視距離推定技術 Androidアプリ

SDK組み込みマニュアル

株式会社スワローインキュベート

2023年09月22日

■はじめに

・視距離推定技術 Androidアプリは、株式会社スワローインキュベートが提供しています。

・本書は、予告なしに変更されることがあります。

・本書を無断で、複製、転用、公衆送信、貸与等を行わないようお願いします。

・本サンプルアプリをご利用いただくには、あらかじめ当社利用規約に同意いただく必要があります。
詳しくは営業担当までお問い合わせください。

■ご注意事項

・SDKをご購入いただいていない場合は、本アプリのソースコードはすべて当社に帰属します。

・当社は本プログラムに関する保証は一切行っておりません。お客様の責任にてご使用ください。
本プログラム使用による問題や損害が生じた場合にも当社は何ら責任を負いませんので、あらかじめご了承ください。

お問い合わせ

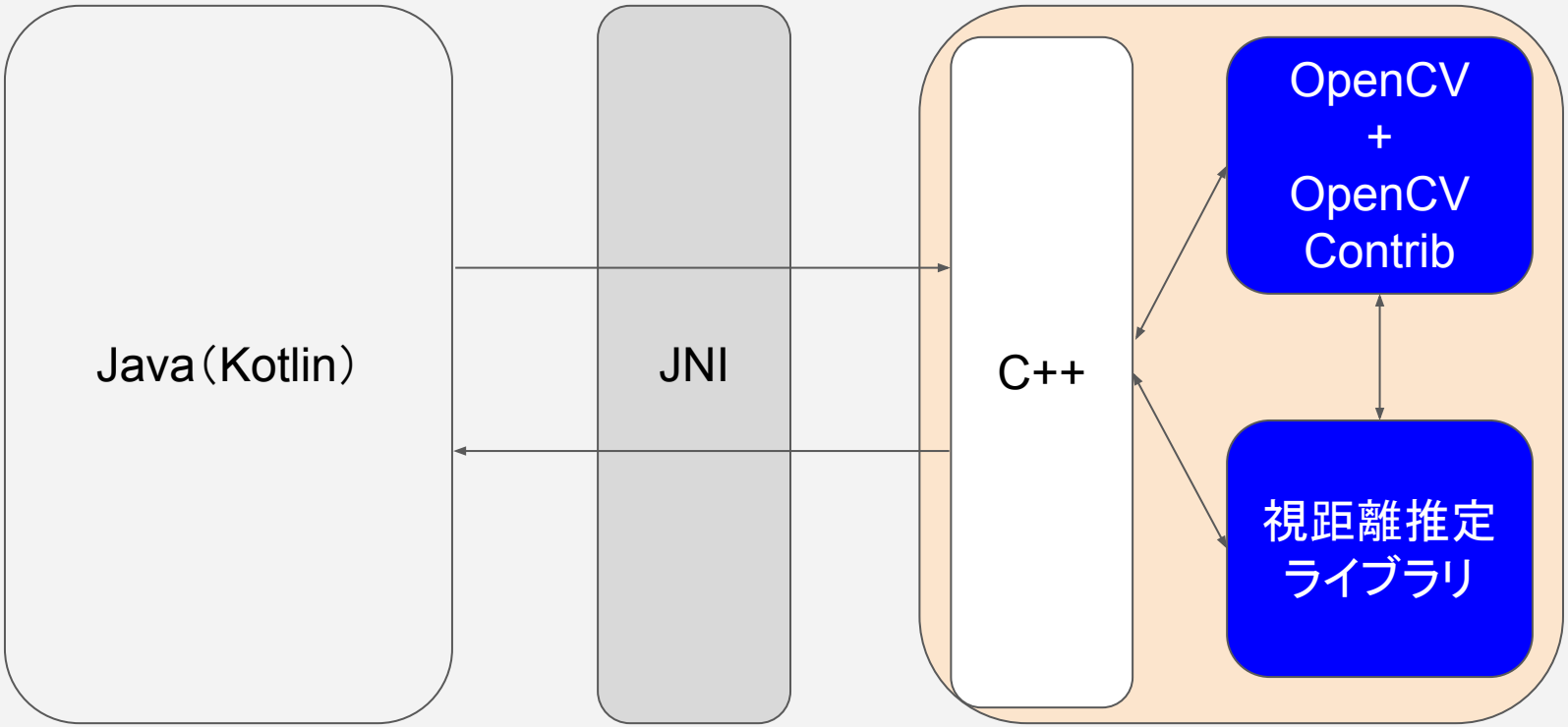
株式会社スワローインキュベート

視距離推定技術　テクニカルサポート窓口

MAIL: support@swallow-incubate.com

■視距離推定SDK組み込み時の構成

当社のso形式で提供される視距離推定 SDKは、C++インターフェースとなっています。
SDKの各インターフェースを呼び出すためには、JNI(Java Native Interface)を介してC++言語(C++11)から呼び出していただく必要があります。
また本SDKはOpenCVにも依存しているため、当社より提供予定の OpenCV Frameworkも同様の形式で呼び出していただきます。詳しくはサンプルアプリのファイル構成及びソースコードを参照してください。

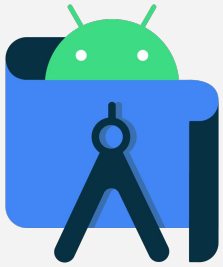


※JNI・・・Java Native Interface

同梱内容

■同梱内容

以下のファイルが提供時の主要なファイルとなります。



Android Studioプロジェクトファイル
viewdist-android-sample

視距離推定
Androidサンプルアプリ

soファイル
libViewDist.so

視距離推定ライブラリ

soファイル
libopencv_***.framework

OpenCVライブラリ

期限付きトークン
(試用版)

アクティベーションツール

ご利用マニュアル
(本書)

事前準備

■事前準備 - ご利用環境

現在のバージョンでは、以下のご利用環境を推奨しています。

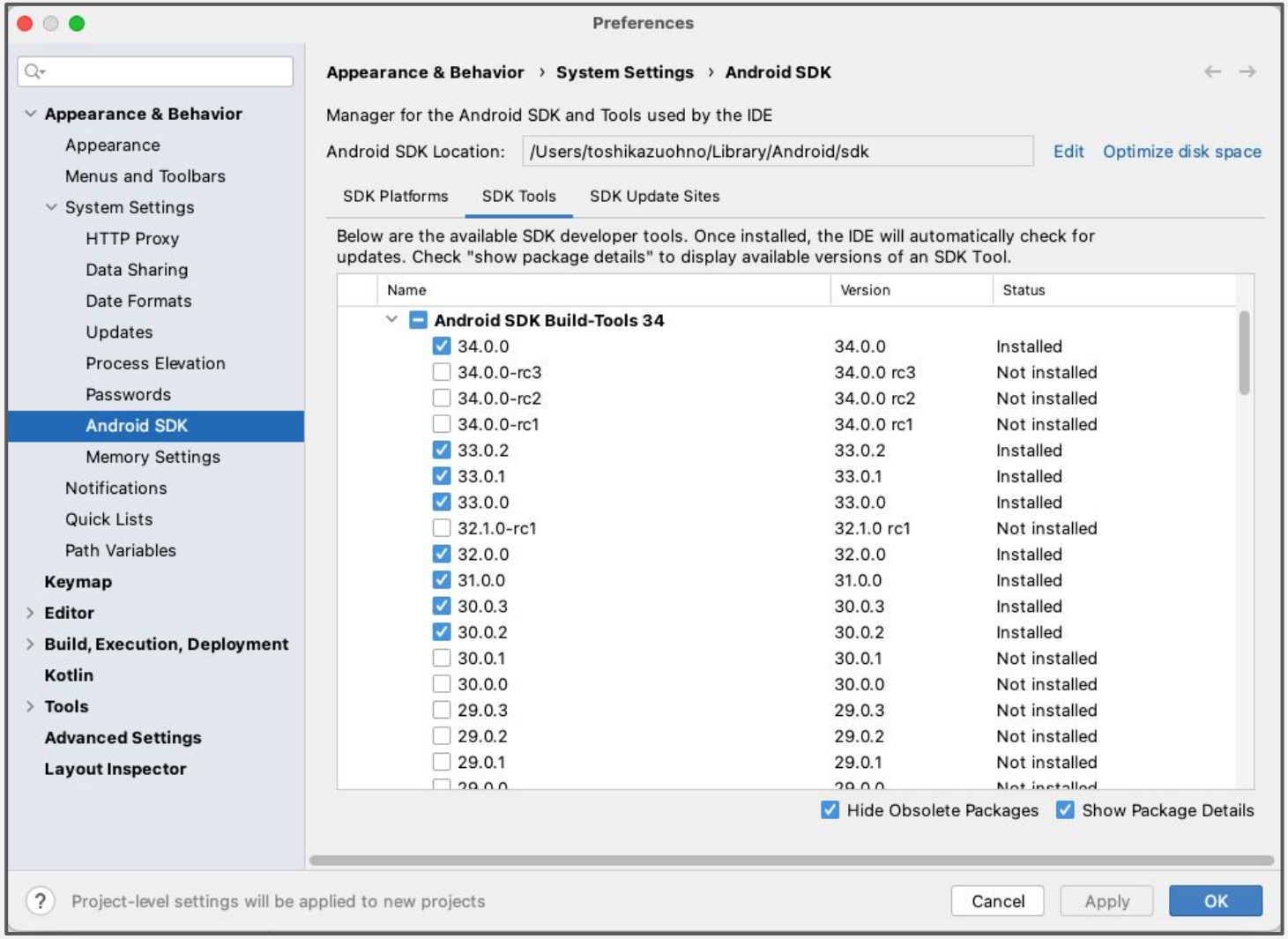
項目		内容
開発環境	Android SDK	Android 7.0以降（APIレベル 24以降）
	Android NDK	25.0.8775105（Android NDK 21以降）
	Java環境	JDK 17
	プログラム言語	Java（一部Kotlinを含む） / C++11
	開発環境	Android Studio 最新版を推奨
	開発マシンOS	Windows OS または macOS
	依存ライブラリ	ViewDist ライブラリ OpenCV 4.7.0 / OpenCV Contrib 4.7.0 （一部OpenCV_Contribモジュール入り）
動作環境	OS	Android 7.0以降
	CPUアーキテクチャ	arm64-v8a、armeabi-v7a、x86、x86_64
	推奨メモリ	2GB以上を推奨
	カメラ	フロントカメラ（90万画素以上）

■事前準備 - Android Studioのインストール

お使いのWindosOS/macOSマシン上に、Android Studioアプリをあらかじめインストールしておいてください。すでにインストール済みの方は、比較的新しいバージョンにアップデートしておくことを推奨します。

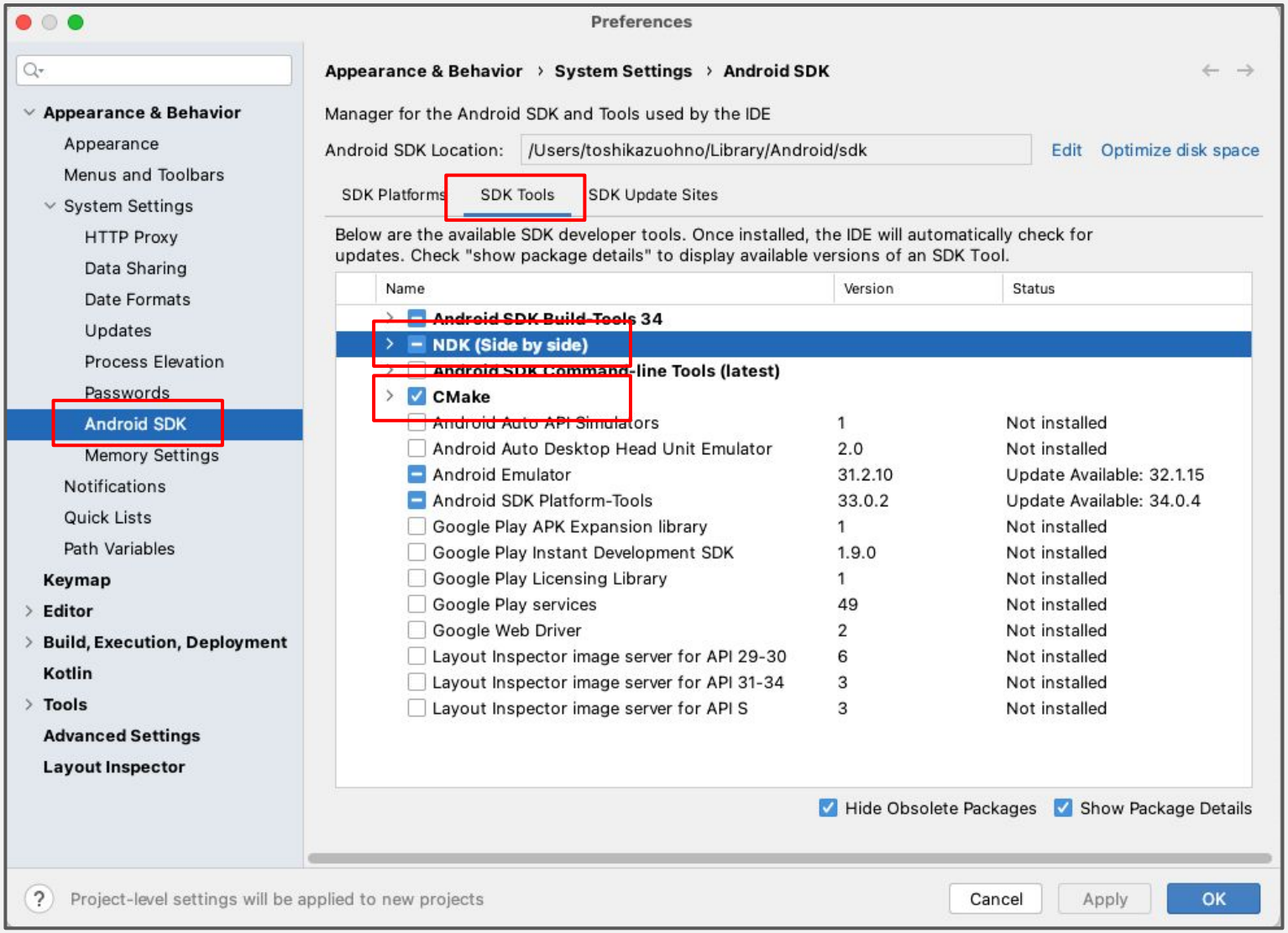
■事前準備 - Android NDK / CMakeの導入

本サンプルアプリをビルドするには、Android NDK及びCMakeが必要となります。
[Tools] → [SDK Manager] を選択し、下記のウィンドウを表示します。



