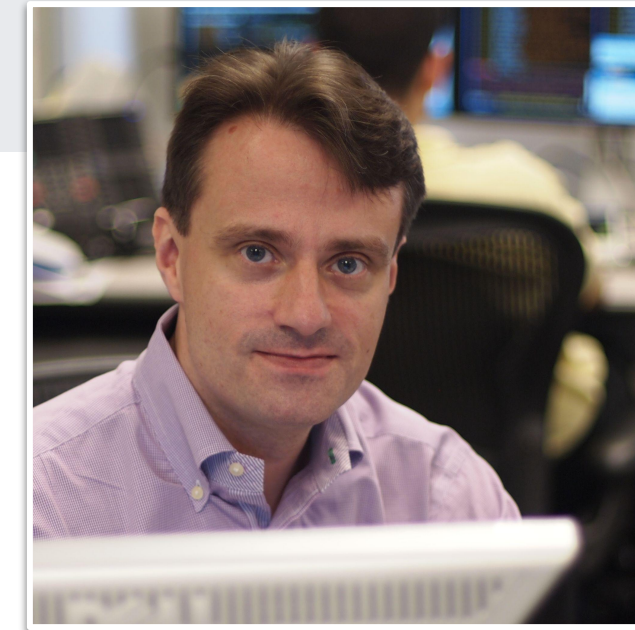


# MediaPipe with a bit of Bard

# About Me



- Google Developer Expert for Machine Learning and Deep Learning (2017-2022)
- Deep Learning R&D :
  - Language & Dialogue systems
  - Generative Models
  - Text-to-Speech
- MeetUp Co-organiser:
  - "Machine Learning Singapore"



Martin Andrews



# About Red Dragon AI



**RED DRAGON AI**

- Founded 2017
- Google Partner
- Consulting, Prototyping & Building
- Research - NeurIPS, EMNLP, COLING, NAACL
- Interactive Digital Personas

