# 端侧轻量级人脸检测模型的对比研究与性能评估

董霁兴　杨成鑫

山东师范大学信息科学与工程学院

2024 年 12 月 17 日

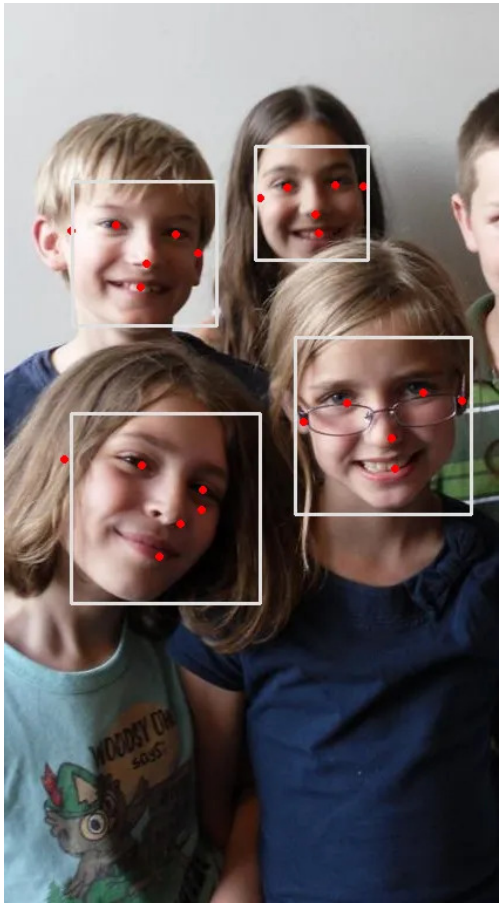
# 实验内容概述

1. 理论研究与综述
   - 人脸检测技术的基本原理与研究现状
   - 经典轻量化卷积神经网络的介绍
   - 本次实验使用的轻量人脸检测模型介绍
2. 实验研究与评估
   - 统一导出 ONNX 格式并使用 ONNX Runtime 推理
   - 在 WIDER FACE 和自建数据集上进行测试
   - 评估检测精度、计算效率和存储开销、针对不同场景给模型选择建议
3. 实验分工与贡献
   - 杨成鑫：负责本次实验所用模型的调研及综述，参与实验方案设计与优化，进行实验结果定量分析
   - 董霁兴：负责实验设计，数据集整理，代码实现与性能评估

# 目录

# 人脸检测任务介绍

- 人脸检测是人脸智能分析应用的核心基础组件
- 应用领域广泛：
  - 智能安防：监控画面中的人脸识别
  - 社交娱乐：特效和互动功能
  - 人脸识别、属性分析的预处理步骤
- 人脸检测与人脸识别的区别：
  - 人脸检测：确定图像中是否存在人脸及其位置
  - 人脸识别：识别和验证人脸的身份(如机场安检、手机解锁)
  - 人脸检测是所有基于人脸特征的分析算法的前提



# 技术发展历程

基于手工特征的传统方法

- 基于知识的方法
- 特征不变方法
- 模板匹配方法
- 基于外观的方法
- Viola-Jones 算法 (2001)
  - Haar 特征
  - Adaboost 算法
- ...

基于深度学习的方法

- 多阶段检测架构
  - Cascade CNN, MTCNN
- 两阶段检测架构
  - Face R-CNN, ScaleFace
- 单阶段检测架构
  - SSD, RetinaNet

# 端侧人脸检测

## 端侧设备特点

- 资源受限
  - 计算能力有限
  - 存储容量受限
  - 功耗要求严格

## 应用场景

- 手机端
  - 自拍美颜
  - 视频通话人脸跟踪
- 嵌入式设备
  - 智能安防摄像头
  - 智能门禁系统

# 目录

# 轻量化卷积网络概述

- 负责提取图像特征
- 直接影响模型效率和准确性
- 主流轻量化卷积网络：
  - SqueezeNet (2016.02)
  - MobileNets (2017.04)
  - ShuffleNet (2017.07)
  - 其他：Xception、EfficientNet、GhostNet 等
- 特别说明：本次介绍仅涉及各网络的初始版本

