PGD-AI

Intelligent Interfaces

Rock – Paper – Scissors

APTS YOERI, FAZELIAN PADIN, TIMMERMANS BART

Introduction

- Our take on “Intelligent” Interfaces:
  - Do NOT use traditional user-interface devices
    - keyboard, mouse, …
  - But use:
    - Face recognition
    - Speech-commands recognition
    - Gesture recognition
    - …

➔ Motivation to:
- Create a real UI experience accessible to all
- Experiment with speech and video recognition

➔ Create a Browser-based rock-paper-scissors game

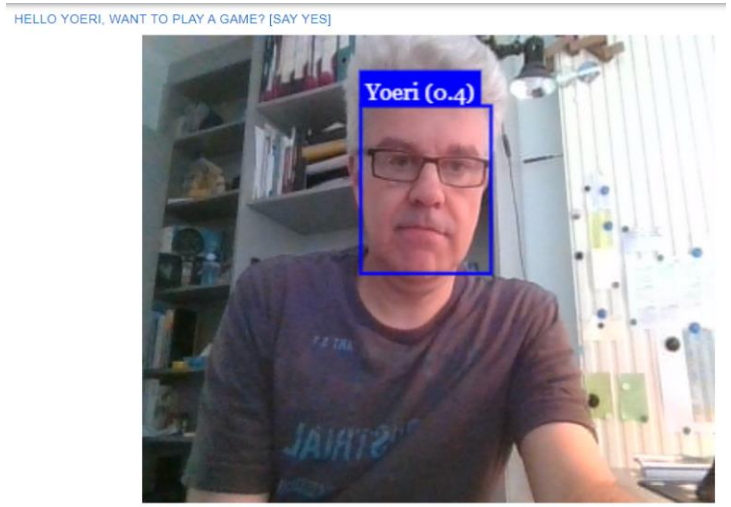
Application Architecture

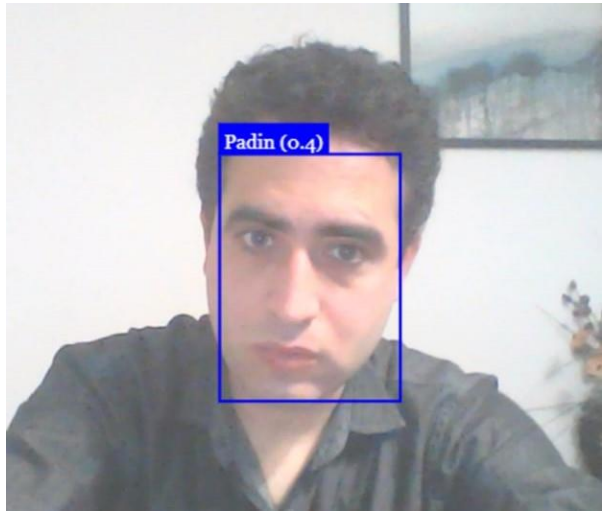
- Complete browser-based solution
  - Base app create with React and MUI
  - Video stream through react-webcam using mediaDevices.getUserMedia()
  - Audio streaming using the browser's mediaDevices.getUserMedia() and WebAudio API

