

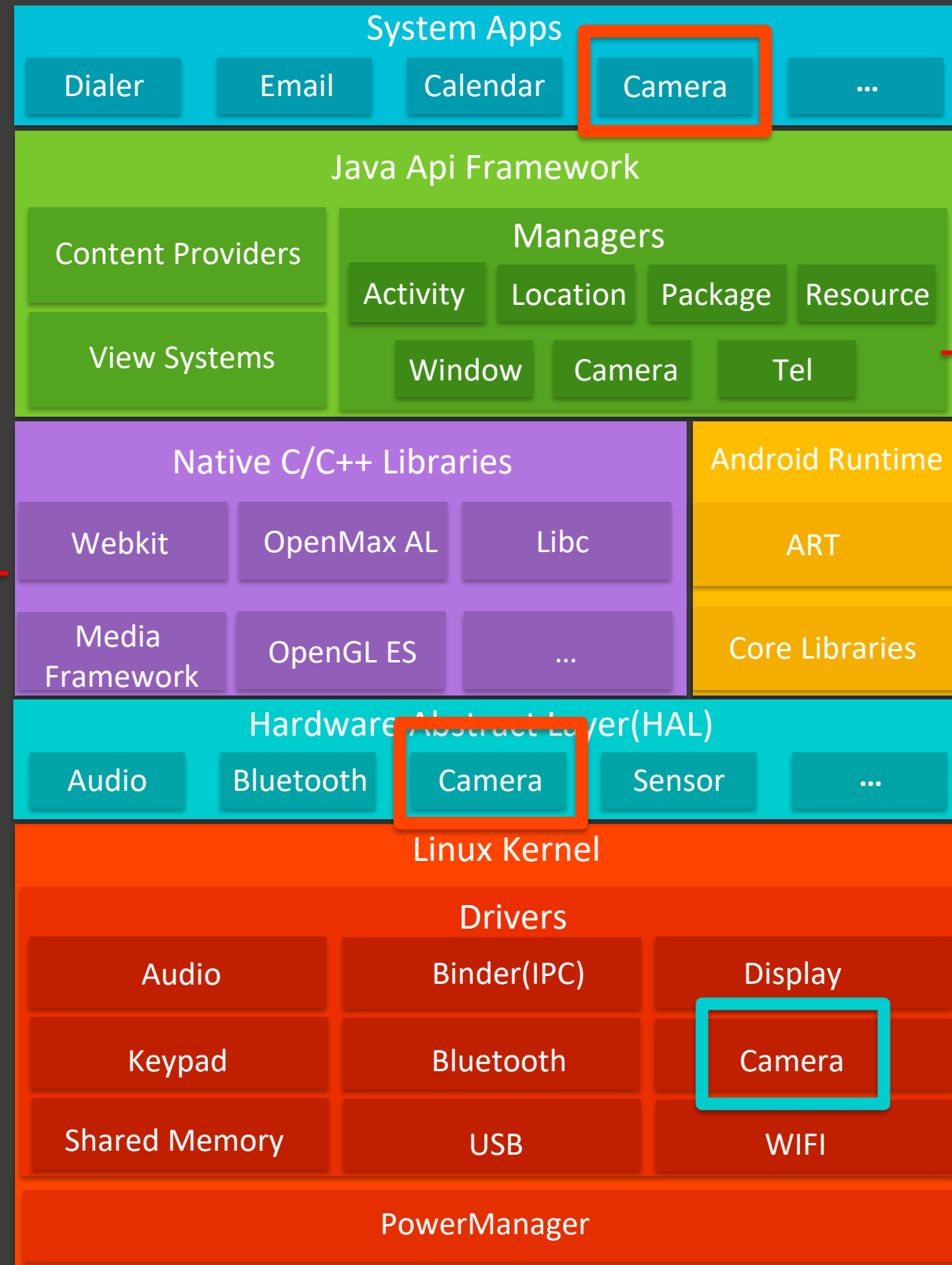
主题：人工智能在移动相机应用中的实战

团队：花椒相机
讲师：余雪亭



- 自定义相机
- 图形图像基础知识
- 相机与人工智能的结合
- 遇到的问题 and 解决方案

系统架构



Camera Client
Camera Service

Camera.java

android_hardware_Camera.cpp

1.调用系统的相机：

调用方式：MediaStore.ACTION_IMAGE_CAPTURE+startActivityForResult

处理方式：在onActivityResult中处理相机返回的图片数据。

2.自定义相机

(1)open

(2)setParameters

(3)setPreviewDisplay(SurfaceHolder)

(4)startPreview

(5)takePicture

需要的步骤

- 离线预览相机数据
- 快速处理相机数据

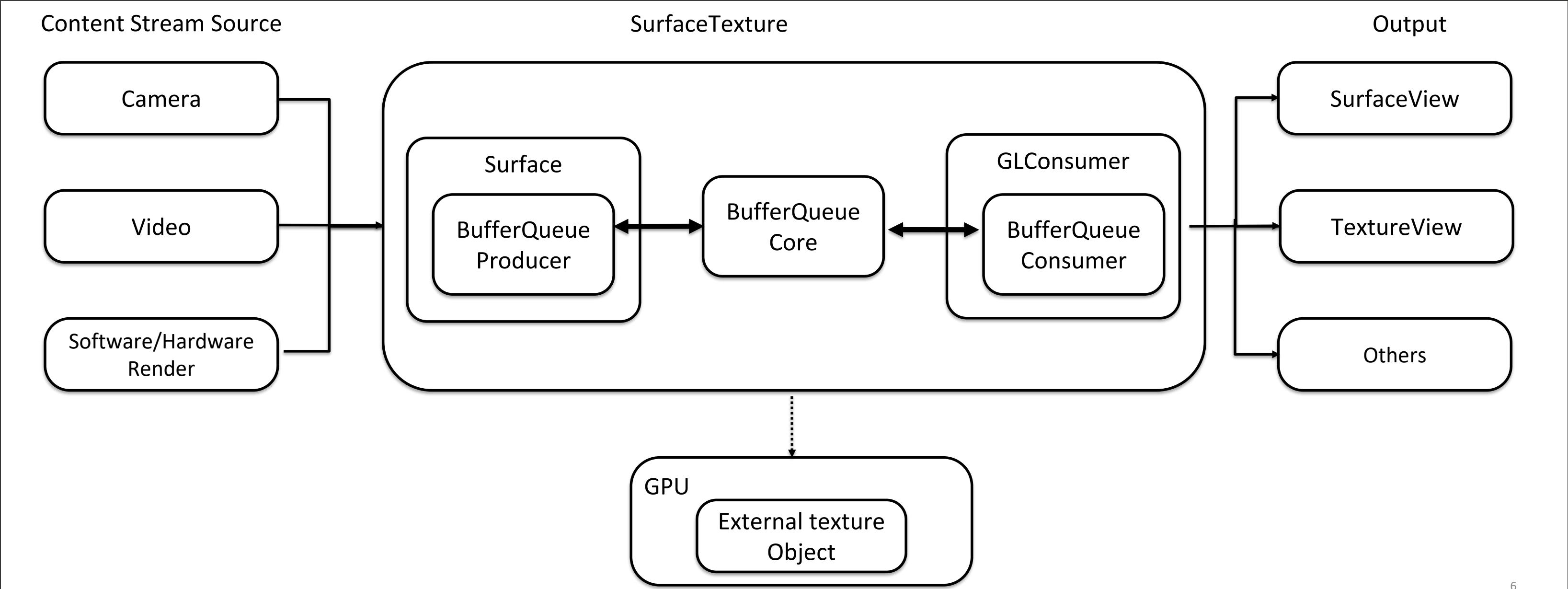
需要的基础知识

- SurfaceTexture
- OpenGL

SurfaceTexture

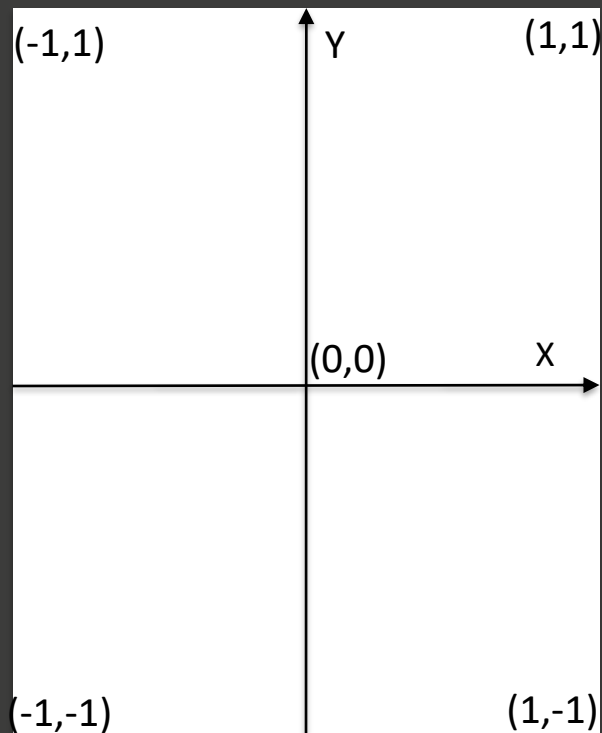


SurfaceTexture从Android 3.0加入。和SurfaceView不同的是，它对图像流的处理并不直接显示，而是转为GL外部纹理，因此可用于图像流数据的二次处理。