# MediaPipe with a bit of Bard

- What is MediaPipe?
- Live demos
- Working with Colab & Bard



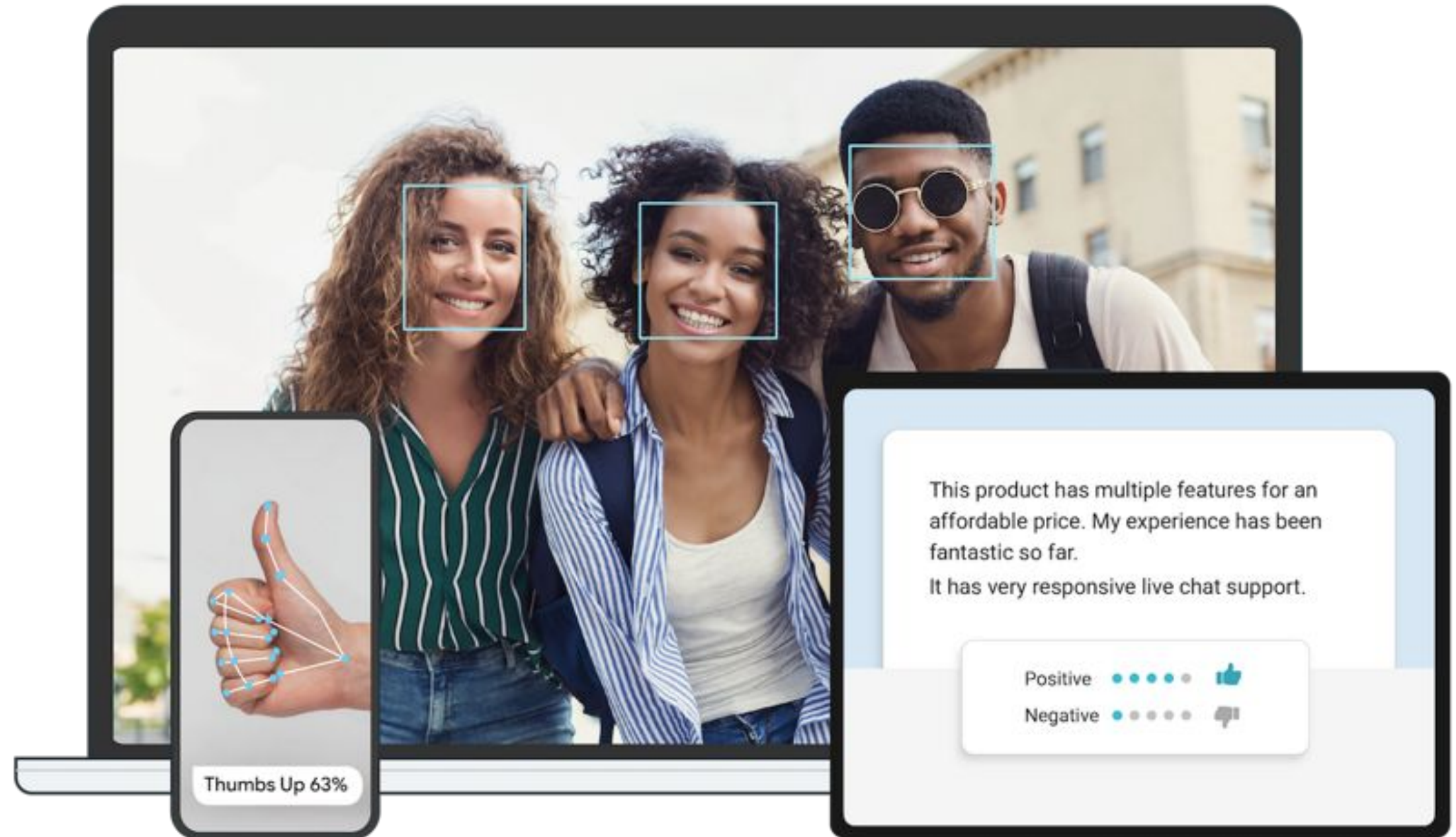
MediaPipe

# What is MediaPipe?

The ecosystem is coming together...

# On-Device Machine Learning

- Mobile (Android, iOS)
- Web
- Desktop
- Edge devices
- IoT



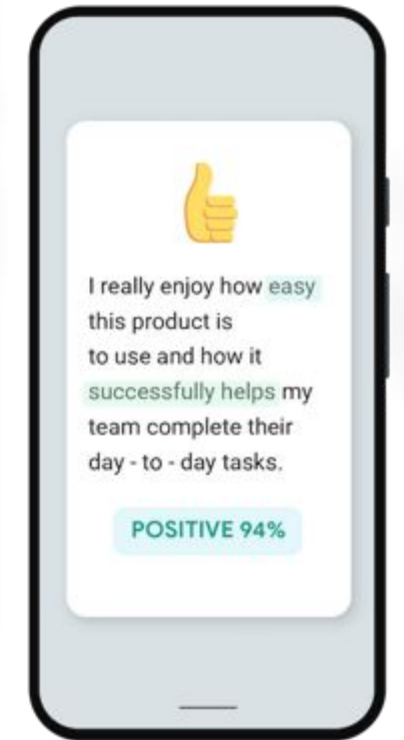
