

AI使用指南

文件标识: RK-KF-YF-E24

发布版本: V1.0.0

日期: 2025-02-13

文件密级: ☐绝密 ☐秘密 ☐内部资料 ☒公开

免责声明

本文档按“现状”提供, 瑞芯微电子股份有限公司(“本公司”, 下同)不对本文档的任何陈述、信息和内容的准确性、可靠性、完整性、适销性、特定目的性和非侵权性提供任何明示或暗示的声明或保证。本文档仅作为使用指导的参考。

由于产品版本升级或其他原因, 本文档将可能在未经任何通知的情况下, 不定期进行更新或修改。

商标声明

“Rockchip”、“瑞芯微”、“瑞芯”均为本公司的注册商标, 归本公司所有。

本文档可能提及的其他所有注册商标或商标, 由其各自拥有者所有。

版权所有 © 2025 瑞芯微电子股份有限公司

超越合理使用范畴, 非经本公司书面许可, 任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部, 并不得以任何形式传播。

瑞芯微电子股份有限公司

Rockchip Electronics Co., Ltd.

地址: 福建省福州市铜盘路软件园A区18号

网址: www.rock-chips.com

客户服务电话: +86-4007-700-590

客户服务传真: +86-591-83951833

客户服务邮箱: fae@rock-chips.com

前言

概述

AI 功能的相关配置使用。

产品版本

芯片名称	内核版本	Android版本
RK3576	Kernel5.10及以上	Android14、15

读者对象

本文档（本指南）主要适用于以下工程师：

技术支持工程师

软件开发工程师

修订记录

版本号	作者	修改日期	修改说明
V1.0.0	韩金锋、王杭	2025-02-13	初始版本

目录

AI使用指南

1. 概述
2. 配置
3. AI应用说明
 - 3.1 AI悬浮球
 - 3.1.1 apk路径
 - 3.1.2 开启使用
 - 3.2 RkAuth授权
 - 3.2.1 源码路径
 - 3.2.2 授权配置
 - 3.3 超分
 - 3.3.1 源码位置
 - 3.3.2 授权配置
 - 3.3.3 开启使用
 - 3.4 MEMC
 - 3.4.1 源码位置
 - 3.4.2 授权配置
 - 3.4.3 开启使用
 - 3.5 PQ
 - 3.5.1 源码位置
 - 3.5.2 授权配置
 - 3.5.3 开启使用
 - 3.6 在线字幕
 - 3.6.1 源码位置
 - 3.6.2 授权配置
 - 3.6.3 开启使用
 - 3.7 AI搜图
 - 3.7.1 源码位置
 - 3.7.2 授权配置
 - 3.7.3 开启使用
 - 3.8 AI橡皮擦
 - 3.8.1 源码位置
 - 3.8.2 授权配置
 - 3.8.3 开启使用

1. 概述

本文档介绍如何在RK3576的14和15 Android平台上，启用AI功能。

2. 配置

对应的mk文件中配置BOARD_RKAI_SUPPORT为true，例如：

```
diff --git a/BoardConfig.mk b/BoardConfig.mk
index db254dc..165bb6b 100644
--- a/BoardConfig.mk
+++ b/BoardConfig.mk
@@ -119,6 +119,8 @@ BUILD_BROKEN_USES_BUILD_HOST_STATIC_LIBRARY := true

BOARD_SHOW_HDMI_SETTING ?= true

+BOARD_RKAI_SUPPORT := true
+
# for dynamaic afbc target
BOARD_HS_DYNAMIC_AFBC_TARGET := false
```

