**By muqadas and neha**

**Main.dart**

import 'package:flutter/material.dart';

import 'package:camera/camera.dart';

import 'package:firebase\_core/firebase\_core.dart';

import 'package:cloud\_firestore/cloud\_firestore.dart';

import 'package:firebase\_storage/firebase\_storage.dart';

import 'package:tflite/tflite.dart';

import 'package:http/http.dart' as http;

import 'dart:io';

import 'Captureimage.dart'; // Your camera detection page

List<CameraDescription> cameras = [];

void main() async {

WidgetsFlutterBinding.ensureInitialized();

await Firebase.initializeApp();

cameras = await availableCameras();

runApp(const MyApp());

}

class MyApp extends StatelessWidget {

const MyApp({super.key});

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'Auto Challan Detection',

theme: ThemeData(primarySwatch: Colors.blue),

home: const SplashScreen(),

);

}

}

class SplashScreen extends StatefulWidget {

const SplashScreen({super.key});

@override

State<SplashScreen> createState() => \_SplashScreenState();

}

class \_SplashScreenState extends State<SplashScreen> {

@override

void initState() {

super.initState();

initializeApp();

}

Future<void> initializeApp() async {

await Tflite.close();

await Tflite.loadModel(

model: "assets/model.tflite",

labels: "assets/labels.txt",

);

FirebaseFirestore.instance.collection('uploads').snapshots().listen((snapshot) {

for (var doc in snapshot.docs) {

if (doc.data().containsKey('imageUrl')) {

String imageUrl = doc['imageUrl'];

processImageFromUrl(imageUrl);

}

}

});

await Future.delayed(const Duration(seconds: 3));

Navigator.pushReplacement(

context,

MaterialPageRoute(

builder: (context) => CaptureImagePage(

model: 'assets/model.tflite',

cameras: cameras, // ✅ Pass camera list here

),

),

);

}

Future<void> processImageFromUrl(String imageUrl) async {

final response = await http.get(Uri.parse(imageUrl));

final tempDir = Directory.systemTemp;

final tempFile = File('${tempDir.path}/temp.jpg');

await tempFile.writeAsBytes(response.bodyBytes);

var recognitions = await Tflite.runModelOnImage(

path: tempFile.path,

numResults: 2,

threshold: 0.5,

);

if (recognitions != null && recognitions.isNotEmpty) {

for (var rec in recognitions) {

if (rec['label'] == 'wrong\_parking') {

print('🚨 Wrong parking detected! Starting countdown...');

// ✅ Challan ke attributes

String challanNumber = DateTime.now().millisecondsSinceEpoch.toString();

String challanDate = DateTime.now().toString();

String challanReason = 'Wrong Parking';

String ownerName = 'Ali Khan';

String ownerContact = '03001234567';

String ownerAddress = 'House 123, Lahore';

String vehicleNumber = 'LEA-1234';

String vehicleColour = 'Black';

String vehicleType = 'Car';

String parkingLocation = 'Mall Road, Lahore';

String parkingTime = TimeOfDay.now().format(context); // Using context from State

String parkingDuration = '20 minutes';

String challanAmount = '500 PKR';

// ✅ Challan ko Firebase Firestore me save karein

await FirebaseFirestore.instance.collection('challans').add({

'challan\_number': challanNumber,

'challan\_date': challanDate,

'challan\_reason': challanReason,

'owner\_name': ownerName,

'owner\_contact': ownerContact,

'owner\_address': ownerAddress,

'vehicle\_number': vehicleNumber,

'vehicle\_colour': vehicleColour,

'vehicle\_type': vehicleType,

'parking\_location': parkingLocation,

'parking\_time': parkingTime,

'parking\_duration': parkingDuration,

'challan\_amount': challanAmount,

'image\_url': imageUrl,

'timestamp': FieldValue.serverTimestamp(),

});

print('✅ Challan saved to Firebase!');

}

}

}

}

@override

Widget build(BuildContext context) {

return const Scaffold(

body: Center(child: CircularProgressIndicator()),

);

}

}

**Capture image.dart**

import 'dart:async';

import 'package:flutter/material.dart';

import 'package:camera/camera.dart';

import 'package:tflite/tflite.dart';

class CaptureImagePage extends StatefulWidget {

final String model;

final List<CameraDescription> cameras; // ✅ Accept cameras

const CaptureImagePage({

Key? key,

required this.model,

required this.cameras,

}) : super(key: key);

@override

State<CaptureImagePage> createState() => \_CaptureImagePageState();

}

class \_CaptureImagePageState extends State<CaptureImagePage> {

CameraController? \_controller;

bool \_isDetecting = false;

@override

void initState() {

super.initState();

loadModel();

initializeCamera();

}

Future<void> loadModel() async {

await Tflite.close();

await Tflite.loadModel(

model: widget.model,

labels: "assets/labels.txt",

);

}

Future<void> initializeCamera() async {

\_controller = CameraController(

widget.cameras[0], // ✅ Use widget.cameras here

ResolutionPreset.medium,

);

await \_controller!.initialize();

if (!mounted) return;

setState(() {});

}

@override

void dispose() {

\_controller?.dispose();

Tflite.close();

super.dispose();

}

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(title: const Text('Live Detection')),

body: \_controller == null || !\_controller!.value.isInitialized

? const Center(child: CircularProgressIndicator())

: CameraPreview(\_controller!),

);

}

}