# SqueezeNet 架构

## 主要特点

- 2016 年 2 月由伯克利和斯坦福研究人员提出，是较早提出的一个轻量化神经网络
- 在保持和 AlexNet 相同准确率的情况下，将模型参数减少到原来的 50 倍
- 压缩后仅 0.47MB

## 三个设计策略

1. 用 1×1 卷积替代 3×3 卷积
2. 减少 3×3 卷积的输入通道数
3. 延迟下采样，保持较大特征图尺寸

# Fire Module

- 核心组成部分:
  - squeeze 层: 使用 1×1 卷积压缩特征图通道数
  - expand 层: 使用 1×1 和 3×3 卷积提升通道数并拼接结果

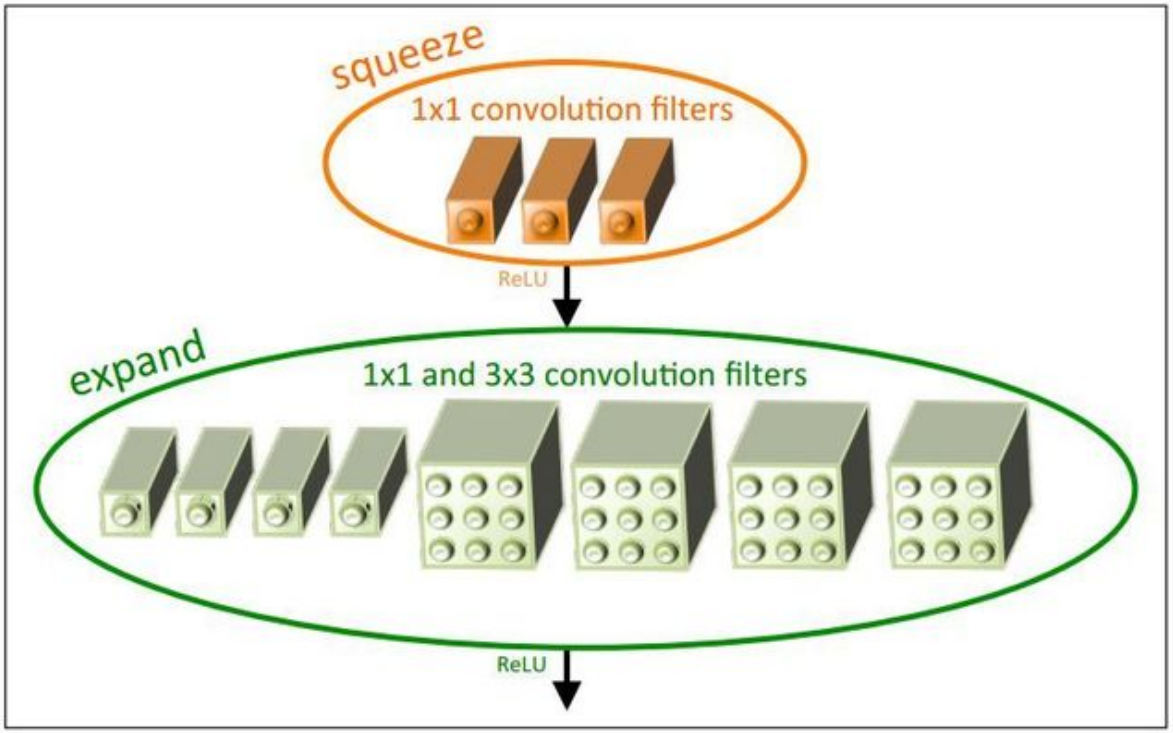



图 1: Fire Module 结构

# SqueezeNet 性能对比

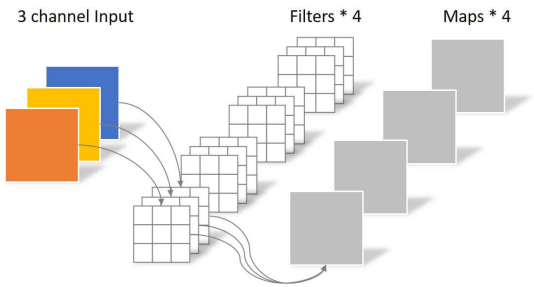
| CNN architecture | Compression Approach | Data Type | Original → Compressed Model Size | Reduction in Model Size vs. AlexNet | Top-1 ImageNet Accuracy | Top-5 ImageNet Accuracy |
|---|---|---|---|---|---|---|
| AlexNet | None (baseline) | 32 bit | 240MB | 1x | 57.2% | 80.3% |
| AlexNet | SVD (Denton et al., 2014) | 32 bit | 240MB → 48MB | 5x | 56.0% | 79.4% |
| AlexNet | Network Pruning (Han et al., 2015b) | 32 bit | 240MB → 27MB | 9x | 57.2% | 80.3% |
| AlexNet | Deep Compression (Han et al., 2015a) | 5-8 bit | 240MB → 6.9MB | 35x | 57.2% | 80.3% |
| SqueezeNet (ours) | None | 32 bit | 4.8MB | **50x** | 57.5% | 80.3% |
| SqueezeNet (ours) | Deep Compression | 8 bit | 4.8MB → 0.66MB | **363x** | 57.5% | 80.3% |