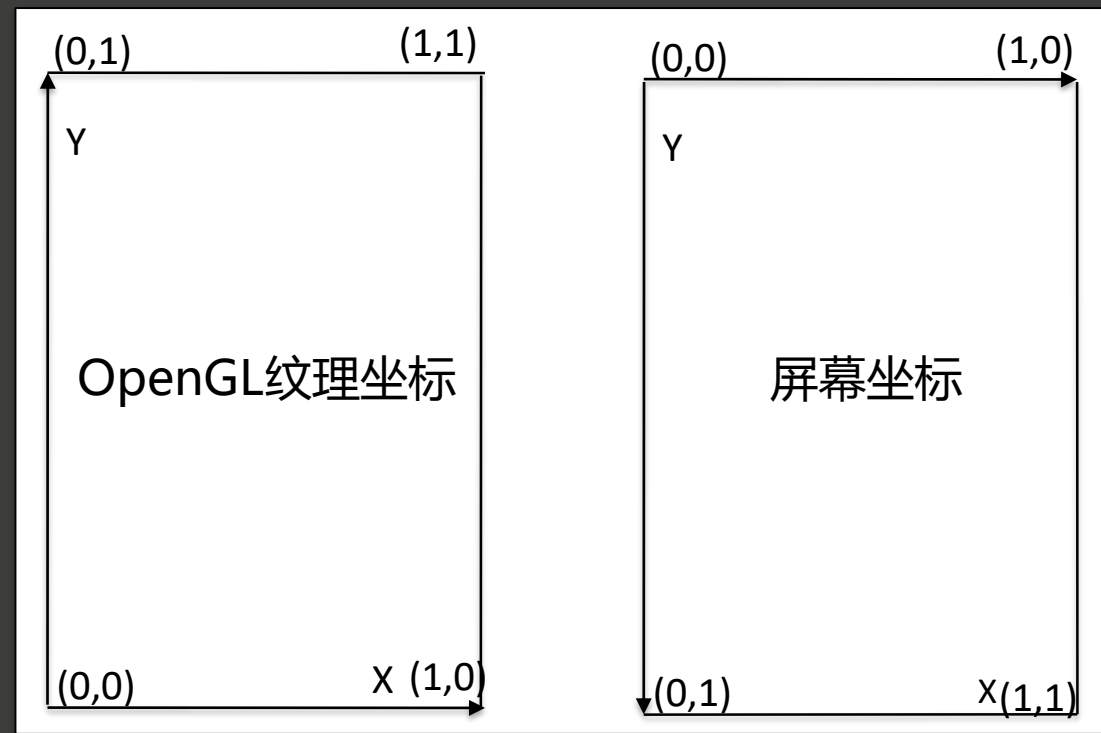


OpenGL (Open Graphics Library) 是指定义了一个跨编程语言、跨平台的编程接口规格的专业图形程序接口。它用于三维图像 (二维的亦可) , 是一个功能强大, 调用方便的底层图形库。

OpenGL应用与移动端使用的是一个特殊的嵌入式版本: OpenGL ES (OpenGL for Embedded System)



世界坐标



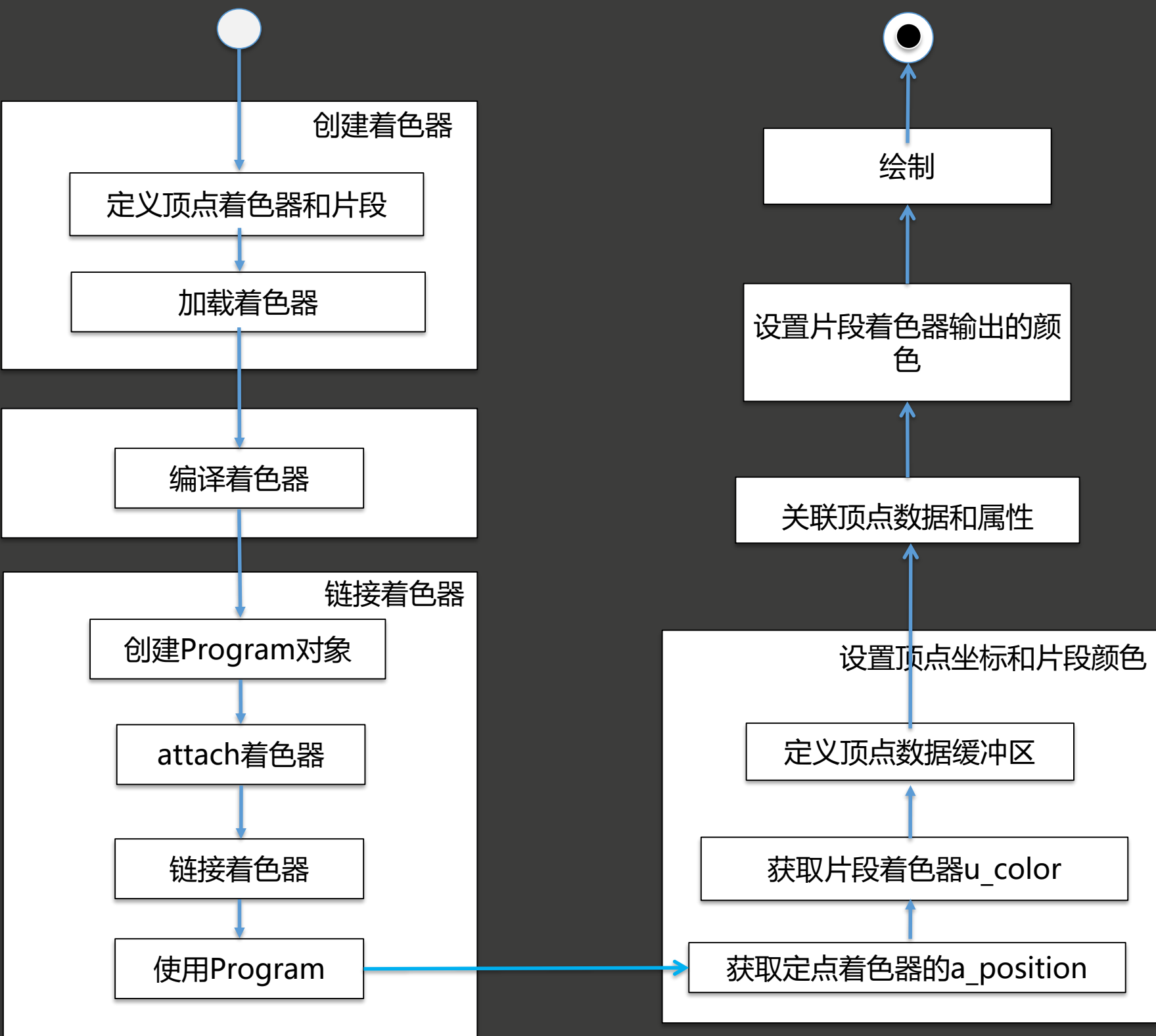
屏幕&纹理坐标

```
attribute vec4 a_Position;
void main() {
    gl_Position = a_Position;
}

precision mediump float;
uniform vec4 u_Color;
void main() {
    gl_FragColor = u_Color;
}
```

Shader

绘制流程

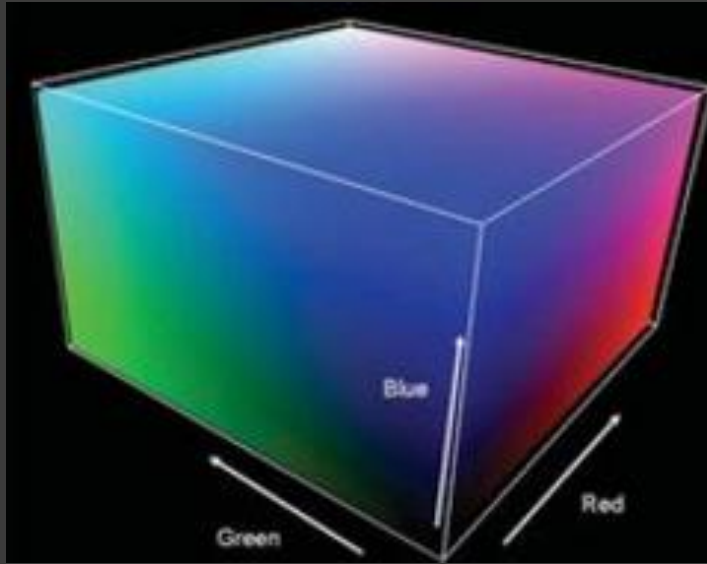


目标：检测，匹配，分割，变形，增强，平滑

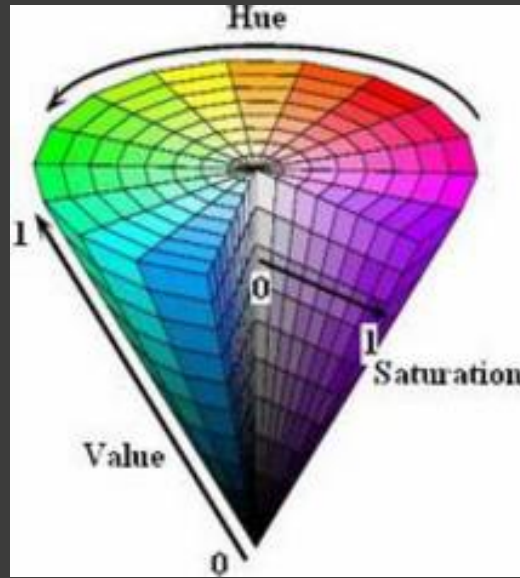


深度学习

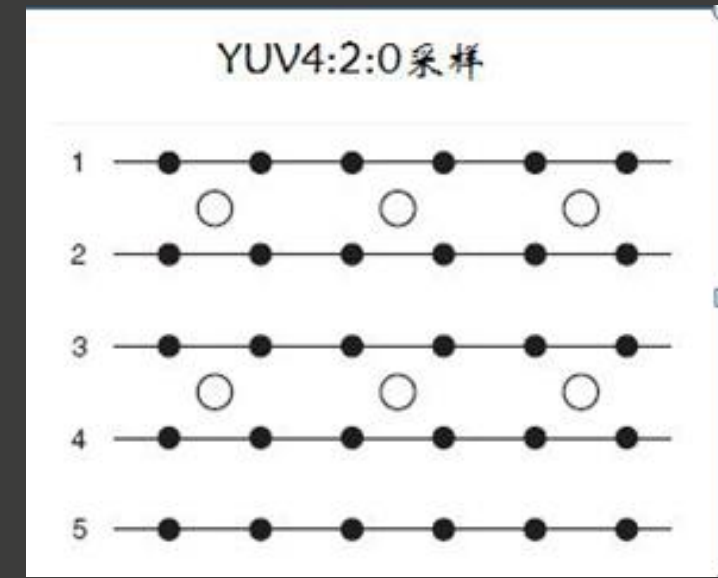
颜色空间



RGB



HSV



YUV

$\max = \max(R, G, B)$
 $\min = \min(R, G, B)$
 $V = \max(R, G, B)$
 $S = (\max - \min) / \max$
if $(R == \max)$ $H = (G - B) / (\max - \min) * 60$
if $(G == \max)$ $H = 120 + (B - R) / (\max - \min) * 60$
if $(B == \max)$ $H = 240 + (R - G) / (\max - \min) * 60$
if $(H < 0)$ $H = H + 360$

