HCMUS - FIT

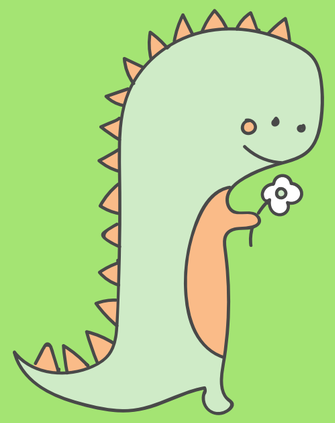
Facial Action Tracking

Group 10

Group 10

- 20127121 - Trần Ngô Gia Bảo
- 20127369 - Lê Quốc Trung
- 18127042 - Lê Phương Đào

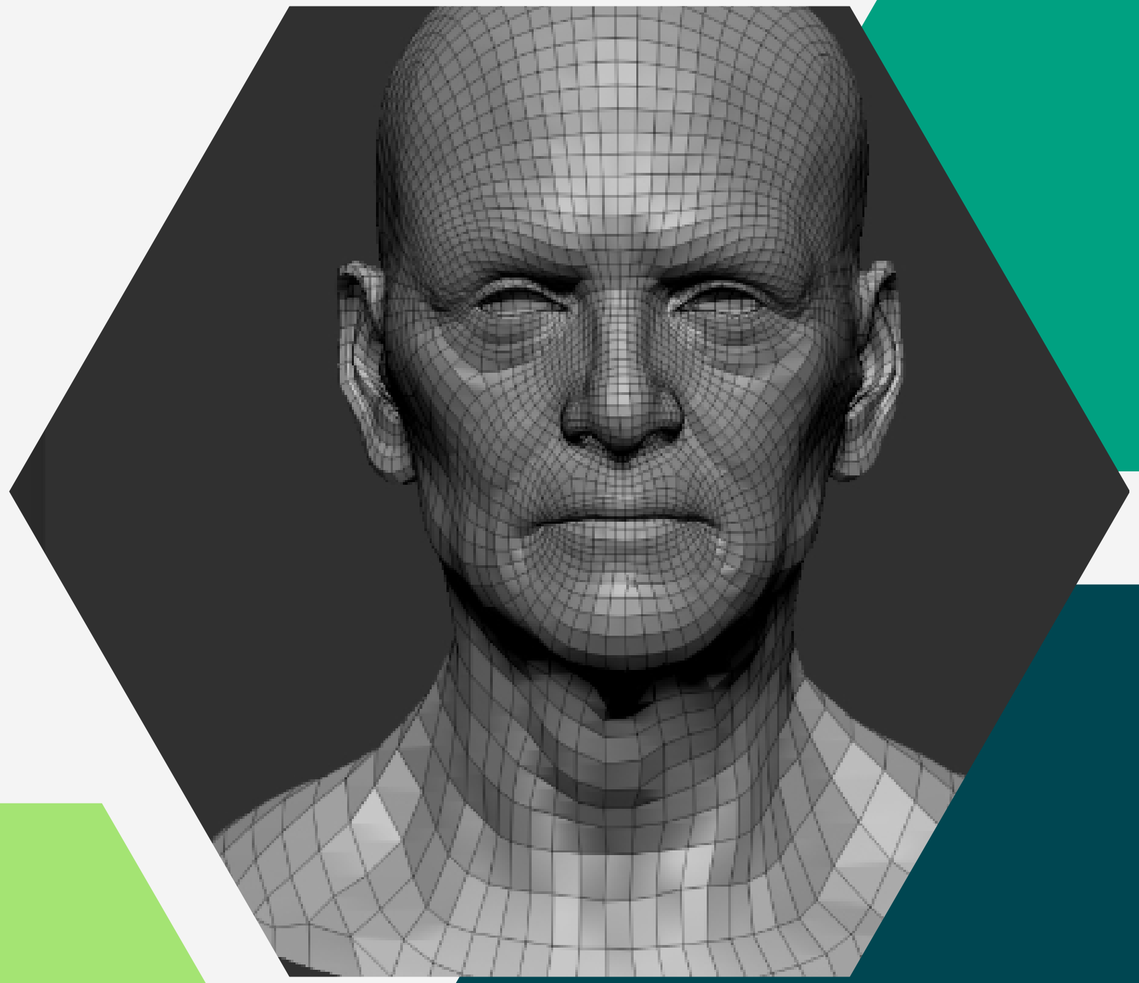
Presentation Content



- Book
- SOTA paper
- Demo

Introduction

Applications: face animation, facial expression analysis, and human-computer interfaces.



Parametric face modeling

Many ways to parameterize and model the appearance and behavior of the human face

Parametric face modeling

Three-dimensional motion and pose

the dynamic, 3D position and rotation of the head

Facial action

Facial feature motion such as lip and eyebrow motion

Shape and feature configuration

the shape of the head, face and the facial features

Parametric face modeling

Illumination

the variability in appearance due to different lighting conditions

---

Texture

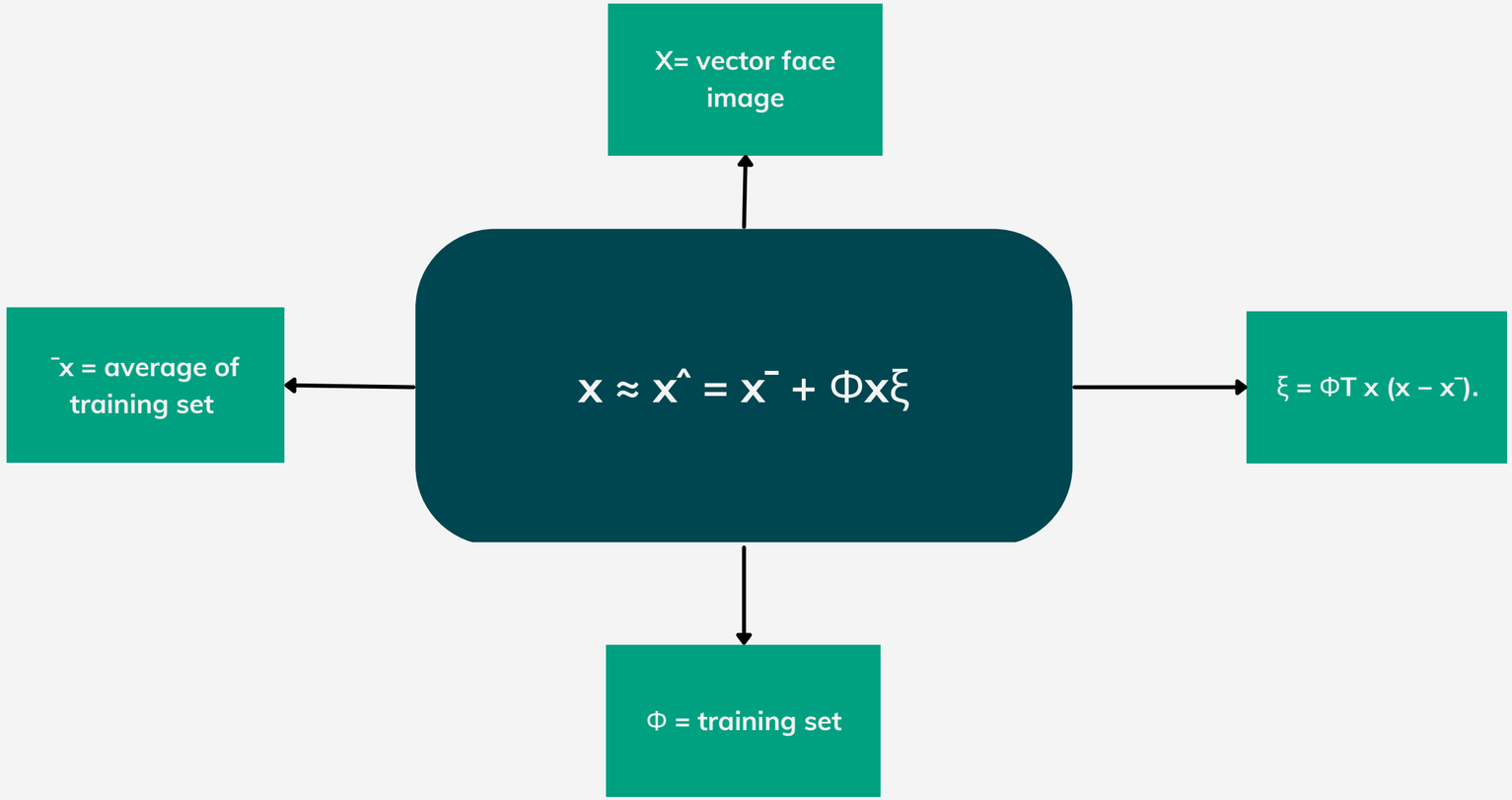
the image pattern describing the skin

Expression

muscular synthesis of emotions making the face look, for example, happy or sad

Eigenfaces

A statistical texture models:

