**人脸几何特征提取与分析技术文档**

**1. 项目概述**

本项目旨在开发一套鲁棒的人脸几何特征提取与分析系统。该系统能够从给定的图像中准确识别并勾勒出人脸轮廓，定位关键面部标志点，并在此基础上计算和可视化人脸中线与面部切线的夹角。这些技术在**生物识别、医疗美容辅助分析、表情识别**等领域具有广泛的应用潜力。

**2. 核心技术方法**

本项目主要依赖于以下计算机视觉库和算法：

* **OpenCV (Open Source Computer Vision Library)**: 用于图像处理、颜色空间转换、轮廓检测与绘制等基础操作。
* **Dlib**: 一个强大的C++库，其中包含了高效的机器学习算法。本项目主要利用其预训练的**人脸检测器 (HOG + SVM)** 和**人脸关键点检测器 (68点模型)** 来准确识别和定位面部特征。
* **NumPy**: 用于高效的数值计算，特别是在处理图像像素数据、向量运算和几何计算方面。
* **SciPy**: 提供了科学计算工具，本项目中主要使用其 scipy.stats.linregress 函数进行**线性回归**，以拟合出人脸的精确中线。

**3. 技术流程与实现细节**

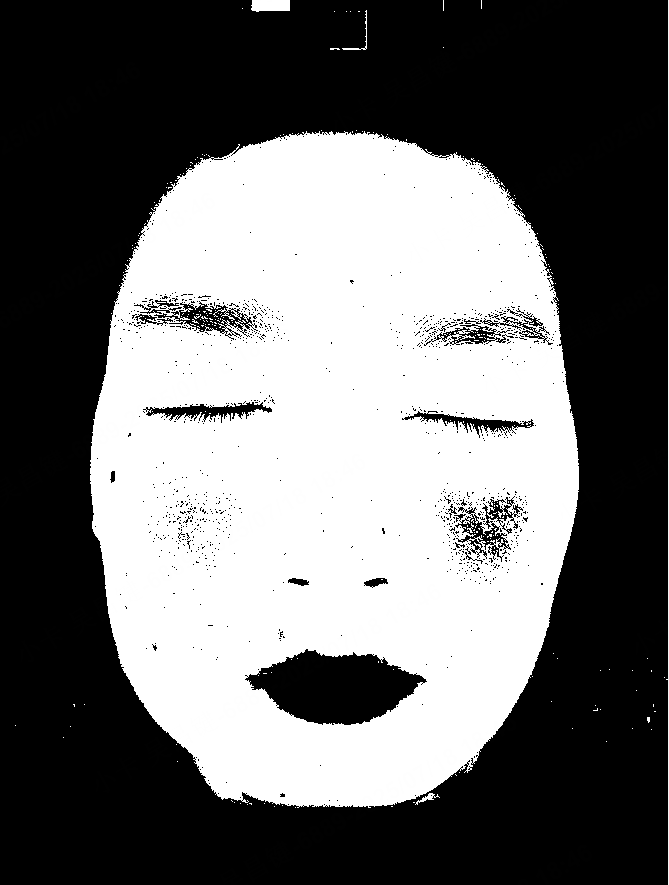
整个系统的处理流程可分为以下几个主要阶段：

**3.1 人脸检测与关键点定位**

1. **人脸检测**: 利用 Dlib 的人脸检测器在输入图像中定位人脸区域。这一步是后续所有分析的基础。
2. **关键点检测**: 对检测到的人脸区域，Dlib 的 68 点人脸关键点模型被用于精确定位眉毛、眼睛、鼻子、嘴巴和下巴等部位的标志点。这些点以 (x, y) 坐标的形式存储，是后续几何计算的基石。