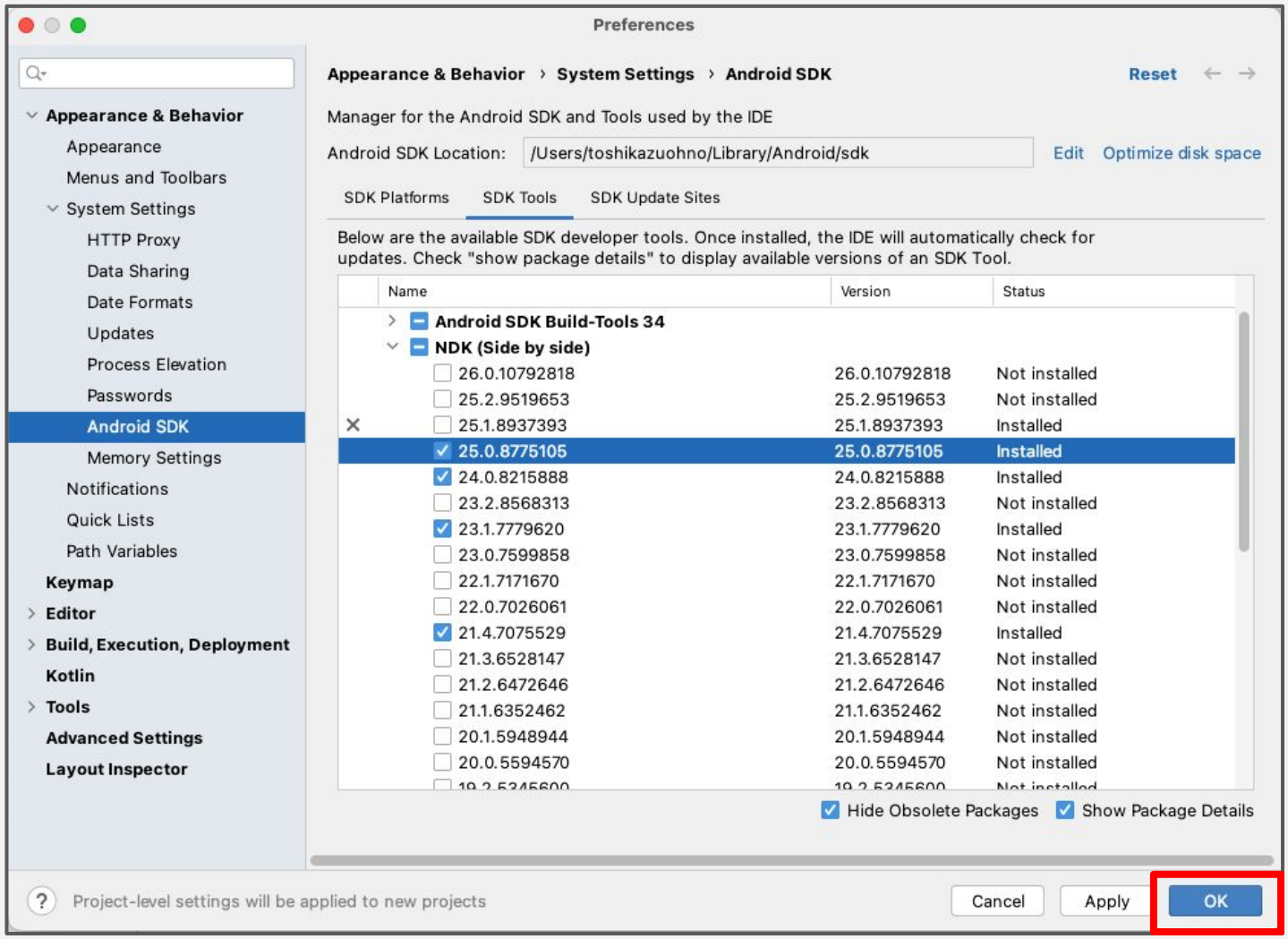
■事前準備 - Android NDK / CMakeの導入

[Android SDK] → [SDK Tools]を選択し **NDK(Side by side)とCMake**にチェックをつけます。バージョンを指定する場合は「**Show Package Details**」をクリックします。



■事前準備 - Android NDK / CMakeの導入

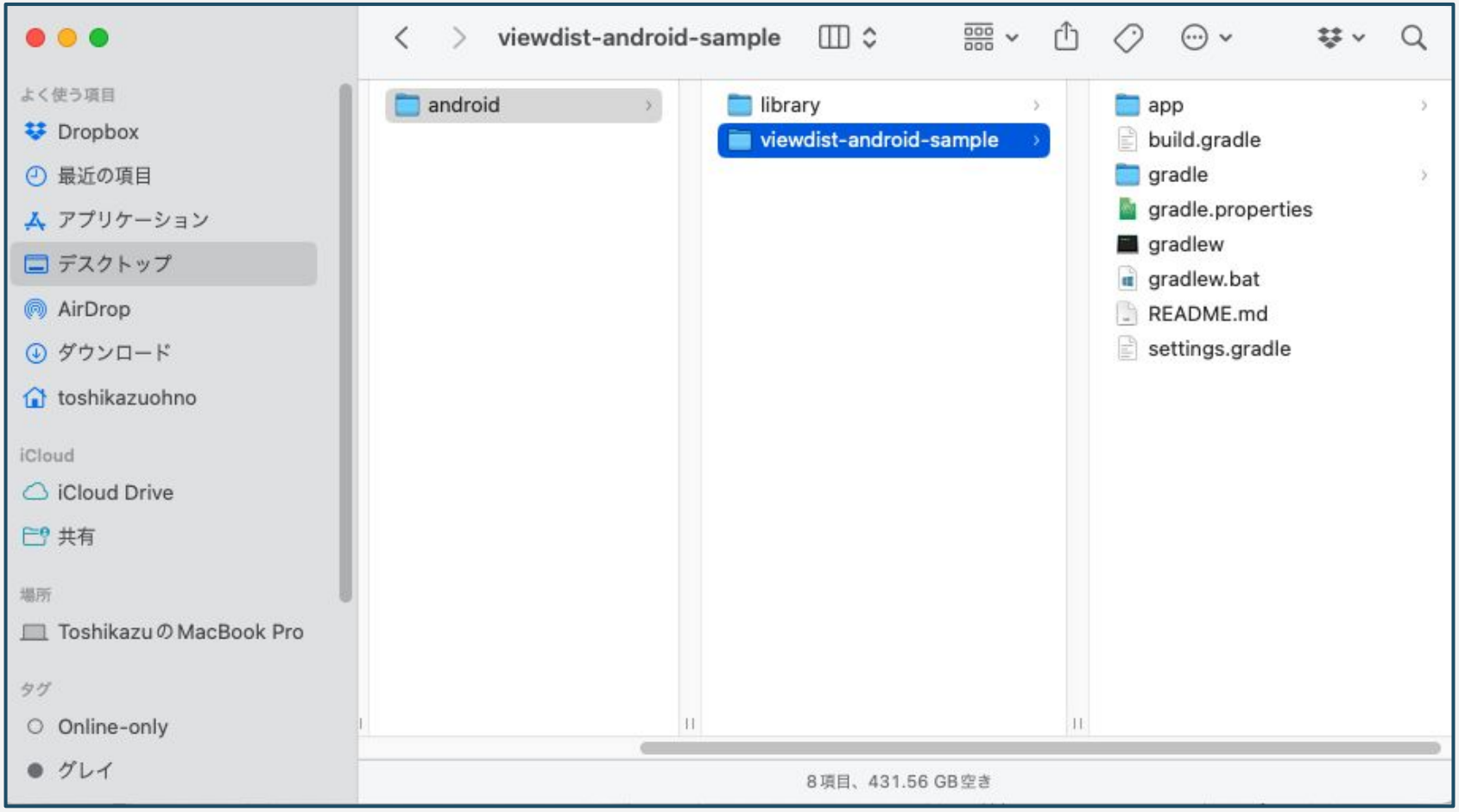
NDKはバージョン21以降を使用してください。本ガイドでは、**NDK ver.25.0.8775105**を使うものとして進めていきます。「OK」をクリックするとダウンロードがはじまりますので、導入はこれで完了となります。



プロジェクトを開く

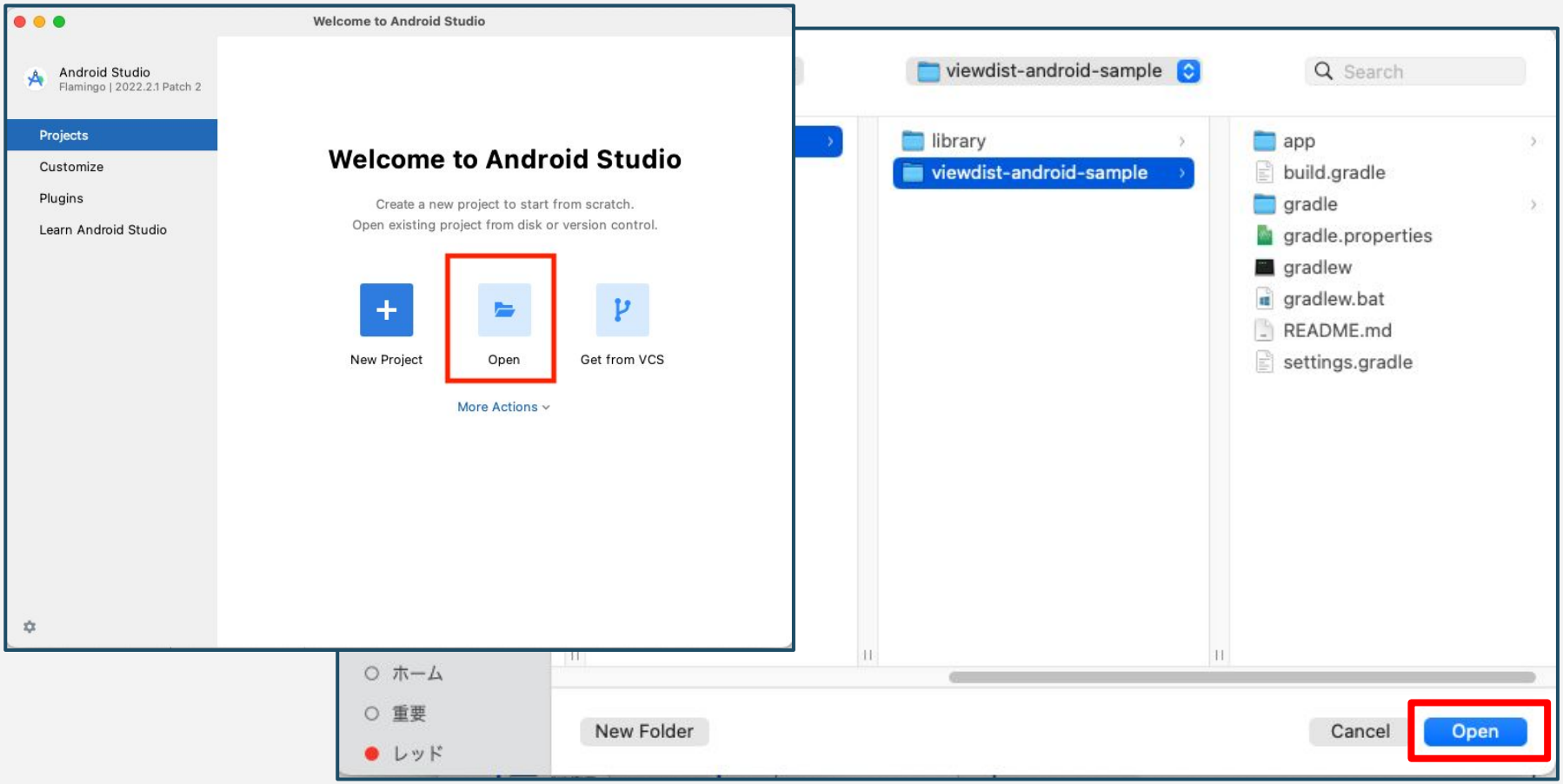
■プロジェクトを開く - 手順①

ご提供したAndroidサンプルアプリフォルダを任意の場所に設置してください。
本マニュアルでは、mackBookProのマシンのデスクトップのandroidディレクトリに置いたものとして説明をします。