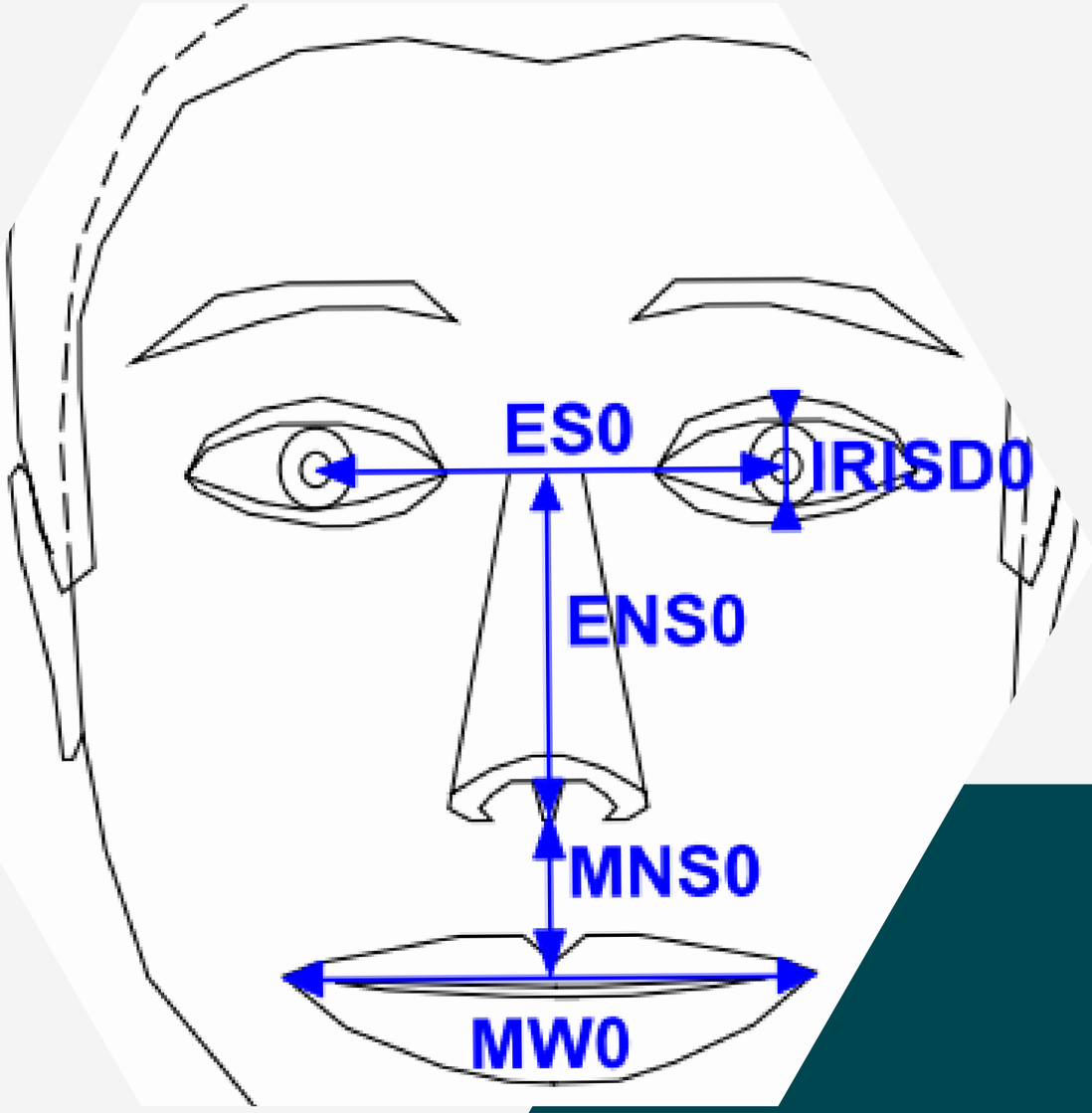


Facial action coding system (FACS)

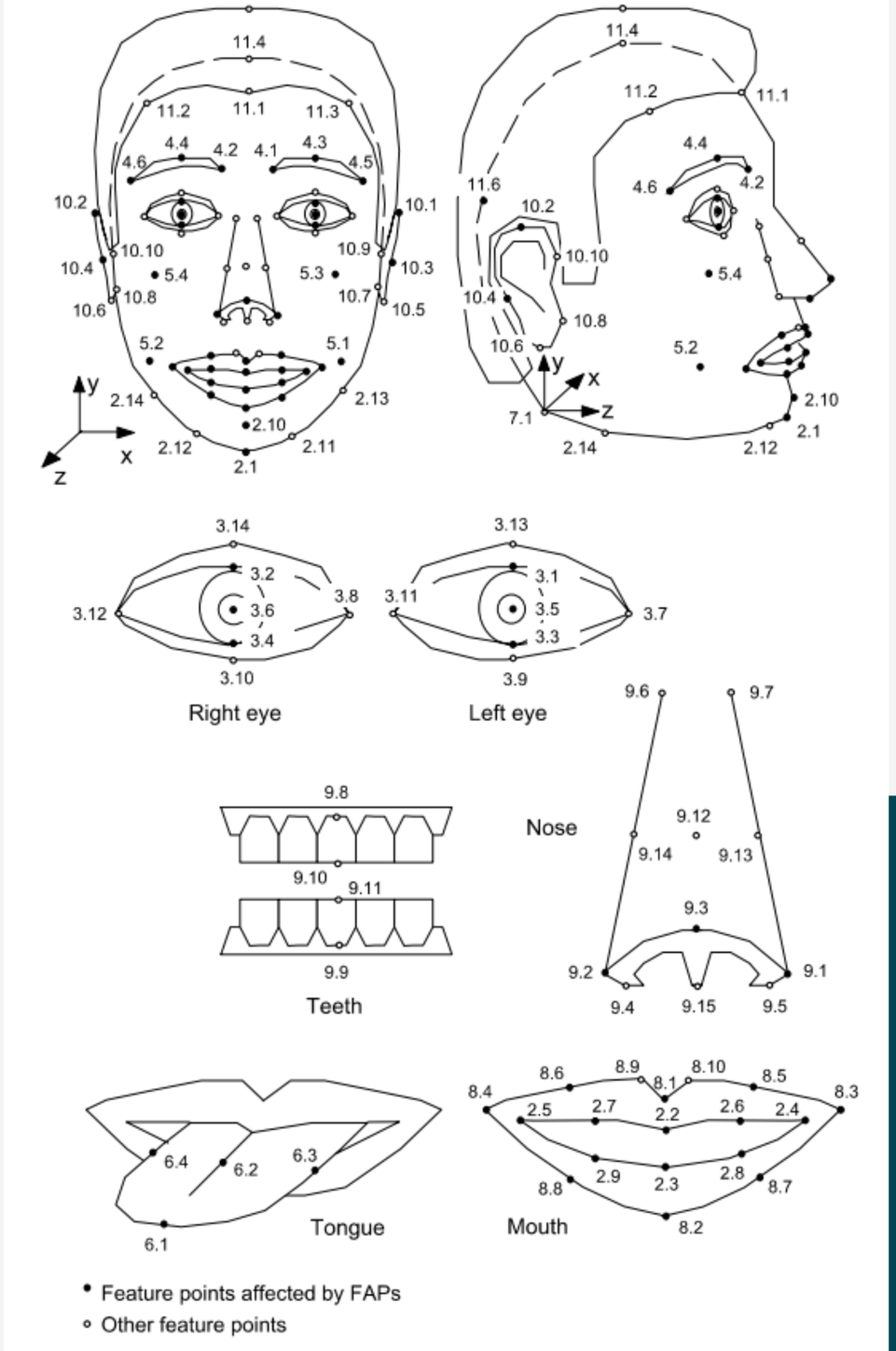
FACS describe a combination of around 50 action units (AUs), each AU represents the activation of one facial muscle

MPEG4

- MPEG-4 is an international standard for coding and representation of audiovisual object