**3.2 肤色分割 (非皮肤区域排除)**

为了确保后续轮廓提取的准确性，本项目采用了两种互补的颜色空间进行肤色分割：

1. **HSV (Hue, Saturation, Value)** 颜色空间:
   * 将 RGB 图像转换到 HSV 空间。
   * 通过定义肤色的 **H、S、V 阈值范围** (lower\_skin, upper\_skin) 来创建二值掩膜。
   * **优化策略**: 针对初期白色衣物被误识别的问题，通过**提高 S\_min (最小饱和度)** 和**降低 V\_max (最大亮度)**，有效排除了低饱和度、高亮度的非肤色区域（如白色衣物）。
   * 
2. **YCbCr 颜色空间**:
   * 将 RGB 图像转换到 YCbCr 空间。
   * YCbCr 空间在肤色聚类方面表现良好，通过设定 **Cb 和 Cr 的阈值范围** (lower\_skin\_ycbcr, upper\_skin\_ycbcr) 创建另一份二值掩膜。
   * **优化策略**: 同样针对白色衣物问题，通过**提高 Cb\_min** 和**降低 Cr\_max**，使得肤色检测更为精确。



1. **掩膜融合**: 两种颜色空间生成的二值掩膜通过**逻辑与 (cv2.bitwise\_and)** 操作进行融合，生成一个更可靠的 final\_skin\_mask。这结合了两种方法的优势，最大限度地保留肤色区域并抑制非肤色背景。



**3.3 人脸轮廓提取与平滑**

1. **最大连通域**: 在 final\_skin\_mask 上，利用 cv2.findContours 找到所有轮廓，并识别出其中**面积最大的轮廓**，这通常代表了完整的人脸区域。
2. **轮廓近似平滑**: 使用 cv2.approxPolyDP 对提取到的原始人脸轮廓进行多边形近似。
   * **参数 epsilon**: 这个参数决定了近似的精度。epsilon 值越小，近似后的轮廓点越多，曲线越平滑，越接近原始轮廓；反之则点越少，轮廓越简化。
   * **优化策略**: 通过**减小 epsilon 的乘数（例如从 0.005 减小到 0.002 或更小）**，显著增加了轮廓点的密度，使得最终绘制的脸型轮廓更加平滑和精细。



**3.4 精确人脸中线拟合**

为了克服人脸偏转导致中线不准确的问题，本项目采用了**线性回归**来拟合更精确的人脸中线：

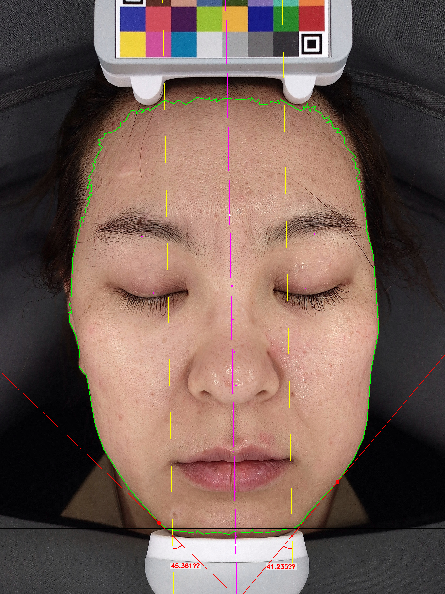
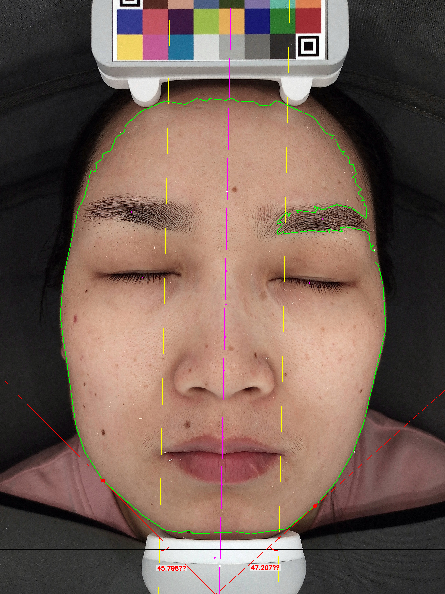
1. **关键点选择**: 收集 Dlib 关键点中被认为位于人脸中轴线上的点，包括：
   * 左右眉毛的**中心点**。
   * 左右眼睛**内眼角和外眼角连线的中心点**。
   * **鼻梁上的所有点** (pts[27] 到 pts[30])，这些是中轴上最重要的点。
   * **上唇珠** (pts[51]) 和**下唇中点** (pts[57])。
   * **下巴最底部中心点** (pts[8])。
2. **线性回归 (X,Y 坐标反转)**:
   * 将上述关键点的 (x, y) 坐标作为数据点。
   * 为了更好地拟合垂直或接近垂直的直线（避免斜率趋于无穷大），本项目创新性地**反转了 X 和 Y 坐标**，使用 scipy.stats.linregress 拟合 x = m'y + c' 形式的直线。这种方法在处理人脸偏转时，能更稳定地找到最佳拟合的二维中轴线。
3. **中线端点确定**: 根据拟合出的直线方程，结合预设的 Y 坐标范围（从眉毛上方到下巴下方），计算出中线的两个端点。这些端点被裁剪到图像边界内，确保中线可见且合理。
4. **avg\_midline\_x 保留**: 为了兼容后续需要一个简单垂直中心线判断（如左右脸颊划分）的逻辑，项目在执行线性回归之前，**保留了基于所有拟合点 X 坐标平均值的 avg\_midline\_x 计算**。这使得两种中线定义各司其职，互不干扰。



