Task 2: synopsis

1. Drowsiness detection:

Introduction:

A countless number of people drive on the highway day and night. Taxi drivers, bus drivers, truck drivers and people traveling long-distance suffer from lack of sleep. Due to which it becomes very dangerous to drive when feeling sleepy. The majority of accidents happen due to the drowsiness of the driver.

We will be using OpenCV for gathering the images from webcam and feed them into a Deep Learning model which will classify whether the person's eyes are 'Open' or 'Closed'.

Approach:

There are numerous techniques to detect drowsiness. In this, we have put forward a deep learning-based approach to detect the drowsiness of the drivers. We have used convolutional neural networks, which is a class of deep learning.

DEEP LEARNING: Deep learning eliminates some of data pre-processing that is typically involved with machine learning. These algorithms can ingest and process unstructured data, like text and images, and it automates feature extraction, removing some of the dependency on human experts. For example, let's say that we had a set of photos of different pets, and we wanted to categorize by "cat", "dog", "hamster", etc. Deep learning algorithms can determine which features (eg. ears) are most important to distinguish each animal from another. In machine learning, this hierarchy of features is established manually by a human expert.

CNN (CONVOLUTIONAL NEUTRAL NETWORKS): used primarily in computer vision and image classification applications, can detect features and patterns within an image, enabling tasks, like object detection or recognition.

The approach we will be using for this is:

1. Take image as input from a camera: We use the method provided by OpenCV, cv2.VideoCapture(0) to access the camera and set the capture object (cap). cap.read() will read each frame and we store the image in a frame variable.

2. Detect the face in the image and create a Region of Interest (ROI): To detect the face in the image, we need to first convert the image into grayscale as the OpenCV algorithm for object detection takes gray images in the input. We don’t need colour information to detect the objects. We will be using classifier to detect faces.

3. Detect the eyes from ROI and feed it to the classifier: The same procedure to detect faces is used to detect eyes. detect the eyes using left_eye = leye.detectMultiScale(gray)**.** l_eye only contains the image data of the eye. This will be fed into our CNN classifier which will predict if eyes are open or closed. Similarly, we will be extracting the right eye into r_eye**.**

4. Classifier will categorize whether eyes are open or closed.

5. Calculate score to check whether the person is drowsy: The score is basically a value we will use to determine how long the person has closed his eyes. So if both eyes are closed, we will keep on increasing score and when eyes are open, we decrease the score.

2. Natural Language Processing:

Reviews for a restaurant on a food delivery app

Introduction:

Analysing customer feedback, such as opinions in survey responses and social media conversations, allows brands to learn what makes customers happy or frustrated, so that they can tailor products and services to meet their customers' needs.

Since customers express their thoughts and feelings more openly than ever before, sentiment analysis is becoming an essential tool to monitor and understand that sentiment.

Sentiment Analysis:

Sentiment analysis is the process of detecting positive or negative sentiment in text. It's often used by businesses to detect sentiment in social data, gauge brand reputation, and understand customers.

Sentiment analysis models focus on polarity (positive, negative, neutral) but also on feelings and emotions (angry, happy, sad, etc), and even intentions (interested v. not interested)

Depending on how you want to interpret reviews and queries, you can define and tailor your categories to meet your sentiment analysis needs.

Sentiment analysis algorithms:

1. Rule Based Approaches: Usually, a rule-based system uses a set of human-crafted rules to help identify subjectivity, polarity, or the subject of an opinion. These rules may include various NLP techniques developed in computational linguistics. Defines two lists of polarized words (eg. negative words such as bad, worst, ugly, etc and positive words such as good, best, beautiful, etc). Counts the number of positive and negative words that appear in a given text.

If the number of positive word appearances is greater than the number of negative word appearances, the system returns a positive sentiment, and vice versa. If the numbers are even, the system will return a neutral.

2. Automatic Approaches: Automatic methods, contrary to rule-based systems, don't rely on manually crafted rules, but on machine learning techniques. A sentiment analysis task is usually modeled as a classification problem, whereby a classifier is fed a text and returns a category, eg. positive, negative, or neutral.

3. Hybrid: Hybrid systems combine the desirable elements of rule-based and automatic techniques into one system. One huge benefit of these systems is that results are often more accurate.

   Ratings:

   For each textual review, we want to predict if it corresponds to a positive review (the customer is happy) or to a negative one (the customer is not satisfied). The reviews overall ratings can range from 1/5 to 5/5. In order to simplify the problem we will split those into two categories:

   Negative (n) < 2.5   Positive(p) >= 2.5