Project Proposal

This project consists of developing an app that classifies road conditions (dry, wet, snowy, etc.) and outputs other weather-related parameters such as temperature by processing user-uploaded images of traffic webcams. The aim is to help improve navigation systems and technology related to driver safety.

1 Choice of Dataset

The selected dataset for this project is the "Road CCTV images with associated weather data," which is publicly available on the Harvard Dataverse website. This dataset is particularly relevant to the project since it evaluates a variety of weather-related metrics, such as temperature, dew, humidity, precipitation, and more. It has 11 metrics in total while of the other traffic webcam datasets only included one or two specific metrics, such as road conditions or time of day. This dataset also has more road condition classes (seven in total) compared to most other similar datasets (which are typically composed of three or four classes). Moreover, the records are collected over multiple stations in Poland, during different seasons, and at different times during the day and night. Consequently, the dataset is more diverse and thus can lead the model to make better insights and generalizations. There are more than 3,400,000 images included in the dataset, which means there are enough examples to perform a robust analysis. This dataset has also been used for an academic paper published in the peer-reviewed Journal of Intelligent Information Systems, which makes it a reliable source of information.

2 Methodology

2.1 Data Preprocessing

The chosen dataset has around 3,400,000 different images distributed across 556 files, so training a model on a dataset of this scale would be too computationally expensive for the scope of this project. Instead, a small subset of the original dataset will be taken to train the model, starting with 10,000 to 20,000 images and increasing the number of images if the computation time allows it. To preserve the diversity of the dataset, the selected subset will use images from multiple files. Indeed, the files are organized chronologically, which means that taking all the images from one file to train the model leads to a loss of diversity regarding seasons, which in turn might decrease the model's accuracy. The images from the dataset will need to be preprocessed to ensure consistency in size and format by normalizing pixel values. The data columns on station IDs, image paths, dew, temperature below five cm, temperature above 20 cm, and temperature above two meters will be removed since they are irrelevant to the prediction model. Afterwards, the dataset will be filtered to include only the images in the subset with their corresponding data rows, which then need to be parsed to check for any irrelevant or missing information and any inconsistencies in formatting. The date column will be modified to indicate the season, and columns with numerical data will be normalized so the data can later be inputted in an appropriately-scaled feature vector. The road condition classes are the most useful type of information in the dataset.

2.2 Machine Learning Model

The aim of the project is to accurately predict the following metrics/classes:

1. Humidity (on a scale from 0 to 100)
2. Precipitation (in mm)
3. Road Conditions (classes: dry, wet, snow, ice, saline)
4. Ground Temperature (in Celsius)
5. Warnings (classes: none, low warning, medium warning, slippery danger, high danger)
6. Wind Direction (in degrees)
7. Wind Speed (in meters/second)

The selected model is a convolutional neural network (CNN). One of the reasons for choosing a CNN model is that the research paper that used the same dataset to analyze road conditions also used a CNN model. Also, more generally, CNNs are designed for image processing and are considered quite strong for image classification tasks, which is particularly convenient as the model will need to analyze webcam traffic images and extract information (such as wet surfaces or road visibility) based on visual cues. Similarly, CNNs are proficient at recognizing important patterns (which can include water puddles or snow accumulation) due to applying layers of filters across the image. Moreover, CNN models are strong at modelling complex patterns and correlations, which is helpful considering that weather parameters are interconnected, and perform well on large datasets. Other models that were considered included the random forest model, which is less computationally expensive but might not be as efficient for pattern detection since it requires manual feature extraction. Vision transformers were also considered since they perform exceptionally well on image classification tasks, but it is even more computationally expensive than CNN models, which makes its implementation inefficient.

2.3 Evaluation Metrics

The following evaluation metrics will be used:

1. Accuracy ($\frac{\text{Number of correct predictions}}{\text{Total number of predictions}} * 100$)
2. Logistic Loss
3. Confusion Matrix (to get a score for accuracy, precision, and sensitivity)
4. F1-Score (based on precision and sensitivity metrics from the confusion matrix)
5. Top-K Accuracy
6. Brier Score

The baseline performance will be based on the classification of road conditions since it is the most important metric. The accuracy of road conditions should be higher than 70% and the corresponding F1 score should be higher than 0.6. Further baseline results might be added or increased after more experimentation with the CNN model.

3 Application

The front-end of the web application will be designed using React while the back-end will be made in Flask. The basic functionality of the application is that the user clicks on an input button, after which they will be prompted to upload an snapshot of webcam traffic footage. After having uploaded the picture, the model will analyze it and the user will be directed to a different web page that displays the output metrics in a neat and organized format. Given enough testing with evaluation metrics, it is also possible to implement a certainty qualifier. For instance, the program could return "snowy (85% certainty)" under the road conditions section.