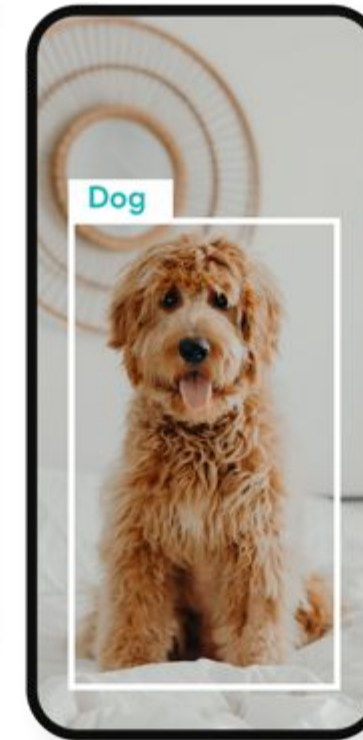
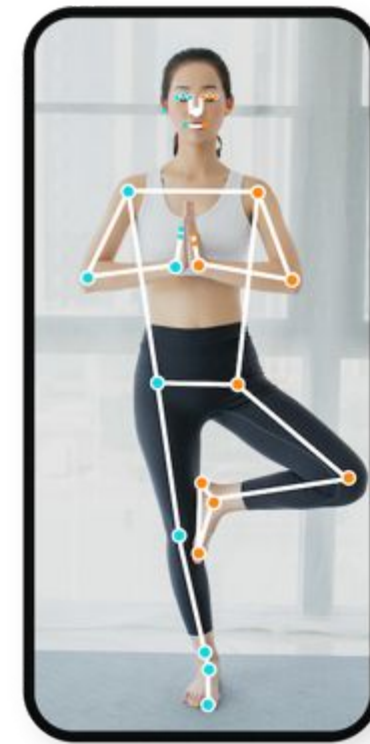


# On-Device Benefits

- No Cloud API calls required
- Data privacy for users
- Lower latency

# Big picture now coming together

- Ecosystem of compatible models
- Pipelined together into tasks
- Use existing components
  - or customise piecewise...