3. AI应用说明

3.1 AI悬浮球

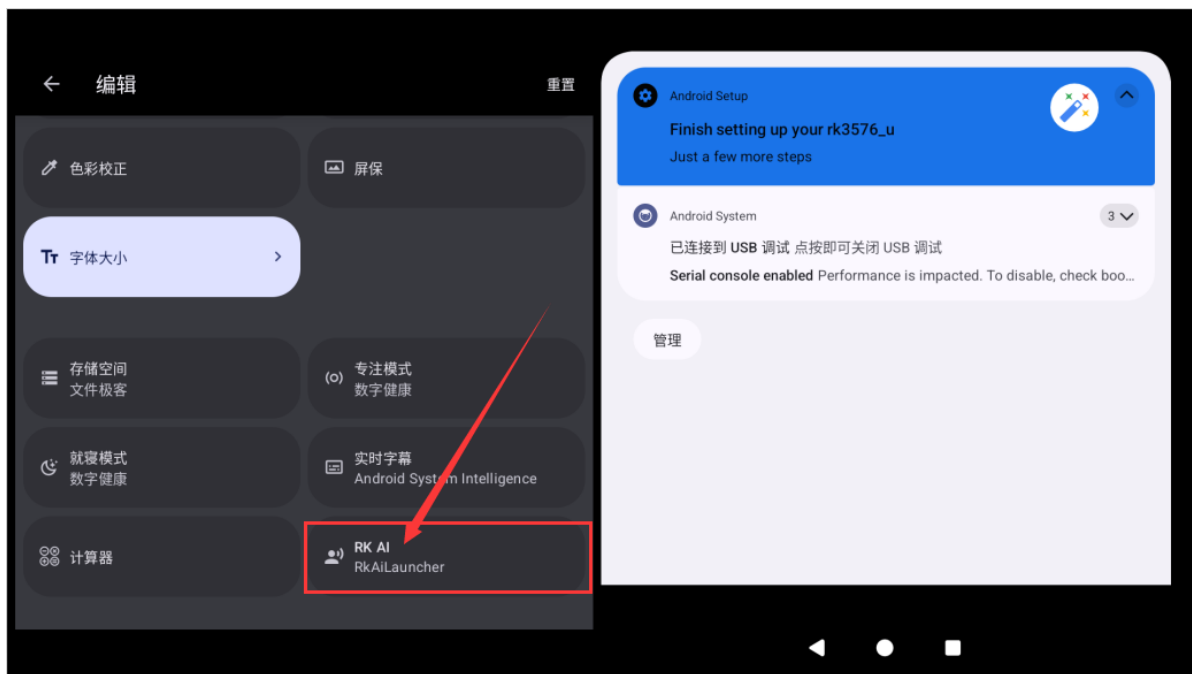
3.1.1 apk路径

```
vendor/rockchip/common/apps/RkAiLauncher/
```

