damian
Kohan Alexandru
Name, specialisation, group, email
Scooby Doo, Software Engineering, 258, raul.gabor@stud.ubbcluj.ro

2023-2024

Chapter 1

Basic Flow Description

1.0.1 Setting up the Environment

- **Install TFF from GitHub**: Install TensorFlow Federated (TFF) from the GitHub repository. Follow the installation instructions provided in the TFF documentation.

```
pip install --upgrade tensorflow-federated
```

- **Import Libraries**: Import necessary libraries and modules, including TensorFlow and TFF.

```
import tensorflow as tf
import tensorflow_federated as tff
```

1.0.2 Data Collection and Preprocessing

- **Collect Diverse Dataset**: Gather a diverse dataset of road sign images, including images captured under different weather conditions, from various geographic locations, and with varying lighting.
- **Data Preprocessing**: Preprocess the data to ensure it's ready for model training. This includes resizing images, normalizing pixel values, and categorizing road signs based on their classes.

1.0.3 Federated Learning Setup

- **Federated Context**: Define the federated context for the road sign recognition task. This includes specifying the global model and setting up the communication structure for federated learning.
- **Server**: Configure the server to hold the global model.

1.0.4 Model Architecture

- **Design NN Model**: Design a Neural Network (NN) model for road sign recognition. Ensure the model architecture is capable of handling various road sign types and environmental conditions.

```
    def create_road_sign_model():
        model = tf.keras.Sequential([...])
        return model
```

1.0.5 Federated Training Loop

- **Global Model Initialization**: Initialize the global model with the designed NN architecture.

```
    global_model = create_road_sign_model()
    for round_num in range(num_rounds):
        # Distribute global model to client devices
        ...

        # Local Training on Client Devices
        ...

        # Secure Aggregation of Updates
        ...

        # Model Evaluation on a Validation Dataset
        ...
```

- **Iterative Federated Training**:

    1. **Local Training**: Training the model using various data sets which include images of road signs in different environments.
    2. **Secure Aggregation**: Implement secure aggregation techniques to protect model updates during aggregation.

3. **Model Evaluation**: Evaluate the global model's performance after each federated training round. Calculate metrics such as recognition accuracy.

```
evaluation_metrics = ...
evaluation_results = evaluation_metrics(global_model)
```

1.0.6 Testing and Validation

- Test the system for identifying various road signs, including different weather conditions, varying lighting, and geographic locations. Collect data from real-world scenarios to validate model performance.

1.0.7 Description of Application Functionalities

- **Data Collection**: The application collects a diverse range of data, including images of road signs captured under various conditions (day, night, rain, snow, etc.) and from different geographical locations. It also takes into account different types of road signs, such as regulatory, warning, and informational signs.
- **Real-time Recognition**: The system provides real-time recognition of road signs from live camera feeds or static images. It leverages AI to analyze the visual data and identify the type and meaning of each road sign.
- **Data Storage**: The application securely stores data about recognized road signs.

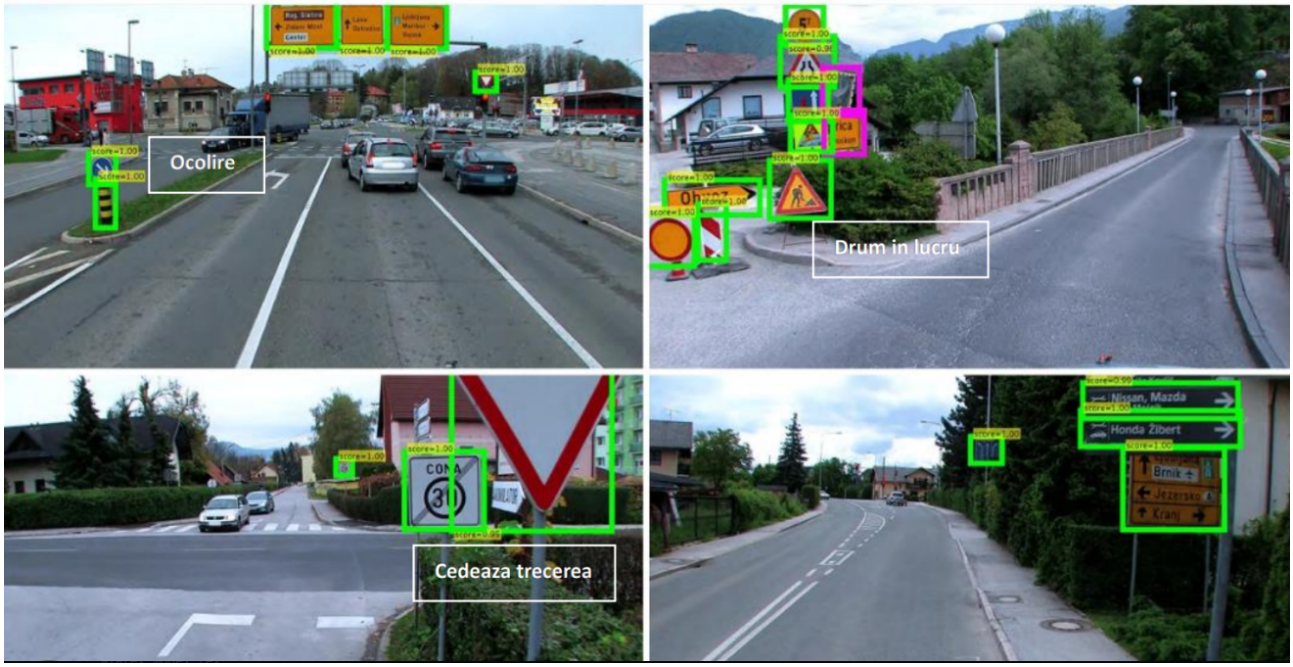

Figure 1.1: Interface

Chapter 2

Report

2.0.1 Description of Application Functionalities

- **Data Collection**: The application collects a diverse range of data, including images of road signs captured under various conditions (day, night, rain, snow, etc.) and from different geographical locations. It also takes into account different types of road signs, such as regulatory, warning, and informational signs.
- **Real-time Recognition**: The system provides real-time recognition of road signs from live camera feeds or static images. It leverages AI to analyze the visual data and identify the type and meaning of each road sign.
- **Data Storage**: The application securely stores data about recognized road signs, enabling post-trip analysis and report generation.

2.0.2 Description of the Problem Solved with AI (Plastic and Formal)

- **Layman's Explanation**: The application solves the problem of assisting drivers in recognizing and understanding road signs in real-time. It uses AI to "see" and "interpret" the road signs, under various weather conditions.
- **Formal Definition**: The problem addressed is road sign recognition, a computer vision task. The application employs deep learning, specifically neural networks (NNs), to classify road signs into predefined categories. The problem is characterized by the need for real-time recognition of road signs from visual data, with a focus on accuracy, robustness, and the ability to handle diverse environmental conditions.

2.0.3 Related Work & Useful Tools and Technologies

- **Related Work**: Several related works have explored computer vision and object recognition, including road sign recognition. Notable papers and research efforts include "Traffic Sign Recognition with Multi-Scale Convolutional Networks" by Sermanet et al. (2011) and "Deep Residual Learning for Image Recognition" by He et al. (2015). These studies laid the foundation for effective CNN-based recognition systems.

- **Useful Tools and Technologies**:

  - **TensorFlow**: TensorFlow provides the core deep learning framework for building and training the CNN model.

  - **OpenCV**: OpenCV is used for image preprocessing, manipulation, and feature extraction.

  - **Secure Data Handling**: Security measures ensure the protection of personal data, especially in systems where live camera feeds are involved.