| SqueezeNet (ours) | Deep Compression | 6 bit | 4.8MB → 0.47MB | **510x** | 57.5% | 80.3% |

图 2: SqueezeNet 与 AlexNet 在 ImageNet 上的对比

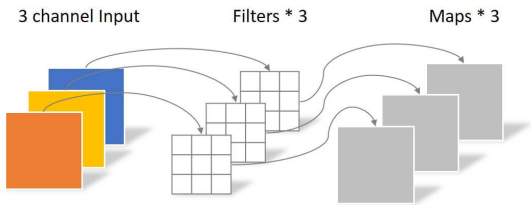
- 在 ImageNet 数据集上的 TOP-1 和 TOP-5 的准确率都与 AlexNet 相似
- SqueezeNet 模型参数量仅为 AlexNet 的 1/50，压缩后模型文件仅 0.47MB

# MobileNet 架构

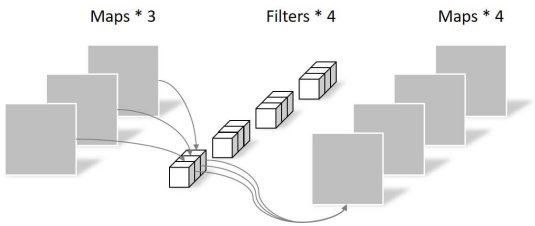
- 2017 年 4 月由 Google 团队提出
- 基于流线型架构，提出深度可分离卷积构建轻量级深层神经网络，在减少模型参数两的同时保持良好的特征提取能力。
- 深度可分离卷积的两个步骤：
  - 深度卷积 (Depthwise Conv)：每个通道独立卷积，实现空间信息的提取
  - 逐点卷积 (Pointwise Conv)：1×1 卷积融合通道信息，确保每个输出特征图包含所有输入特征图的信息




(a) 标准卷积




(b) 深度卷积




(c) 逐点卷积

# 深度可分离卷积参数量分析

## 参数量

$$P_{std} = K \times K \times C \times N = K^2 \times C \times N$$

$$P_{ds} = (K \times K \times N) + (1 \times 1 \times C \times N) = (K^2 + C) \times N$$

## 压缩比

$$\frac{P_{ds}}{P_{std}} = \frac{(K^2 + C) \times N}{K^2 \times C \times N} = \frac{1}{C} + \frac{1}{K^2}$$

当 $K = 3$ 时，压缩比约为 9 倍

# MobileNet 性能对比

Table 8. MobileNet Comparison to Popular Models

| Model | ImageNet Accuracy | Million Mult-Adds | Million Parameters |
|---|---|---|---|
| 1.0 MobileNet-224 | 70.6% | 569 | 4.2 |
| GoogleNet | 69.8% | 1550 | 6.8 |
| VGG 16 | 71.5% | 15300 | 138 |