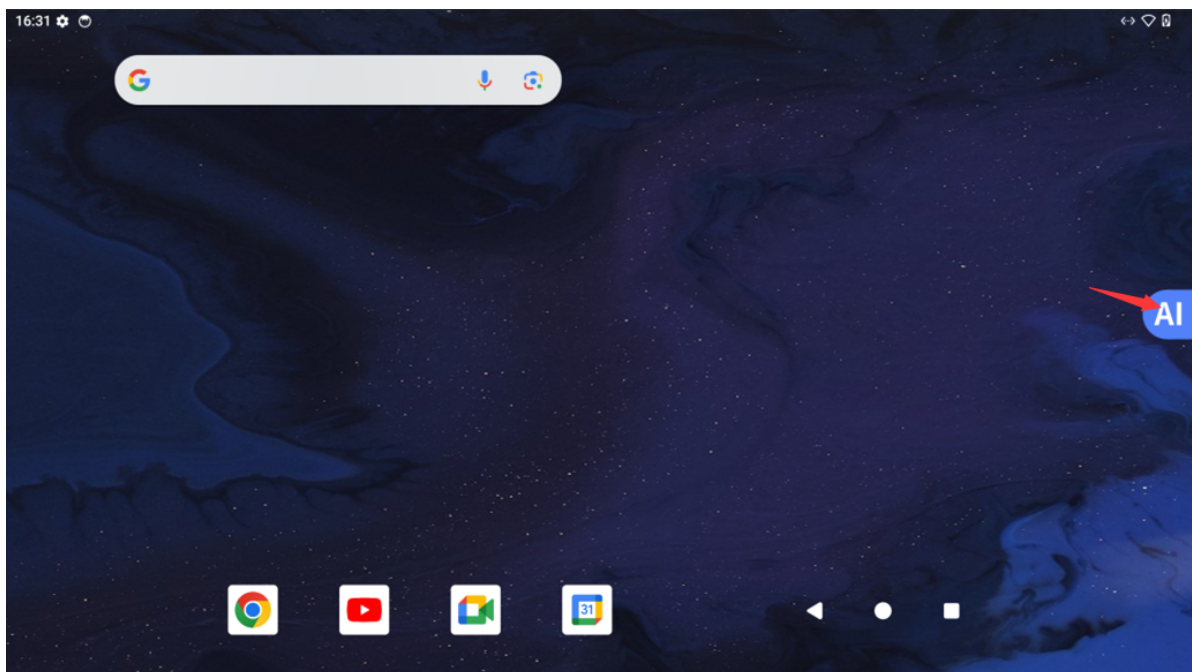
该目录为编译好的悬浮球apk文件，如需修改布局，请提redmine申请源码。

3.1.2 开启使用

启用AI功能后，开机下拉状态栏，可以看到RK AI的快速按钮



拖动到顶部后点击开启，出现"AI"悬浮球，点击悬浮球进行后续AI模块功能的启用。



3.2 RkAuth授权

3.2.1 源码路径

```
packages/apps/RkAuth
```

3.2.2 授权配置

rkauth_config.json文件会在编译时，生成在固件的vendor/etc/rkauth_config.json和/system/etc/rkauth_config.json路径下，供vendor和system进程的程序调用，进行模块授权。有新算法功能，可自行新增。以下是json字段说明，后文会在单独的模块中再次说明。

```
packages/apps/RkAuth/etc/rkauth_config.json
{
    "rkauth_modules_config": [{
        "module": "sr", //模块名: 超分
        "activate_code_vendor_storage_id": 60, //该模块的在线密钥存储的id, 已有的id
        请勿修改
        "activate_code_vendor_storage_size": 256, //在线密钥的长度
        "license_vendor_storage_id": 1060, //离线密钥存储位置, 开机联网后, RkAuth会
        读取该模块的在线密钥进行授权, 授权成功后, 将离线密钥写入到该位置
        "license_vendor_storage_size": 1024, //离线密钥长度
        "activate_code_path": "", //留空就好
        "license_path": "/sdcard/svep_key.lic" //在线授权成功, 也会将离线密钥也生成
        在该位置, 供读取不到vendorstorage的应用进行单独调用
    },
    {
        "module": "memc",
        "activate_code_vendor_storage_id": 62,
        "activate_code_vendor_storage_size": 256,
        "license_vendor_storage_id": 1062,
        "license_vendor_storage_size": 1024,
        "activate_code_path": "",
        "license_path": "/sdcard/memc_key.lic"
    },
    ... //根据算法功能可自行新增
    ]
}
```

3.3 超分

3.3.1 源码位置

```
hardware/rockchip/libsvpe/libsvpepsr
```

3.3.2 授权配置

```
packages/apps/RkAuth/etc/rkauth_config.json
{
    "rkauth_modules_config": [{
        "module": "sr", //模块名: 超分
        "activate_code_vendor_storage_id": 60, //超分在线密钥存储在vendorstorage
        的60位置, 这个id请勿修改。
        "activate_code_vendor_storage_size": 256, //在线密钥的长度
        "license_vendor_storage_id": 1060, //离线密钥存储位置, 超分成功授权后, 离线密
        钥会自动写入到vendorstorage的1060位置
        "license_vendor_storage_size": 1024, //离线密钥长度
        "activate_code_path": "",
        "license_path": "/sdcard/svep_key.lic" //超分成功授权后, 也会在该位置生成
        lic文件。像图片超分这类自己写app, 调用libsvpe库API的操作, 需要配置app能访问的路径。
    },
    ...
    ]
}
```

