

**DPI**

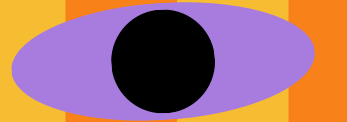
# **DISABILITY PARKING IDENTIFIER**



**-BY:DANAH ALHAMLAN,NJOD ALANSARI-**

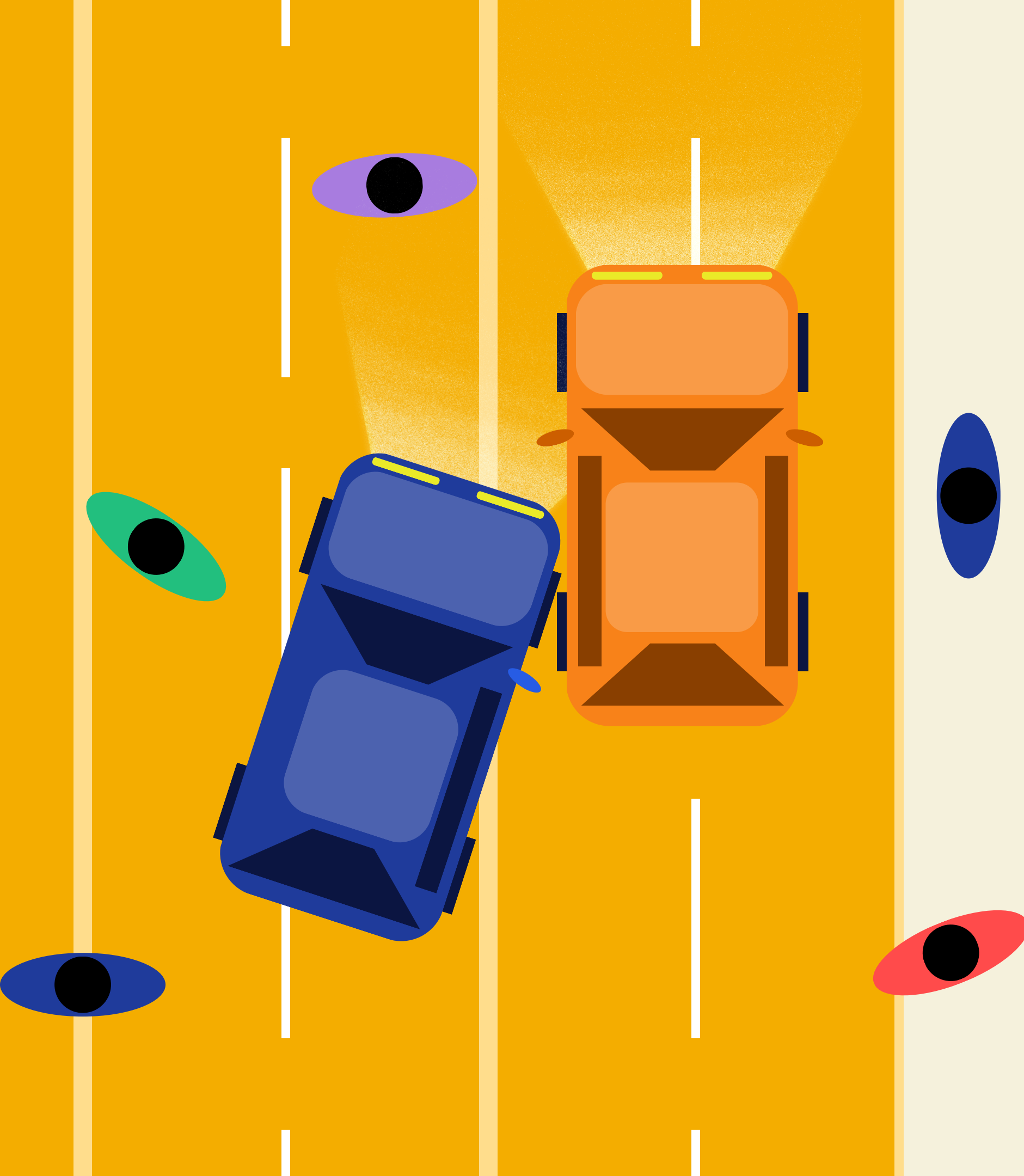
# فقرات المشروع:

- 1 تحديد المشكلة
- 2 أهداف المشروع
- 3 بيانات المشروع
- 4 تقديم النتائج
- 5 تطبيق واقعي



## تحديد المشكلة

مشكلة معرّف المواقف لذوي الإحتياجات الخاصة هي أن بعض الأشخاص يستخدمونها بشكل خاطئ أو يعتدون على حقوق الأشخاص ذوي الإعاقة. هذه المشكلة تسبب إزعاجاً وضراً للمعاقين الذين يحتاجون إلى مواقف قريبة من أماكن الخدمات والمرافق. وفقاً للإدارة العامة للمرور بالمملكة، فإن المخالفين يواجهون غرامات مالية تتراوح بين 500 و 900 ريال، ويمكن حجز المركبة في حالة عدم تجاوب السائق. ولحل هذه المشكلة، يجب على الجهات المسؤولة تطبيق العقوبات بصرامة وتوعية المجتمع بحقوق وحاجات ذوي الإعاقة، وتشجيعهم على المشاركة في الحياة الطبيعية.



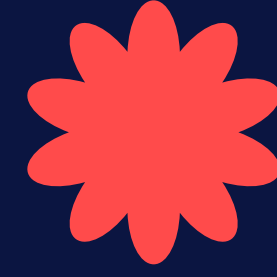


2

## أهداف المشروع

- يهدف هذا المشروع إلى تطوير جهاز يساعد ذوي الإحتياجات الخاصة على التعرف على المواقف
- يساعد المشروع على ضمان حقوق ذوي الاحتياجات الخاصة بحيث لا يصطف شخص غير محتاج في مواقفهم
- يفرق المشروع بين المواقف العادية والمواقف الخاصة بذوي الاحتياجات الخاصة





3

## بيانات المشروع

تم استخدام GOOGLE  
وبعض محركات البحث الأخرى  
ومواقع الإنترنت في جمع  
الصور



4

## تقديم النتائج



```

from keras.models import load_model # TensorFlow is required for Keras to work
from PIL import Image, ImageOps # Install pillow instead of PIL
import numpy as np

# Disable scientific notation for clarity
np.set_printoptions(suppress=True)

# Load the model
model = load_model("821.h5", compile=False)

# Load the labels
class_names = open("2448.txt", "r").readlines()

# Create the array of the right shape to feed into the keras model
# The 'length' or number of images you can put into the array is
# determined by the first position in the shape tuple, in this case 1
data = np.ndarray(shape=(1, 224, 224, 3), dtype=np.float32)

# Replace this with the path to your image
image = Image.open('89.jfif').convert("RGB")
# resizing the image to be at least 224x224 and then cropping from the center
size = (224, 224)
image = ImageOps.fit(image, size, Image.Resampling.LANCZOS)

# turn the image into a numpy array
image_array = np.asarray(image)

# Normalize the image
normalized_image_array = (image_array.astype(np.float32) / 127.5) - 1

# Load the image into the array
data[0] = normalized_image_array

# Predicts the model
prediction = model.predict(data)
index = np.argmax(prediction)
class_name = class_names[index]
confidence_score = prediction[0][index]

# Print prediction and confidence score
print("Class:", class_name[2:], end="")
print("Confidence Score:", confidence_score)