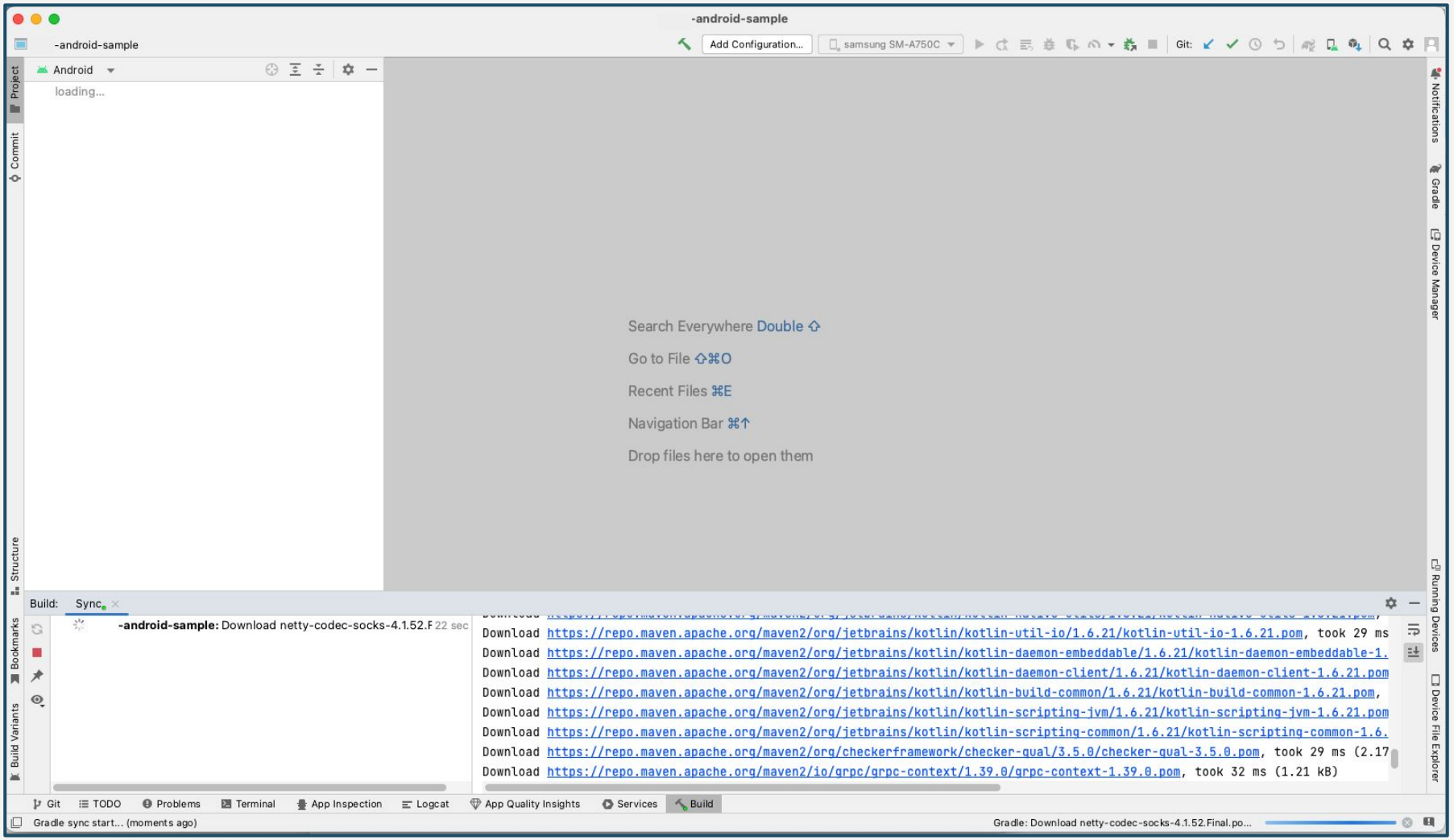
■プロジェクトを開く - 手順②

Android Studioを開き、「Open」を押すとプロジェクトディレクトリを選択できる画面が開きます。当社から提供される視距離推定Androidサンプルアプリのプロジェクトファイルを指定して「Open」をクリックして開いてください。視距離推定Androidサンプルアプリには、ソースコードやlayoutファイル、学習モデルファイル等が入っています。



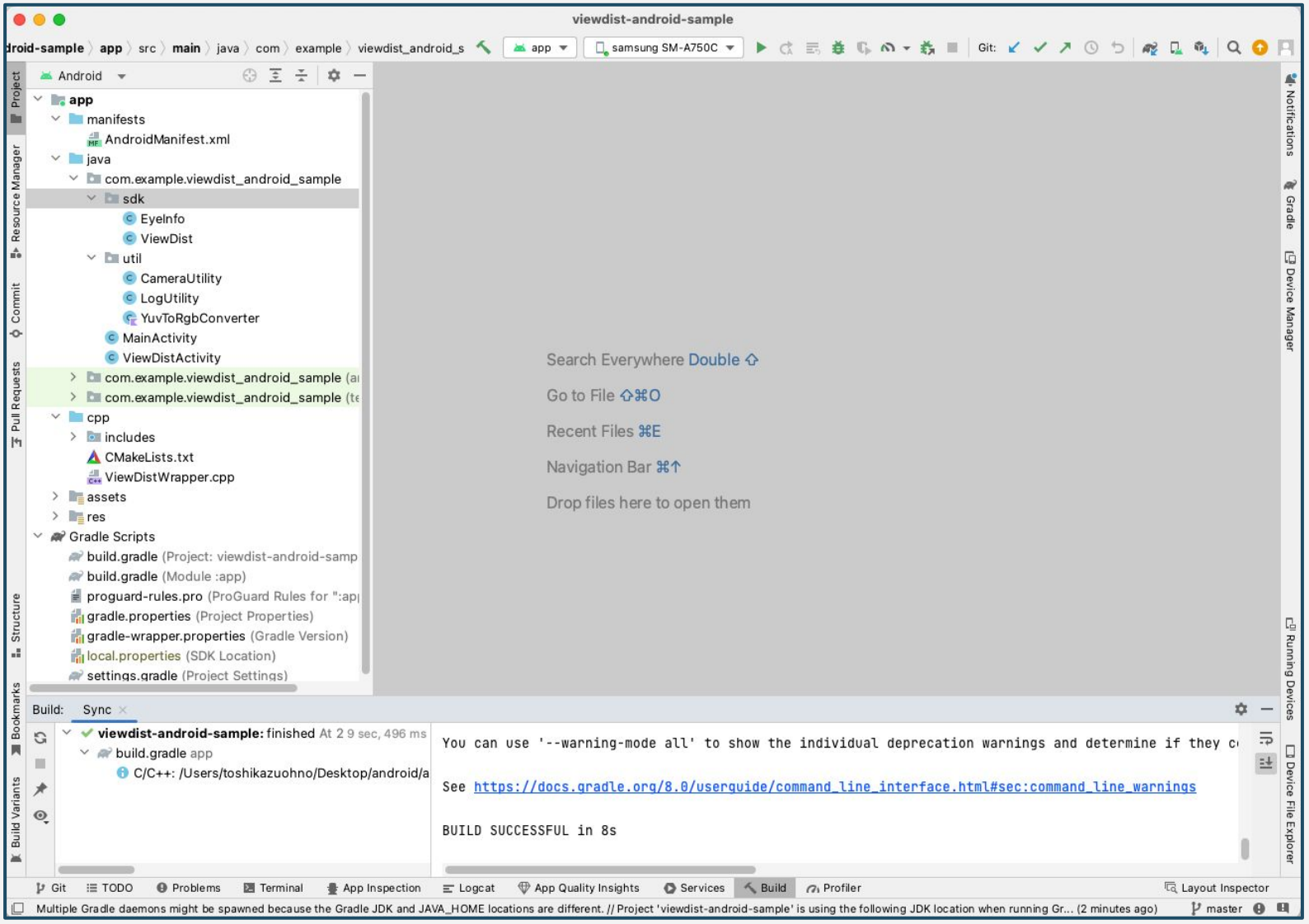
■プロジェクトを開く - 手順③

Android Studioで視距離推定サンプルアプリを開くと、以下のようなウィンドウが立ち上がります。自動的にビルドがはじまりますので、しばらく待ちます。



■プロジェクトを開く - 手順④

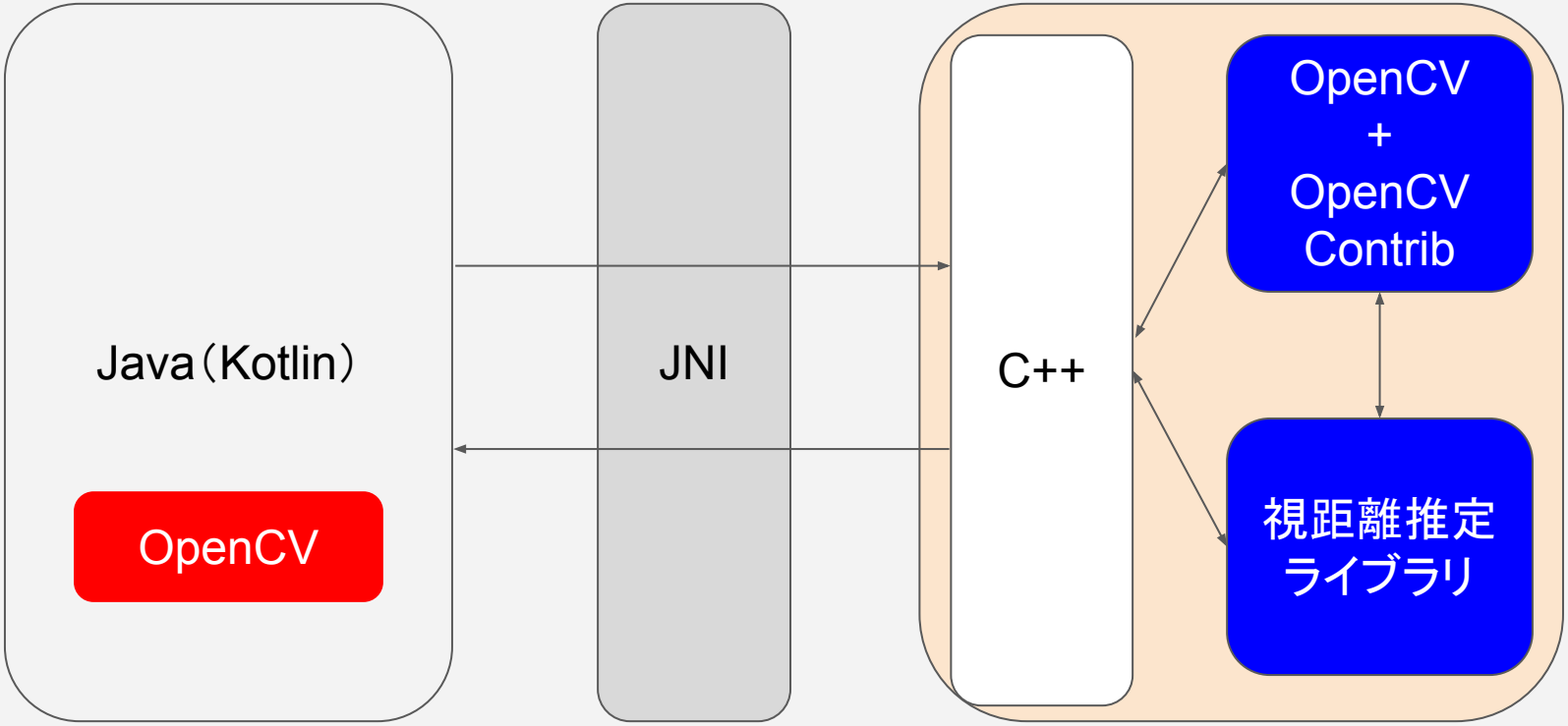
Buildログ画面に「BUILD SUCCESSFUL」と表示されれば、操作が可能になります。
以下のような画面になれば、プロジェクトの立ち上げは完了となります。



OpenCV for Javaの導入

■OpenCV for Javaの導入（補足）

本サンプルアプリでは、フレーム画像をMat型に変換する処理等を、Java側で記述している為、OpenCV for Javaを利用しておりますが、C++側で同等の処理を行う場合は、OpenCV for Javaのインポートは不要です。ここではまず、下図赤枠の部分のOpenCVの導入方法について説明をしていきます。