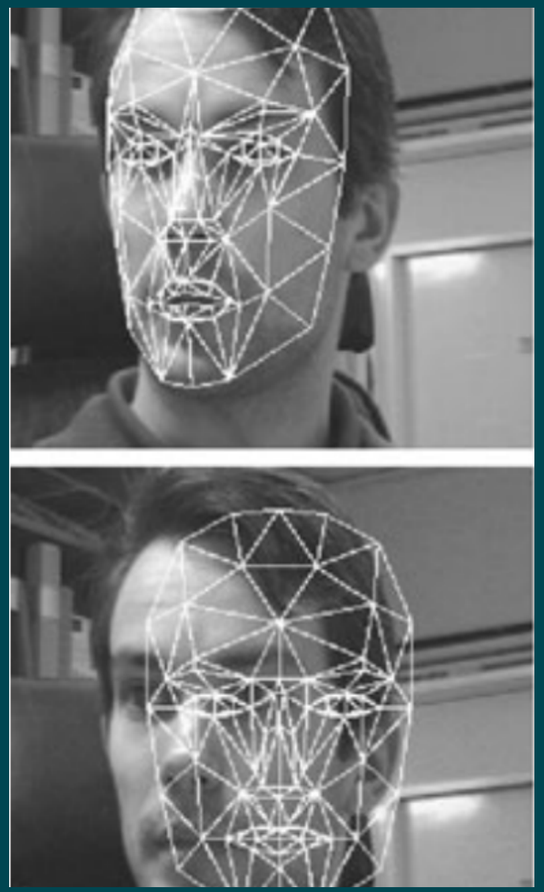


Facial feature point (FP)



Computer graphic model

When synthesizing face using computer graphics, the most common models is a wireframe model also known as polygonal mesh



Tracking strategies

Motion-Based vs. Model-Based Tracking.

Estimating displacements of pixels, while model-based trackers use a model to change pose parameters

Model-Based Tracking: First Frame Models vs. Pre-trained Models

uses a reference image of the object captured from the first frame to compensate for changes in scale and rotation.

Features-based tracking example.

Using pose and structure parameterization, EKF parameter estimation, and face detection to initialize the tracker and extract feature points.

Appearance-Based Tracking Example

Improving by feature/appearance-based tracker, Inverse Compositional Algorithm, and particle filter.

Fused Trackers

Motion- and model-based Tracking combination

Devision hybrid method combining adaptive and trained methods.

Appearance- and features-based tracking

Combining feature-based and appearance-based approaches for facial landmark localization.

Commercial available trackers

Face and facial feature tracking is used in commercial and end-user products.

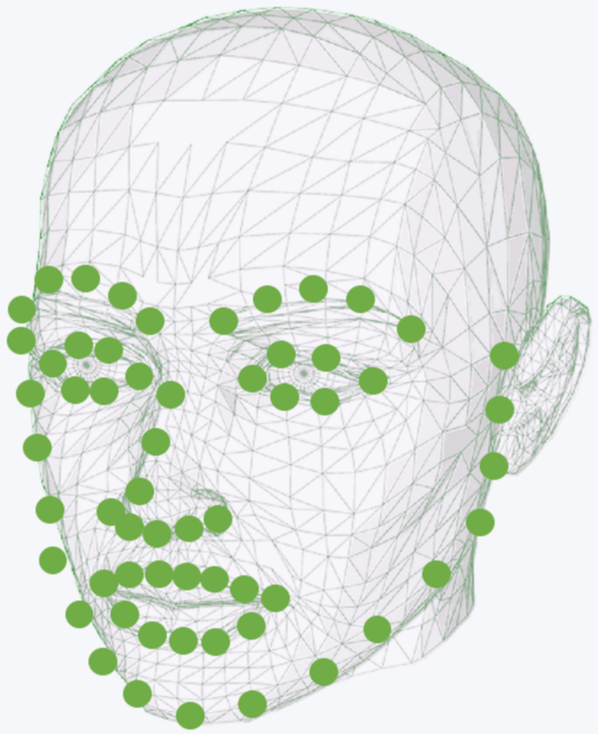
MediaPipe

OpenCV

mediapipe.solution

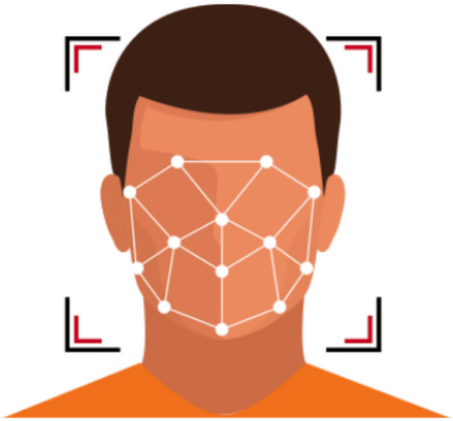
face_mesh()

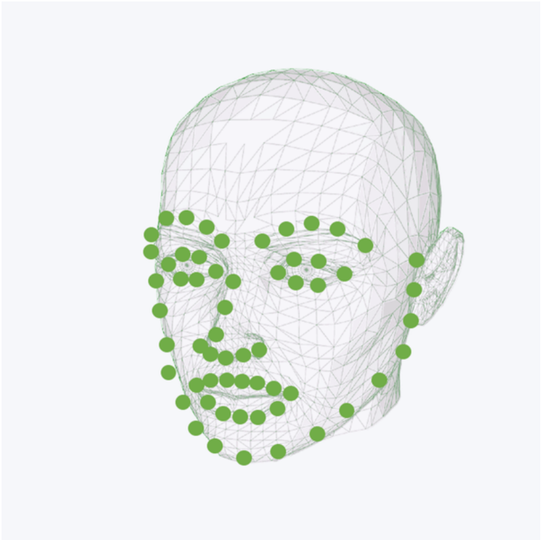
468 facial landmark



PIPELINE

Face detection → Face landmark





Step 1: Face Detection

BlazeFace: Sub-millisecond Neural Face Detection on Mobile GPUs

Valentin Bazarevsky Yury Kartynnik Andrey Vakunov Karthik Raveendran Matthias Grundmann
Google Research
1600 Amphitheatre Pkwy, Mountain View, CA 94043, USA
{valik, kartynnik, vakunov, krav, grundman}@google.com

arXiv:1907.05047v2 [cs.CV] 14 Jul 2019

Abstract

We present BlazeFace, a lightweight and well-performing face detector tailored for mobile GPU inference. It runs at a speed of 200–1000+ FPS on flagship devices. This super-realtime performance enables it to be applied to any augmented reality pipeline that requires an accurate facial region of interest as an input for task-specific models, such as 2D/3D facial keypoint or geometry estimation, facial features or expression classification, and face region segmentation. Our contributions include a lightweight feature extraction network inspired by, but distinct from MobileNetV1/V2, a GPU-friendly anchor scheme modified from Single Shot MultiBox Detector (SSD), and an improved tie resolution strategy alternative to non-maximum suppression.

1. Introduction

In recent years, a variety of architectural improvements in deep networks ([4, 6, 8]) have enabled real-time object detection. In mobile applications, this is usually the first step in a video processing pipeline, and is followed by task-specific components such as segmentation, tracking, or geometry inference. Therefore, it is imperative that the object detection model inference runs as fast as possible, preferably with the performance much higher than just the standard real-time benchmark.

We propose a new face detection framework called BlazeFace that is optimized for inference on mobile GPUs, adapted from the Single Shot Multibox Detector (SSD) framework [4]. Our main contributions are:

1. Related to the inference speed:
   - 1.1. A very compact feature extractor convolutional neural network related in structure to MobileNetV1/V2 [3, 9], designed specifically for lightweight object detection.
   - 1.2. A novel GPU-friendly anchor scheme modified from SSD [4], aimed at effective GPU utilization. *Anchors* [8], or *priors* in SSD terminology, are predefined static bounding boxes that serve as the basis for the adjustment by network predictions and determine the prediction granularity.
2. Related to the prediction quality: A tie resolution strategy alternative to non-maximum suppression [4, 6, 8] that achieves stabler, smoother tie resolution between overlapping predictions.

2. Face detection for AR pipelines

While the proposed framework is applicable to a variety of object detection tasks, in this paper we focus on detecting faces in a mobile phone camera viewfinder. We build separate models for the front-facing and rear-facing cameras owing to the different focal lengths and typical captured object sizes.

In addition to predicting axis-aligned face rectangles, our BlazeFace model produces 6 facial keypoint coordinates (for eye centers, ear tragions, mouth center, and nose tip) that allow us to estimate face rotation (roll angle). This enables passing a rotated face rectangle to later task-specific stages of the video processing pipeline, alleviating the requirement of significant translation and rotation invariance in subsequent processing steps (see Section 5).

3. Model architecture and design

BlazeFace model architecture is built around four important design considerations discussed below.