}

3.3.3 开启使用

打开视频，点击AI悬浮球中“超分”按钮，开启超分：



3.4 MEMC

3.4.1 源码位置

```
hardware/rockchip/libsvep/libsvepmemc
```

3.4.2 授权配置

```
packages/apps/RkAuth/etc/rkauth_config.json
{
  "rkauth_modules_config": [
    . . .
    {
      "module": "memc", //模块名: memc
      "activate_code_vendor_storage_id": 62, //memc在线密钥存储在vendorstorage
      的62位置, 这个id请勿修改。
      "activate_code_vendor_storage_size": 256, //在线密钥的长度
      "license_vendor_storage_id": 1062, //离线密钥存储位置, memc成功授权后, 离线
      密钥会自动写入到vendorstorage的1062位置
      "license_vendor_storage_size": 1024, //离线密钥长度
      "activate_code_path": "",
      "license_path": "/sdcard/memc_key.lic" //memc成功授权后, 也会在该位置生成
      lic文件。
    },
    . . .
  ]
}
```

3.4.3 开启使用

打开视频, 点击AI悬浮球中“MEMC”按钮, 开启“MEMC”:



3.5 PQ

3.5.1 源码位置

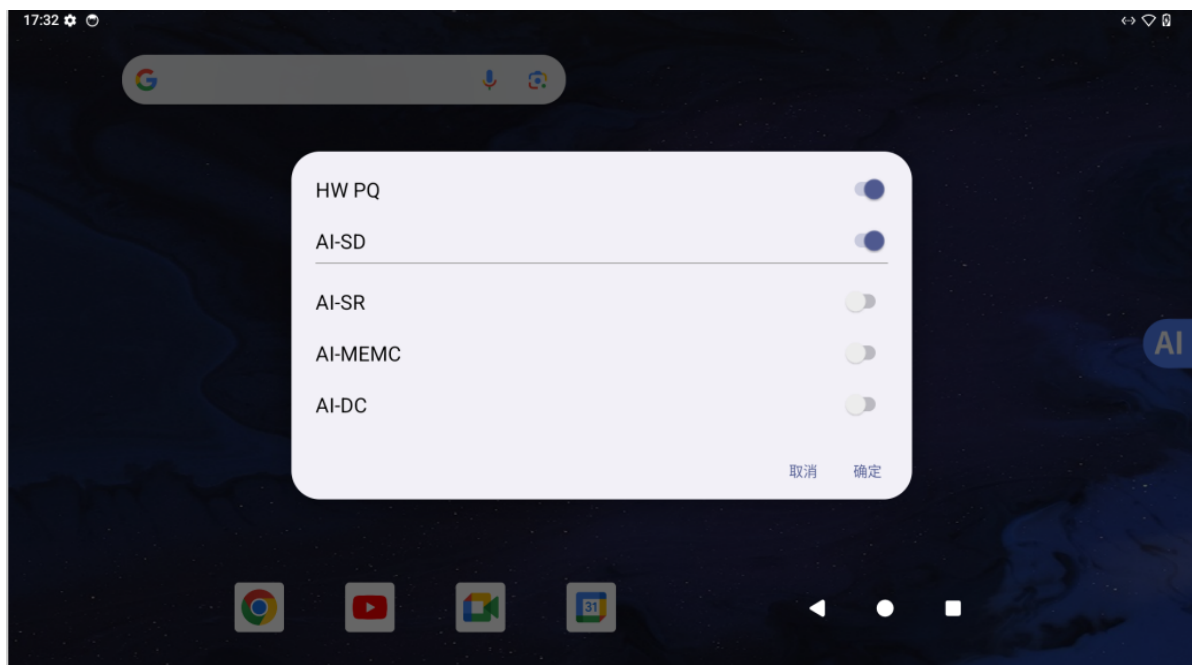
```
hardware/rockchip/libpq
```

3.5.2 授权配置

```
packages/apps/RkAuth/etc/rkauth_config.json
{
  "rkauth_modules_config": [
    . . .
    {
      "module": "pq", //模块名: pq
      "activate_code_vendor_storage_id": 64, //pq在线密钥存储在vendorstorage的
      64位置, 这个id请勿修改。
      "activate_code_vendor_storage_size": 256, //在线密钥的长度
      "license_vendor_storage_id": 1064, //离线密钥存储位置, pq成功授权后, 离线密
      钥会自动写入到vendorstorage的1064位置
      "license_vendor_storage_size": 1024, //离线密钥长度
      "activate_code_path": "",
      "license_path": "/sdcard/pq_key.lic" //pq成功授权后, 也会在该位置生成lic文
      件。
    },
    . . .
  ]
}
```