RGB \rightarrow YUV
 $Y = 0.299R + 0.587G + 0.114B$
 $U' = (BY) * 0.565$
 $V' = (RY) * 0.713$

YUV \rightarrow RGB
 $R = Y + 1.403V'$
 $G = Y - 0.344U' - 0.714V'$
 $B = Y + 1.770U'$

二值化



Gray > T ? 255 : 0

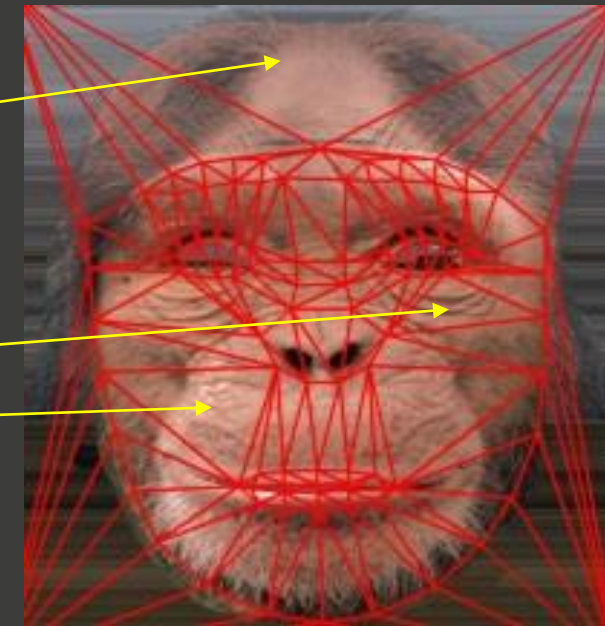
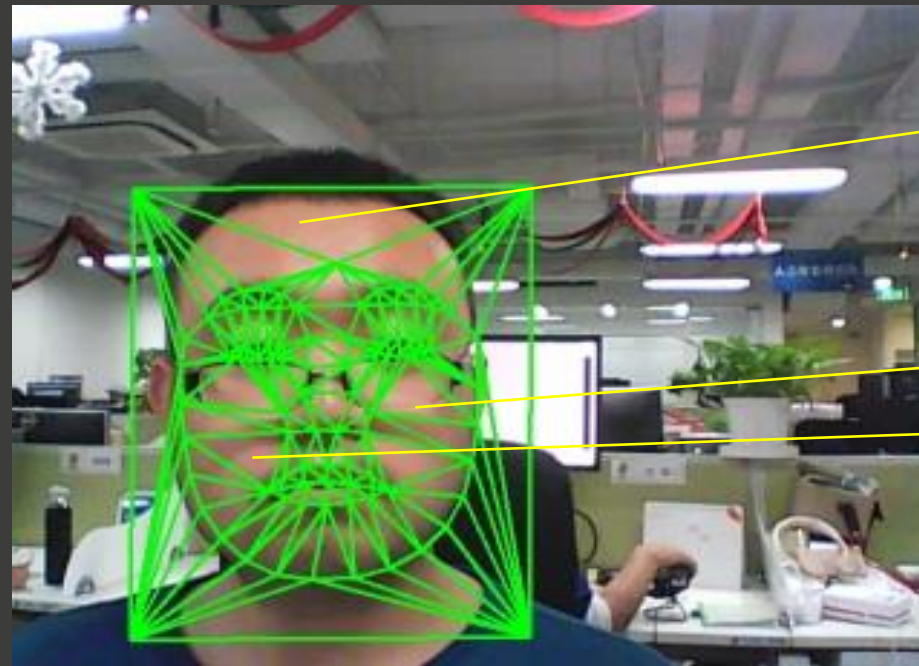
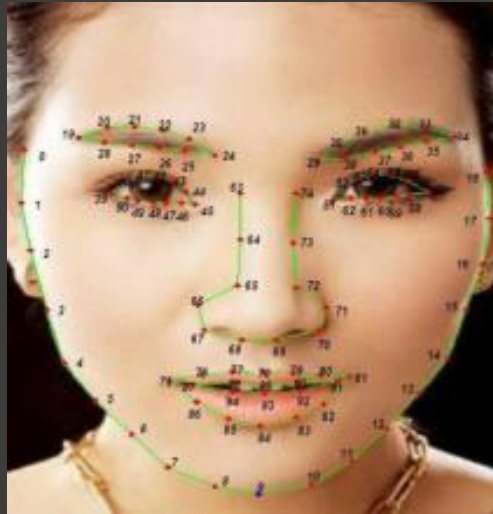
$$\text{Gray} = R \times 0.299 + G \times 0.587 + B \times 0.114$$

高斯模糊

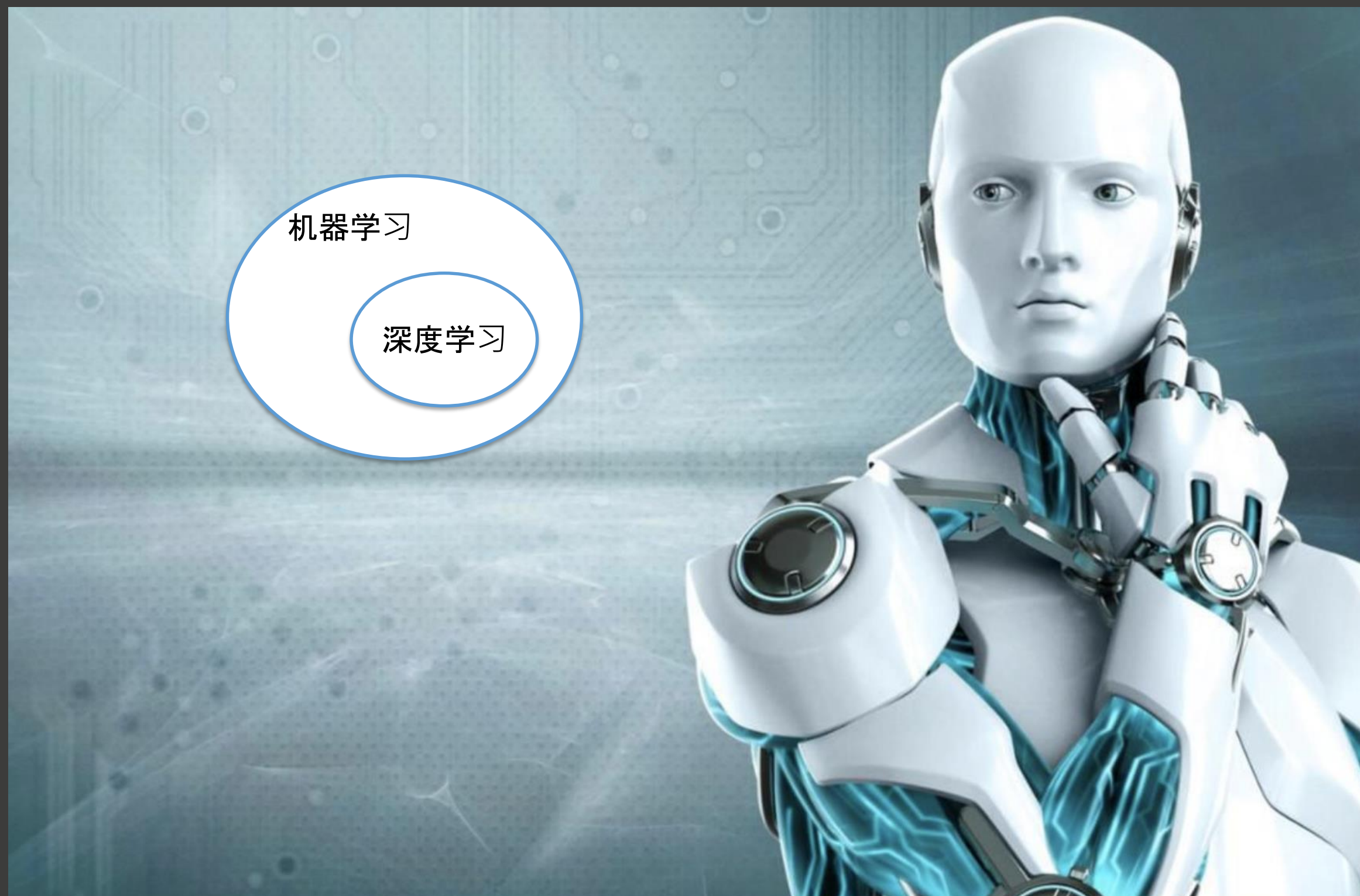


0.0947416	0.118318	0.0947416
0.118318	0.147761	0.118318
0.0947416	0.118318	0.0947416