Enlarging the receptive field sizes. While most of the modern convolutional neural network architectures (including both MobileNet [3, 9] versions) tend to favor 3×3 convolution kernels everywhere along the model graph, we note that the depthwise separable convolution computations are dominated by their pointwise parts. On an $s \times s \times c$ input tensor, a $k \times k$ depthwise convolution involves s^2ck^2 multiply-add operations, while the subsequent 1×1 convolution into d output channels is comprised of s^2cd such operations, within a factor of d/k^2 of the depthwise part.

In practice, for instance, on an Apple iPhone X with the Metal Performance Shaders implementation [1], a 3×3

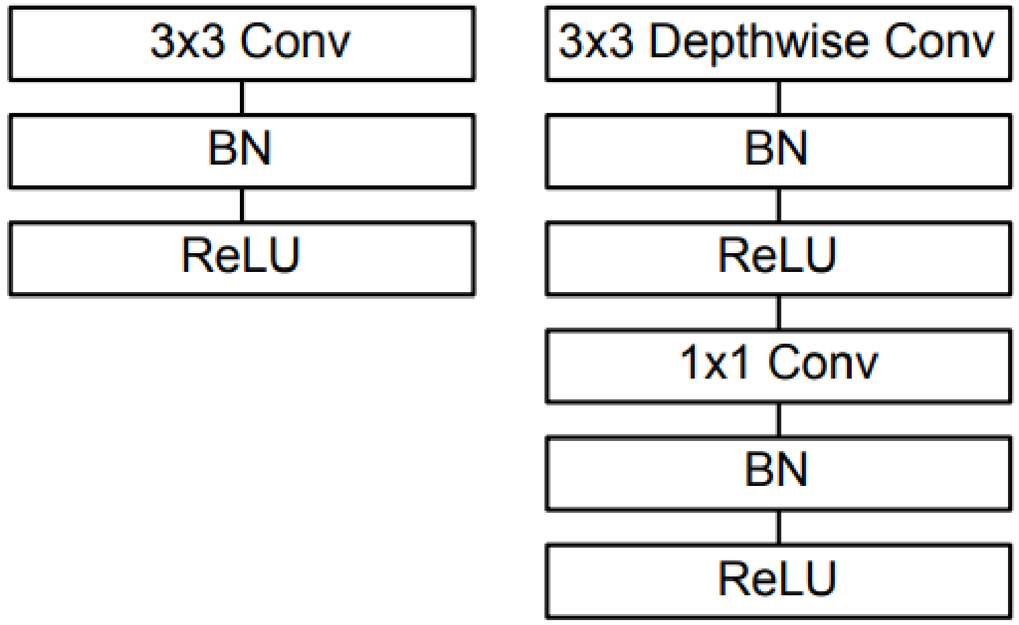
1

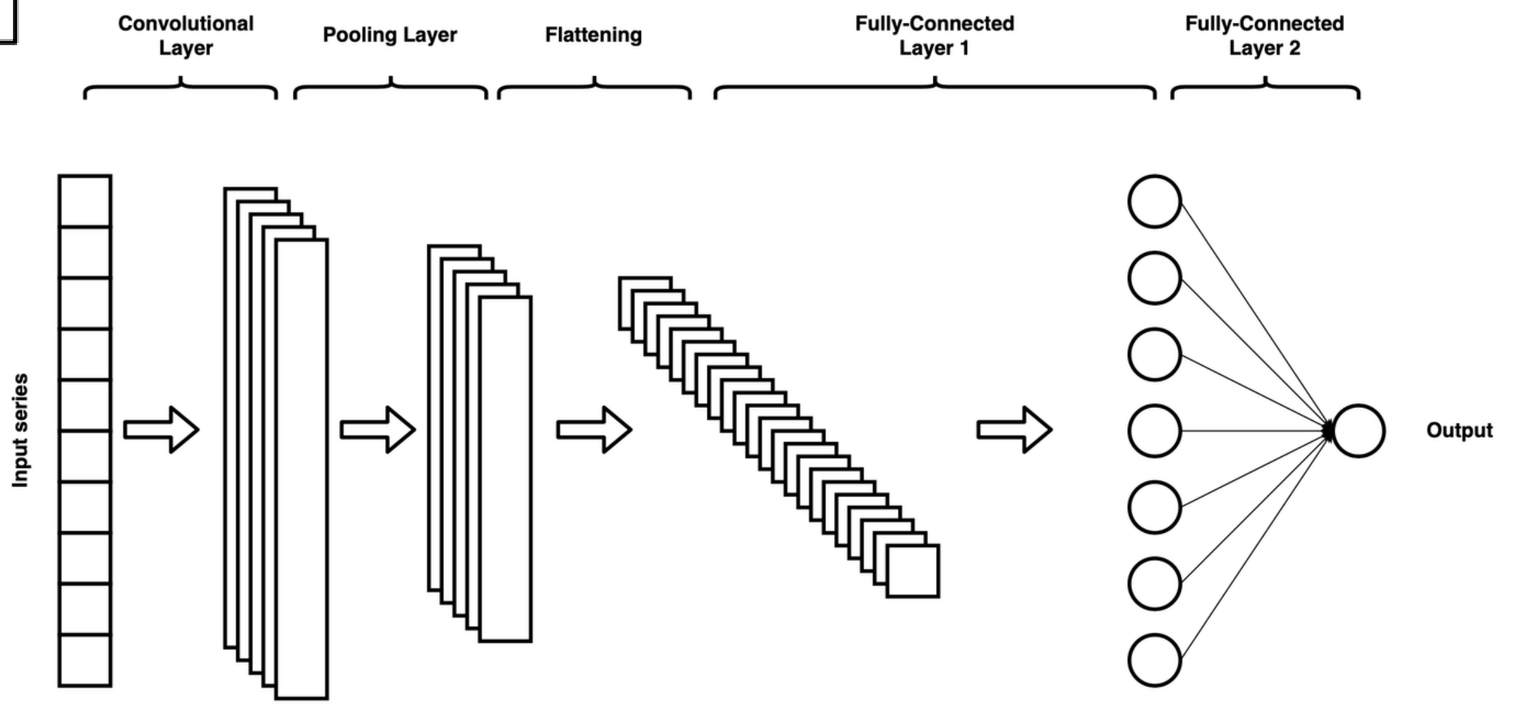
BlazeFace

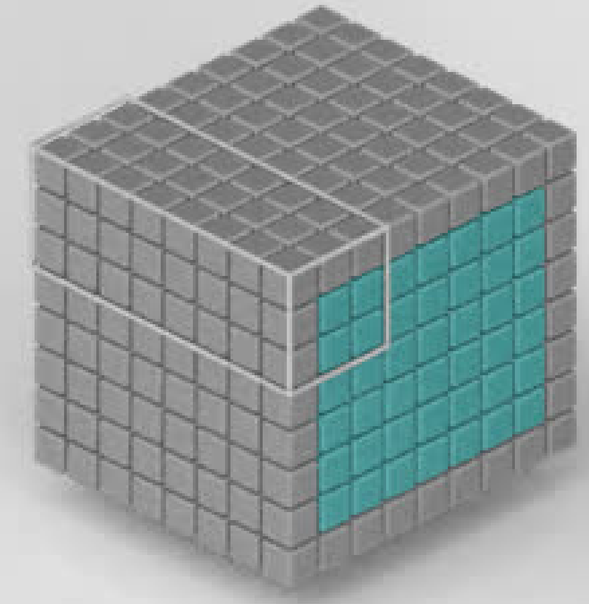
lightweight and well-performing face detector