```

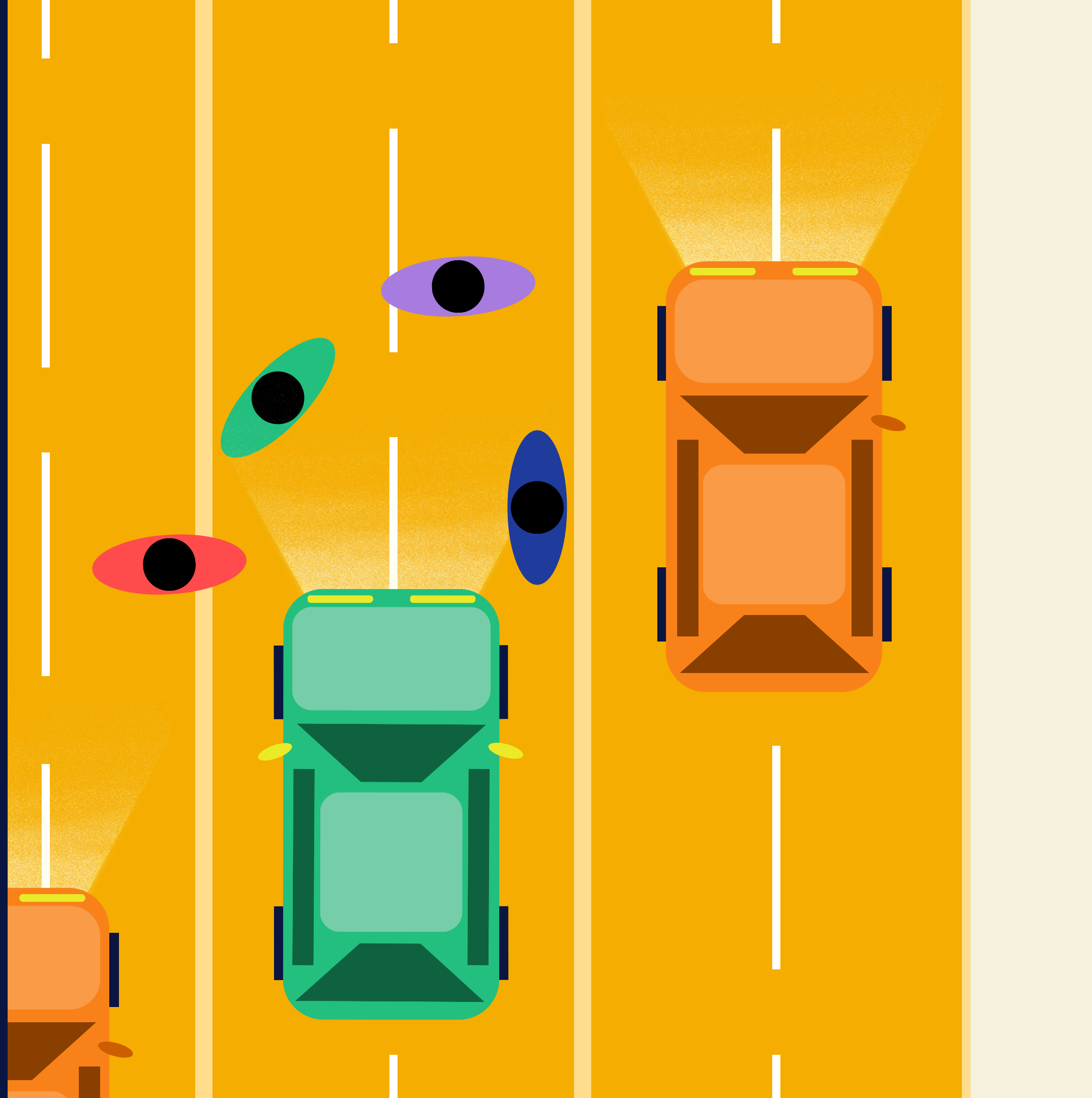
```

1/1 [=====] - 0s 461ms/step
Class: Normal attitude
Confidence Score: 0.99999166

```

5

## تطبيق واقعي





# ختامًا:

شكرا لحسن الاستماع