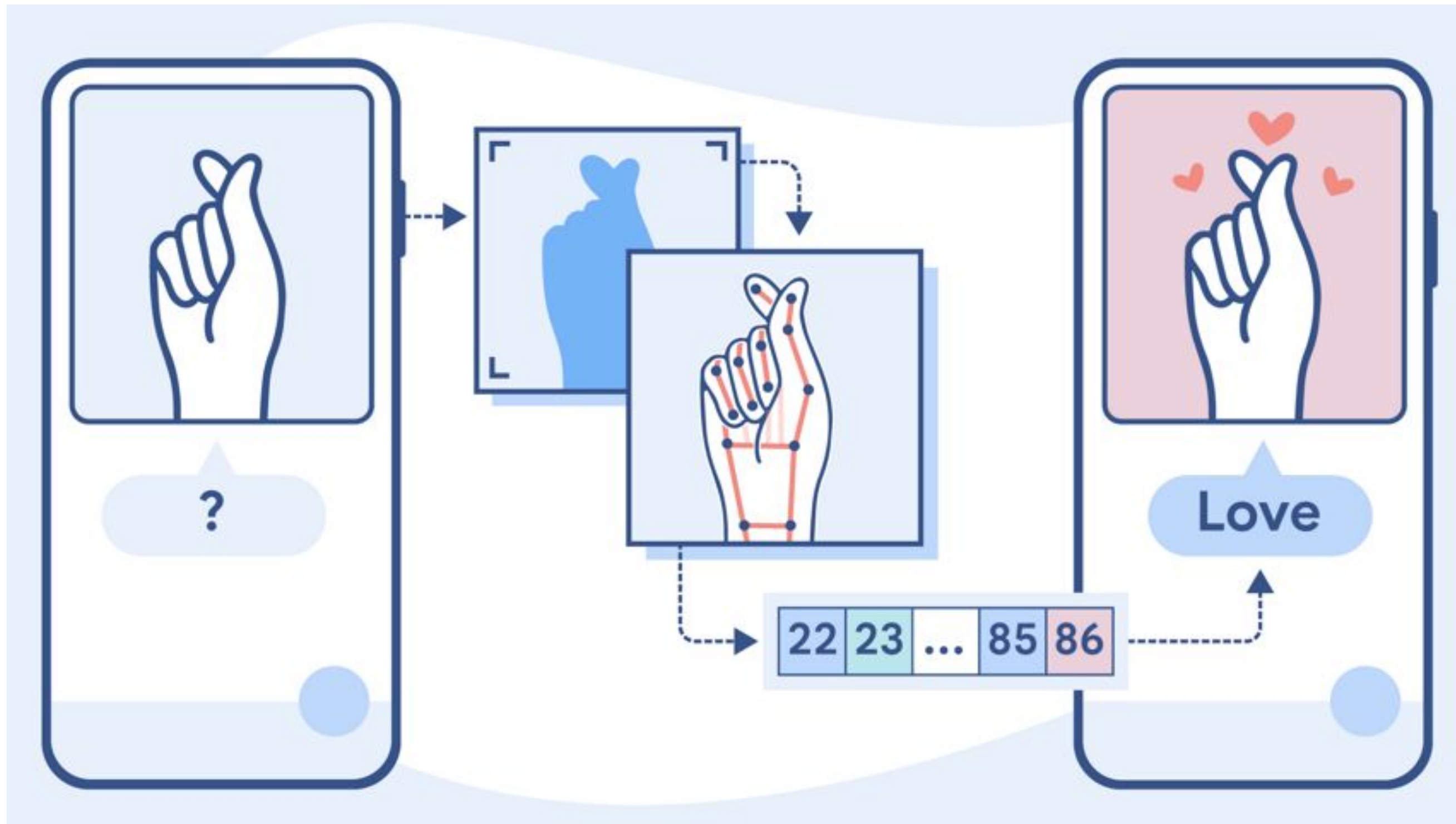
MediaPipe

Example Task = Gesture Recognition

Thumbs up 63%




## Example Task = Pipeline of Models




# MediaPipe

## VISION



**Object Detection**  
Track and label objects in images.

[See demo](#) [Customize](#)