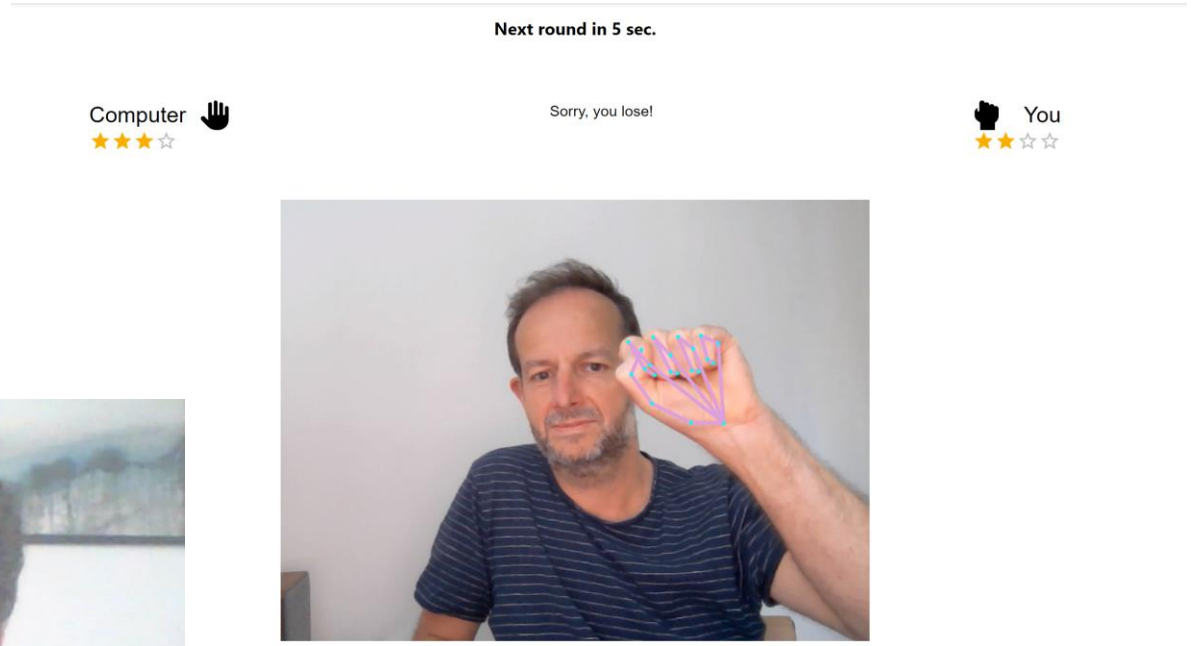
- Tensorflow.js
  - Develop and/or use ML models directly in the browser
  - Convert Python models to be used in the browser (not used)
  - Retrain existing models using transfer learning (not used)
  - Use WebGL for speeding up visual and speech recognition
  - Use predefined tensorflow.js models
    - Hand pose detection
    - Speech command recognition
  - Custom models for face recognition: face-api package