分析与应用-三角变换



人工智能 (Artificial Intelligence)

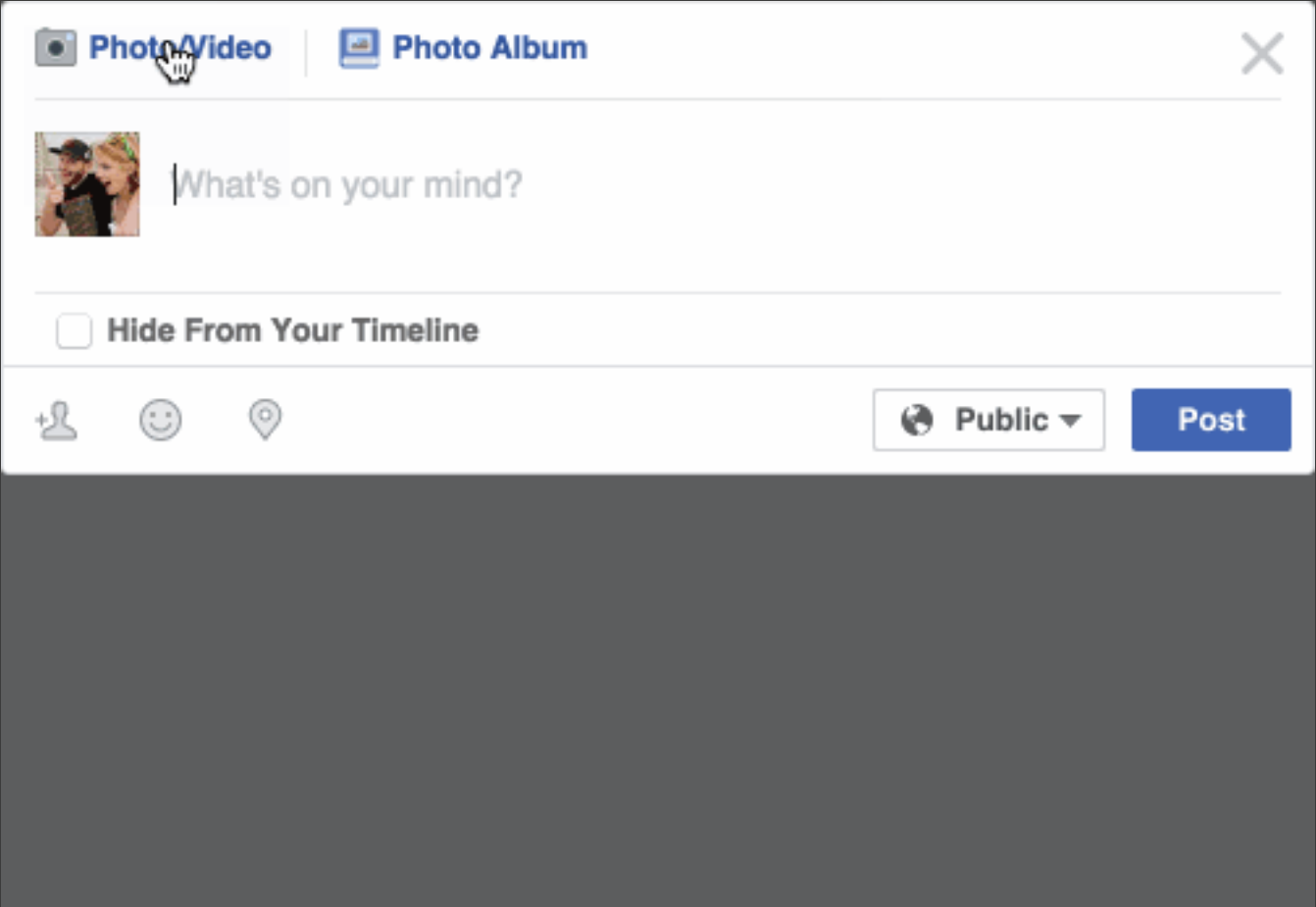


人脸识别

手势识别

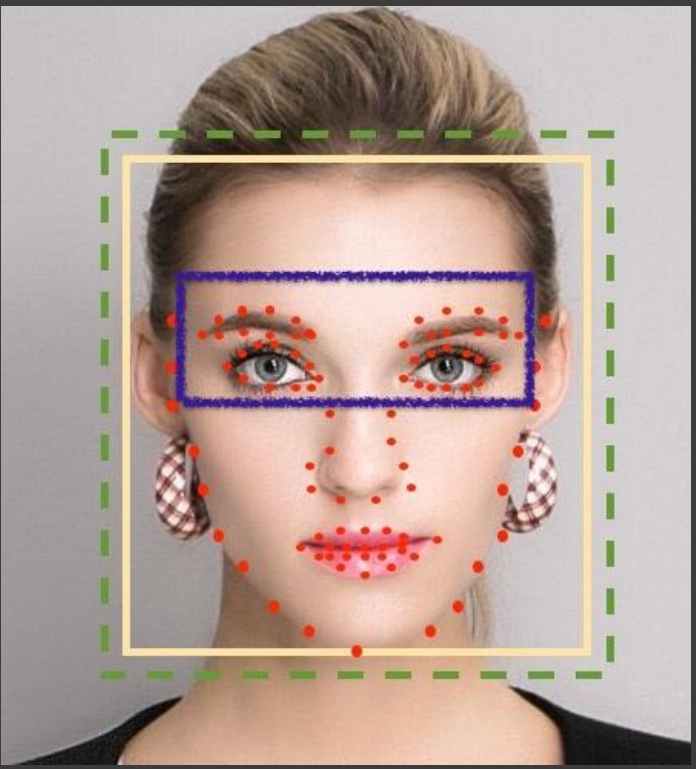
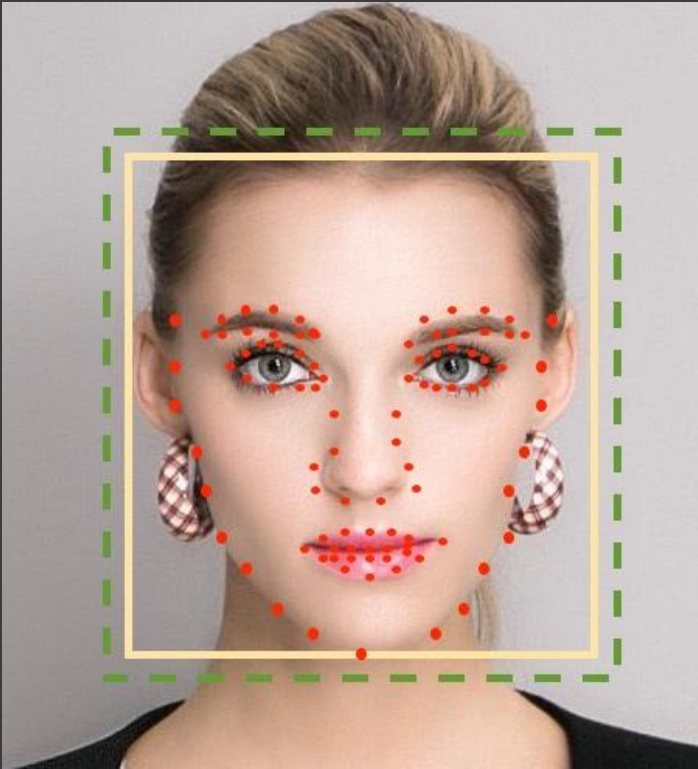
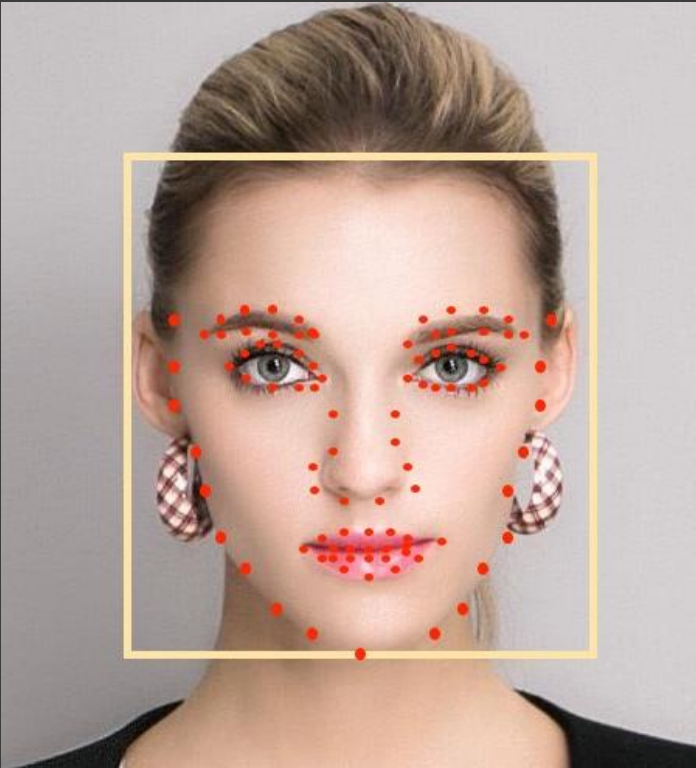
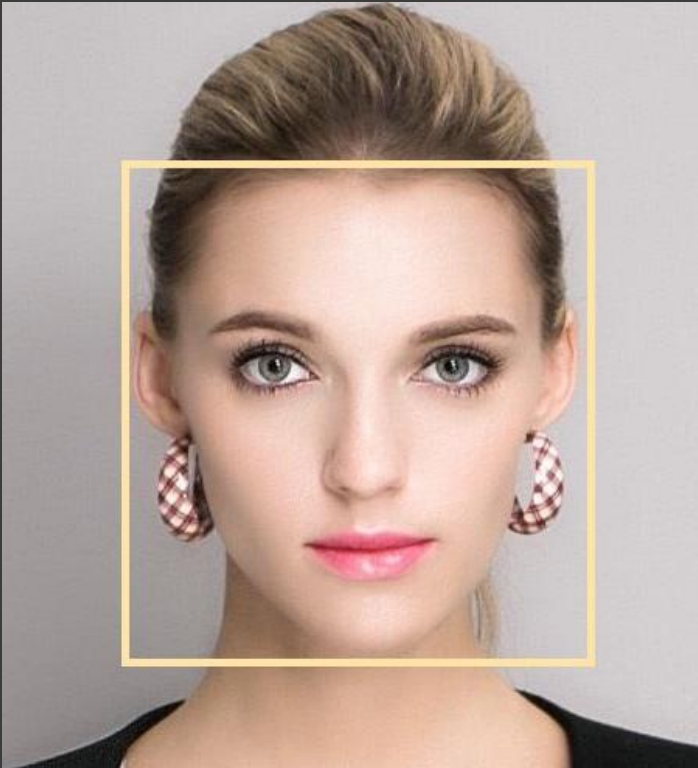
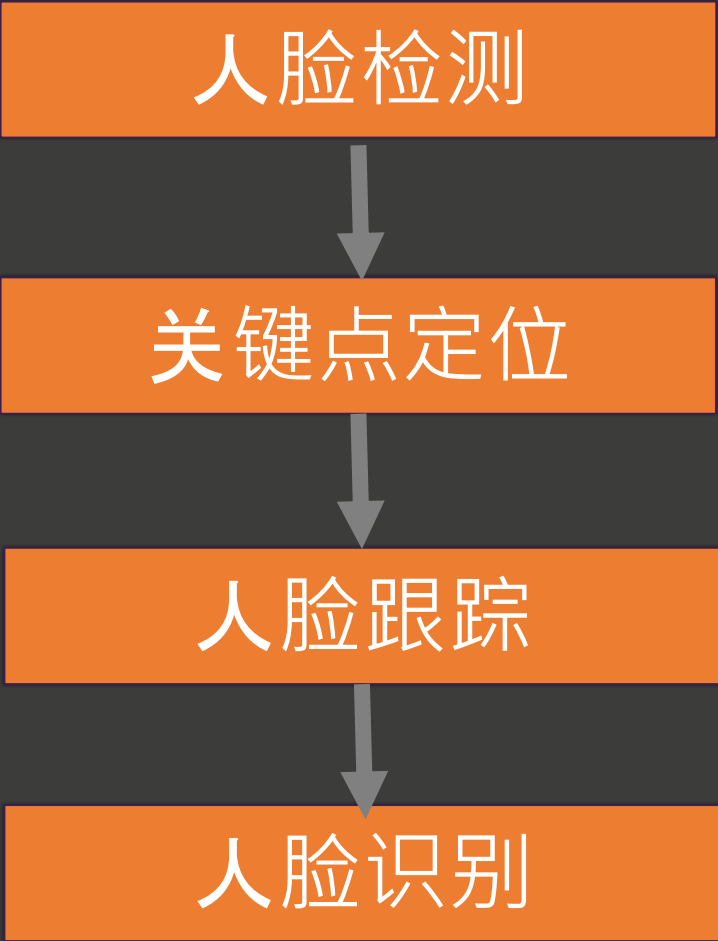
人体抠图