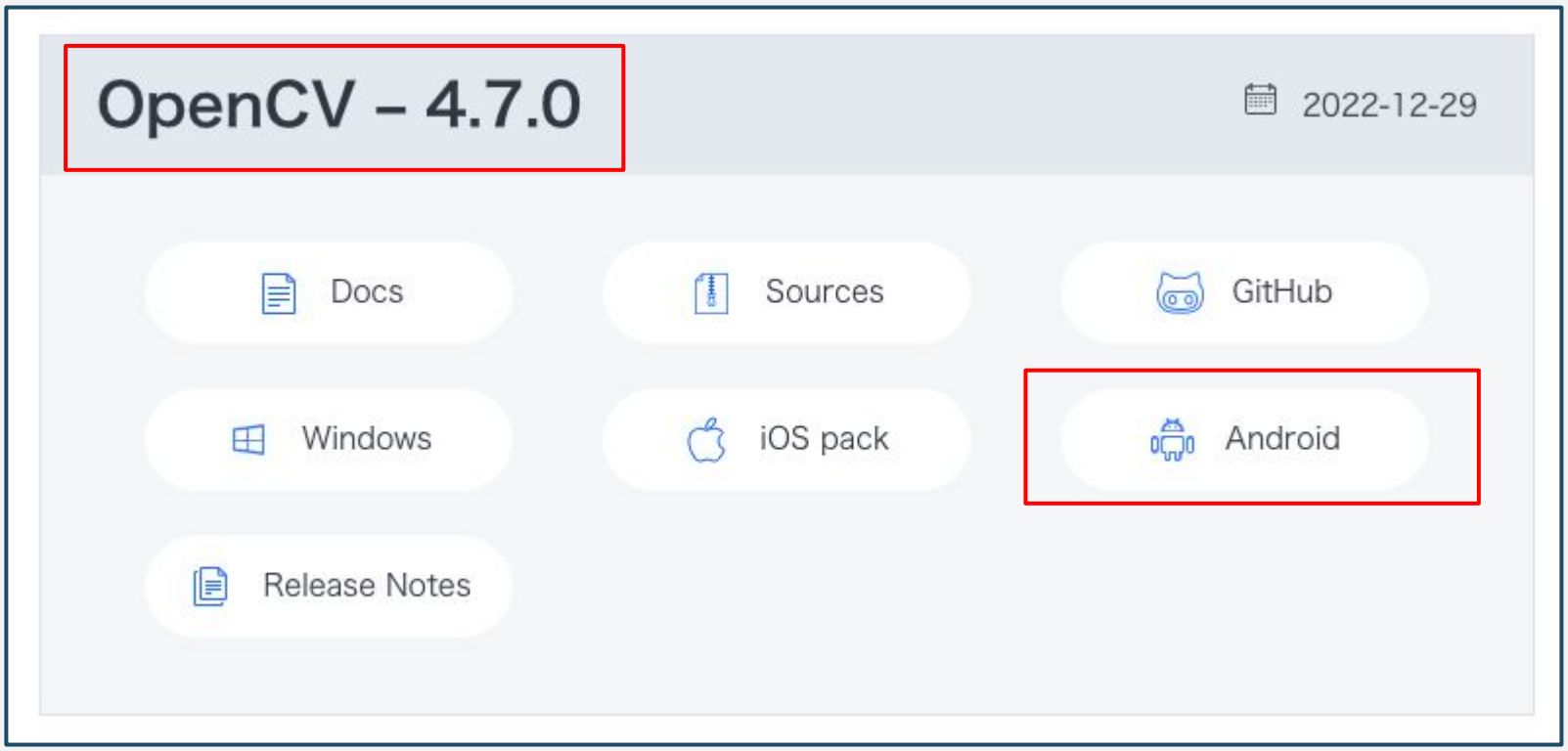


※JNI・・・Java Native Interface

■OpenCV for Javaの導入①

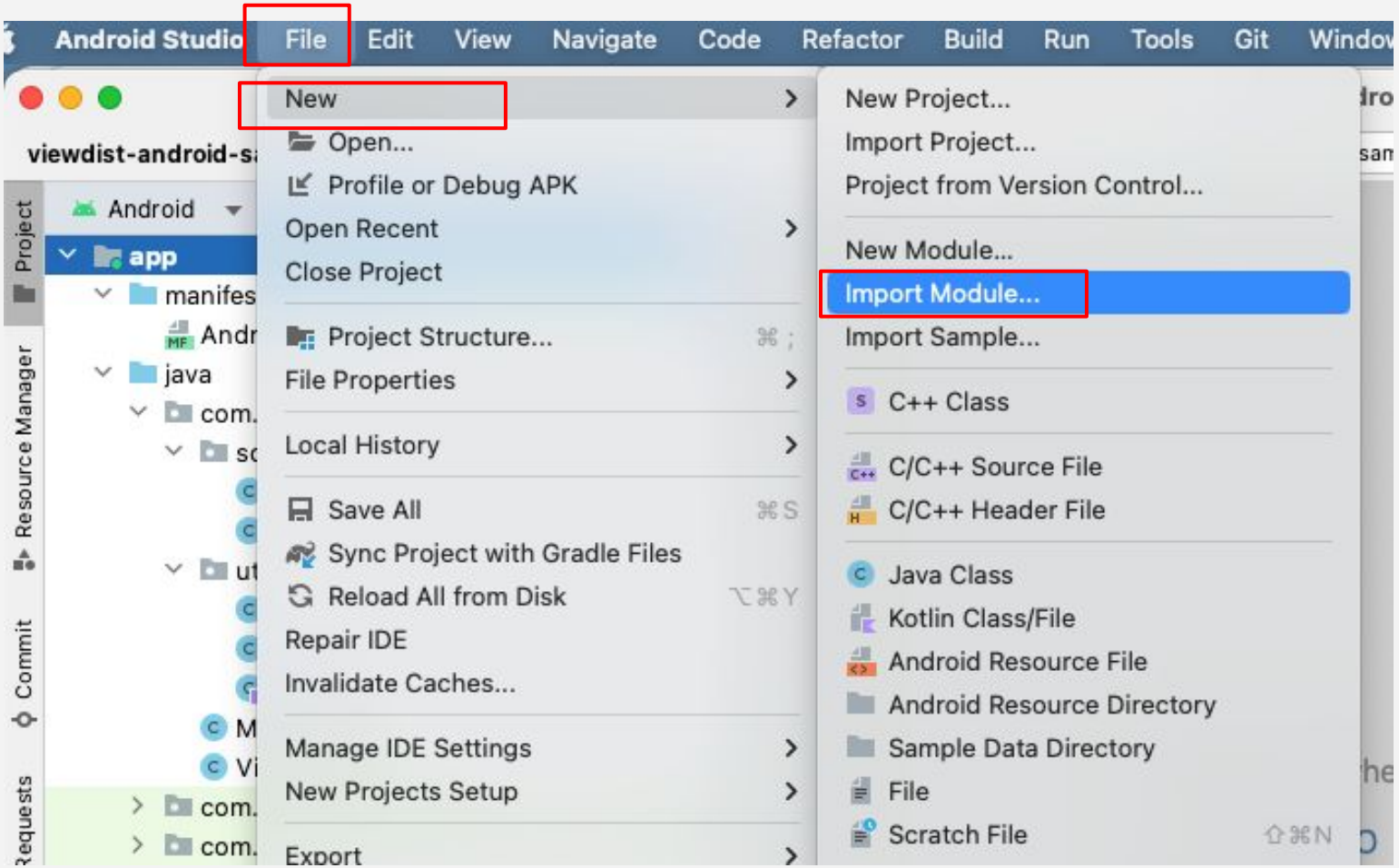
公式サイトよりOpenCVのAndroid向けモジュール一式（zip形式）をダウンロードし、解凍後、ご利用のAndroid Studioプロジェクトにインポートします。本SDKをご利用の際は、**OpenCV 4系**をダウンロードして下さい。（本書では4.7.0を使用します。）
格納先はプロジェクト内の任意のフォルダを指定して下さい。

公式サイトURL：https://opencv.org/releases/



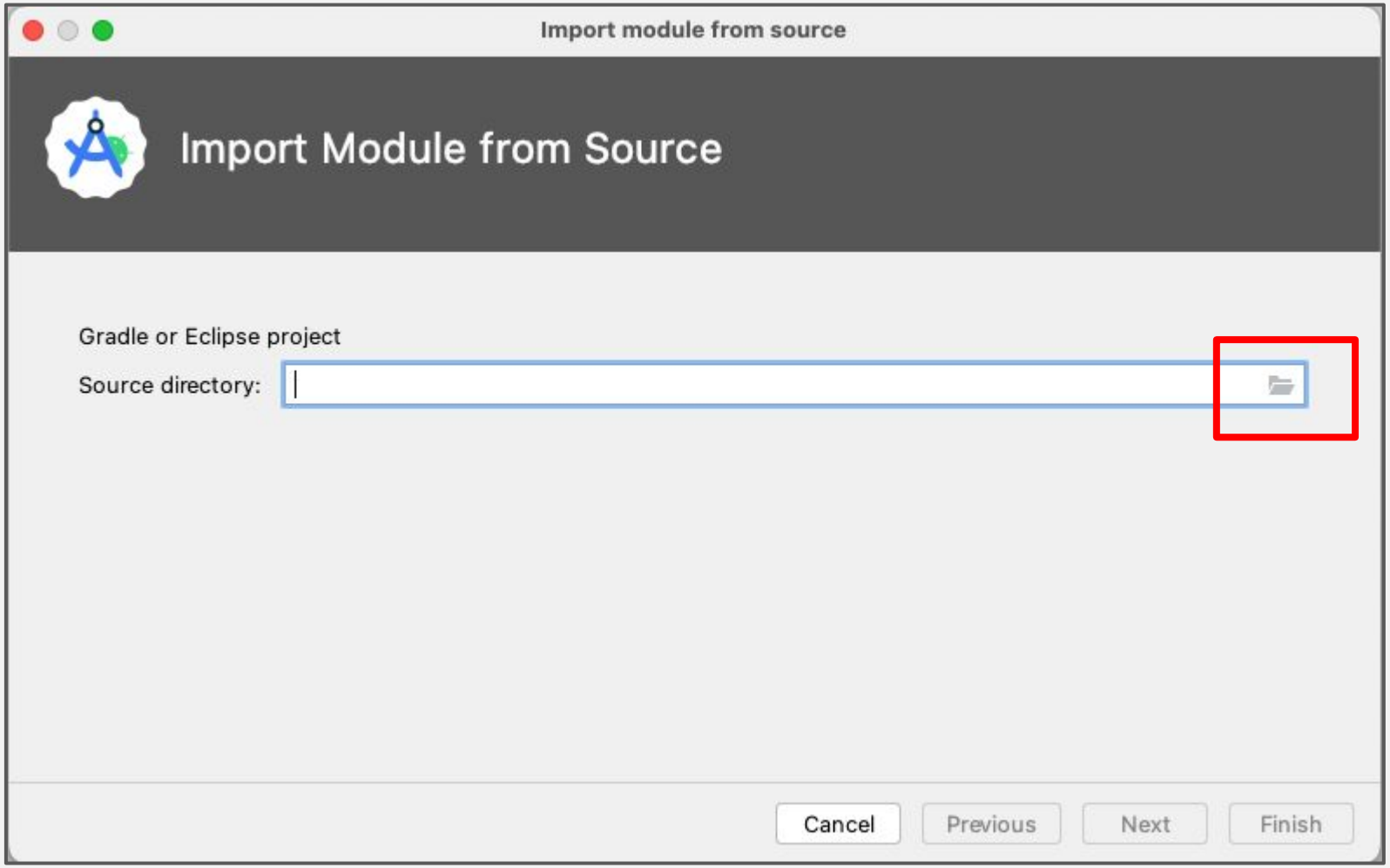
■OpenCV for Javaの導入②

OpenCVをAndroid Studioにインポートします。Android Studioのメニューから
[File] → [New] → [Import Module] を選択するとモジュールをインポートする為のダイアログが開きます。