- Deployment using GitHub-pages

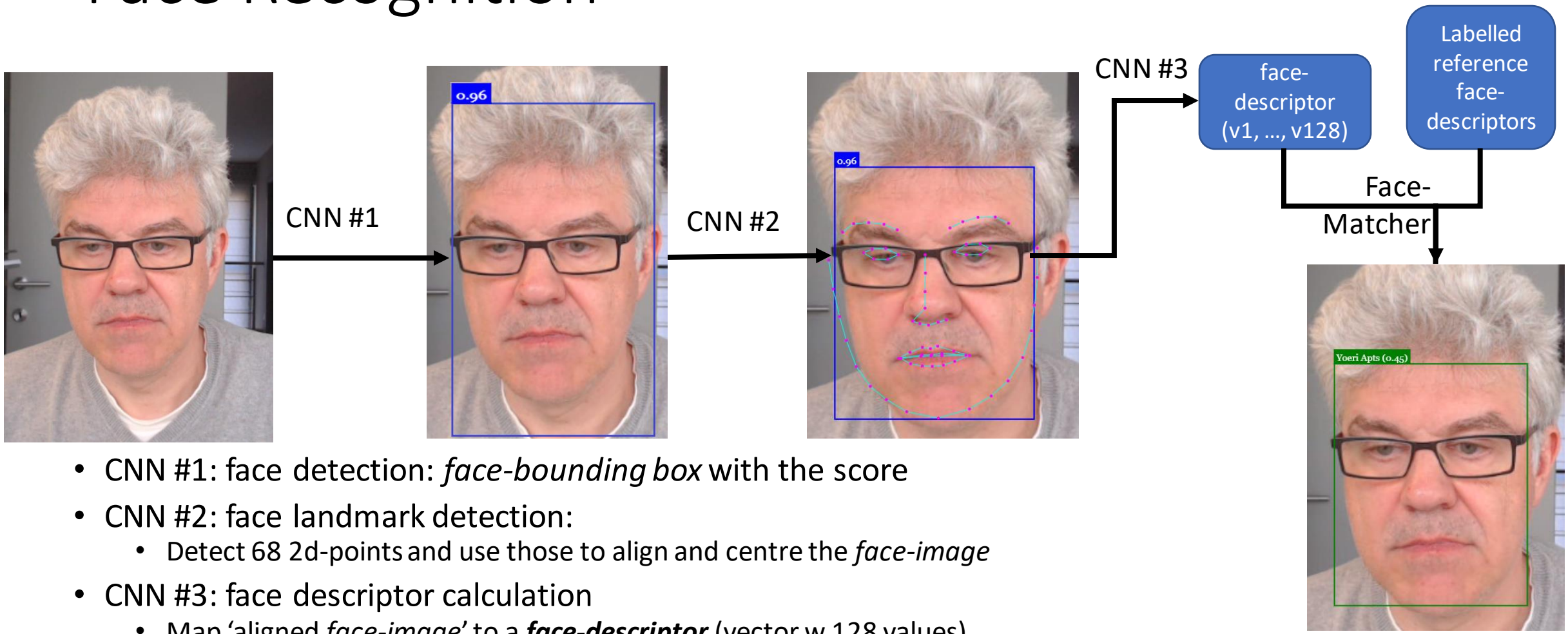
Demo


HELLO YOERI, WANT TO PLAY A GAME? [SAY YES]
Yoeri (0.4)


Padin (0.4)


Next round in 5 sec.
Computer
Sorry, you lose!
You

Face Recognition



- CNN #1: face detection: *face-bounding box* with the score
- CNN #2: face landmark detection:
  - Detect 68 2d-points and use those to align and centre the *face-image*
- CNN #3: face descriptor calculation
  - Map 'aligned *face-image*' to a ***face-descriptor*** (vector w 128 values)
- Face Matcher: Classifier; finds the shortest distance to labelled reference face-descriptors

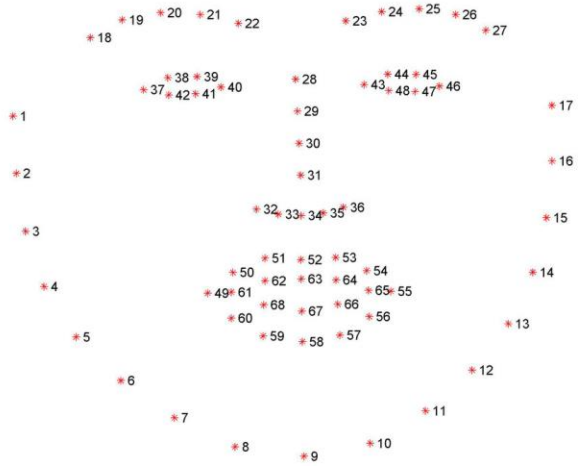
Face Recognition

- CNN #1: SSD MobileNet v1
  - Single Shot Multibox Detector
    - Input: full image
    - output: face-bounding box(es)
    - multiple convolutional layers, 1 fully connected layer
    - 4M trainable parameters
    - trained with WIDERFACE image dataset; 400K labelled faces in 32K pictures

- CNN #2: Face Landmark 68 CNN
    - input: face-bounding box
    - output: 68 x 2d-landmarks
    - multi-layer CNN (uses underlying C++ dlib implementation)