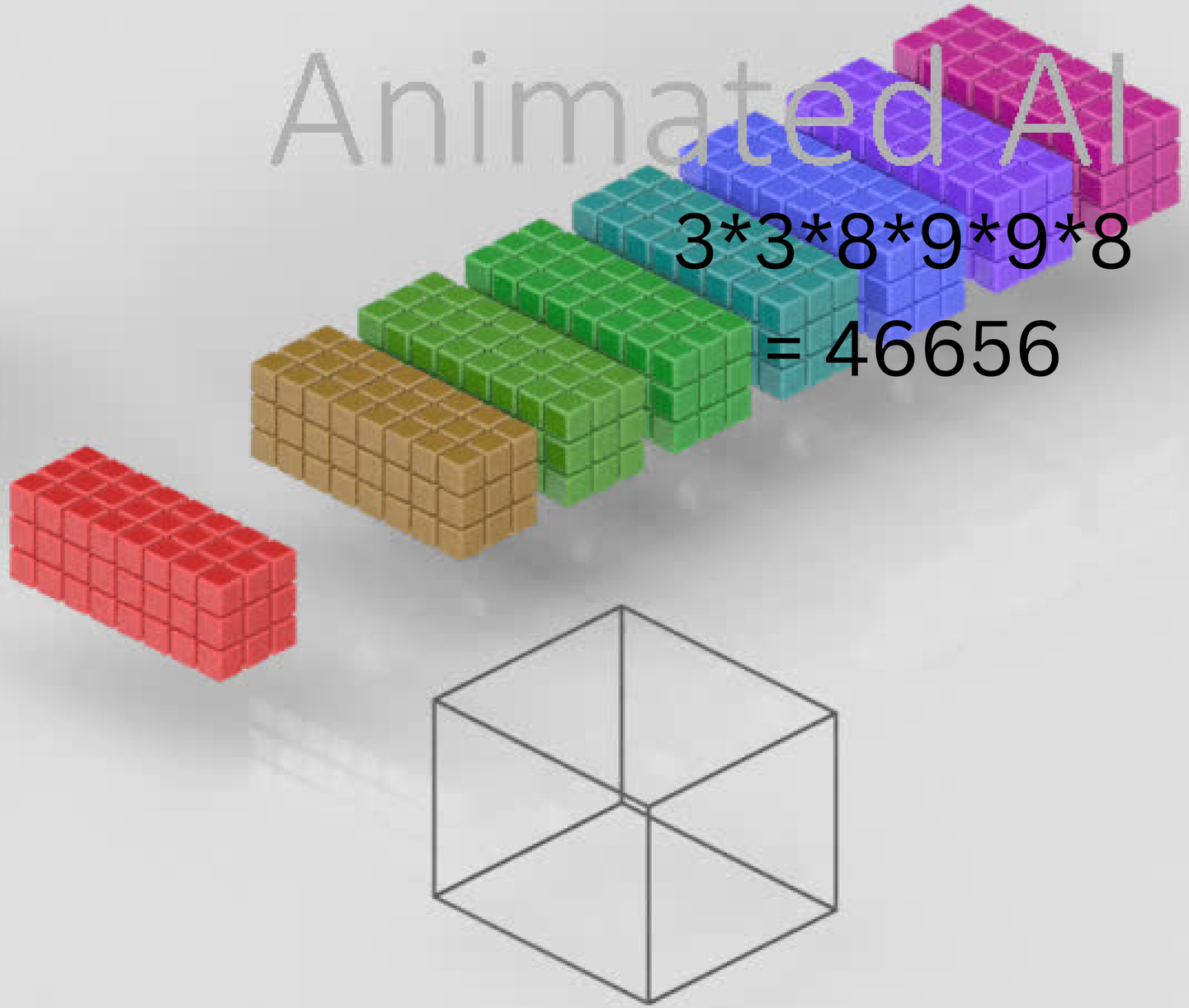
a speed of 200–1000+ FPS

MobiNet

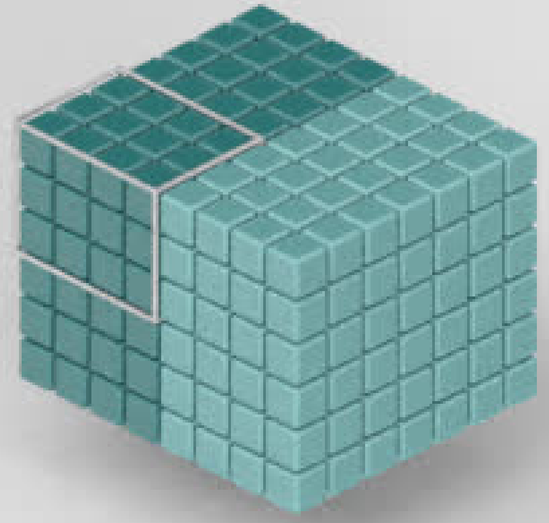

3x3 Conv
BN
ReLU
3x3 Depthwise Conv
BN
ReLU
1x1 Conv
BN
ReLU


Convolutional Layer
Pooling Layer
Flattening
Fully-Connected Layer 1
Fully-Connected Layer 2
Input series
Output

