Documentación de Aplicación Móvil en Android Studio (Kotlin).

Descripción General Descripción General.

La aplicación es una herramienta de detección de objetos mediante el uso de modelos de inteligencia artificial. Su propósito es identificar y clasificar objetos en tiempo real, proporcionando información relevante al usuario. Está dirigida a desarrolladores, investigadores, y personas interesadas en aplicaciones prácticas de visión por computadora.

Características Principales:

Detección de objetos en tiempo real utilizando modelos de inteligencia artificial.

Identificación y clasificación automática de objetos.

Interfaz de usuario intuitiva para mostrar los resultados de detección.

Soporte para múltiples modelos de detección, ajustables según las necesidades del usuario.

Funcionalidad offline para procesar imágenes locales sin conexión a internet.

Integración con APIs para actualizar modelos o compartir resultados.

Manejo de errores y notificaciones para mejorar la experiencia del usuario.

2. Estructura del Proyecto

Estructura de Directorios:

/app: Contiene el código fuente de la aplicación, incluidos los recursos y archivos de configuración.

/src: Contiene el código fuente dividido en main y test.

/main: Incluye todo el código principal de la aplicación.

/java: Contiene los archivos .kt con el código Kotlin de la aplicación.

/res: Contiene los recursos de la aplicación, como layouts, cadenas de texto, imágenes, etc.

/test: Incluye las pruebas unitarias y de instrumentación para la aplicación.

/res: Contiene recursos estáticos de la aplicación.

/anim: Archivo XML utilizado para definir una animación que hace que un elemento cambie su opacidad, o transparencia, de forma progresiva

/layout: Archivos XML que definen la estructura de las pantallas de la interfaz de usuario.

/drawable: Imágenes y otros elementos gráficos utilizados en la aplicación.

/values: Archivos XML que contienen valores como cadenas, colores y estilos.

build.gradle: Archivo de configuración que especifica las dependencias y configuraciones del proyecto.

Componentes Principales:

Cada paquete está organizado para cumplir una función específica dentro de la aplicación, facilitando la separación de responsabilidades y el mantenimiento del código. Para visualizar esta parte, desde el IDE de Android Studio selecciona el modo Android.

org.openbot: Contiene los componentes principales de la aplicación, como los modelos de datos, controladores de lógica, y la interfaz de usuario. Es el paquete base donde se encuentran las clases principales que gestionan la funcionalidad general de la aplicación.

org.openbot.env (test): Incluye las clases y componentes necesarios para las pruebas y el entorno de prueba de la aplicación. Este paquete se enfoca en realizar pruebas automatizadas y de integración para garantizar la calidad del código.

Paquetes Principales:

org.openbot.autopilot: Contiene clases y componentes relacionados con la funcionalidad de conducción autónoma de la aplicación.

org.openbot.common: Incluye utilidades y clases comunes que se comparten en diferentes módulos de la aplicación para evitar la duplicación de código.

org.openbot.controller: Maneja la lógica de control de los robots o vehículos, proporcionando funcionalidades de control remoto y lógica de movimiento.

org.openbot.customview: Contiene vistas personalizadas utilizadas en la interfaz de usuario, extendiendo o mejorando las vistas estándar de Android.

org.openbot.env: Define el entorno de ejecución, incluidos aspectos como la configuración de sensores y otros elementos del sistema.

org.openbot.googleServices: Contiene integraciones con servicios de Google, como mapas o autenticación.

org.openbot.logging: Maneja el registro y monitoreo de eventos importantes en la aplicación, para facilitar el diagnóstico y la resolución de problemas.

org.openbot.main: Punto de entrada principal de la aplicación, incluyendo la actividad inicial y las configuraciones generales.

org.openbot.model: Contiene los modelos de datos utilizados en la aplicación, que representan la información procesada y mostrada al usuario.

org.openbot.modelManagement: Se encarga de la gestión de los modelos de inteligencia artificial, incluyendo la carga, actualización y optimización.

org.openbot.objectNav: Contiene clases para la navegación de objetos, utilizando los datos recogidos por los modelos de IA para guiar el vehículo.

org.openbot.original: Almacena código original o versiones anteriores del proyecto para referencia.

org.openbot.playground: Incluye funcionalidades experimentales o en desarrollo, que aún no forman parte del núcleo de la aplicación.

org.openbot.pointGoalNavigation: Clases relacionadas con la navegación basada en puntos de destino, para dirigir el vehículo a ubicaciones específicas.

org.openbot.profile: Maneja la gestión de perfiles de usuario, incluyendo preferencias y configuraciones personalizadas.

org.openbot.projects: Contiene proyectos relacionados o ejemplos que ayudan a extender la funcionalidad principal.

org.openbot.robot: Maneja la comunicación y control directo del robot, incluyendo los comandos enviados a través del hardware.