3.5.3 开启使用

点击AI悬浮球中“PQ”按钮。PQ功能分为HWPQ和AIPQ 2个模块。其中, 启用AIPQ中AI-SD的前提是HWPQ要打开, 因此弹窗配置界面上对AI-SD和HW PQ做了联动处理。



3.6 在线字幕

3.6.1 源码位置

```
vendor/rockchip/common/apps/RkAiService/
```

该目录为编译好的apk文件，如需修改，请提redmine申请源码

3.6.2 授权配置

在线字幕需要用到asr语音转文字，以及llm文字翻译功能

```
packages/apps/RkAuth/etc/rkauth_config.json
{
    "rkauth_modules_config": [
        . . .
        {
            "module": "asr", //模块名: asr 语音转文字
            "activate_code_vendor_storage_id": 84, //asr在线密钥存储在vendorstorage
            的84位置，这个id请勿修改。
            "activate_code_vendor_storage_size": 256,
            "license_vendor_storage_id": 1084,
            "license_vendor_storage_size": 1024,
            "activate_code_path": "",
            "license_path": "/sdcard/rkai/key_asr.lic" //RkAiService会读取这个文件
            进行asr的初始化
        },
        {
            "module": "llm", //模块名: llm 文字翻译
            "activate_code_vendor_storage_id": 86, //llm在线密钥存储在vendorstorage
            的84位置，这个id请勿修改。
            "activate_code_vendor_storage_size": 256,
            "license_vendor_storage_id": 1086,
            "license_vendor_storage_size": 1024,
            "activate_code_path": "",
            "license_path": "/sdcard/rkai/key_llm.lic" //RkAiService会读取这个文件
            进行llm的初始化
        },
        . . .
    ]
}
```