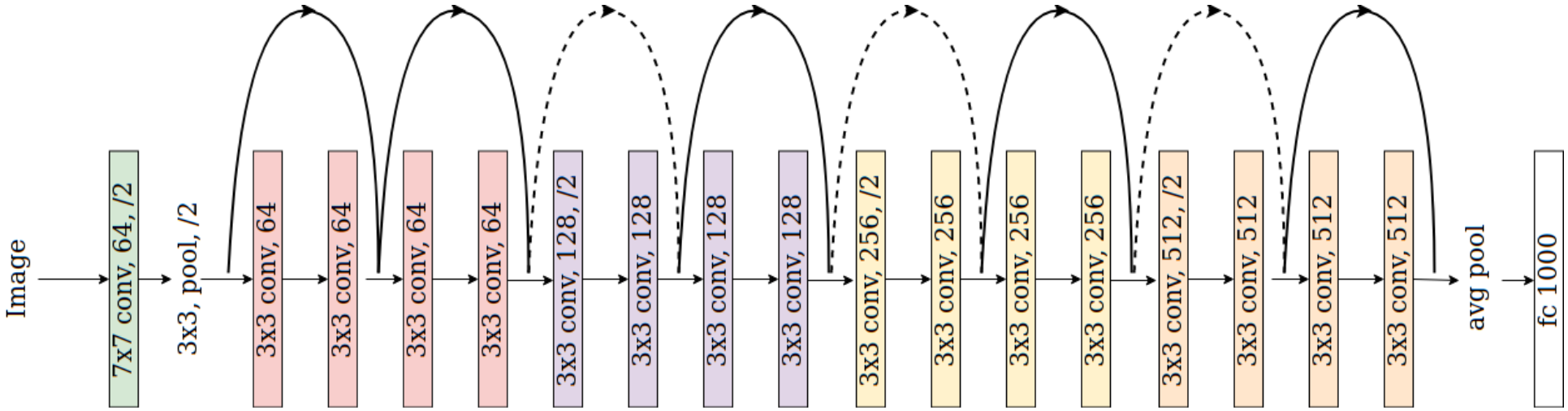
Table 1. MobileNet Body Architecture

Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
$5\times$ Conv dw / s1	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
$5\times$ Conv / s1	$1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$



Face Recognition

- CNN #3: ResNet-34 to calculate face descriptor
  - **Res**idual **Net**work with
    - **33** convolutional layers
    - **1** fully connected last layer
  - has been trained to learn to map the characteristics of a human face to a **face descriptor** (a feature vector with 128 values)
  - input: centered and aligned face image 150x150
  - output: face-descriptor (vector w 128 values)



Face Recognition

- Implementation:
  - Build with the face-api.js an open-source library
    - based on tensorflow.js, C++ dlib, …
  - OpenGL based GPU acceleration is used
  - Uses ‘Lightweight models’
    - Pro: allow real-time processing in the browser
    - Con: less accurate

Speech Command Recognition

- Keyword Spotting (KWS) vs. Speech Recognition
  - Runs usually natively on-device
  - Must be energy-efficient
  - Most of input is silence or background noise
  - Unit of recognition: single word
- Pretrained TensorFlow.js model working on WebAudioAPI
- Dataset
  - Limited vocabulary: digits 0 to 9, "Yes", "No", "Up", "Down", "Left", "Right", "Stop", and "Go"
  - Challenge: ignoring audio that does not contain speech
  - Size: 105,829 utterances of 35 words recorded by 2,618 crowdsource speakers

Speech Command Recognition

- Tensorflow.js model is based on a CNN Architecture proposed earlier by Google
- CNN preference over DNN in KWS
  - Deals with input typology
  - Shifts in frequency due to different speaking styles
  - Reducing parameters without sacrificing performance
- CNN Architecture
  - 2 convolutional, 1 linear low-rank, 1 DNN layer.