FACEBOOK

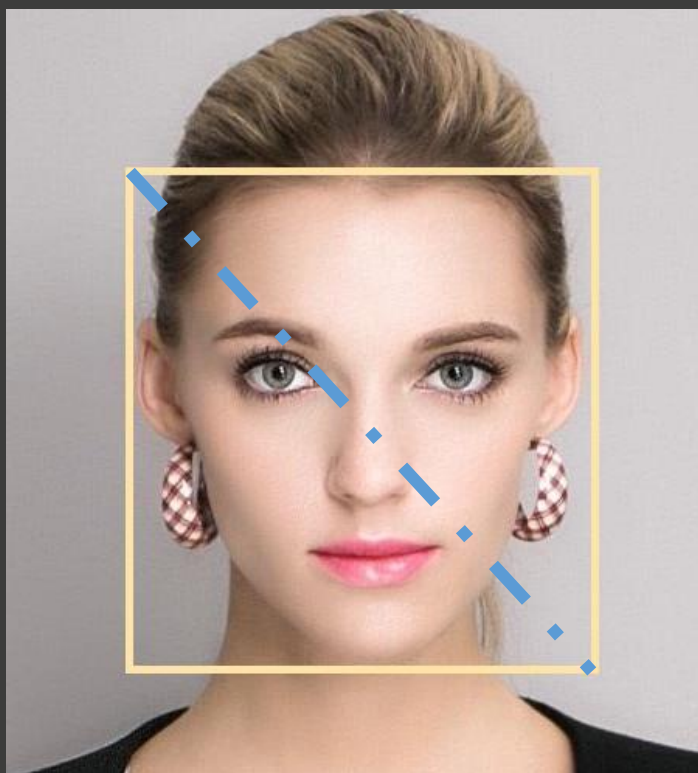


正确率：98%

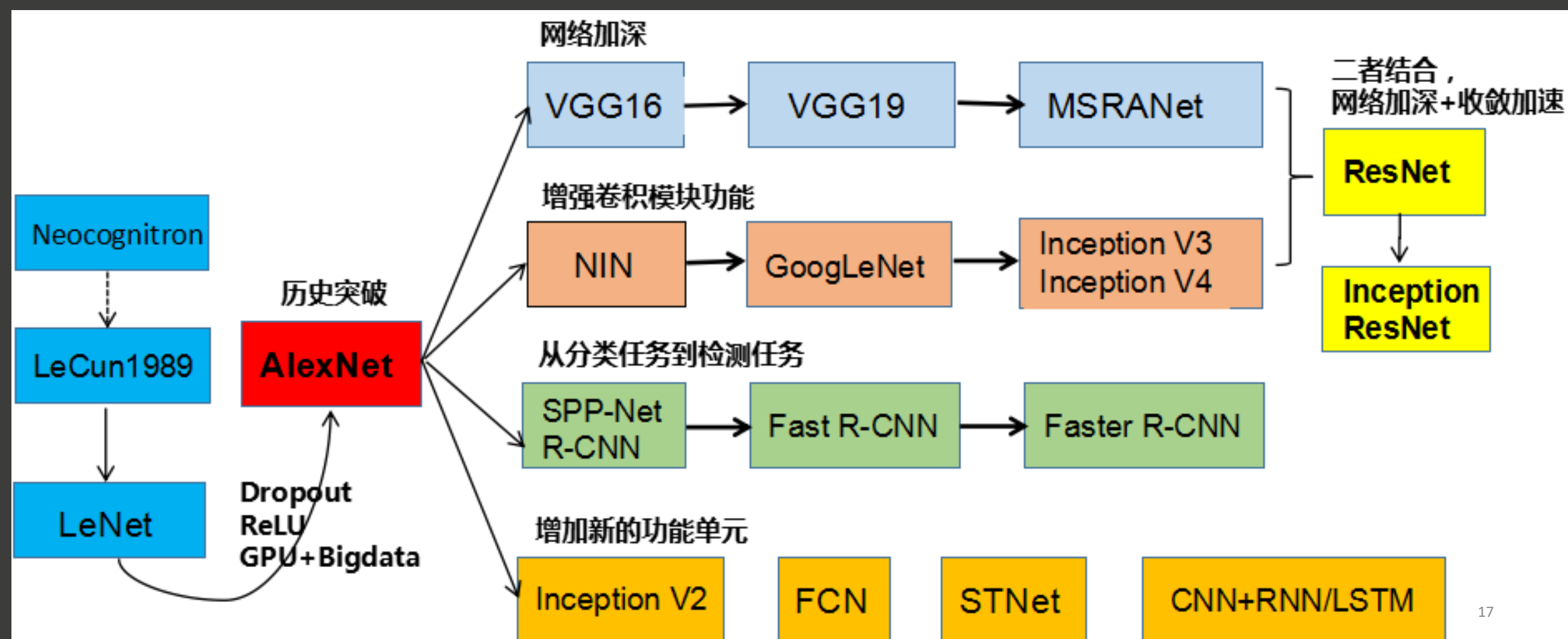
360AI方法-人脸检测及识别

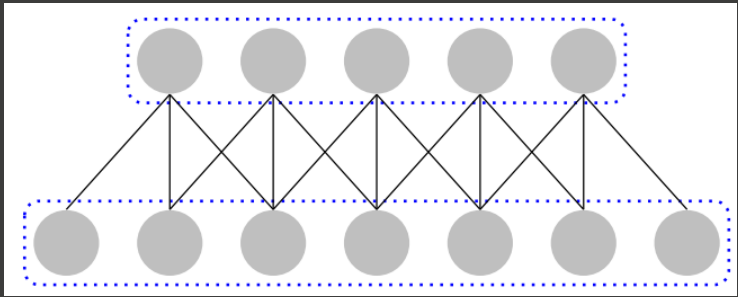
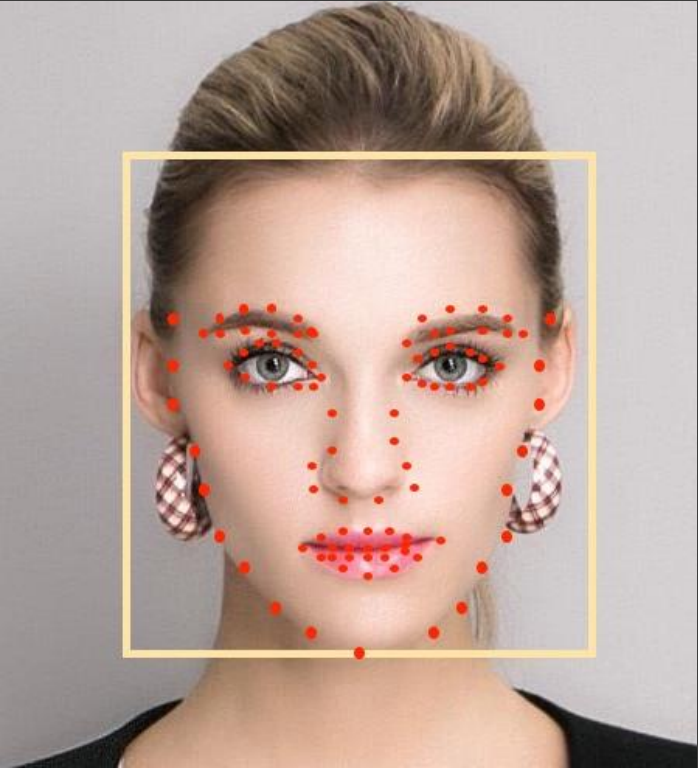