**Image Classification**  
Identify content in images.

[See demo](#) [Customize](#)



**Image Segmentation**  
Locate objects and create image masks with labels.

[See demo](#)



**Interactive Segmentation**  
Segment the object of interest in an image.

[See demo](#)




**Gesture Recognition**  
Identify and recognize hand gestures.

[See demo](#) [Customize](#)



**Hand Landmark Detection**  
Detect hand landmarks.

[See demo](#)



**Image Embedding**  
Convert images into embedding vectors.

[See demo](#)




**Face Detection**  
Detect faces in real time.

[See demo](#)



**Face Landmark Detection**  
Detect face landmarks and blendshape scores in real time.


[See demo](#)



**Pose Landmark Detection**  
Identify key points on the body in real time.


[See demo](#)

## TEXT



**Text Classification**  
Classify text into relevant tags.

[See demo](#) [Customize](#)



**Text Embedding**  
Convert text into an embedding vector.


[See demo](#)



**Language Detection**  
Identify the language of a given text.

[See demo](#)

## AUDIO



**Audio Classification**  
Identify sounds in audio clips.

[See demo](#) [Customize](#)

MediaPipe

# Demo Time!

Extreme Confidence Mode : <https://mediapipe-studio.webapps.google.com/home>

Demo

# Audio Classification

- ( Using the UI )

## Audio Classification

Categorize audio clips based on a defined set of classes. The default model, Yamnet, was trained on the [AudioSet](#) dataset to predict 521 classes, such as speech, music, bird chirp, and waves. For more information on labels, performance, etc., see the [documentation](#).

The sample parameters below can be changed. See [documentation](#) for more details

Inference delegate: GPU inference

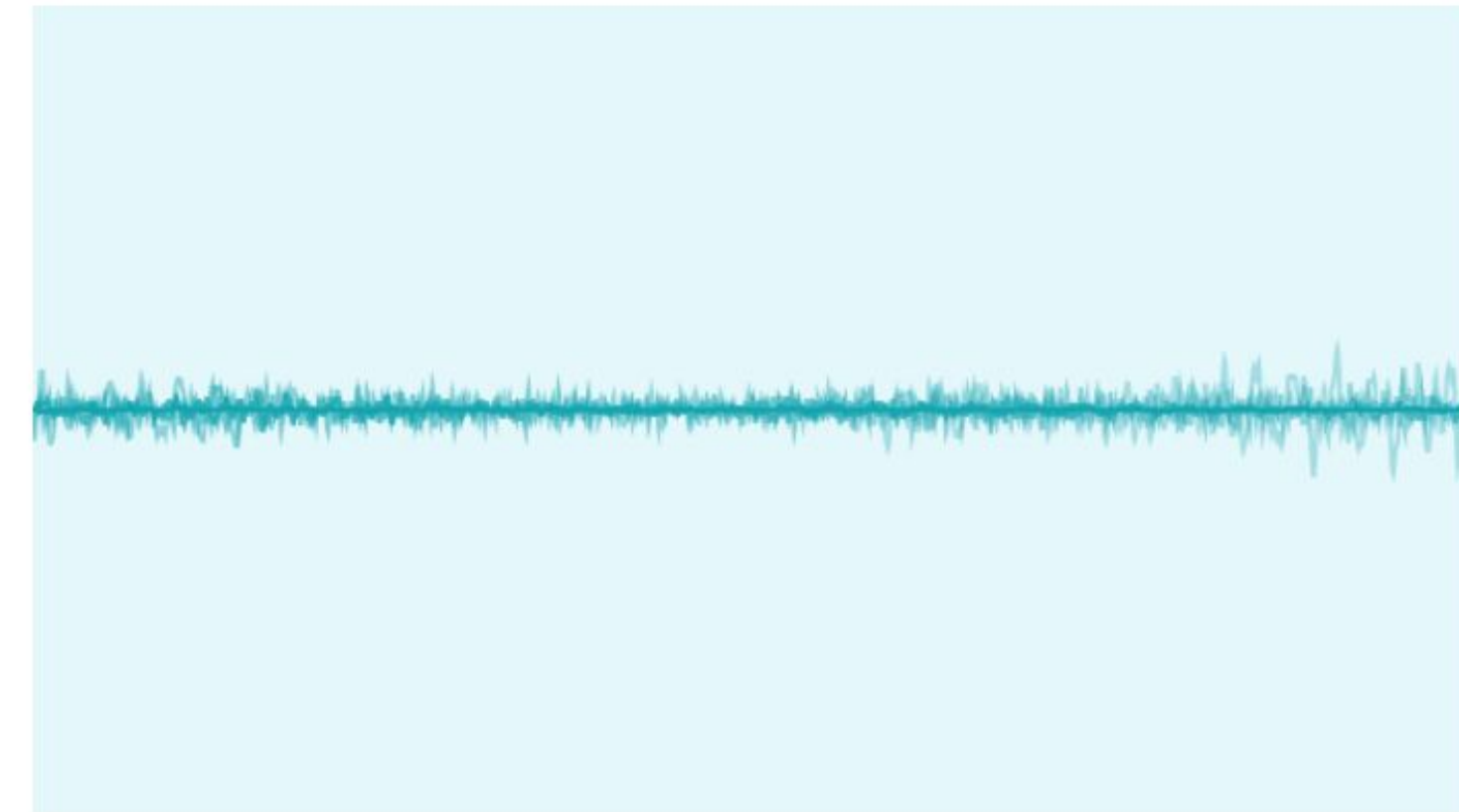
Model selections: Yamnet audio classifier