图 4: MobileNet 与其他模型在 ImageNet 上的性能对比

- 精度略高于 GoogLeNet，但计算量仅为其 1/3
- 相比 VGG-16 精度降低 0.9%，但参数量和计算量大幅降低

# ShuffleNet

- 2017 年 7 月由 Face++ 团队（旷世科技）提出
- MobileNet 中 $1 \times 1$ 卷积耗费了 94.8% 的计算时间
- 提出逐点分组卷积 (Pointwise Group Convolution) 减少计算时间
- 通过通道重排 (Channel Shuffle) 解决分组卷积信息交互问题
- 在相同参数量规模下，ShuffleNet 在 ImageNet 数据集上的 TOP-1 准确率高于 MobileNet

| Model | Complexity (MFLOPs) | Cls err. (%) | Δ err. (%) |
|---|---|---|---|
| 1.0 MobileNet-224 | 569 | 29.4 | - |
| ShuffleNet 2× ($g = 3$) | 524 | **26.3** | 3.1 |
| ShuffleNet 2× (with *SE*[13], $g = 3$) | 527 | **24.7** | 4.7 |
| 0.75 MobileNet-224 | 325 | 31.6 | - |
| ShuffleNet 1.5× ($g = 3$) | 292 | **28.5** | 3.1 |
| 0.5 MobileNet-224 | 149 | 36.3 | - |
| ShuffleNet 1× ($g = 8$) | 140 | **32.4** | 3.9 |
| 0.25 MobileNet-224 | 41 | 49.4 | - |
| ShuffleNet 0.5× ($g = 4$) | 38 | **41.6** | 7.8 |
| ShuffleNet 0.5× (shallow, $g = 3$) | 40 | 42.8 | 6.6 |

图 5: ShuffleNet 与其他模型在 ImageNet 上的性能对比

# 目录

# 轻量级人脸检测模型概述

- 本研究选择三个代表性模型：
  - RetinaFace (2019)
  - BlazeFace (2019)
  - YOLO5Face (2021)
- 选择标准：
  - 模型轻量化，实时性能好
  - 在如今端侧轻量化模型实践中应用广泛
  - 在 WiderFace 数据集取得较高排名

# RetinaFace

## 主要特点

- 2019 年由 Insight Face 团队提出，曾长期保持 SOTA 水平
- 采用多任务学习框架，结合监督学习和自监督学习
- 支持不同 backbone 选择，可平衡精度和速度
- 支持人脸检测、关键点定位和密集人脸对应关系预测

## 实验选用版本

- RetinaFace-MobileNet0.25
- RetinaFace-MobileNetV2
- 可直接对原始图片进行推理，无需预处理缩放

# BlazeFace 特点

- Google 专为移动 GPU 优化设计
- Google ML Kit 和 MediaPipe 默认人脸检测模型
- 在 iPhone XS 上推理时间仅需 0.6ms
- 基于 MobileNetV1/V2 定制特征提取网络
- 提出适合 GPU 运算的新型 anchor 方案
- 改进 NMS 策略，提高视频检测稳定性

# BlazeFace 实验选用版本

- BlazeFace-128 为前置摄像头设计，接收 128×128 的输入
- BlazeFace-320 和 BlazeFace-640 分别接受 320×320 和 640×640 的输入
- 采用 padding 方式等比例缩放
- 比 MobileNetV2-SSD 快近 4 倍
- 支持人脸关键点预测和旋转角度估计

# YOLO5Face

## 主要特点

- 将人脸检测视为通用目标检测任务
- 基于 YOLOv5 增加关键点回归功能并改进网络结构
- 在 WiderFace 数据集上性能优异，超过许多专门的人脸检测模型