**3.5 切线计算与绘制**

1. **交点识别**: 在人脸中线和左右嘴角平行线的交汇处确定交点 P。
2. **候选切点筛选**:
   * 从之前提取的完整脸颊轮廓点 (jaw\_pts，即 face\_contour\_pts) 中筛选出适合作为切点的**候选点 Q**。
   * **优化策略**: 引入更严格的垂直和水平范围限制。候选点 Q 必须：
     + 位于交点 P 的下方。
     + 位于由 jawline\_bottom\_y (下巴最低点向上移动一定像素) 和 cheek\_upper\_bound\_y (脸颊上部点向上移动一定像素) 定义的**精确脸颊垂直区域内**。这有效避免了切线落在下巴或过高的颧骨区域。
     + 根据交点 P 的左右位置，位于脸颊的相应一侧。
3. **支持线判断**: 实现 is\_supporting\_line(P, Q, contour) 函数，判断连接交点 P 和候选切点 Q 的直线是否是轮廓 contour 的支持线。支持线意味着轮廓上的所有其他点都位于该直线的同一侧。
4. **切点选择**: 在所有满足支持线条件的候选点中，选择**距离交点 P 最近**的点作为最终的切点 Q\_sel。
5. **切线绘制**: 从交点 P 沿着 P 到 Q\_sel 的方向延长绘制切线，并在切点 Q\_sel 处绘制标记。

**3.6 角度计算与可视化**

1. **平行方向向量**: 使用新拟合的人脸中线 (pt\_midline\_top 和 pt\_midline\_bottom) 来计算其**方向向量** (para\_vec)。
2. **切线方向向量**: 基于交点 P 和选定的切点 Q\_sel 计算**切线的方向向量** (tan\_vec)。
3. **夹角计算**: 利用两个方向向量的**点积**，计算它们之间的锐角。
4. **角度可视化**:
   * 通过 draw\_angle 函数，在交点 P 处绘制一个表示夹角的弧线。
   * 在弧线附近清晰地标注计算出的角度数值，并带有半透明背景，确保可读性。

**4. 优势与鲁棒性**

* **多源融合**：结合 HSV 和 YCbCr 两种颜色空间的肤色检测，增强了复杂光照和背景下的鲁棒性。
* **精确轮廓**: cv2.approxPolyDP 的精细调整使得人脸轮廓更加平滑和逼真。
* **自适应中线**: 线性回归（尤其通过 X,Y 坐标反转）能够适应人脸偏转，提供更准确的面部中轴线，避免了传统垂直中线的局限性。
* **智能切线选择**: 结合 Dlib 关键点和几何约束，精确筛选脸颊区域的切点，避免了切线误落在下巴或非脸颊区域。
* **模块化设计**: 代码结构清晰，各功能模块独立，便于调试、维护和未来扩展。

**5. 潜在应用**

* **医疗美容**: 精准测量面部角度，辅助整形方案设计和效果评估。
* **生物识别**: 提取独特的几何特征，用于身份验证或人脸识别系统。
* **姿态估计**: 通过中线和关键点的相对位置，推断人脸的头部姿态。
* **表情分析**: 结合面部几何变化，辅助识别和量化表情特征。