3.6.3 开启使用

播放视频的时候，点击AI悬浮球的“字幕”按钮开启在线字幕功能：



3.7 AI搜图

3.7.1 源码位置

```
vendor/rockchip/common/apps/RkAiGallery/
```

该目录为编译好的apk文件，如需修改，请提redmine申请源码

3.7.2 授权配置

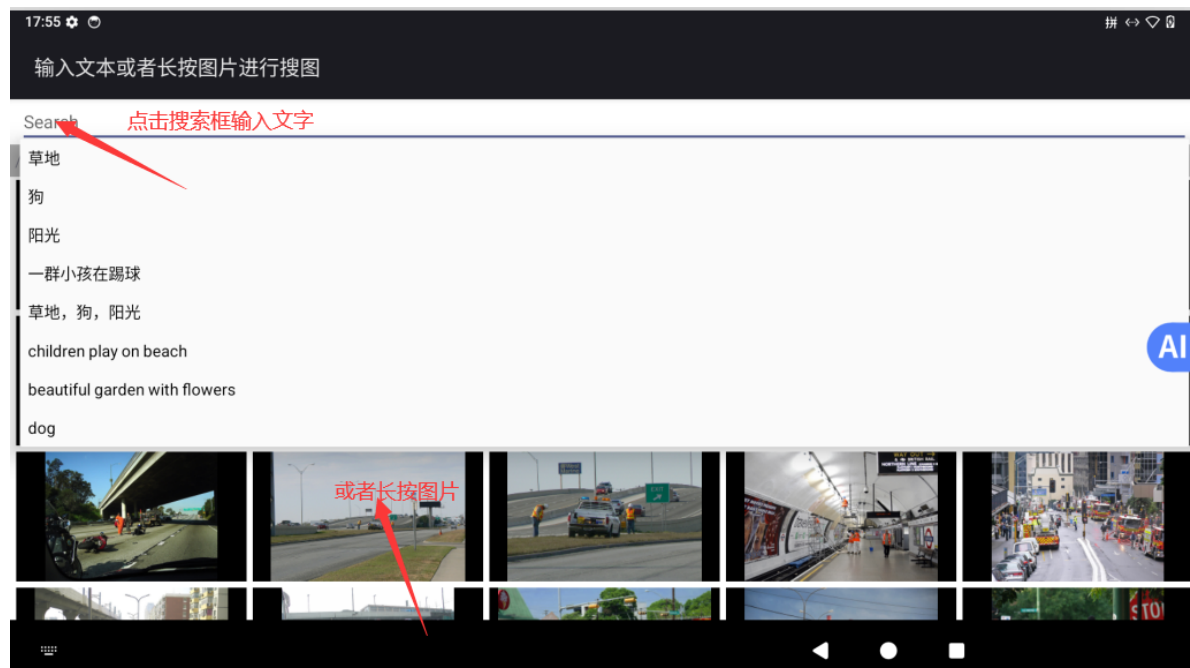
```
packages/apps/RkAuth/etc/rkauth_config.json
{
  "rkauth_modules_config": [
    . . .
    {
      "module": "face", //注意：搜图和人脸共用face这个模块名，在vendorstorage存储位置是不同的
      "activate_code_vendor_storage_id": 98, //在线密钥存储在vendorstorage的98位置，这个id请勿修改。
      "activate_code_vendor_storage_size": 256,
      "license_vendor_storage_id": 1098,
      "license_vendor_storage_size": 1024,
      "activate_code_path": "",
      "license_path": "/sdcard/rkai/key.lic" //RkAiGallery会读取这个文件进行初始化
    },
    . . .
  ]
}
```