org.openbot.server: Contiene clases para configurar y gestionar un servidor local para comunicaciones en red.

org.openbot.tflite: Integra modelos TensorFlow Lite para el procesamiento de datos de IA en dispositivos móviles.

org.openbot.tracking: Clases relacionadas con el seguimiento de objetos o personas a través de la cámara del dispositivo.

org.openbot.utils: Utilidades generales que soportan diferentes aspectos del desarrollo, como el manejo de archivos, cálculos matemáticos, etc.

org.openbot.vehicle: Contiene clases que representan el vehículo, sus atributos y métodos para interactuar con sus componentes

3. Dependencias y Configuración

Dependencias Utilizadas:

com.android.tools.build:gradle: Utilizada para la construcción del proyecto Android.

org.tensorflow:tensorflow-lite: TensorFlow Lite para la integración y uso de modelos de inteligencia artificial en dispositivos móviles.

org.opencv:opencv-android: OpenCV para funcionalidades de visión por computadora y procesamiento de imágenes.

com.google.firebase:firebase-analytics: Firebase Analytics para monitorear eventos y datos de uso de la aplicación.

Archivos de Configuración:

build.gradle: Contiene configuraciones globales para el proyecto, incluyendo las dependencias y los plugins utilizados para compilar la aplicación.

4. Componentes de la Aplicación

Actividades.

MainActivity (paquete main): Actividad principal de la aplicación que gestiona la navegación entre los fragmentos y carga el contenido relevante.

CameraActivity (paquete original): Actividad que maneja la conexión con la cámara y proporciona la vista de captura.

DefaultActivity (paquete original): Actividad utilizada como plantilla para las

Fragmentos.

AutopilotFragment (paquete autopilot): Implementa la funcionalidad de conducción autónoma, utilizando los datos capturados por la cámara y otros sensores.

CameraFragment (paquete common): Fragmento que muestra la vista de la cámara y gestiona la conexión a dispositivos de captura de video.

ControlsFragment (paquete common): Fragmento encargado de proporcionar al usuario controles para operar el vehículo y manipular la entrada.

FeatureList (paquete common): Fragmento que lista las características o funcionalidades disponibles en la aplicación.

ControllerMappingFragment (paquete controller): Proporciona la interfaz que permite a los usuarios mapear los controles entre el dispositivo móvil y el robot.

PlayGroundViewFragment (paquete playground): Utilizado para probar funcionalidades experimentales que aún no forman parte del núcleo de la aplicación.

BluetoothFragment, UsbFragment (paquete main): Fragmentos para gestionar las conexiones Bluetooth y USB.

ProfileFragment, EditProfileFragment (paquete profile): Fragmentos responsables de la visualización y edición del perfil de usuario.

RobotInfoFragment, FreeRoamFragment (paquete robot): Fragmentos dedicados a mostrar información del robot y permitir la conducción en modo libre.

PointGoalNavigationFragment (paquete pointGoalNavigation): Fragmento para la navegación basada en objetivos puntuales, ayudando al vehículo a dirigirse hacia puntos específicos.

ViewModels.

MainViewModel (paquete main): Gestiona la lógica principal de la aplicación, asegurando la comunicación entre la UI y la lógica de negocio, y permitiendo la persistencia de datos relevantes.

ObjectDetectionViewModel (propuesto, para detección de objetos): Responsable de la gestión de los datos de detección de objetos, utilizando TensorFlow Lite para procesar imágenes y devolver los resultados.

NavigationViewModel (paquete pointGoalNavigation): Controla la lógica relacionada con la navegación del vehículo hacia un objetivo específico, coordinando la posición y orientación.

Servicios:

NetworkServiceConnection (paquete env): Gestiona las conexiones de red, estableciendo comunicación entre la aplicación y servicios externos para obtener información actualizada.

SensorService (paquete logging): Servicio encargado de capturar los datos de sensores (acelerómetro, giroscopio, etc.) y enviarlos al módulo de registro para análisis y monitoreo.

NsdService (paquete server): Servicio que permite descubrir y conectarse a otros dispositivos en la red local para coordinar la comunicación.

RtspServer (paquete env): Implementa un servidor RTSP que permite transmitir video en tiempo real a otros dispositivos en la misma red.

NearbyConnection (paquete env): Maneja la comunicación cercana entre dispositivos, permitiendo el descubrimiento y la conexión rápida a través de Bluetooth o WiFi.

Repositorios.

UserRepository (propuesto): Centraliza el acceso a la información del perfil de usuario, interactuando con Firebase y la base de datos local.

ModelRepository (paquete modelManagement): Se encarga de gestionar los modelos de IA, incluyendo la descarga, actualización y almacenamiento local de estos modelos.

VehicleRepository (paquete vehicle): Gestiona la información y el estado del vehículo, proporcionando métodos para interactuar con sus atributos (conexiones, estado de sensores, etc.).