360AI方法-人脸检测 Faster R-CNN网络深入学习

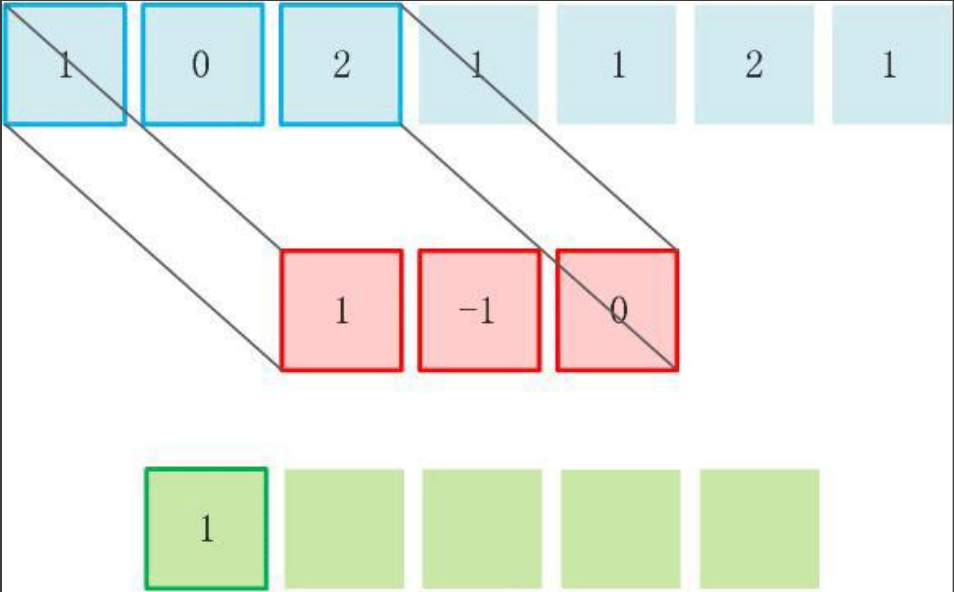


标注人脸框的对角线坐标

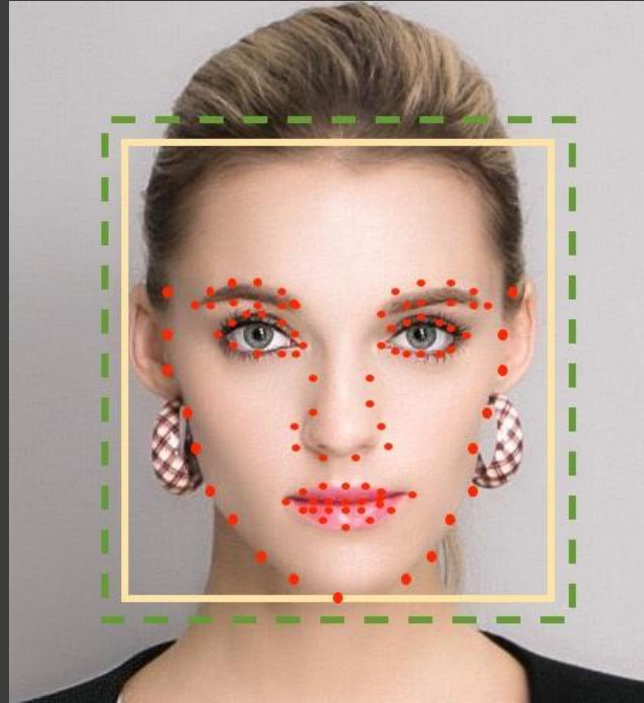




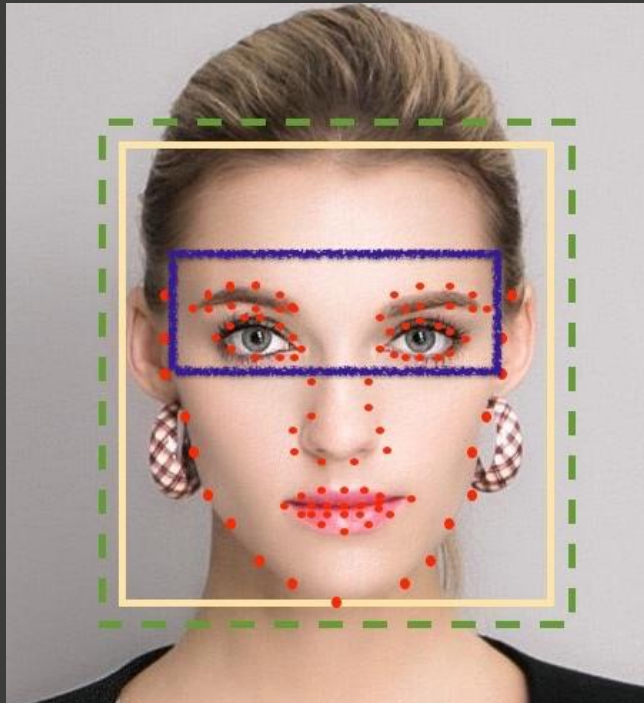
CNN



95个关键点

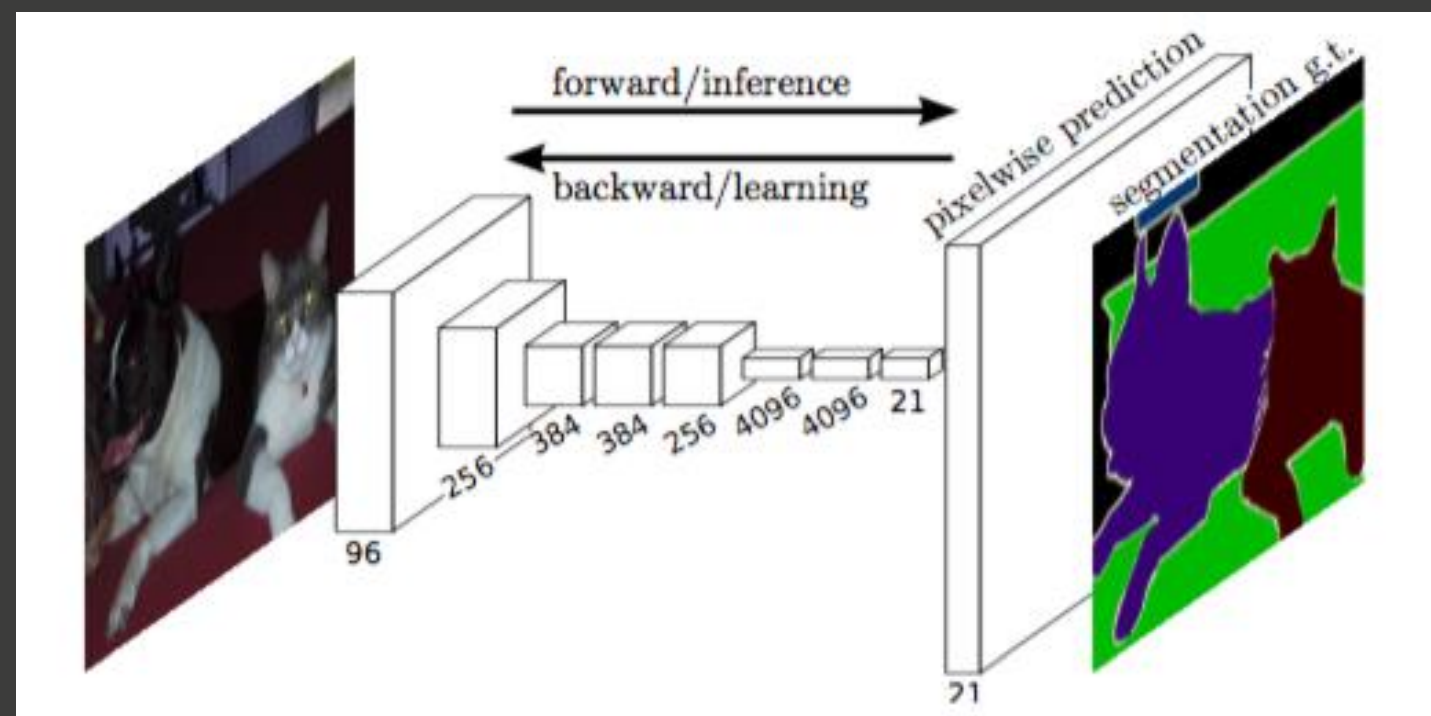


主动把人脸框变大，定位新的关键点

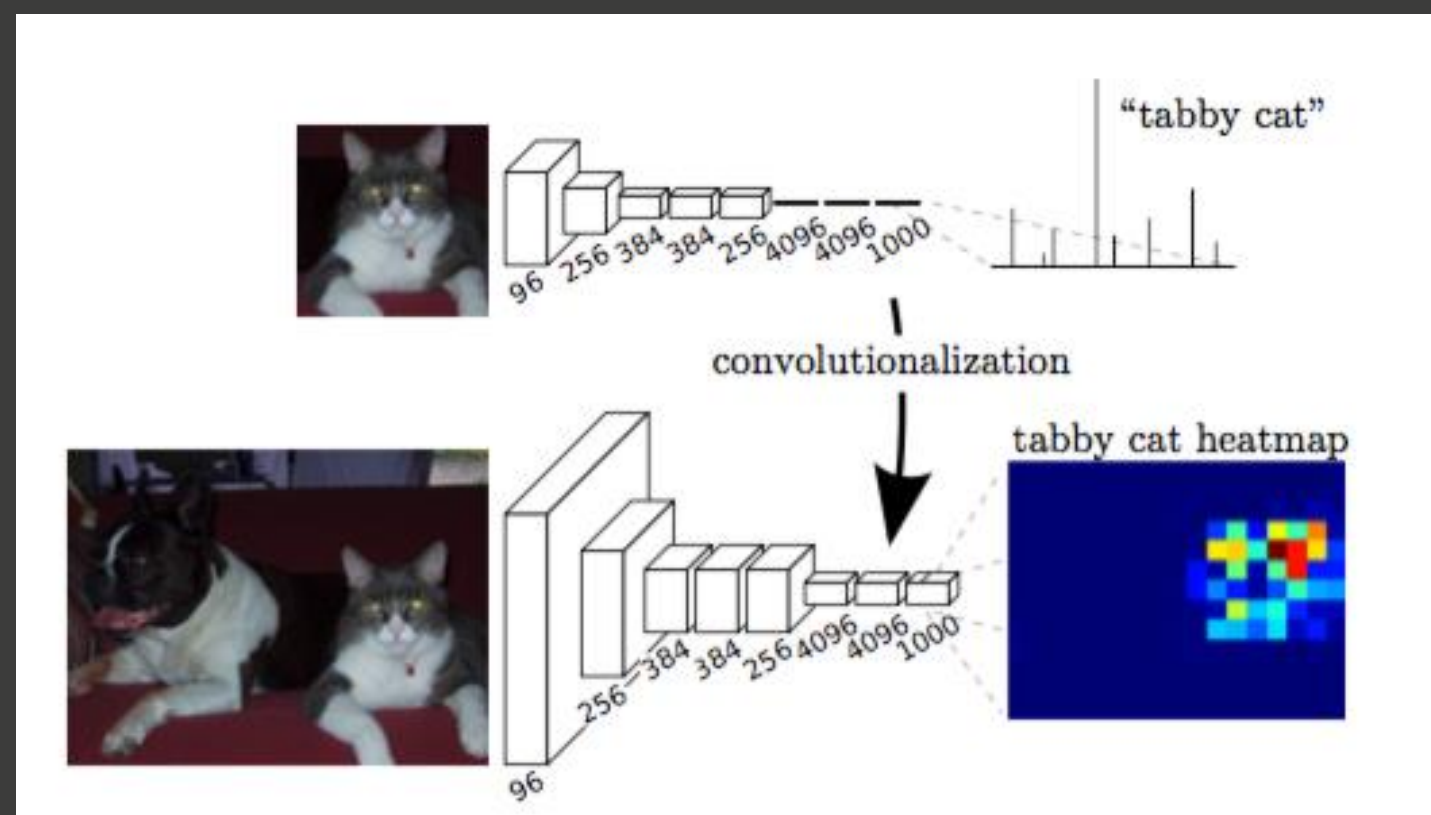


移动端处理有限的特征数据进行识别

360AI方法-人体分割-FCN



FCN(全卷积神经网络)



360AI方法-手势-Faster R-CNN



双手爱心

拜托

誓言

手掌

OK

单手爱心

拳头

摇滚

托手

手枪

点赞

食指平放

胜利

指天

人脸特效

抠图特效

美颜特效

手势特效

应用
特效



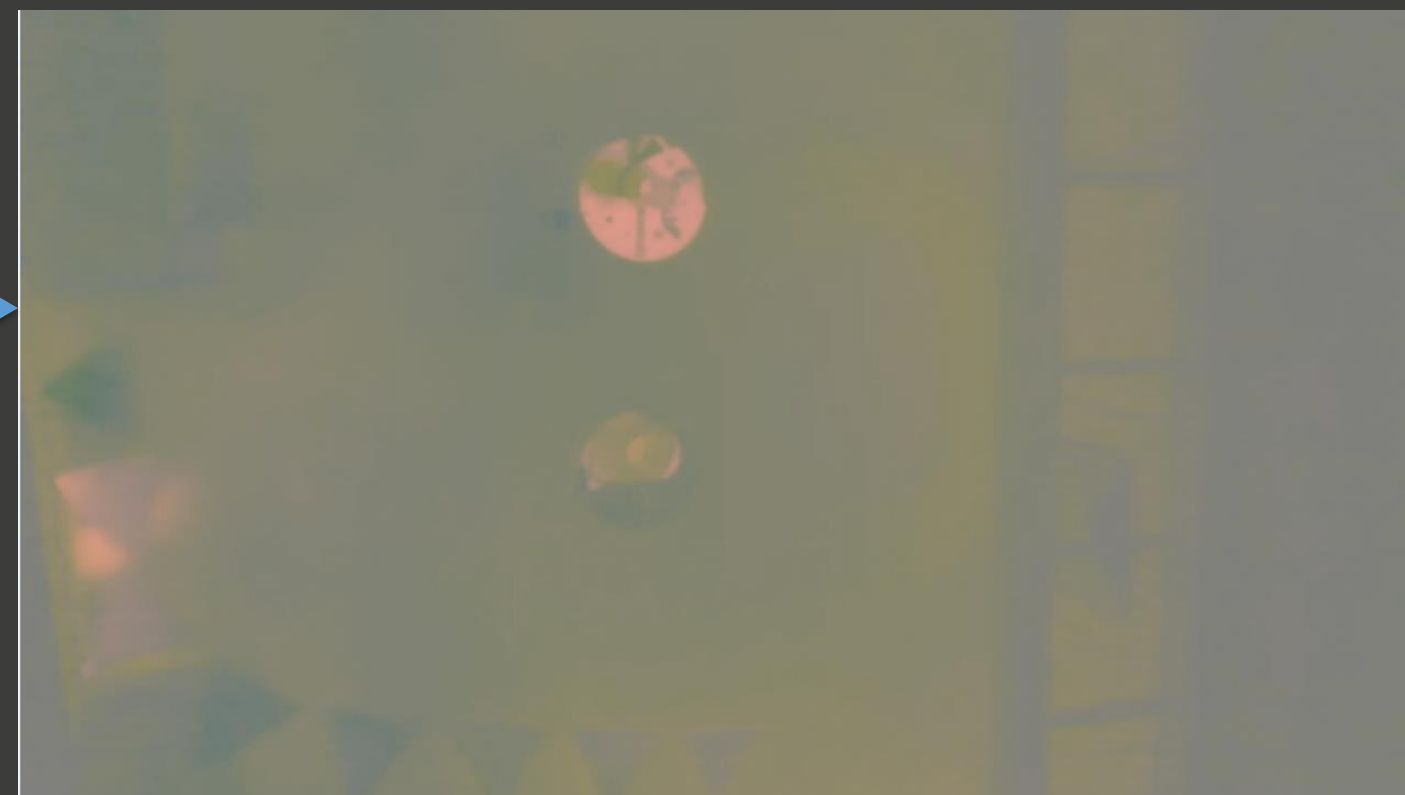
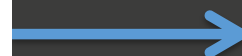
- 内存优化
- 省电
- 单帧耗时

问题-激活纹理单元



```
GLS20. glActiveTexture(GLS20. GL_TEXTURE0 + 10);  
int y = GLS20. glGetUniformLocation(mProgram, "tex_y");  
GLS20. glBindTexture(GLS20. GL_TEXTURE_2D, textureY);  
GLS20. glUniform1i(y, 10);
```

```
GLS20. glActiveTexture(GLS20. GL_TEXTURE0 + 11);  
int u = GLS20. glGetUniformLocation(mProgram, "tex_uv");  
GLS20. glBindTexture(GLS20. GL_TEXTURE_2D, textureUV);  
GLS20. glUniform1i(u, 11);
```