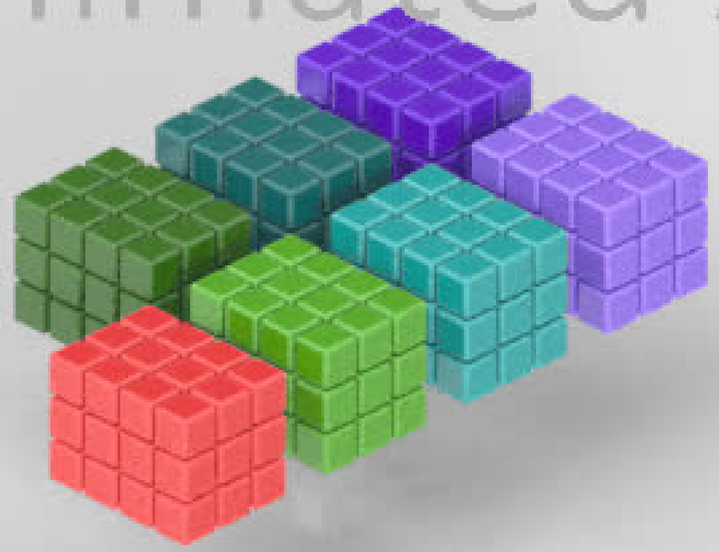


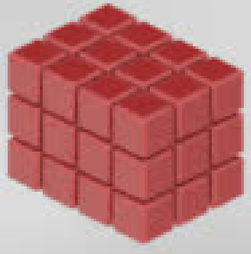


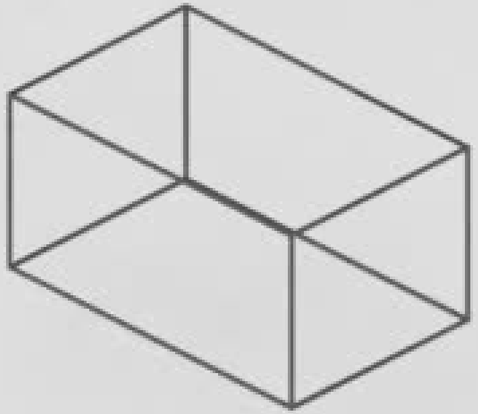
animatedai.github.io



Animated AI

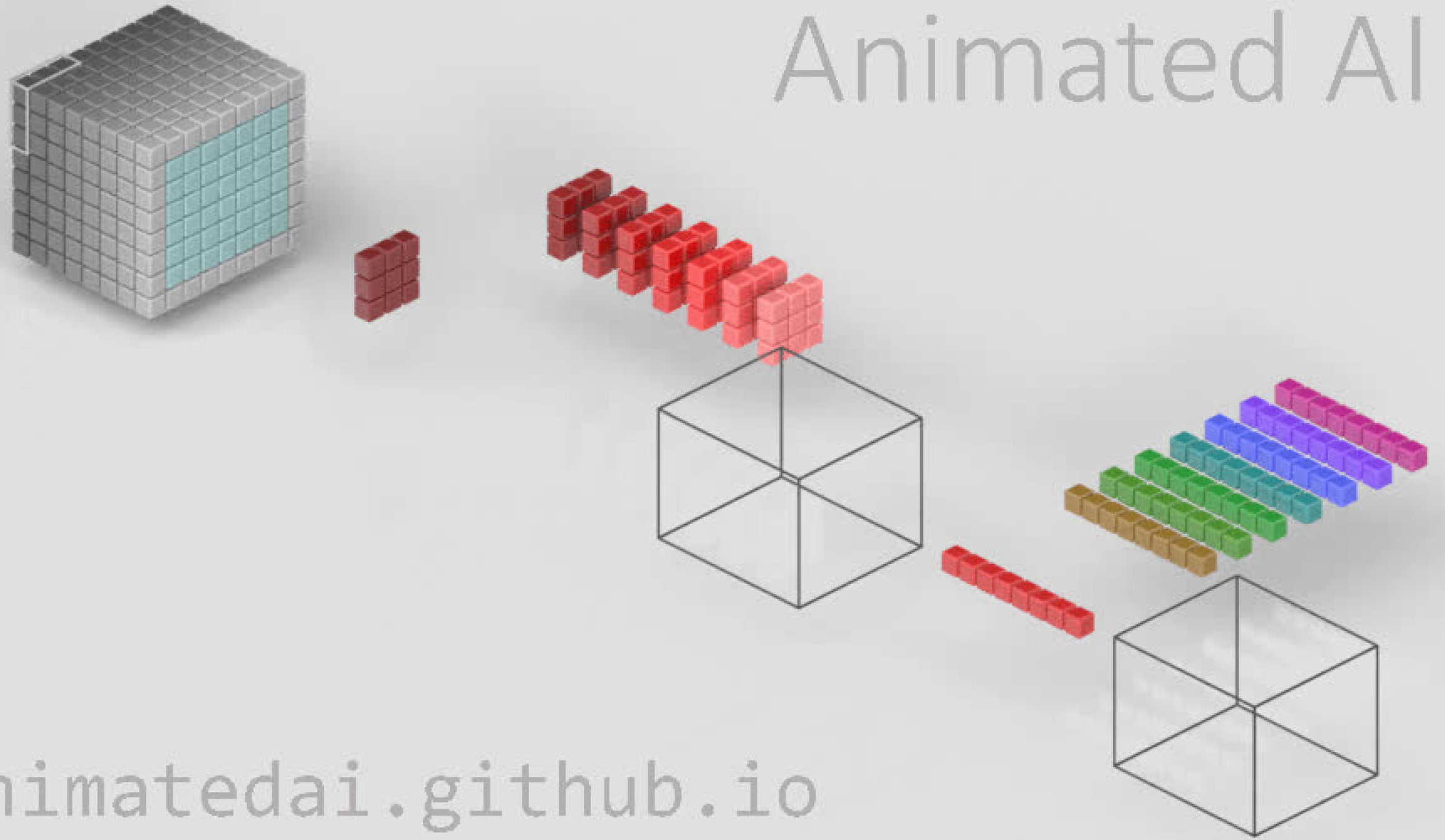






animatedai.github.io

Animated AI



animatedai.github.io

Step 2: Face landmark

arXiv:1907.06724v1 [cs.CV] 15 Jul 2019

Real-time Facial Surface Geometry from Monocular Video on Mobile GPUs

Yury Kartynnik Artsiom Ablavatski Ivan Grishchenko Matthias Grundmann
Google Research
1600 Amphitheatre Pkwy, Mountain View, CA 94043, USA
{kartynnik, artsiom, igrishchenko, grundman}@google.com

Abstract

We present an end-to-end neural network-based model for inferring an approximate 3D mesh representation of a human face from single camera input for AR applications. The relatively dense mesh model of 468 vertices is well-suited for face-based AR effects. The proposed model demonstrates super-realtime inference speed on mobile GPUs (100–1000+ FPS, depending on the device and model variant) and a high prediction quality that is comparable to the variance in manual annotations of the same image.

Figure 1: Face mesh prediction examples

1. Introduction

The problem of predicting the facial geometry by aligning a facial mesh template, also called *face alignment* or *face registration*, has for a long time been a cornerstone of computer vision. It is commonly posed in terms of locating relatively few (typically 68) landmarks, or keypoints. These points either have distinct semantics of their own or participate in meaningful facial contours. We refer the reader to [4] for a good review of related work on both the 2D and 3D face alignment problems.

An alternative approach is to estimate the pose, scale, and the parameters of a 3D morphable model (3DMM) [3]. A 3DMM, such as BFM2017, the 2017 edition of the Basel Face Model [7], is usually obtained through principal component analysis. The resulting mesh typically features many more points (around 50K in the case of BFM), but the range of possible predictions is limited by the linear manifold spanned by the PCA basis, which is in turn determined by the diversity of the set of faces captured for the model. As a concrete example, the BFM is seemingly incapable of reliably representing a face having exactly one eye closed.[1]

We set forth a problem of estimating positions of the 3D mesh vertices with a neural network, treating each vertex as an independent landmark. The mesh topology is comprised of 468 points arranged in fixed quads (see Figure 2a). The points have been manually selected in accordance with the supposed applications, such as expressive AR effects, virtual accessory and apparel try-on and makeup. The areas that are expected to have higher variability and higher importance in human perception have been allocated with higher point density. It allows to build a plausible smooth surface representation with the application of *e.g.* Catmull-Clark subdivision [6] (Figure 2b).

The input to the model is a frame (or, more generally, a stream of frames) of a single RGB camera—no depth sensor information is required. An example of the model output is presented in Figure 1. Our setup targets real-time mobile GPU inference, but we have also designed lighter versions of the model to address CPU inference on the mobile devices lacking proper GPU support. We will call the GPU-targeting model a "full" model, contrasting it to the "lightest" model tailored for CPU in the experiments.

2. Image processing pipeline

We organize the processing of an image as follows:

1. The whole frame from the camera input gets processed by a very lightweight face detector [2] that produces face bounding rectangles and several landmarks (*e.g.* eye centers, ear tragions, and nose tip). The landmarks are used to rotate a facial rectangle to align the line connecting

[1] Among the PCA expression basis components accounted for 97% of the model variance, the one that controls eye closing is acting symmetrically on both eyes.

1

arXiv:2006.10962v1 [cs.CV] 19 Jun 2020

Attention Mesh: High-fidelity Face Mesh Prediction in Real-time

Ivan Grishchenko Artsiom Ablavatski Yury Kartynnik Karthik Raveendran Matthias Grundmann
Google Research
1600 Amphitheatre Pkwy, Mountain View, CA 94043, USA
{igrishchenko, artsiom, kartynnik, krav, grundman}@google.com

Abstract

We present Attention Mesh, a lightweight architecture for 3D face mesh prediction that uses attention to semantically meaningful regions. Our neural network is designed for real-time on-device inference and runs at over 50 FPS on a Pixel 2 phone. Our solution enables applications like AR makeup, eye tracking and AR puppeteering that rely on highly accurate landmarks for eye and lips regions. Our main contribution is a unified network architecture that achieves the same accuracy on facial landmarks as a multi-stage cascaded approach, while being 30 percent faster.

Figure 1. Salient contours predicted by Attention Mesh submodels

1. Introduction

In this work, we address the problem of registering a detailed 3D mesh template to a human face on an image. This registered mesh can be used for the virtual try-on of lipstick or puppeteering of virtual avatars where the accuracy of lip and eye contours is critical to realism.

In contrast to methods that use a parametric model of the human face [1], we directly predict the positions of face mesh vertices in 3D. We base our architecture on earlier efforts in this field [5] that use a two stage architecture involving a face detector followed by a landmark regression network. However, using a single regression network for the entire face leads to degraded quality in regions that are perceptually more significant (*e.g.* lips, eyes).

One possible way to alleviate this issue is a cascaded approach: use the initial mesh prediction to produce tight crops around these regions and pass them to specialized networks to produce higher quality landmarks. While this directly addresses the problem of accuracy, it introduces performance issues, *e.g.* relatively large separate models that use the original image as input, and additional synchronization steps between the GPU and CPU that are very costly on mobile phones. In this paper, we show that it is possible for a single model to achieve the same quality as the cascaded approach by employing region-specific heads that transform the feature maps with spatial transformers [4], while being up to 30 percent faster during inference. We term this architecture as *attention mesh*. An added benefit is that it is easier to train and distribute since it is internally consistent compared to multiple disparate networks that are chained together.

We use an architecture similar to one described in [7], where the authors build a network that is robust to the initialization provided by different face detectors. Despite the differing goals of the two papers, it is interesting to note that both suggest that a combination of using spatial transformers with heads corresponding to salient face regions produces marked improvements over a single large network. We provide the details of our implementation for producing landmarks corresponding to eyes, irises, and lips, as well as quality and inference performance benchmarks.