## 实验选用版本

- YOLO5Face 基于 ShuffleNetv2 设计轻量级模型，优化网络架构大幅缩小模型体积
- 本次实验使用 YOLOv5n 和 YOLOv5n-0.5 两个版本
- 输入尺寸：640×640，采用 padding 方式等比例缩放
- 非常适合在嵌入式设备和移动设备上部署

# 目录

# 实验环境

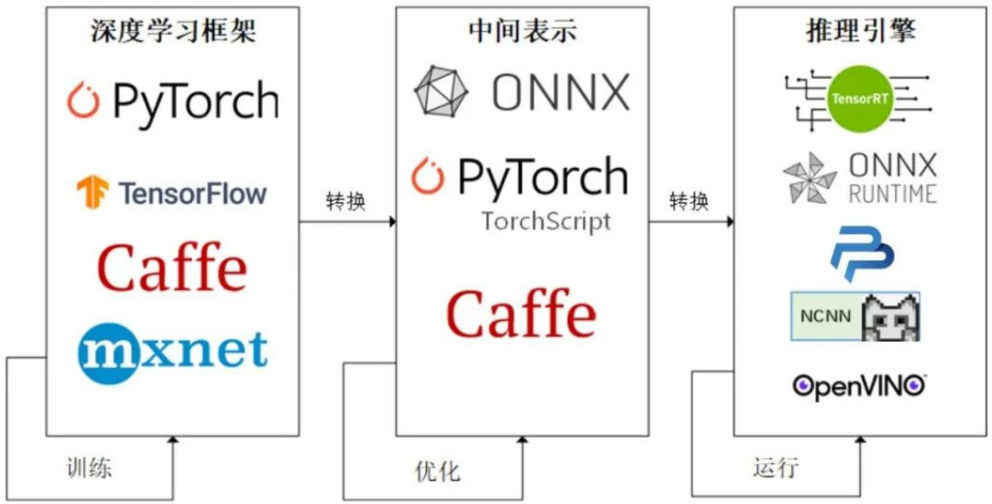



图 6: ONNX 模型转换流程

- 统一导出为 ONNX 格式
- 使用 ONNX Runtime 作为推理引擎
- 硬件: i5-10400 CPU (0.5TFLOPS)

# 评估指标

- 检测精度：使用标准评估代码计算各模型在 WIDER FACE 数据集上的精度，同时在自建数据集 light_face 上计算 mAP
- 计算效率：在 ONNX Runtime 上测试推理时延
- 模型大小：对比导出为 ONNX 格式后各模型的大小

# 实验数据集

## WIDER FACE 验证集

- 人脸检测领域标准基准数据集
- 具有广泛代表性和权威性
- 提供了标准的评估代码

## 自建单人人脸数据集 light_face

- 通过笔记本和手机前置摄像头采集，考虑姿态角度和光照变化
- 约 1000 张图像样本
- 贴近端侧实际应用场景
- 使用腾讯云 API 进行标注

# 模型性能对比

表 1: 各模型性能指标对比

| Model | Size(MB) | FPS | Easy | Medium | Hard | light_face |
|---|---|---|---|---|---|---|
| retinaface_mv1 | 1.66 | 8.68 | 0.91 | 0.88 | 0.73 | 0.99 |
| retinaface_mv2 | 11.93 | 4.15 | 0.94 | 0.92 | 0.82 | 0.99 |
| yolov5n_0.5_face | 5.65 | 22.23 | 0.91 | 0.88 | 0.75 | 0.99 |
| yolov5n_face | 10.51 | 12.90 | 0.94 | 0.92 | 0.81 | 0.99 |
| blazeface_128 | 0.44 | 70.79 | 0.18 | 0.10 | 0.04 | 0.67 |
| blazeface_320 | 0.68 | 46.46 | 0.60 | 0.46 | 0.20 | 0.93 |
| blazeface_640 | 0.68 | 16.26 | 0.80 | 0.64 | 0.35 | 0.98 |

- RetinaFace 和 YOLOv5 系列精度高但模型较大
- BlazeFace 系列速度快且轻量，但精度有所牺牲
- yolov5n_0.5_face 在模型大小、计算效率和精度上表现均衡

