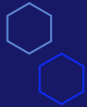# OBJECT TRACKING + COLOR DETECTION

Vera Gnezdilova, Vadim Khanin, Sofya Rozhok

The purpose of the project was to track the movement of vehicles in a traffic surveillance and detect colors of cars

PROCESSING VIDEO FRAMES

The main loop iterates over each frame of the video.

- Each frame is converted to grayscale for car detection

- Car detection is performed using the detectMultiScale method of the Haar Cascade classifier

- Rectangles are drawn around detected cars

- Dominant color within each detected car region is determined using K-means clustering

- Text displaying the closest color name to the dominant color is added to the frame

CLOSEST_COLOUR FUNCTION

```python
def closest_colour(requested_colour):
    min_colours = {}
    for key, name in webcolors.CSS3_HEX_TO_NAMES.items():
        r_c, g_c, b_c = webcolors.hex_to_rgb(key)
        rd = (r_c - requested_colour[0]) ** 2
        gd = (g_c - requested_colour[1]) ** 2
        bd = (b_c - requested_colour[2]) ** 2
        min_colours[(rd + gd + bd)] = name
    return min_colours[min(min_colours.keys())]
```

- Iterates through CSS3 color names

- Calculates Euclidean distance between requested color and each CSS3 color

- Returns the name of the closest matching color

GET_COLOUR_NAME FUNCTION

- Attempts direct color matching using webcolors.rgb_to_name

- If direct matching fails, finds the closest matching color using closest_colour

- Returns both the actual and closest color names

```python
def get_colour_name(requested_colour):
    try:
        closest_name = actual_name = webcolors.rgb_to_name(requested_colour)
    except ValueError:
        closest_name = closest_colour(requested_colour)
        actual_name = None
    return actual_name, closest_name
```

HOW TO INCREASE ACCURACY?

- train on more data K-Nearest Neighbors Classifier

- consider color features such as Color Moments or Color Correlogram

- try other classifiers

- find and determine white color on the video