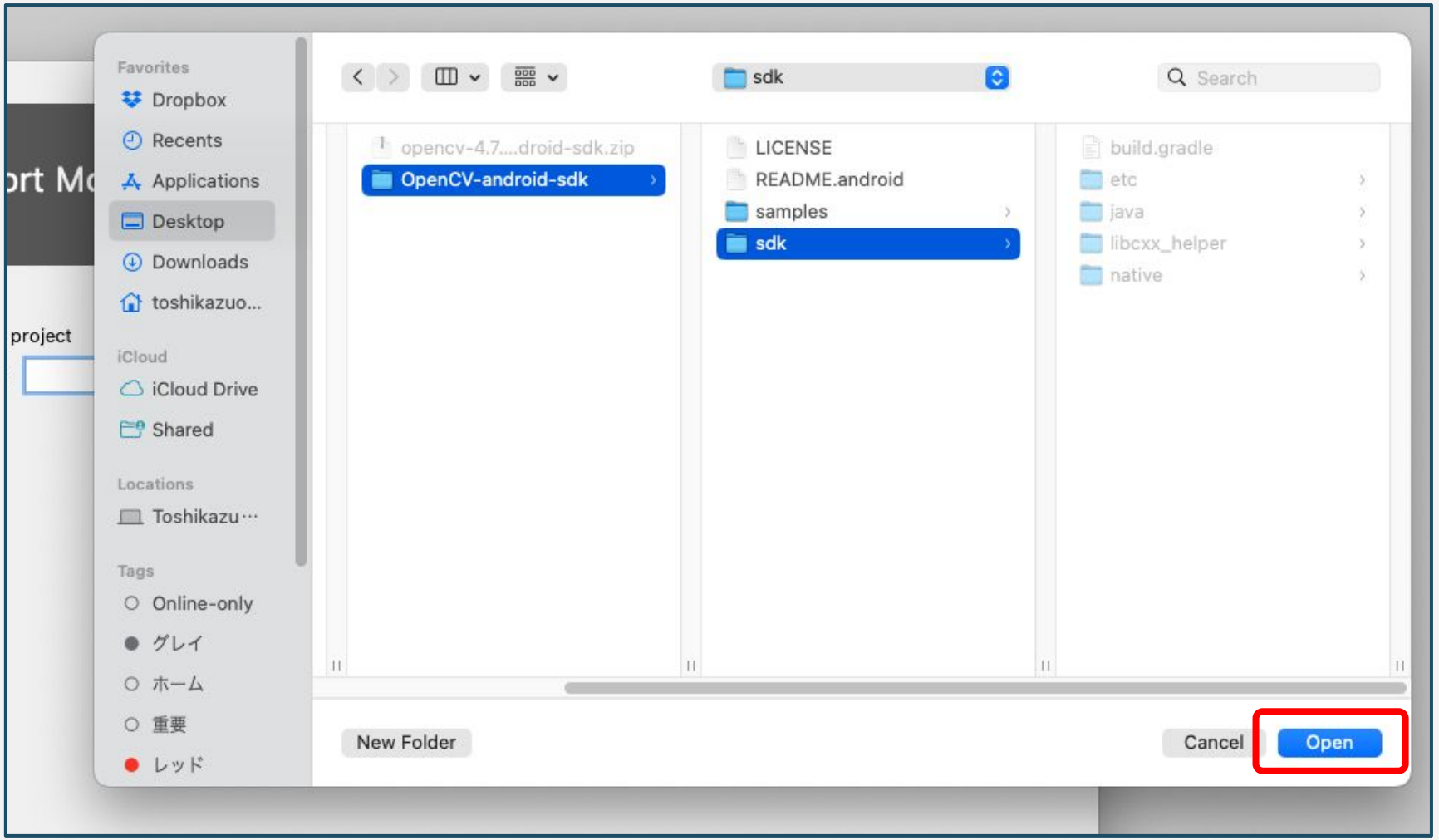
■OpenCV for Javaの導入③

Import module from sourceダイアログが表示されたら、Source directory欄の右端にあるフォルダ選択ボタンをクリックします。



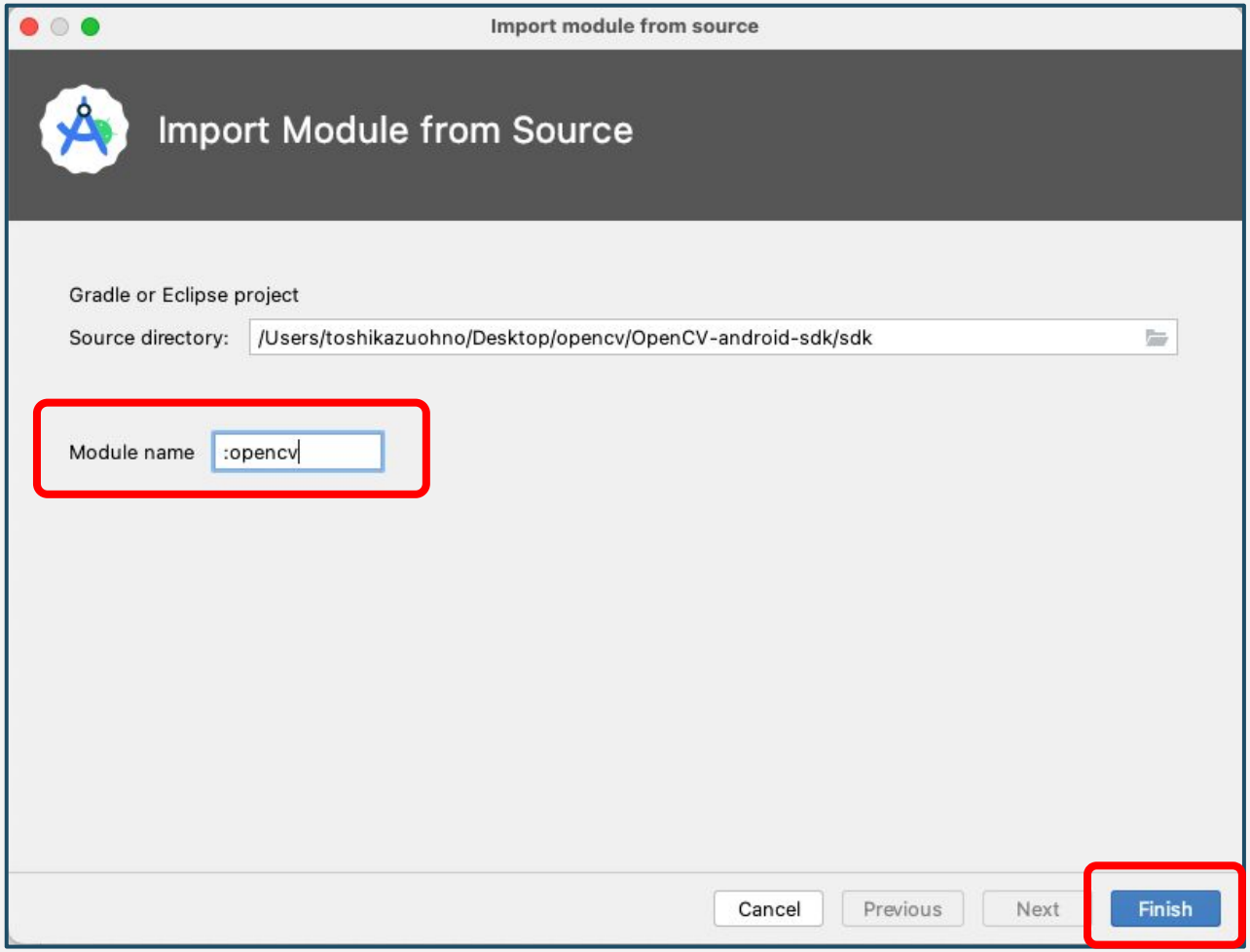
■OpenCV for Javaの導入④

先ほどダウンロードして解凍したフォルダOpenCV-android-sdkの中の「sdk」フォルダを選択して「Open」をクリックします。



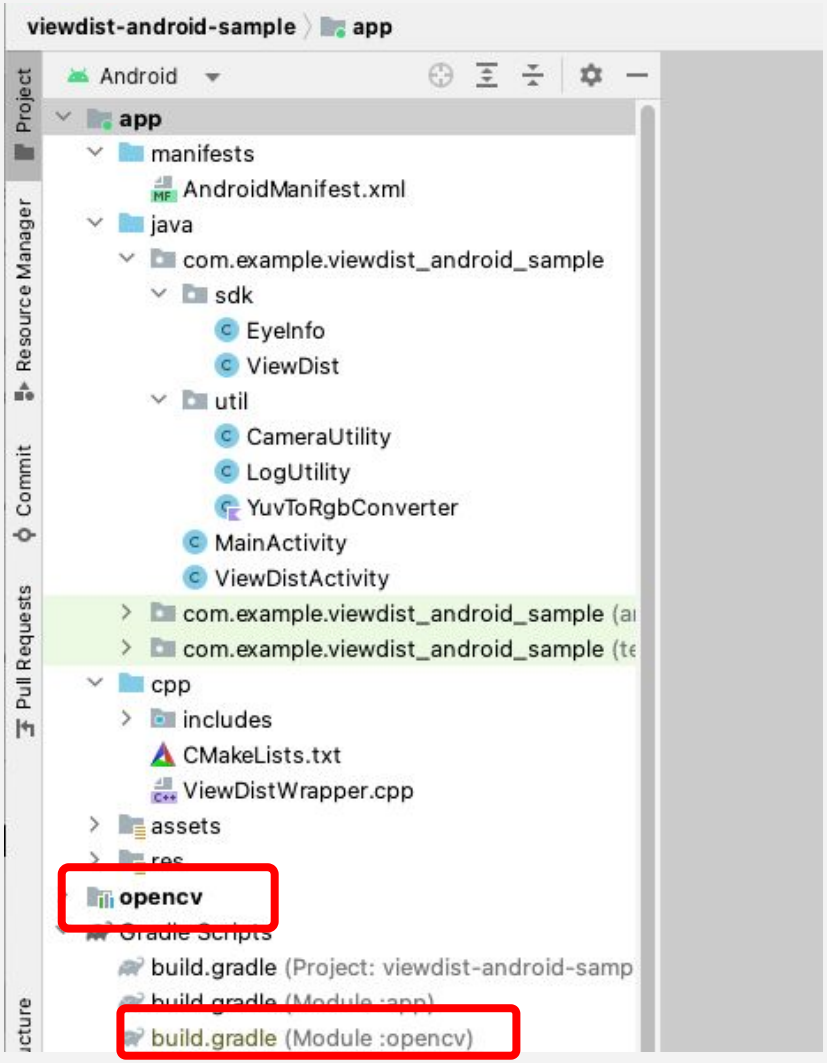
■OpenCV for Javaの導入⑤

選択が完了したら、以下のような画面となりディレクトリパスが入力されます。
Module nameを変更できますので、分かりやすい任意の名前に変更します。
（ここでは「:opencv」とします。）最後に[Finish]をクリックします。



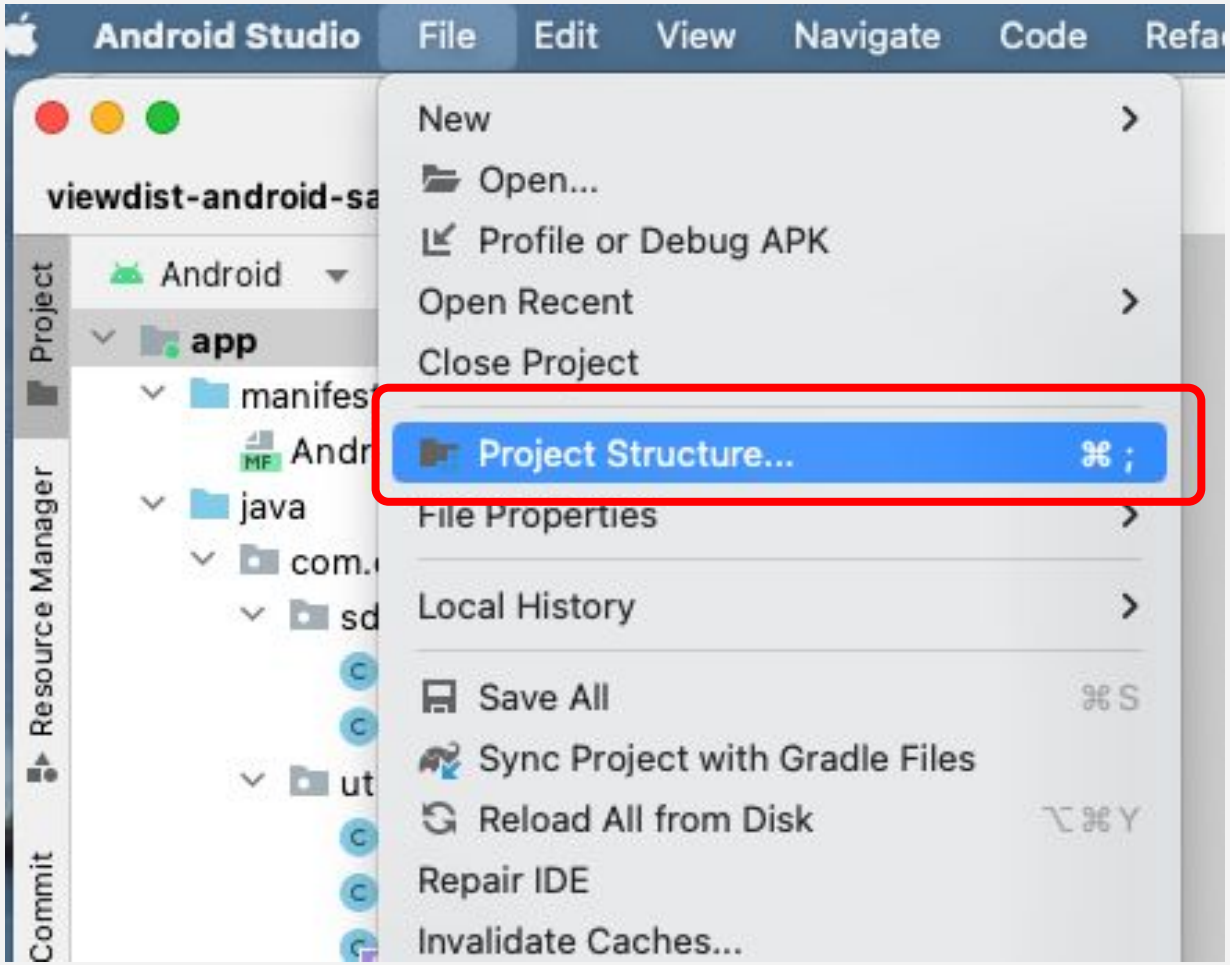
■OpenCV for Javaの導入⑥

最初の画面に戻り、Androidビューのツールウィンドウにopencvが表示されており、
OpenCV用のbuild.gradleを編集できるようになっています。