# 模型性能数据分析 - 模型大小与速度

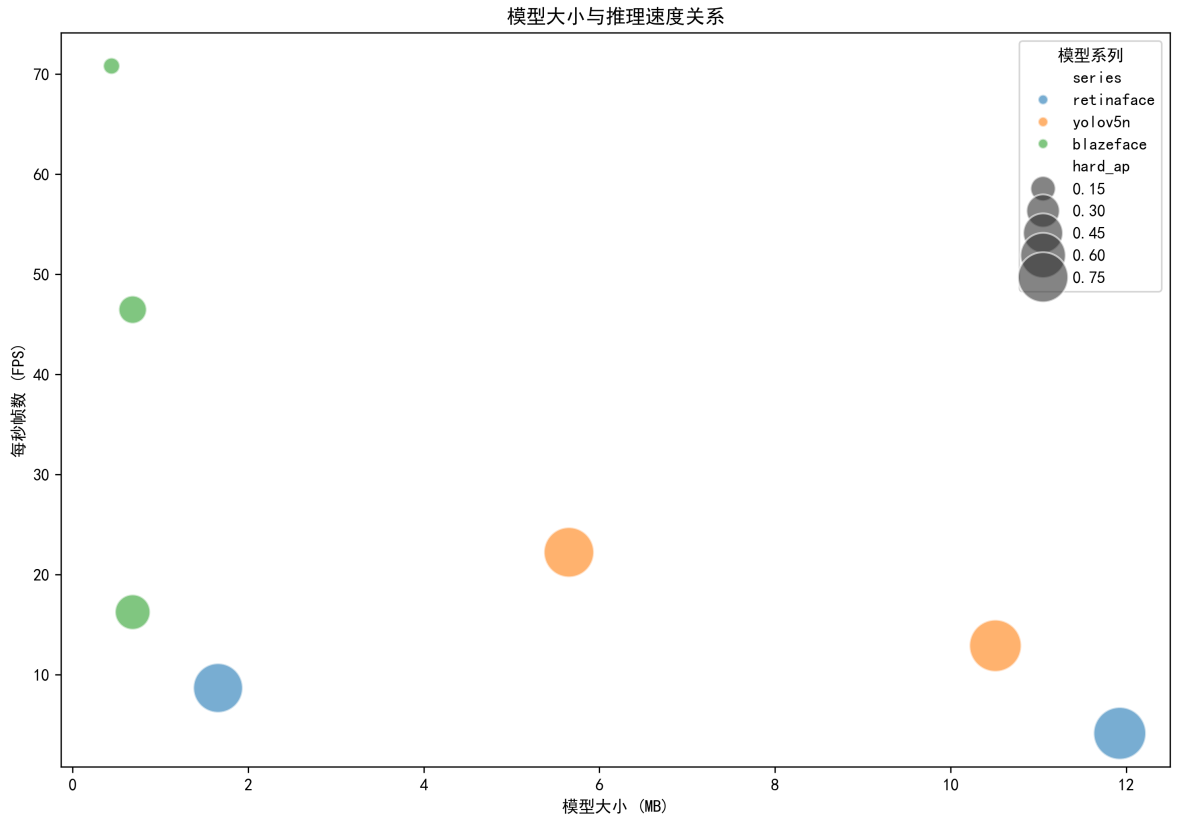



图 7: 模型大小与推理速度对比

# 模型性能数据分析 - 检测精度

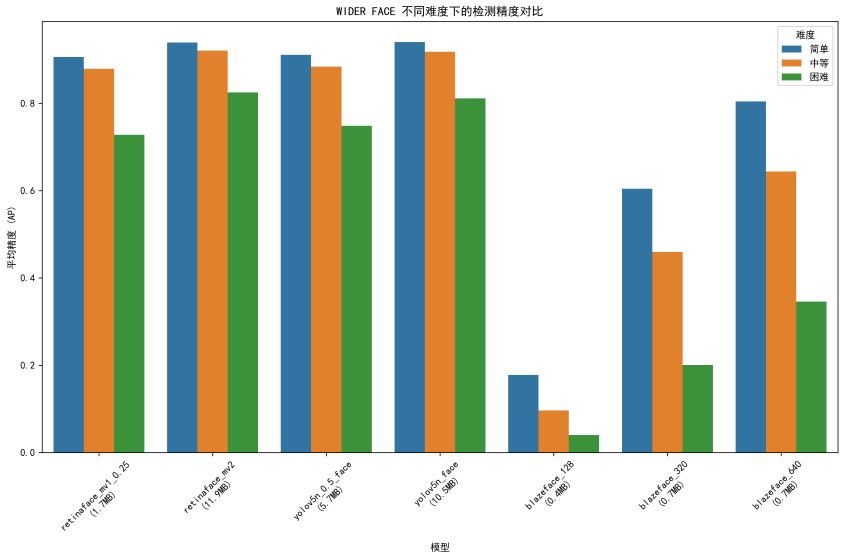



图 8: 不同难度级别下的检测精度对比

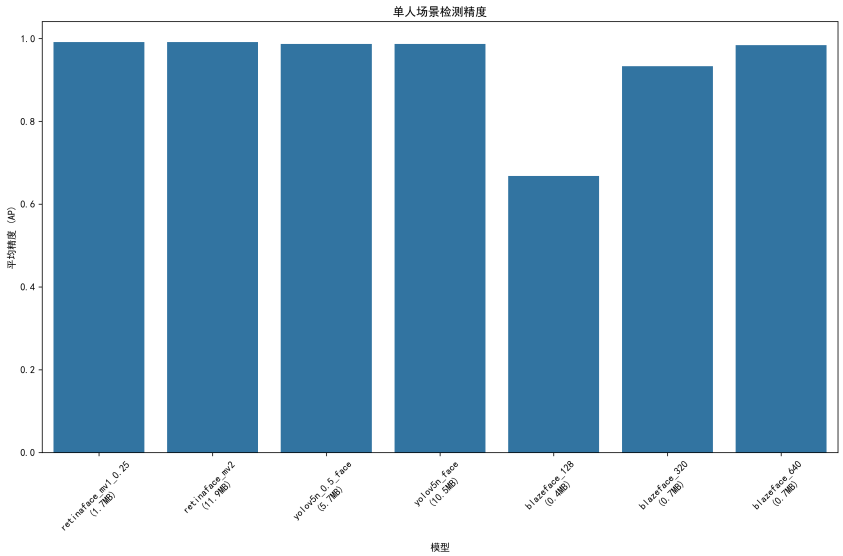



图 9: 单人人脸场景下的检测精度对比

# 应用场景建议

## 高精度场景

- 推荐: YOLOv5n_face 或 RetinaFace_mv2
- 特点: 各难度级别 AP 值均高 (>0.81)
- 代价: 模型较大 (10-12MB)，速度较慢 (4-13FPS)
- 适用: 安防监控，门禁考勤等

## 移动端前置摄像头

- 推荐: BlazeFace_320
- 特点: 体积小 (0.68MB)，速度快 (46FPS)，单人人脸识别场景 AP 达 0.93
- 代价: 非单人场景下精度较低 (0.6-0.8)
- 适用: 移动端实时检测场景，相机应用、视频会议、AR 试妆等

# 应用场景建议

## 轻量级场景

- 推荐: BlazeFace_128
- 特点: 最小体积 (0.44MB)，最快速度 (70+FPS)，在 light_face 数据集上精度尚可 (0.67)
- 代价: 精度较低 (0.18-0.6)
- 适用: 资源受限的 IoT 设备，例如智能门锁、手表、玩具等

## 平衡场景

- 推荐: YOLOv5n_0.5_face
- 特点: 中等体积 (5MB)，较快速度 (25FPS)，各难度级别 AP 值均衡 (0.7-0.8)
- 代价: 在各方面都不是最优
- 适用: 需要在资源和性能间取得平衡的场景，如移动端后置摄像头应用

# 感谢各位的聆听