**Data Engineering Report for Accident Detection Project**

Below is a detailed breakdown of the processes and methodologies implemented to prepare the dataset for training:

**1. Data Collection**

* **Sources:**
* **Source 1**: Kaggle Dataset - [Accident Detection Dataset](https://www.kaggle.com/datasets/puyushgupta/dataset)
* **Source 2**: Roboflow Dataset - **Accident Detection Model**
* Ensured consistency in path generation, formats and structure across both datasets.
* **Volume – Kaggle Dataset:**
  + Training set images: 7,125 (overall)
  + Validation set images: 254 (overall)
* **Volume – Roboflow Dataset:**
  + Training set images: 2,517 (overall)
  + Validation set images: 371 (overall)

**2. Exploratory Data Analysis (EDA)**

* Initial data exploration conducted to identify:
  + Number of images and how they are distributed.
  + Class distribution (imbalance identified between accident and non-accident images).
  + Format issues, missing files, missing images, noisy images and inconsistencies.
  + Distribution of bounding box sizes and positions.
* Verified the completeness of annotations and identified files missing corresponding labels in both train and validation sets.

**3. Data Cleansing**

* Removed duplicate images and labels to eliminate redundancy.
* Removed images and labels with mixed classes (e.g., [0, 4, 5]) or purely unrelated classes (e.g., [0, 4, 9]), which correspond to categories like car, human, and bike.
* Addressed images with missing or empty labels.
* Normalized the class structure:
  + Merged all accident-related annotations into a single "Accident" class to simplify the project, as the main goal is to detect accidents on the road, regardless of the specific accident details.
* Assigned empty labels to non-accident images for consistency, treating them as null labels. This helps balance and simplify the dataset, as the YOLO architecture will treat these images as background (no object to detect), ultimately enhancing the model's generalization.
* Performed data augmentation on non-accident images to balance the dataset and improve the model's ability to generalize. The augmented (non-accident images) were added along the accident images that have been cleaned up earlier to a single path.
* Updated class zero (Accident) in the YOLO file to reflect all relevant labels for the accident class.
* Finally, resized all images in the training set to 640x640, ensuring the bounding boxes were scaled appropriately to match the new image size, so the model can draw the boxes correctly.
* Validation dataset increased from the augmented images in order to test while training.

**4. Data Validation**

* Ensured that each image had a corresponding .txt annotation file.
* Checked for any mismatched labels and removed unnecessary or irrelevant files from both the labels and images.
* Validated a sample of images to confirm their purity and the accuracy of the associated class labels.

**5. Data Augmentation**

* Augmented the non-accident dataset to address the class imbalance:
  + Applied transformations such as flipping, rotation, brightness adjustment, and cropping.
  + Generated an additional 3,000 non-accident images for improved model generalization and taking a bulk of them to the main data.
* Ensured that each augmented image had a corresponding empty label file.

**6. Final Dataset Structure**

* Prepared a balanced dataset for training:
  + Total accident images: **2,699**
  + Total non-accident images (including augmented): **1,900**
  + Total Trian is **4,700** images.
  + Labels follow the YOLO format with one class (Accident).
  + Empty txt file for non-accident images to insure consistency.
  + Final validation dataset is **506** (mixed between accident and non-accident)

**Data Engineering Team: [Joseph Salem, Oun Awamleh]**

**\*\*\* Python Code used in all phases will be shared on GitHub.**

**Thanks...**