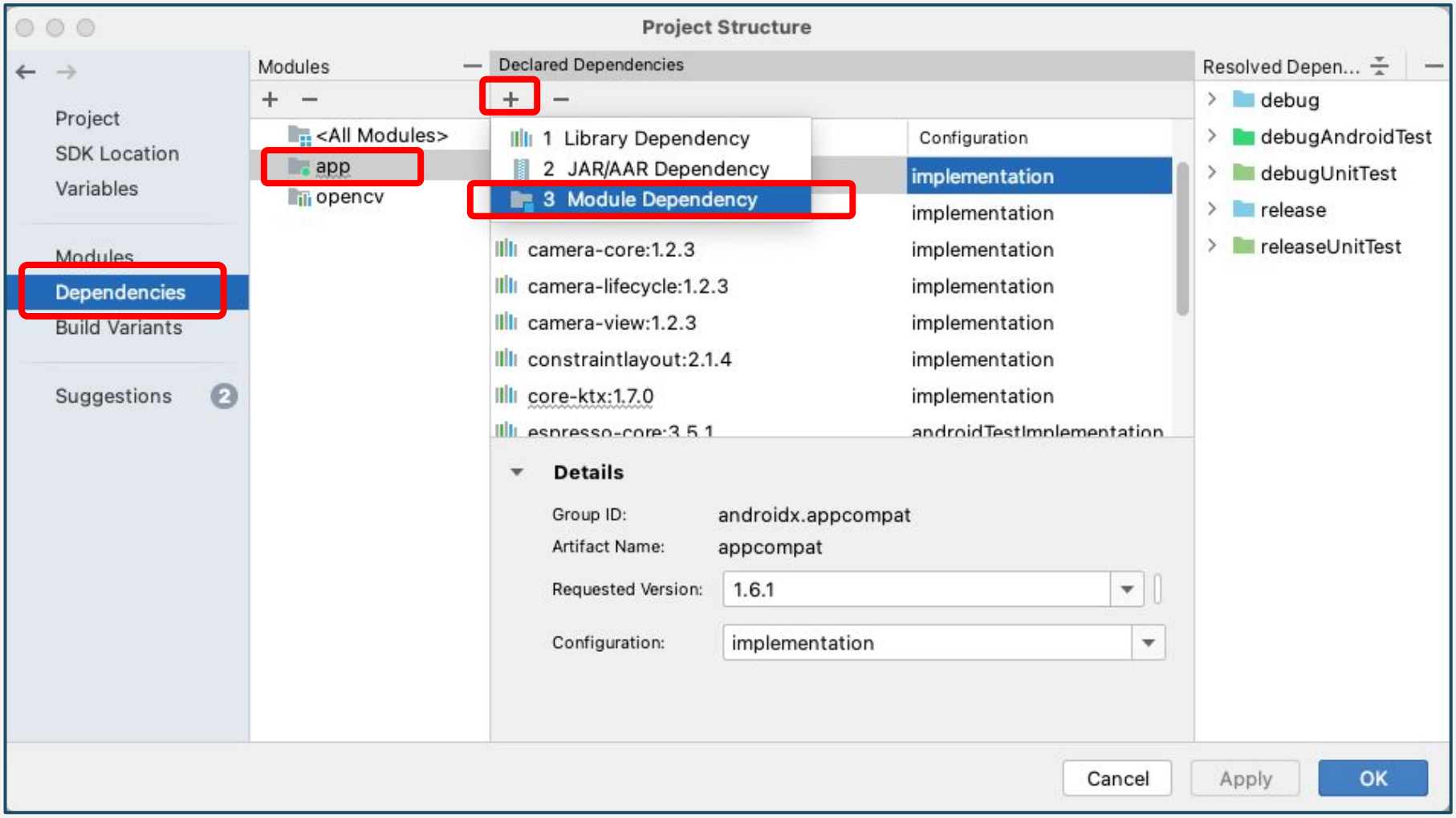
■OpenCV for Javaの導入⑦

続いて依存関係を設定するため、Android Studioのメニューから、
[File] → [Project Structure]をクリックします。



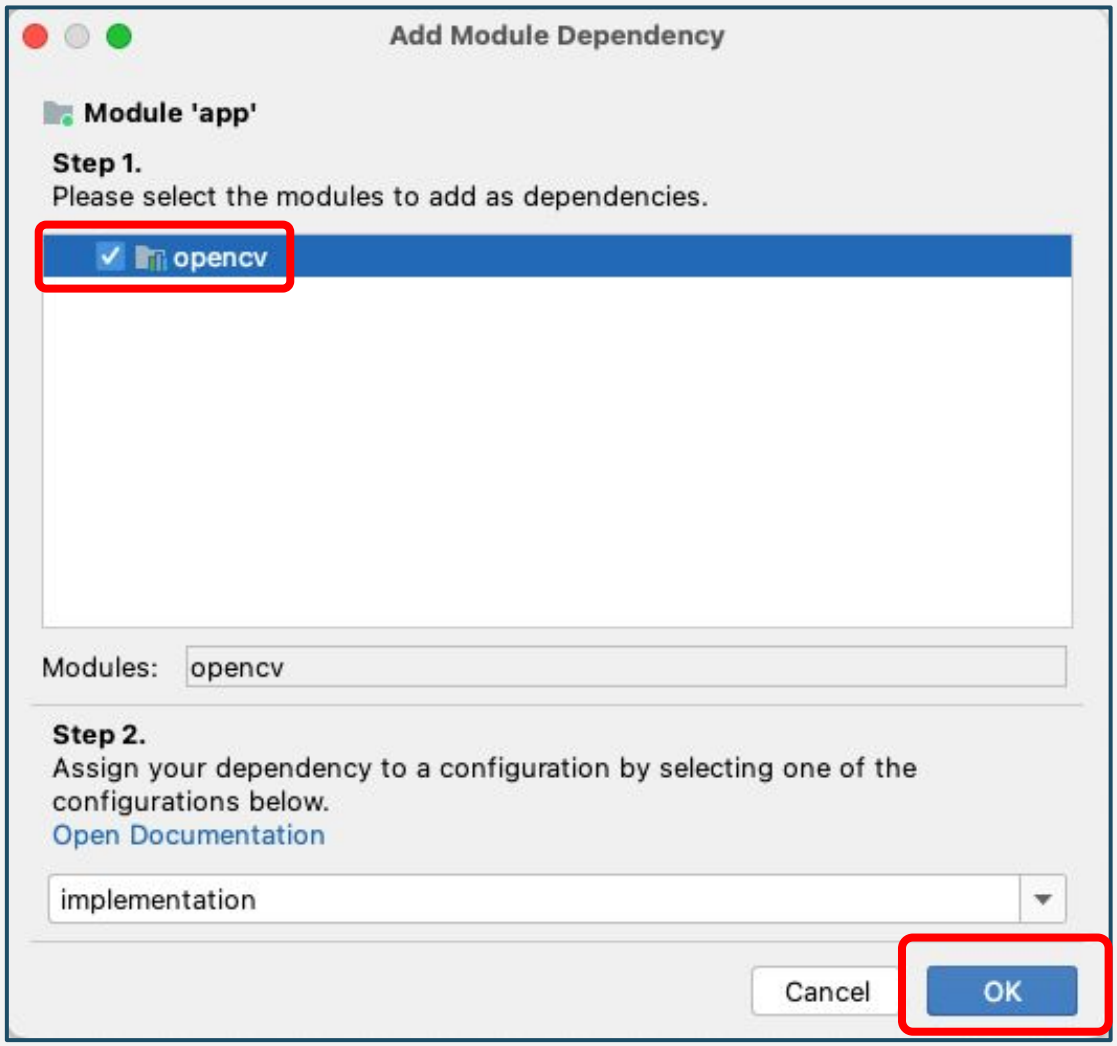
■OpenCV for Javaの導入⑧

表示されたウィンドウで、
[Dependencies] → [app] → [＋] → [3 Module Dependency]を選択します。



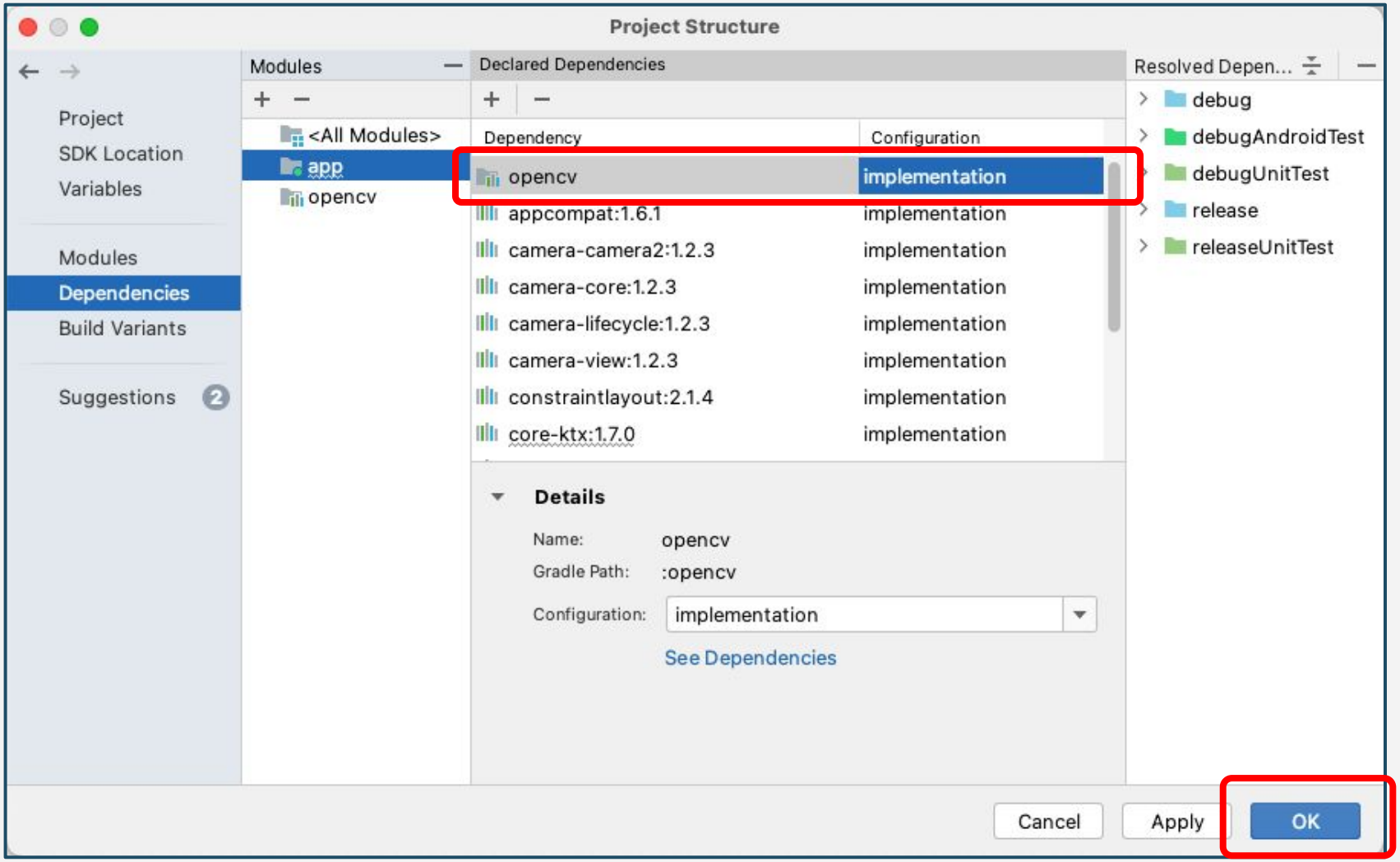
■OpenCV for Javaの導入⑨

Add Module Dependencyウィンドウが表示されるので、「**opencv**」にチェックを入れ、[OK]をクリックします。



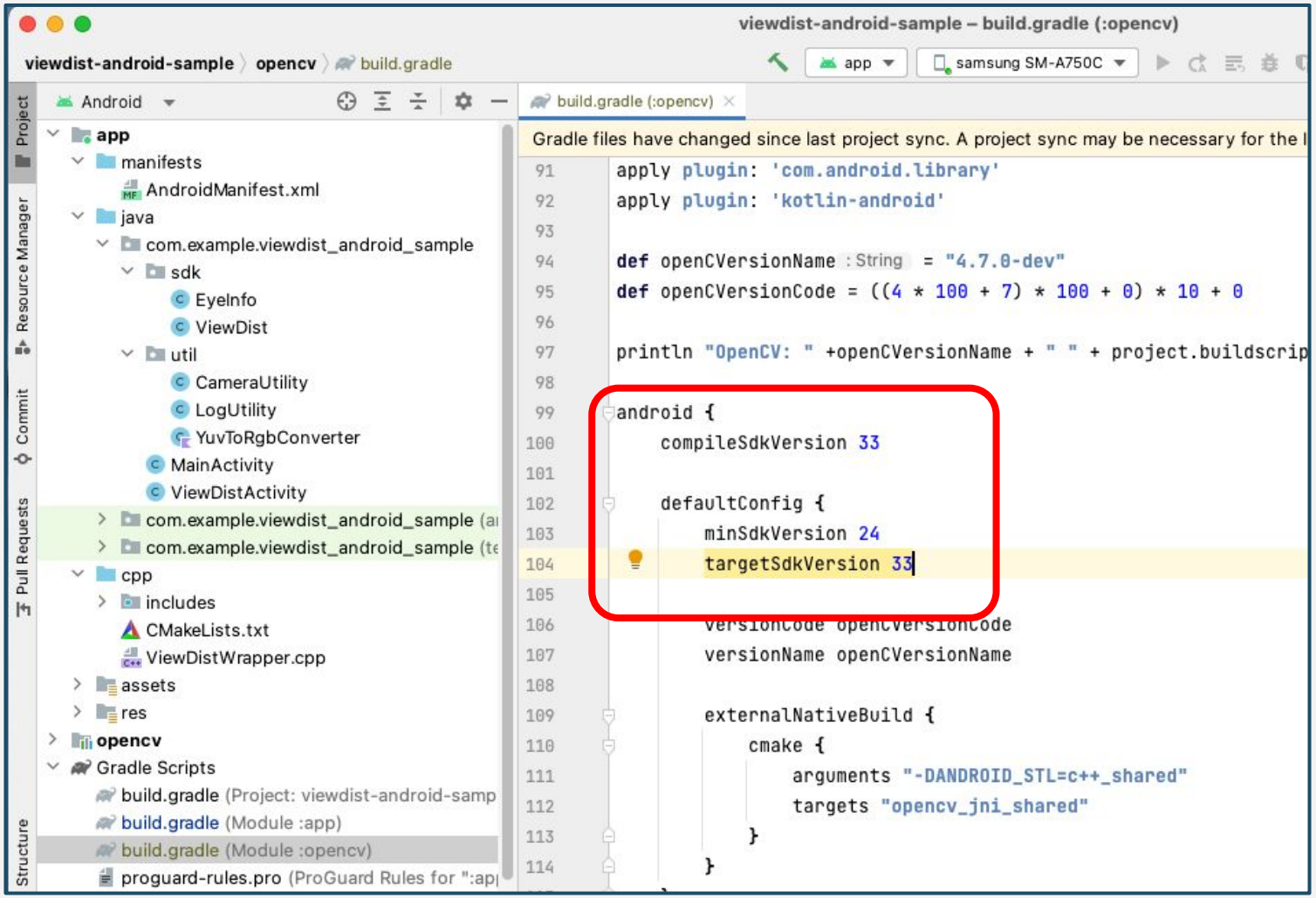
■OpenCV for Javaの導入⑩

「Dependencies」→「app」のところに「**opencv**」が表示されていれば
正しく設定できているので、[OK]をクリックします。



■OpenCV for Javaの導入⑪

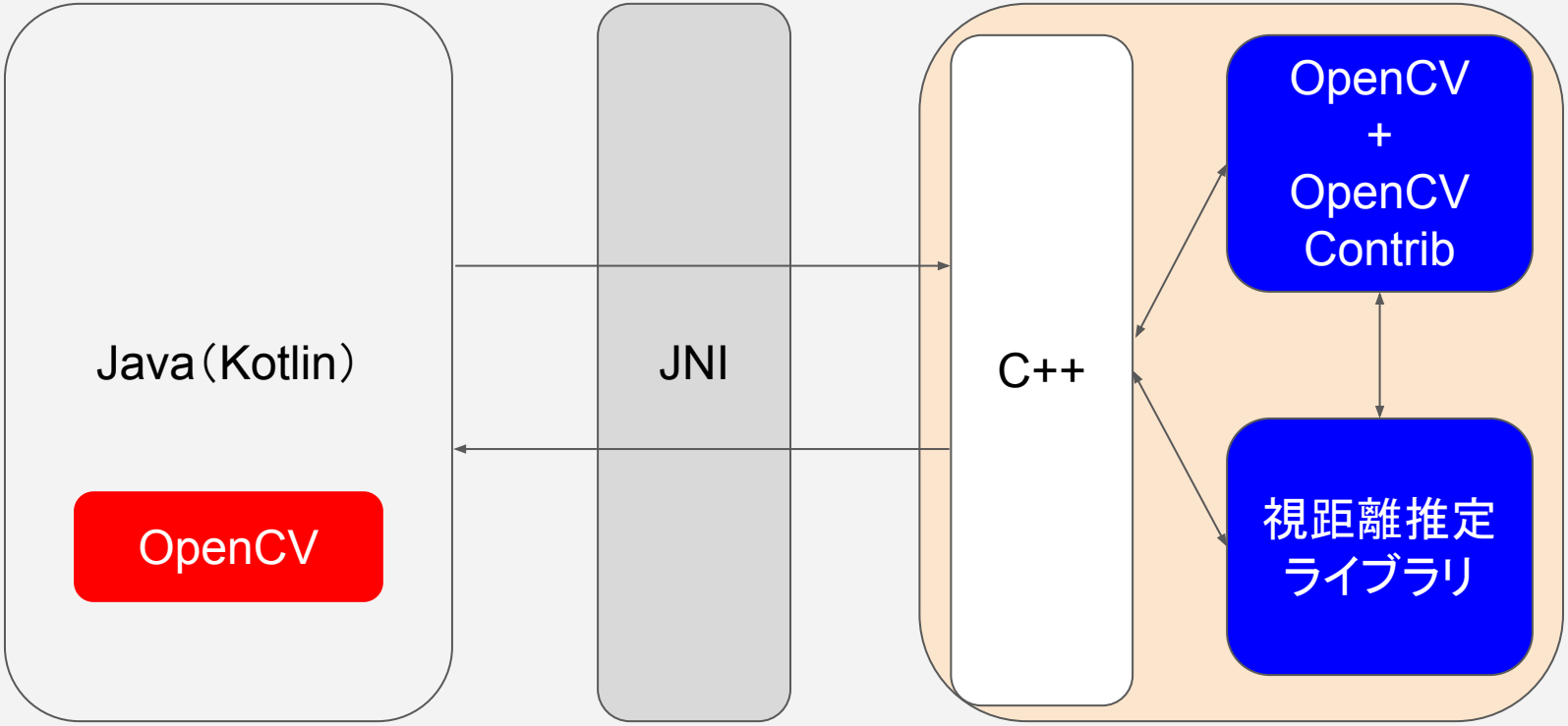
最後に、build.gradle(Module:opencv)の**compileSdkVersion**と**miniSdkVersion**、**targetSdkVersion**を、build.gradle(Module:app)のものと揃えて下さい。変更をした場合は、「Sync now」で同期しておきます。以上でOpenCV for Javaの導入は完了となります。