2. Attention mesh

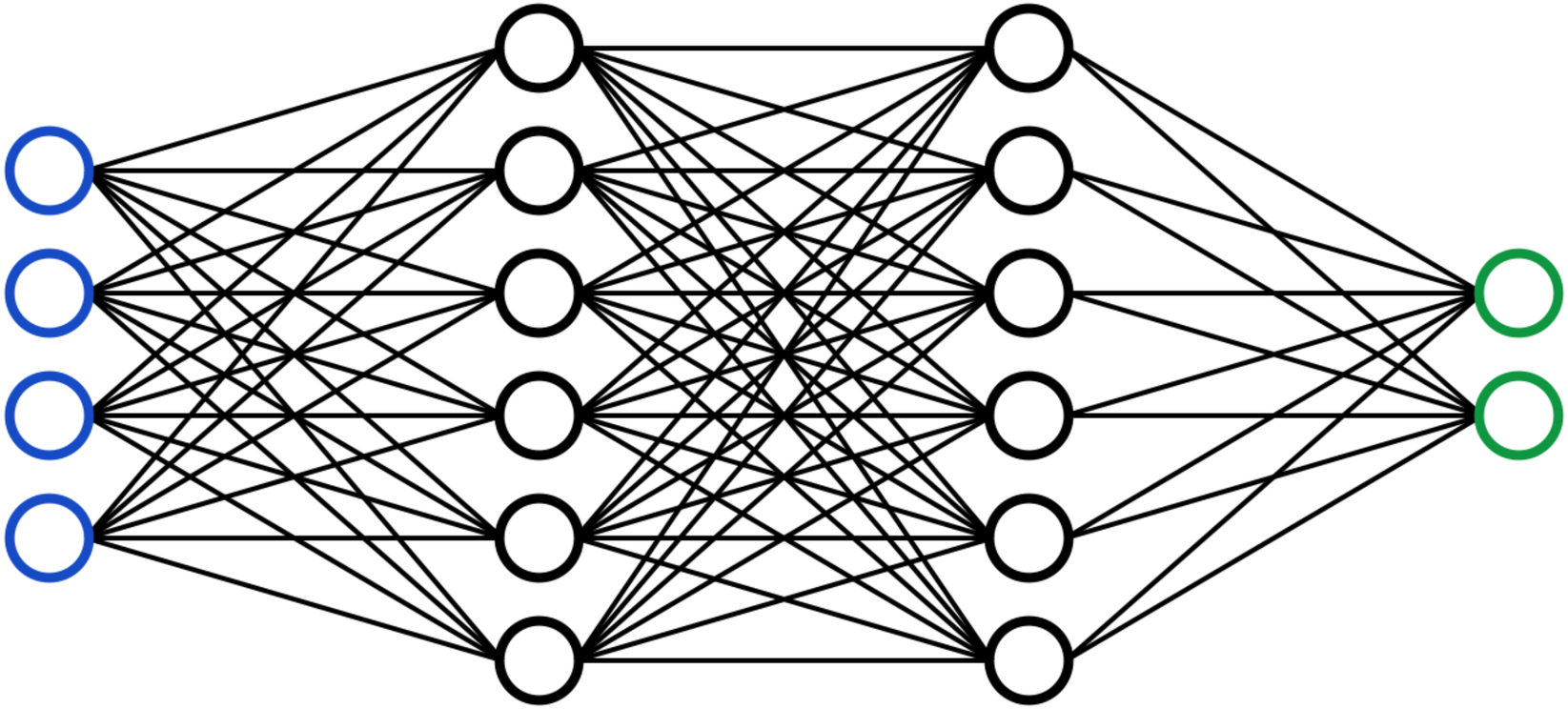
Model architecture The model accepts a 256×256 image as input. This image is provided by either the face detector or via tracking from a previous frame. After extract-

1

Real-time Facial Surface Geometry

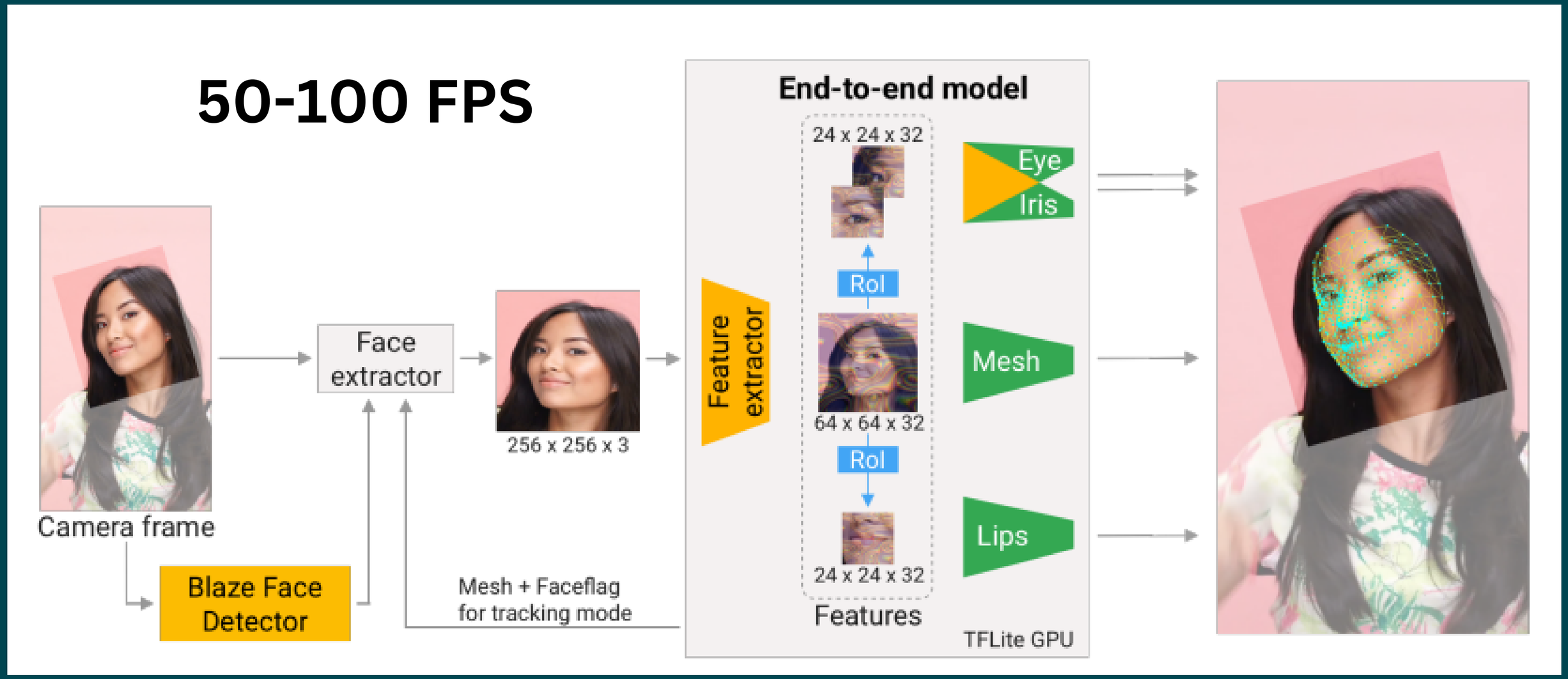
super-realtime inference speed on mobile GPUs (100–1000+ FPS)

128 x 128
input



468 facial landmark
output

Attention mesh pipeline



Datasets

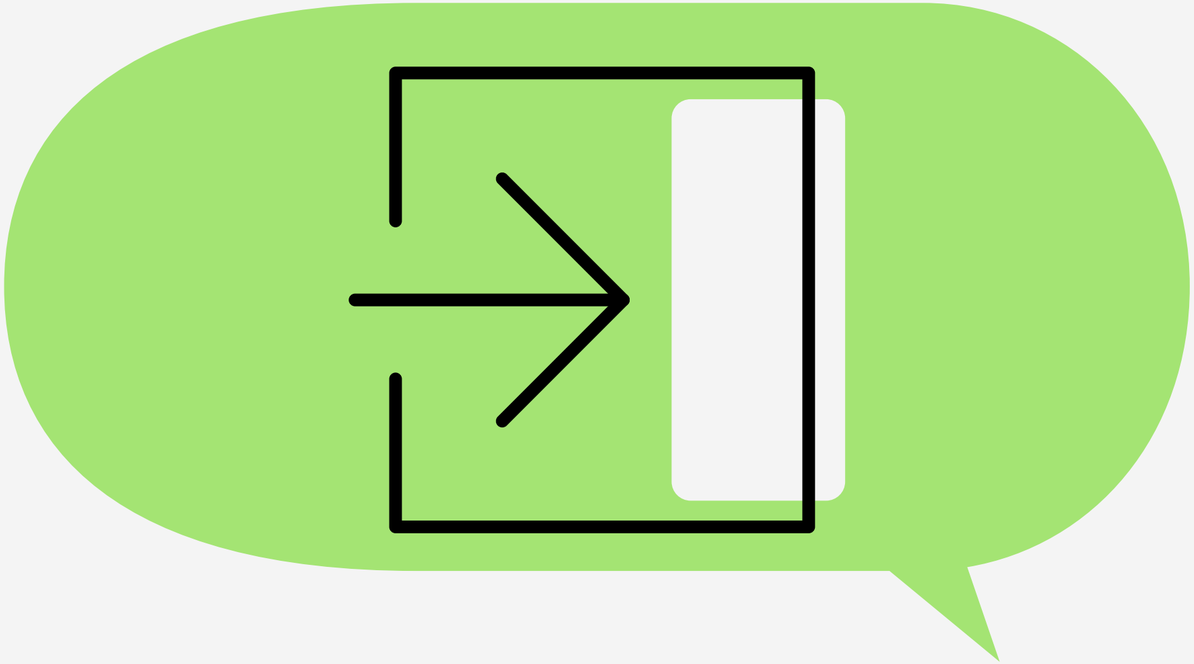
30.000 in-the-wild mobile camera photos taken with numerous camera sensors and in varied conditions

