MediaPipe による12種類のAIタスク実行（Windows上）

https://www.kkaneko.jp/cc/aitasks/mediapipe.html

金子邦彦

構成

- 前半

スライド3～8

準備したサンプルプログラムの実行法や仕組みについて説明

https://www.kkaneko.jp/cc/aitasks/mediapipe.html

- 後半

スライド9～10

応用や別タスクへの拡張時に参照する知識

基本的な流れ

プログラム初回実行時

- モデル・画像の自動ダウンロード
- インターネット接続必須
- 数十秒～数分の待機

実行時の操作

- tkinterダイアログ等での入力や選択
- AI推論実行
- OpenCVウィンドウやターミナルでの結果確認・改善への手がかり取得

AI推論プログラムの共通実行パターン

サンプルプログラムは12個。実行パターンを統一している

5段階の流れ：

- BaseOptions：**モデルパス指定**
- XxxOptions：**タスク固有パラメータ設定**
- create_from_options：**インスタンス生成**
- **推論メソッド**：detect／classify／segment／recognize／embed
- close：**リソース解放**

入力データの型変換

つまずきやすい箇所のため、混同に注意。

画像データ

- 推論用：mp.Image.create_from_file
- 表示用：cv2.imread で別途読み込み

色空間の差異

- MediaPipe内部：RGB
- OpenCV表示：BGR
- cv2.cvtColor で変換

座標表現

- ランドマークは0.0～1.0の正規化座標
- 画素位置は x×width、y×height で算出

結果オブジェクトの読み取り

bounding_box：検出枠

- origin_x、origin_y、width、height

categories：分類結果

- category_name と score の組 •

landmarks：特徴点座標

- x、y、z の正規化値

精度を変える主要パラメータ

実験対象・調整対象となりえるパラメータ

閾値系

score_threshold、min_detection_confidence

- 値を下げると検出と誤検出がともに増加
- 値を上げると両方とも減少

出力数系

max_results、num_hands、num_faces

クラス絞り込み系

category_allowlist、category_denylist

実験結果から改善策を導くヒント

誤検出が多い場合

- score_threshold を上げる、allowlist で対象限定

見逃しが多い場合

- score_threshold を下げる、max_results を増やす

小物体・遠距離で精度低下

- 入力リサイズやトリミング、上位モデルへの切替

クラス外対象が検出されない

- 別モデルへの変更またはカスタム学習

到達目標：ここまでで実行・解釈・改善実験ができる。

タスクとベースモデルの対応

モデル選択や別タスクへの応用時に参照。

- 物体検出：EfficientDet-Lite0（COCO 80クラス）
- 画像分類：EfficientNet-Lite0（ImageNet 1000クラス）
- 画像セグメンテーション：DeepLab V3（Pascal VOC 21クラス）
- 姿勢推定：BlazePose（lite／full／heavy の3段階）
- 手・ジェスチャー：hand_landmarker（21点）
- 顔ランドマーク：face_landmarker（478点と52種ブレンドシェイプ）
- 顔検出：BlazeFace（short_range は約2m以内、遠距離は full_range）
- 画像埋め込み：MobileNet V3 Small
- 音声分類：YAMNet（AudioSet 521クラス）
- テキスト分類：BERT（既定は英語の感情二値分類）
- 言語検出：language_detector（110以上の言語）

モデルファイル形式：単一は.tflite、複数サブモデル内包は.task

到達目標：必要な場合は、モデル選択による精度／速度調整ができる。

音声・テキストについて

音声系タスクのサンプルプログラム

- WAVファイルを int16 で読み出し
- float32 へ正規化（÷32768.0）して AudioData 化
- 16kHzモノラル必須

テキスト系タスクのサンプルプログラム

- 文字列をそのまま渡す
- 出力制御パラメータ（必要な出力のみ有効化して処理負荷を調整）
- output_category_mask、output_face_blendshapes、l2_normalize

ライブラリ補足

- numpy：音声波形と画像配列の操作
- pathlib.Path、urllib.request.urlretrieve：モデルと画像のローカル取得