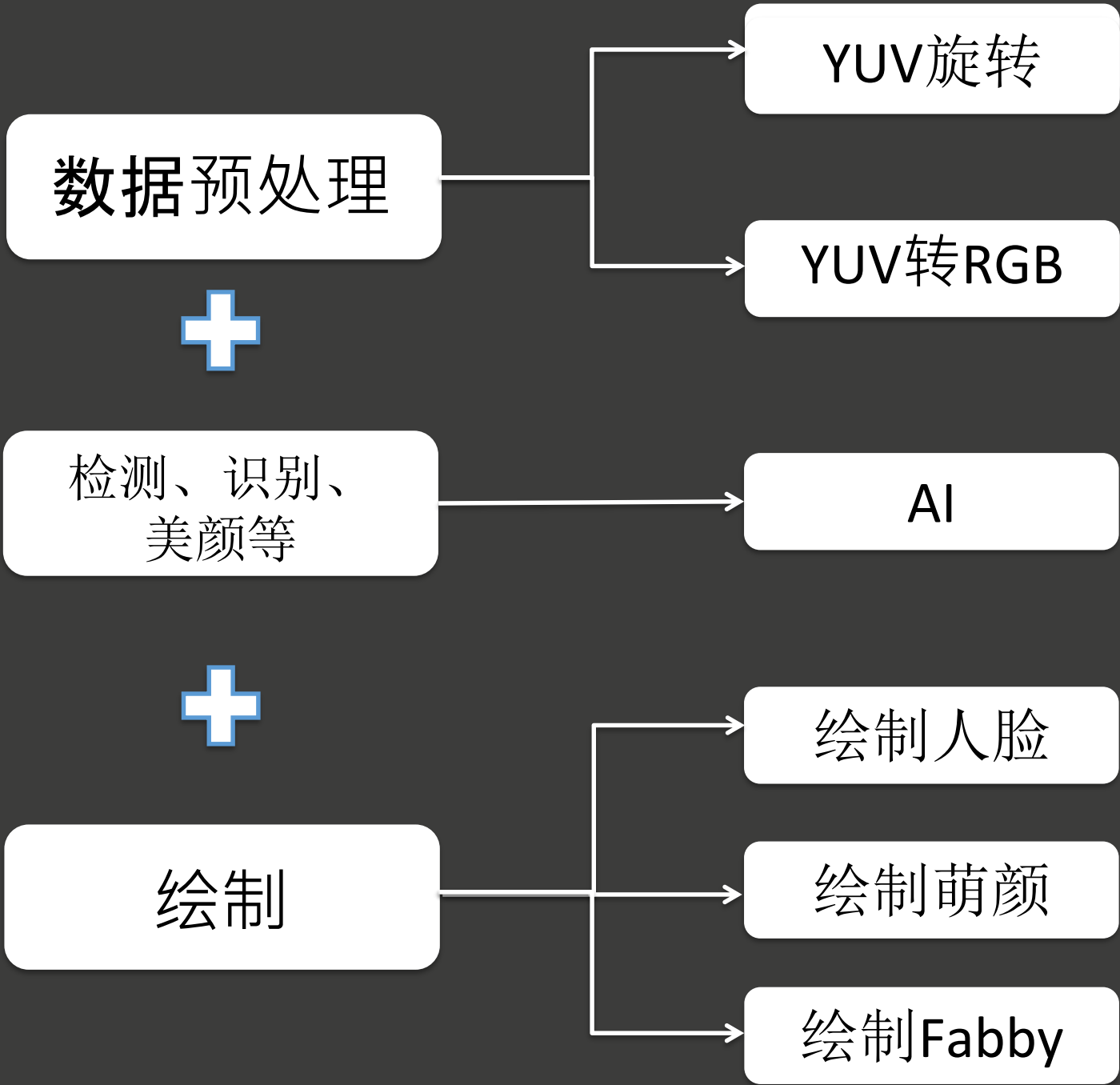


```
GLS20. glGetIntegerv(GLS20. GL_MAX_TEXTURE_IMAGE_UNITS, result);
```

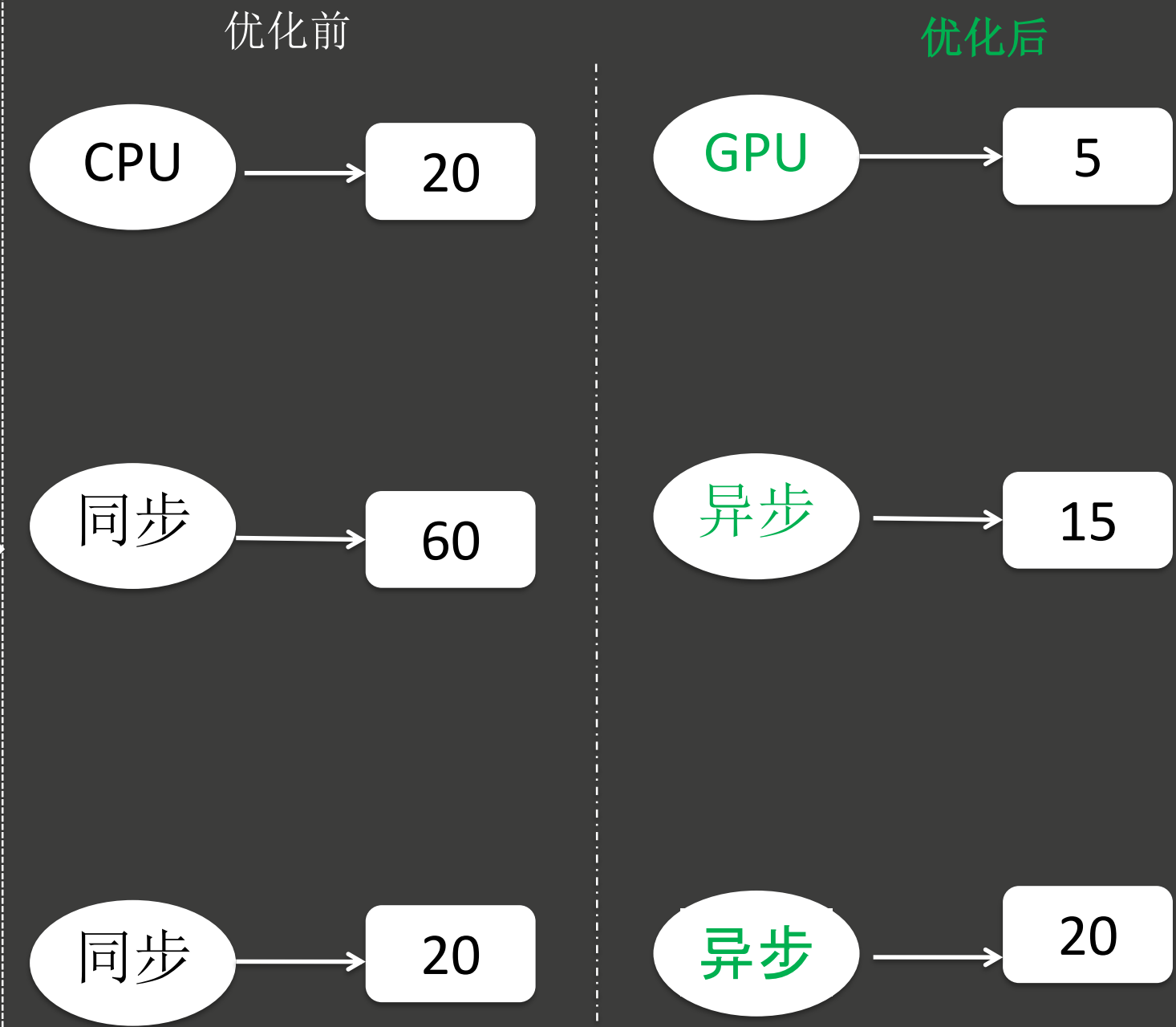

问题-单帧耗时

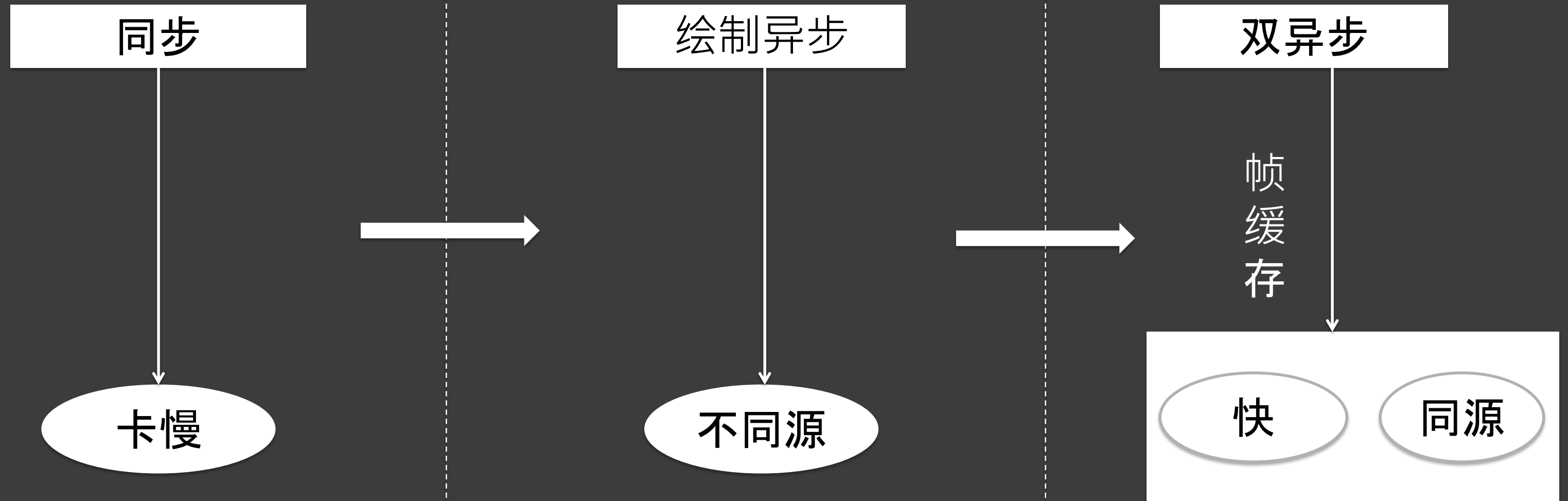


过程



结果







技术交流（干货）：奇卓社（360移动技术微信公众号）

谢谢