各種提供ライブラリの導入

■各種提供ライブラリの導入（補足）

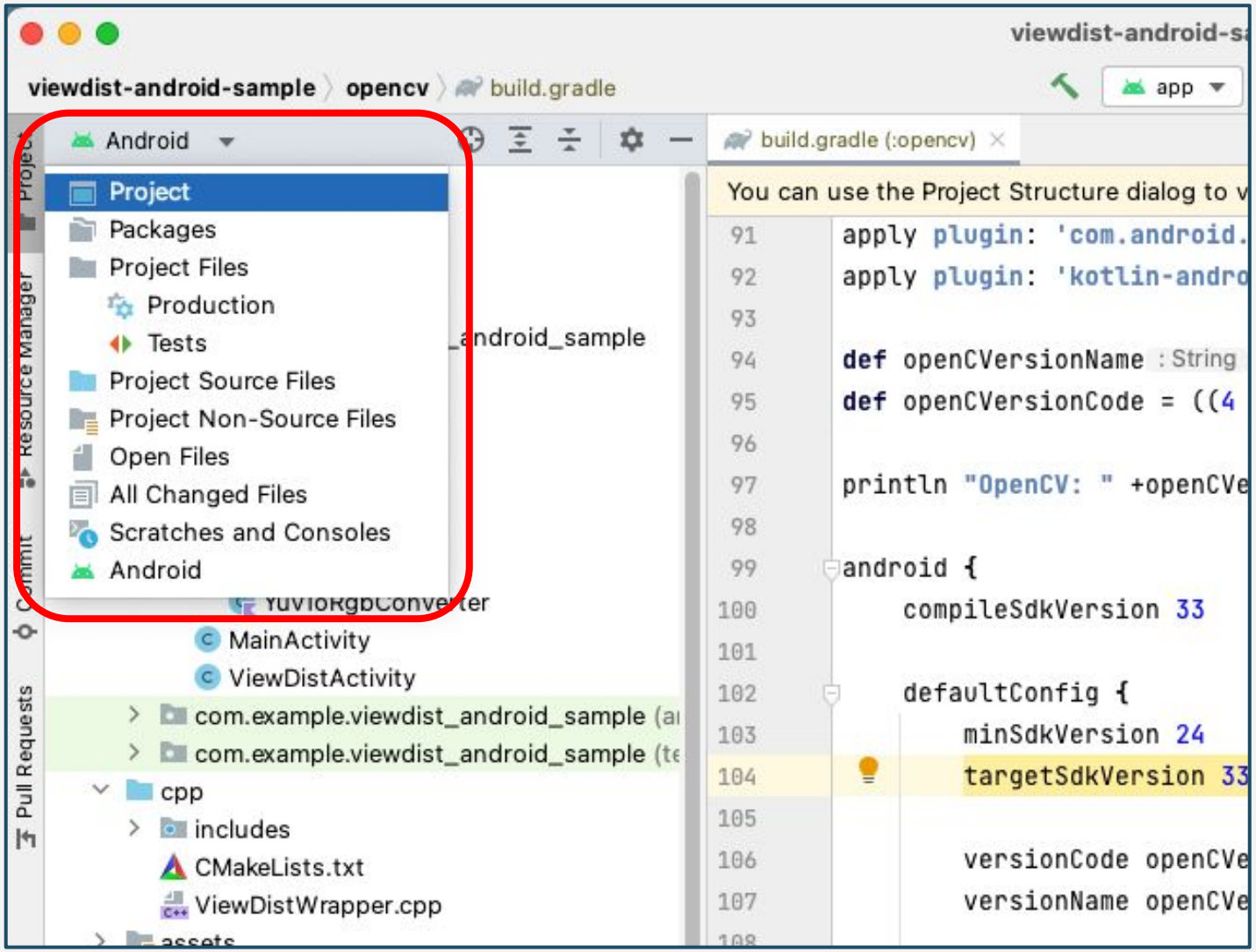
本サンプルアプリでは、視距離推定ライブラリと、OpenCV Contribを取り込んでビルドされたAndroid版OpenCVライブラリが、当社より提供されます。
サンプルアプリ自体はJavaで開発されていますが、当社提供ライブラリはC++言語で呼び出す必要があり、アプリにそのラッパーも含まれています。ここでは、下図青枠の部分の視距離推定ライブラリ及びOpenCVライブラリの導入方法について説明をしていきます。



※JNI・・・Java Native Interface

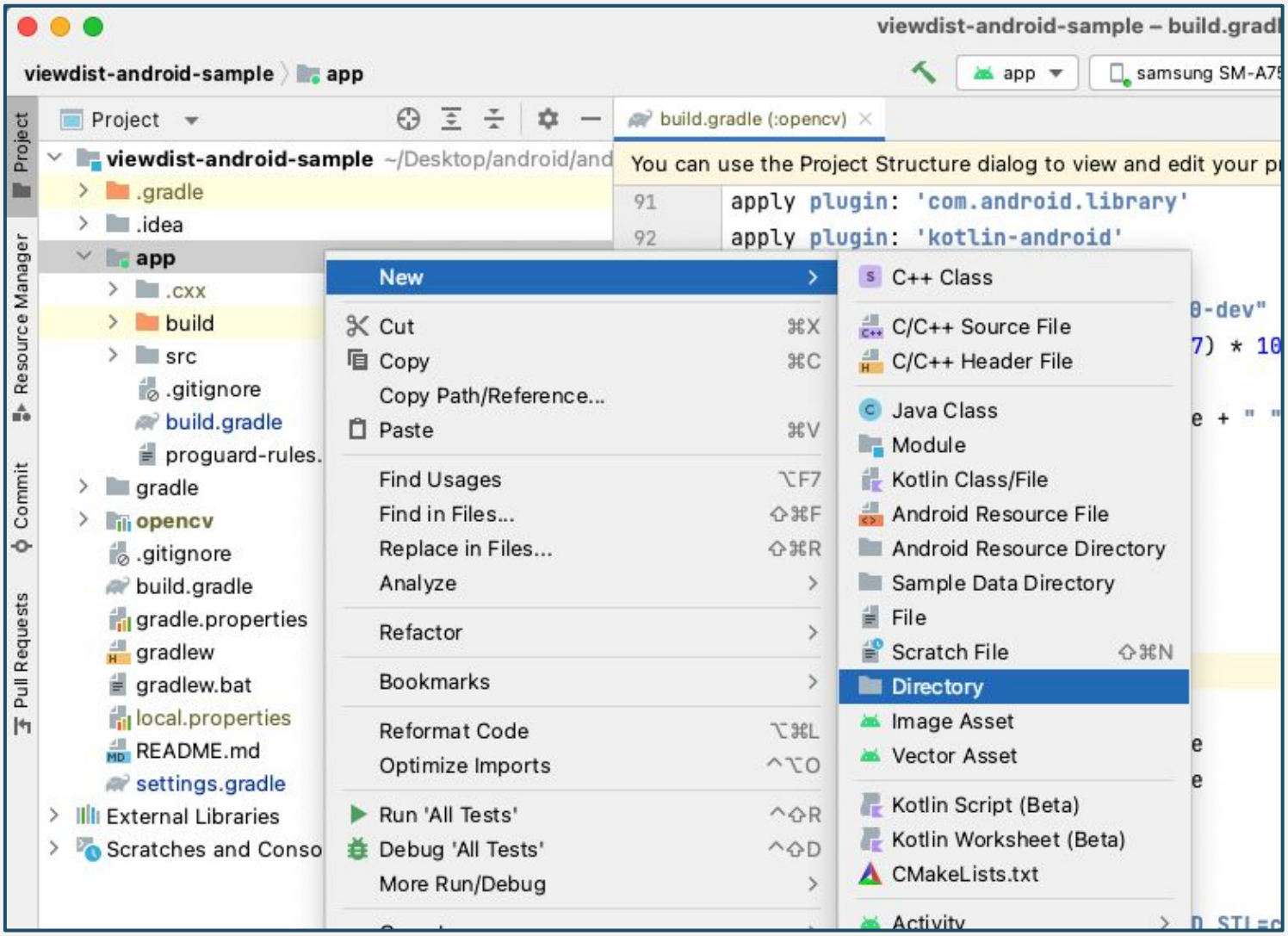
■各種提供ライブラリの導入①

ツールウィンドウから、Projectビューに切り替えを行います。



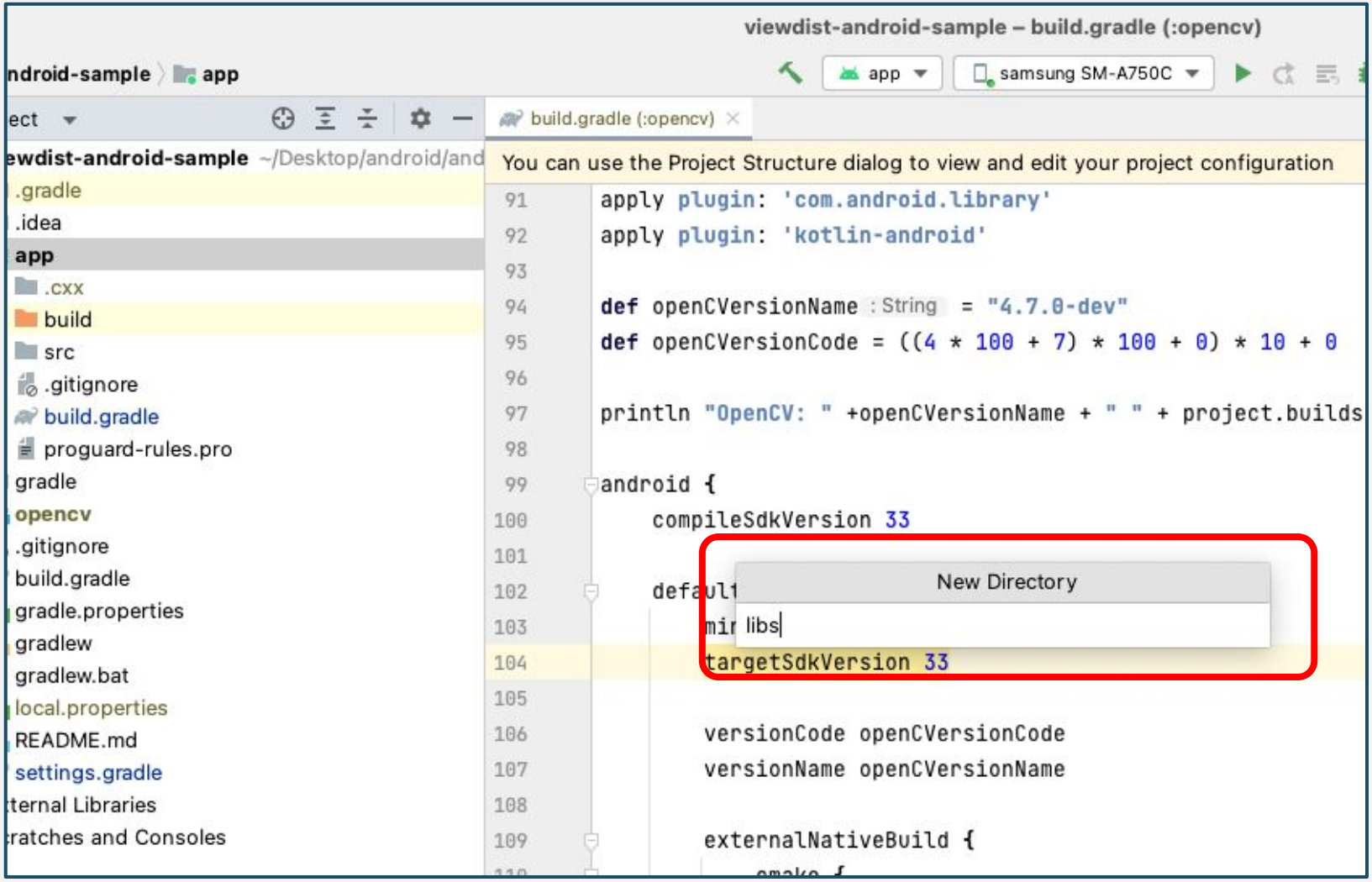
■各種提供ライブラリの導入②

各種ライブラリを設置するためのディレクトリを用意していきます。
[app]ディレクトリを選択し、右クリック→ [New] → [Directory] をクリックします。