Display language: en

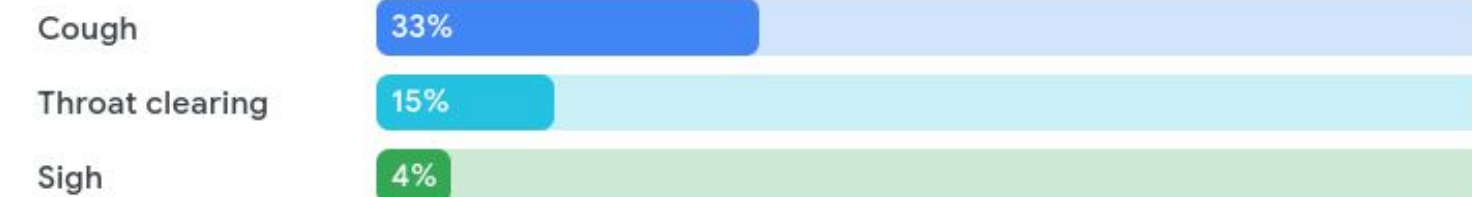
Max results: 1  10

Score threshold: 0%  99%

Input Default



Inference time (ms): 6.6



Demo

# Facial Landmarks

- ( Using the Web )

The screenshot displays the 'MediaPipe Face Landmarker Task for web' interface. At the top, there are navigation buttons for 'Settings', 'Sign Up', and 'Log In'. Below the navigation bar, there are three code editors for HTML, CSS, and JS (TypeScript). The HTML editor shows a copyright notice and a license link. The CSS editor shows a similar copyright notice. The JS editor shows the import statement for the vision module. Below the code editors, there is a video of Barack Obama with a white wireframe overlay on his face. The landmarks are color-coded: red for the eyes, green for the eyebrows, and white for the rest of the face. To the right of the video, there is a list of landmark names and their corresponding values, with horizontal bars indicating the magnitude of each value. The landmarks include: \_neutral (0.0000), browDownLeft (0.0005), browDownRight (0.0073), browInnerUp (0.0003), browOuterUpLeft (0.0003), browOuterUpRight (0.0003), cheekPuff (0.0013), cheekSquintLeft (0.0000), cheekSquintRight (0.0000), eyeBlinkLeft (0.0097), eyeBlinkRight (0.0005), eyeLookDownLeft (0.0000), eyeLookDownRight (0.0000), eyeLookInLeft (0.0288), eyeLookInRight (0.0065), eyeLookOutLeft (0.0000), eyeLookOutRight (0.0627), eyeLookUpLeft (0.0836), eyeLookUpRight (0.0942), eyeSquintLeft (0.0000), eyeSquintRight (0.0000), eyeWideLeft (0.0029), eyeWideRight (0.0043), jawForward (0.0024), jawLeft (0.0016), jawOpen (0.0067), jawRight (0.0002), mouthClose (0.0303), mouthDimpleLeft (0.0178), and mouthDimpleRight (0.0155). At the bottom right, the date 'Saturday 27 May 2023' is displayed.

```
MediaPipe Face Landmarker Task for web
MediaPipe Preview PRO + Follow
HTML
1 <!-- Copyright 2023 The MediaPipe Authors.
2
3 Licensed under the Apache License, Version 2.0 (the "License");
4 you may not use this file except in compliance with the License.
5 You may obtain a copy of the License at
6
7 http://www.apache.org/licenses/LICENSE-2.0
8
9 Unless required by applicable law or agreed to in writing, software
10 distributed under the License is distributed on an "AS IS" BASIS,
11 WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
12 See the License for the specific language governing permissions and
13 limitations under the License. -->
14 <html>
15 <head>
CSS
1 /* Copyright 2023 The MediaPipe Authors.
2
3 Licensed under the Apache License, Version 2.0 (the "License");
4 you may not use this file except in compliance with the License.
5 You may obtain a copy of the License at
6
7 http://www.apache.org/licenses/LICENSE-2.0
8
9 Unless required by applicable law or agreed to in writing, software
10 distributed under the License is distributed on an "AS IS" BASIS,
11 WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
12 See the License for the specific language governing permissions and
13 limitations under the License. */
14
15 /* Copyright 2022 The MediaPipe Authors.
JS (TypeScript)
1 // Copyright 2023 The MediaPipe Authors.
2
3 // Licensed under the Apache License, Version 2.0 (the "License");
4 // you may not use this file except in compliance with the License.
5 // You may obtain a copy of the License at
6
7 // http://www.apache.org/licenses/LICENSE-2.0
8
9 // Unless required by applicable law or agreed to in writing, software
10 // distributed under the License is distributed on an "AS IS" BASIS,
11 // WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
12 // See the License for the specific language governing permissions and
13 // limitations under the License.
14
15 import vision from "https://cdn.jsdelivr.net/npm/@mediapipe/tasks-
_neutral 0.0000
browDownLeft 0.0005
browDownRight 0.0073
browInnerUp 0.0003
browOuterUpLeft 0.0003
browOuterUpRight 0.0003
cheekPuff 0.0013
cheekSquintLeft 0.0000
cheekSquintRight 0.0000
eyeBlinkLeft 0.0097
eyeBlinkRight 0.0005
eyeLookDownLeft 0.0000
eyeLookDownRight 0.0000
eyeLookInLeft 0.0288
eyeLookInRight 0.0065
eyeLookOutLeft 0.0000
eyeLookOutRight 0.0627
eyeLookUpLeft 0.0836
eyeLookUpRight 0.0942
eyeSquintLeft 0.0000
eyeSquintRight 0.0000
eyeWideLeft 0.0029
eyeWideRight 0.0043
jawForward 0.0024
jawLeft 0.0016
jawOpen 0.0067
jawRight 0.0002
mouthClose 0.0303
mouthDimpleLeft 0.0178
mouthDimpleRight 0.0155
Console Assets Comments Shortcuts
Saturday 27 May 2023
```