Result

Model	Inference Time (ms)
Mesh	8.82
Lips	4.18
Eye & iris	4.70
Cascade (sum of above)	22.4
Attention Mesh	**16.6**

Appilcation

Demo



OpenFace 2.0: Facial Behavior Analysis Toolkit

OpenFace 2.0

- Facial landmark detection and tracking
- Head pose estimation
- Eye gaze estimation
- Facial Action Recognition

Openface pipeline

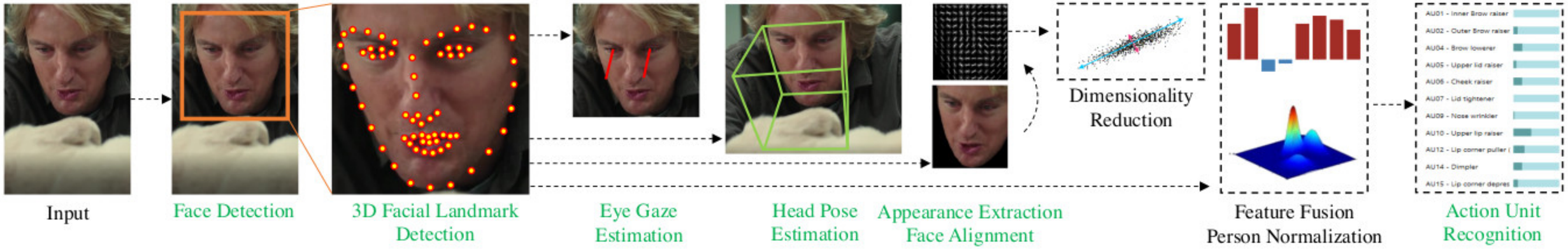


Fig 1: OpenFace 2.0 facial behavior analysis pipeline, including: landmark detection, head pose and eye gaze estimation, facial action unit recognition.

Facial landmark detection and tracking

Deep model simplification

Retrained patch experts using simpler models to reduce model size and improve speed.

Smart multiple hypotheses

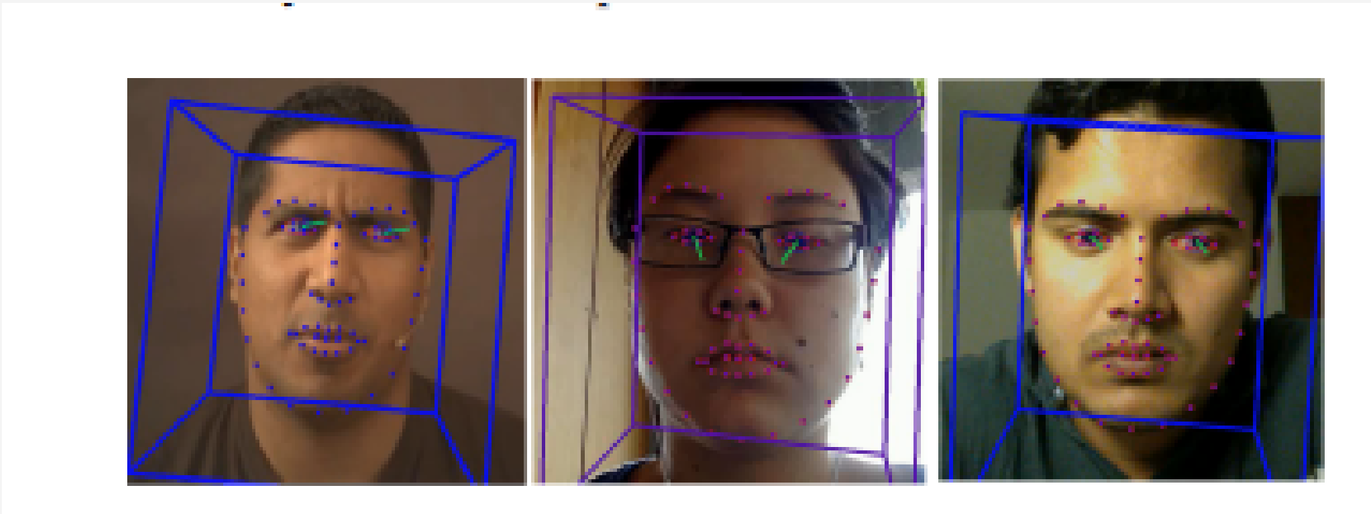
CE-CLM uses multiple initialization hypotheses to speed up landmark detection.

Sparse response maps

Sparse grid reduces model processing speed and accuracy

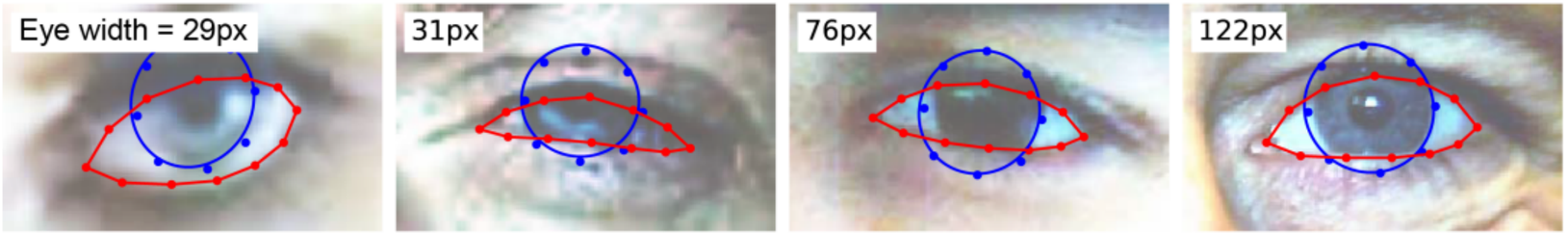
Head pose estimation

CE-CLM is able to accurately estimate head pose by using a 3D representation of facial landmarks and orthographic camera projection.



Eye gaze estimation

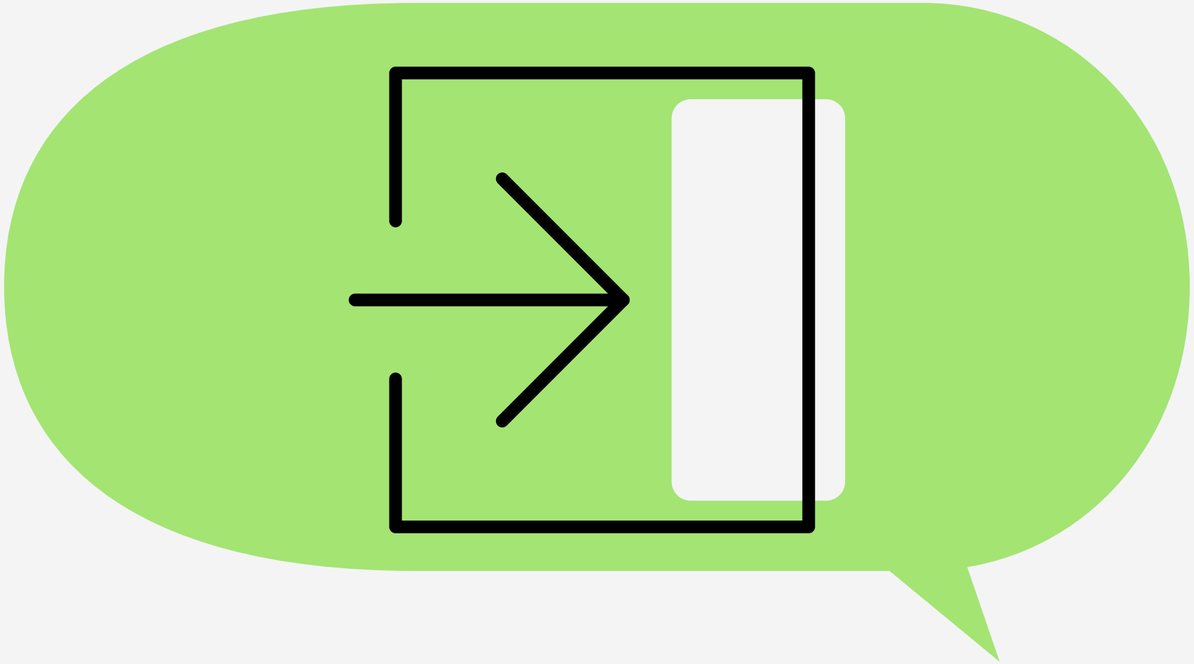
They use the detected pupil and eye location to compute the eye gaze vector individually for each eye.



Facial Action Unit Recognition

OpenFace 2.0 recognizes facial actions using linear kernel Support Vector Machines and person specific normalization and prediction correction.

Demo



THANK YOU FOR LISTENING

Answer questions