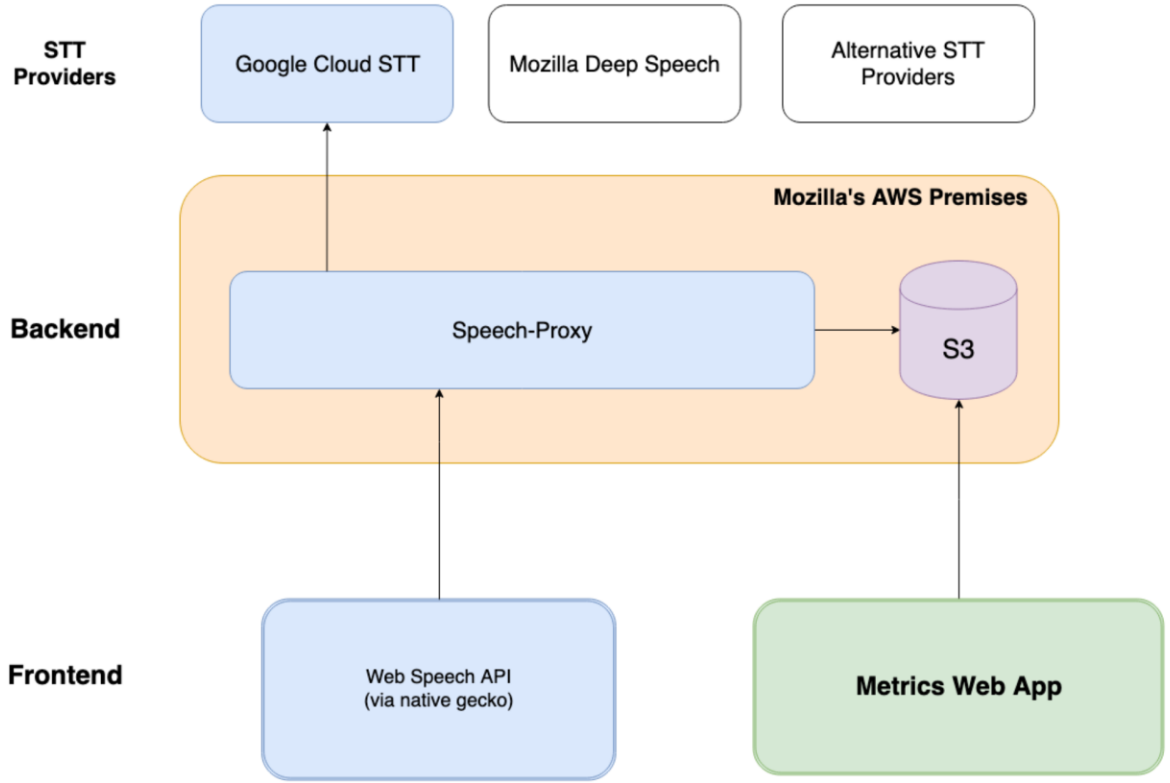
type	m	r	n	p	q	Par.	Mul.
conv	20	8	64	1	3	10.2K	4.4M
conv	10	4	64	1	1	164.8K	5.2M
lin	-	-	32	-	-	65.5K	65.5K
dnn	-	-	128	-	-	4.1K	4.1K
softmax	-	-	4	-	-	0.5K	0.5K
Total	-	-	-	-	-	244.2K	9.7M

Table 1: CNN Architecture for `cnn-trad-fpool3`

Speech Command Recognition

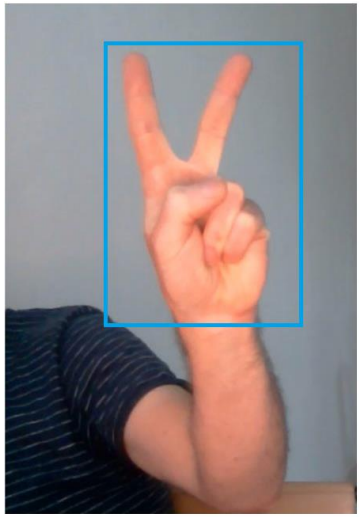
- Seems to be the least developed among TensorFlow.js models
  - Might be related to hold-out method (no k-fold cross-validation)

- Alternative: Web Speech API
  - React Speech Recognition
  - https://github.com/JamesBrill/

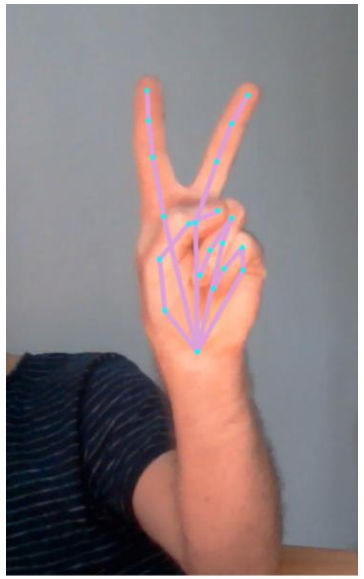


Gesture Recognition
The Pipeline

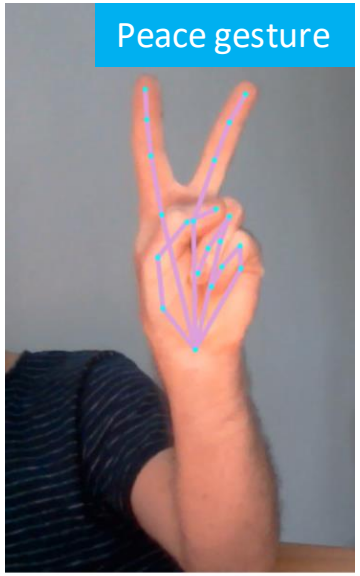
Video (image)