3.7.3 开启使用

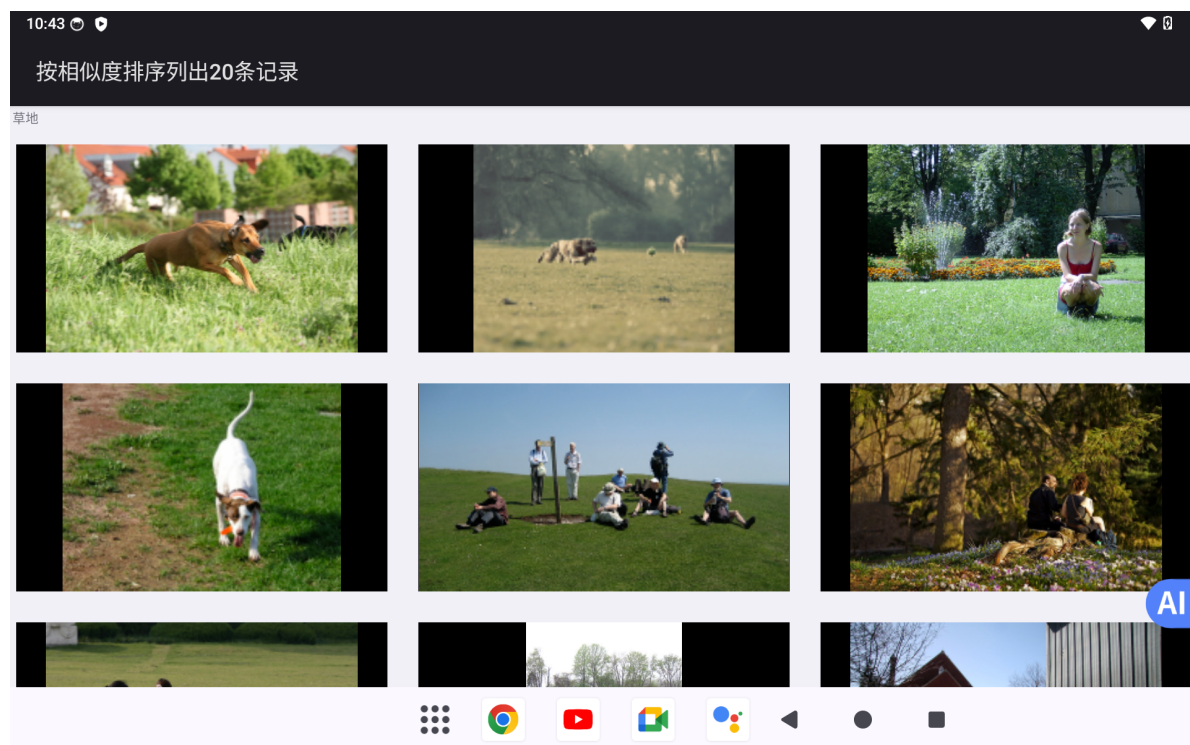
点击AI悬浮球中“搜图”按钮，启动AI搜图应用：

文本搜图：输入文字进行关键词搜图

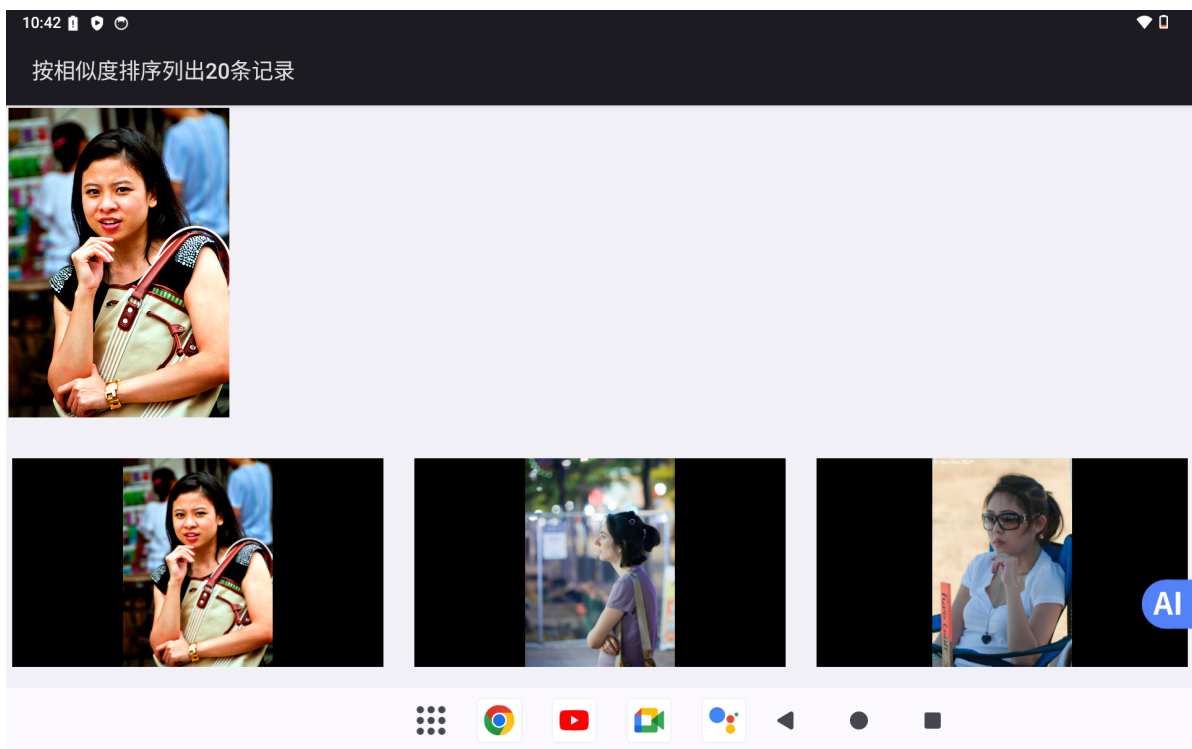
以图搜图：长按图片进行搜图



文本搜图结果：



以图搜图结果：



3.8 AI橡皮擦

3.8.1 源码位置

```
vendor/rockchip/common/apps/RkAiErase/
```

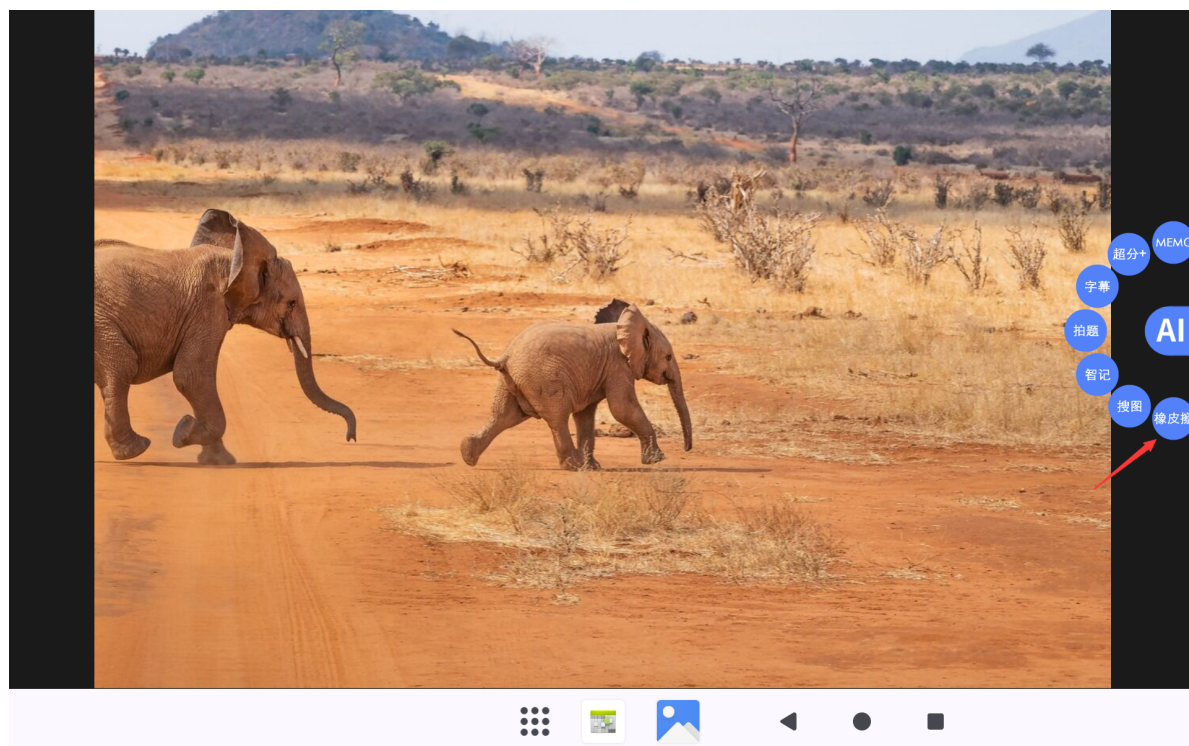
该目录为编译好的apk文件，如需修改，请提redmine申请源码

3.8.2 授权配置

```
packages/apps/RkAuth/etc/rkauth_config.json
{
    "rkauth_modules_config": [
        . . .
        {
            "module": "aieraser", //模块名: aieraser 橡皮擦
            "activate_code_vendor_storage_id": 96, //在线密钥存储在vendorstorage的96
            位置，这个id请勿修改。
            "activate_code_vendor_storage_size": 256,
            "license_vendor_storage_id": 1096,
            "license_vendor_storage_size": 1024,
            "activate_code_path": "",
            "license_path": "/sdcard/rkai/RkAiErase/models_RkAiErase/key.lic"
            //RkAiErase会读取这个文件进行初始化
        },
        . . .
    ]
}
```

3.8.3 开启使用

点击“图库”应用，在图库应用中，选中一张图片，再点击AI悬浮球，悬浮球中就会出现“橡皮擦”的选项：



点击“橡皮擦”就会跳转对应的擦除页面，先选中一块区域，再点击右上角擦除按钮，就可以对该图片进行擦除：