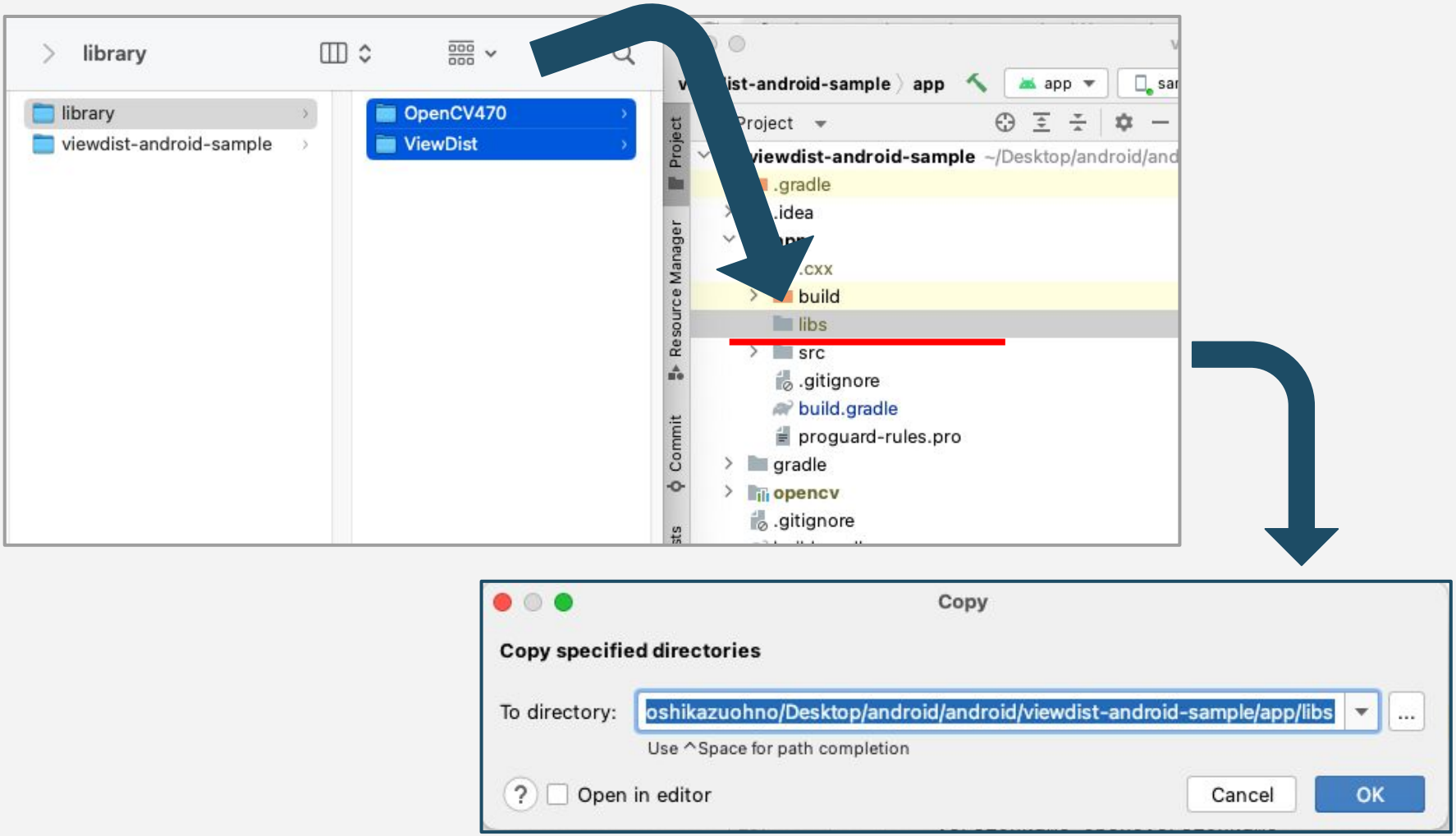
■各種提供ライブラリの導入③

New Directoryウィンドウが表示されたら、ディレクトリ名に「**libs**」を入力し、最後にEnterキーで実行します。



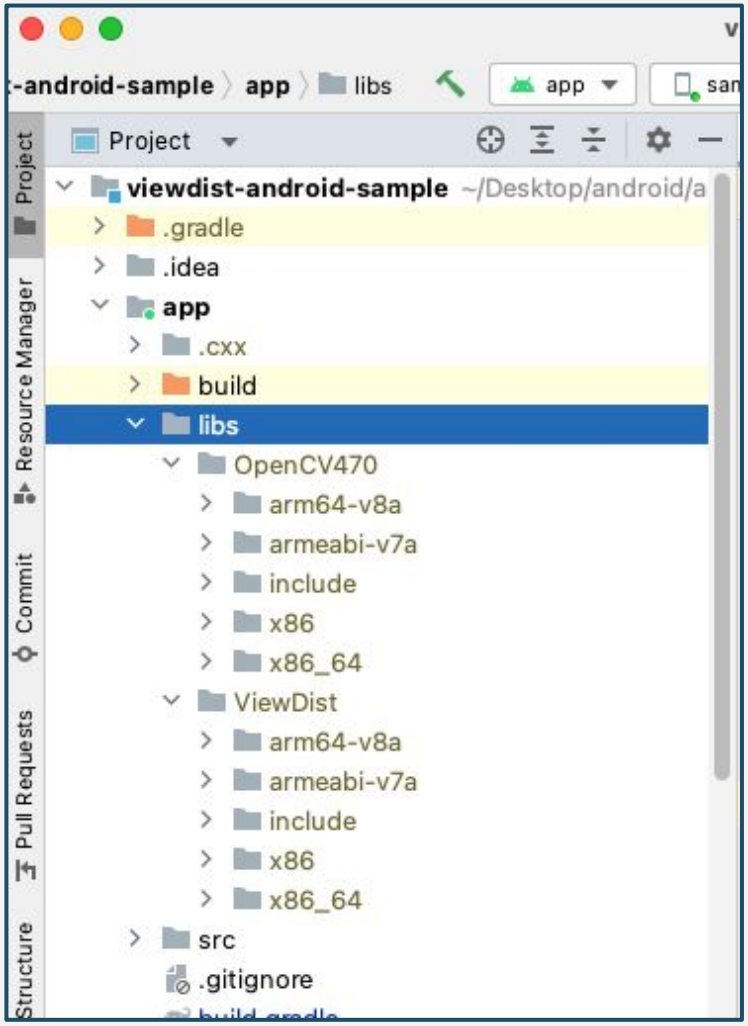
■各種提供ライブラリの導入④

当社提供ライブラリ（**OpenCV470、ViewDist**）を先ほど作成した
libsディレクトリにコピー&ペーストします。確認画面が出たら「OK」をクリックします。



■各種提供ライブラリの導入⑤

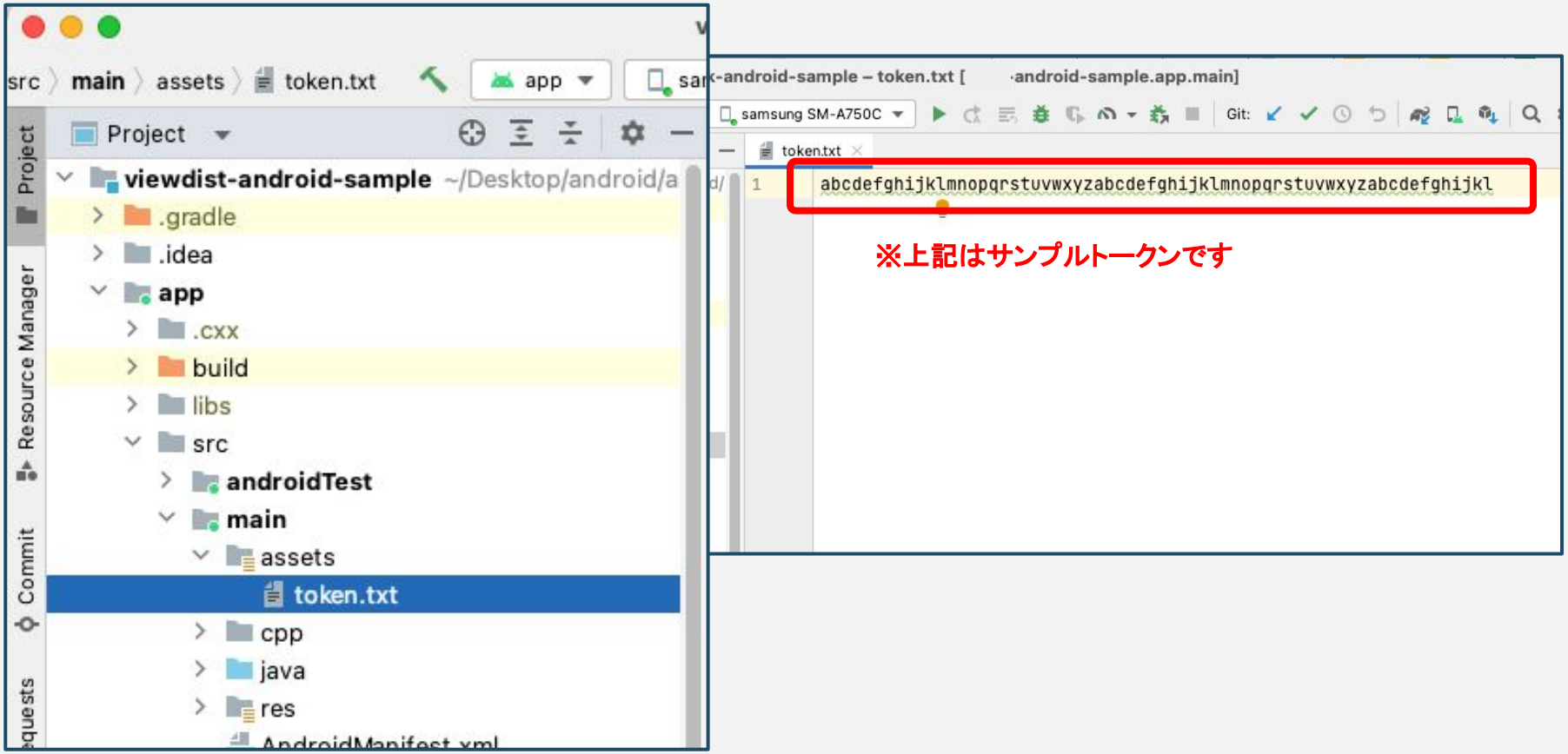
各種提供ライブラリのコピー後、libs配下が下記の様な構成になっていれば
各種提供ライブラリの導入は完了です。
※ ご提供する対応アーキテクチャによっては存在しないディレクトリがある場合があります。



アクティベーショントークンの設定

■アクティベーショントークンの設定

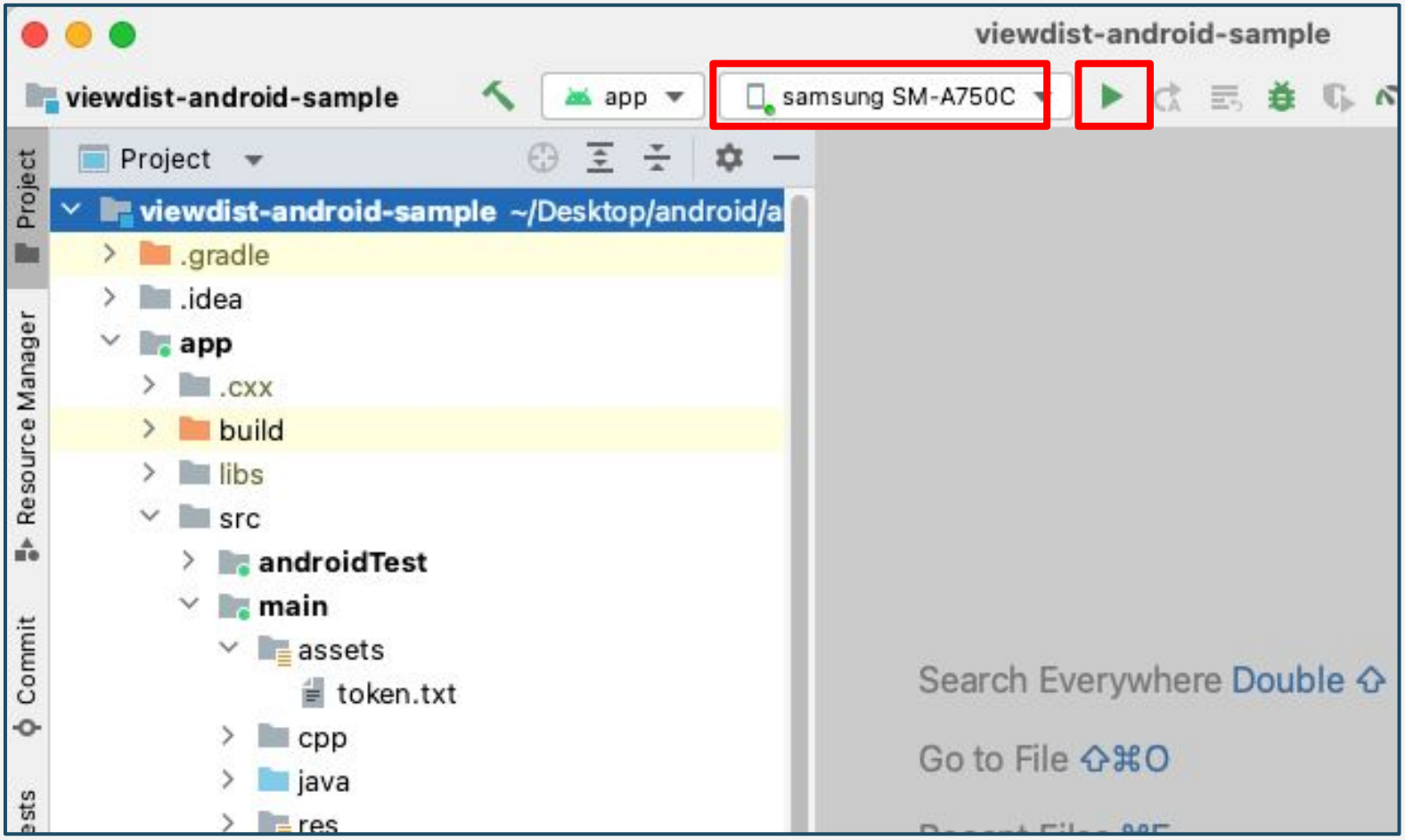
視距離推定ライブラリを実行する際は、当社よりご提供させていただく
アクティベーショントークンが必要となります。
アクティベーショントークンは下記のファイルに記述してください。
viewdist-android-sample / app / src / main / assets / token.txt



アプリのビルド

■アプリのビルド

ここまでの作業を完了したら、現在サンプルアプリを開いているWindowsOS/macOSマシンに、アプリを設置するAndroid端末を接続し、Android Studio上で認識されていることを確認して、選択しておきます。
これでアプリをビルドする準備が整いましたので「▶」マークを押してビルドを行います。



■アプリのビルド

アプリをビルドしたAndroid端末に、アイコン画像なしの「viewdist-android-sample」と表示されたアプリアイコンが表示されていればビルドは完了です。ビルドが完了すると、自動的に以下のようなアプリ画面が立ち上がります。動作を確認しながら、サンプルアプリのソースコードを参照して、開発を進めてください。