Palm detection



Landmark/ keypoint detection

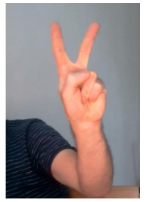


Gesture classification

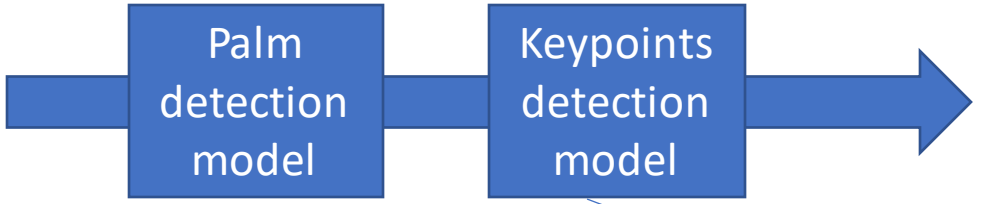
https://ai.googleblog.com/2019/08/on-device-real-time-hand-tracking-with.html

Gesture Recognition

- Handpose-detection model of tensorflow.js
  - Uses mediaPipe hands

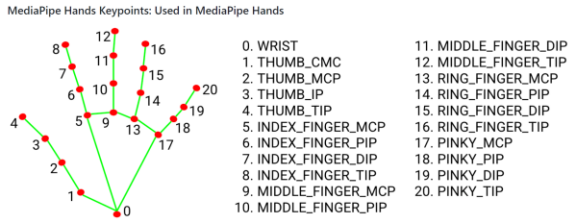


Input



CNN
Single shot detection (SSD)

CNN
Regression model (for direct coordinate prediction)



Output

```
[
  {
    score: 0.8,
    handedness: 'Right',
    keypoints: [
      {x: 105, y: 107, name: "wrist"},
      {x: 108, y: 160, name: "pinky_finger_tip"},
      ...
    ],
    keypoints3D: [
      {x: 0.00388, y: -0.0205, z: 0.0217, name: "wrist"},
      {x: -0.025138, y: -0.0255, z: -0.0051, name: "pinky_finger_tip"},
      ...
    ]
  }
]
```

Gesture Recognition

- Gesture classsification using fingerPose
  - https://github.com/andypotato/fingerpose
  - Simple gesture definition

```
const PaperGesture = new fp.GestureDescription('paper');
PaperGesture.addCurl(fp.Finger.Index, fp.FingerCurl.NoCurl);
PaperGesture.addCurl(fp.Finger.Middle,fp.FingerCurl.NoCurl);
PaperGesture.addCurl(fp.Finger.Ring,  fp.FingerCurl.NoCurl);
PaperGesture.addCurl(fp.Finger.Pinky, fp.FingerCurl.NoCurl);
PaperGesture.addCurl(fp.Finger.Thumb, fp.FingerCurl.NoCurl);
PaperGesture.addCurl(fp.Finger.Thumb, fp.FingerCurl.HalfCurl);
```

  - Use best match of several gesture estimations

Wrap-up / Lessons learned

- Other experiments:
  - Develop a Native app?
    - Tensorflow on Nvidia-GPU/Linux, OpenCV, GUI framework: Qt, …

    == a steep learning curve
- Lesson-learned:
  - Face-, speech-command-, gesture- recognition can today be used in browsers, ready for ‘real-time’, ‘real-world’ applications
  - Still heavy on CPU and GPU use

- Code repository (GitHub):
  - https://github.com/bartimer/pg-ai.git
- Deployed on (GitHub-pages) :
  - https://bartimer.github.io/pg-ai/#/intelligent-interfaces

References

- SSD: Single Shot MultiBox Detector
  - Paper: https://arxiv.org/abs/1512.02325
  - Code: https://github.com/bruceyang2012/Face-detection-with-mobilenet-ssd
- MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications
  - Paper: https://arxiv.org/abs/1704.04861
- Deep Residual Learning for Image Recognition
  - Microsoft Research
  - https://arxiv.org/pdf/1512.03385.pdf
- Face-api.js:
  - https://itnext.io/face-api-js-javascript-api-for-face-recognition-in-the-browser-with-tensorflow-js-bcc2a6c4cf07