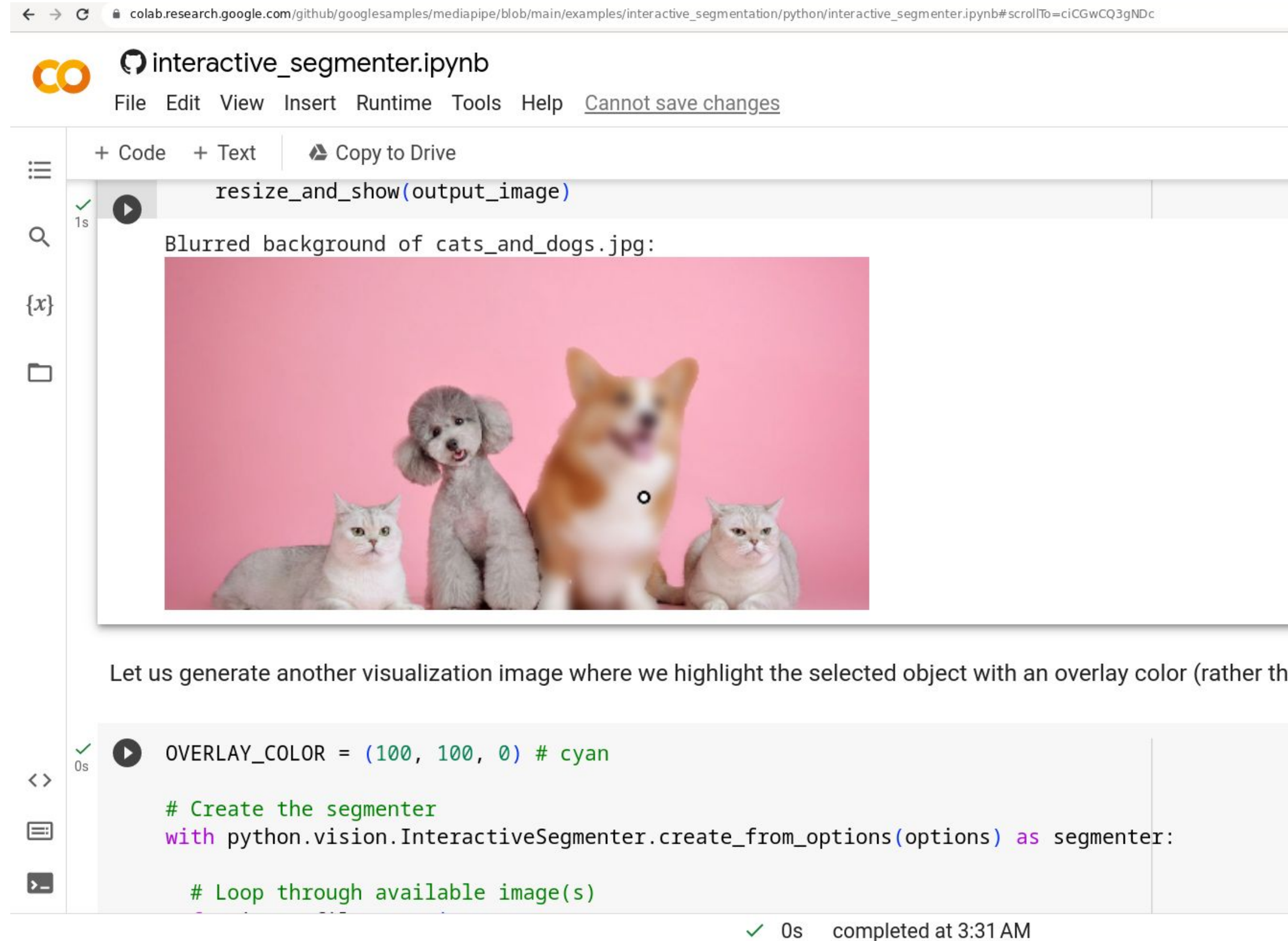
MediaPipe & Bard

# Let's get Bard involved!

Extreme Over-Confidence Mode : <https://mediapipe-studio.webapps.google.com/home>

# Interactive Segmentation

- Get Python Notebook
- Move to Colab
- Bard fix
- "I'm using mediapipe to do image analysis, and the following code blurs the foreground rather than the background : How should I change the code? "



colab.research.google.com/github/googlesamples/mediapipe/blob/main/examples/interactive\_segmentation/python/interactive\_segementer.ipynb#scrollTo=ciCGwCQ3gNDc


interactive\_segementer.ipynb  
File Edit View Insert Runtime Tools Help [Cannot save changes](#)

+ Code + Text Copy to Drive

resize\_and\_show(output\_image)

1s

Blurred background of cats\_and\_dogs.jpg:



Let us generate another visualization image where we highlight the selected object with an overlay color (rather than the background)

```
OVERLAY_COLOR = (100, 100, 0) # cyan

# Create the segmenter
with python.vision.InteractiveSegmenter.create_from_options(options) as segmenter:

    # Loop through available image(s)
```

0s completed at 3:31 AM



# Text Classification

- Show web version
- Documentation : Customisation
- MobileBERT with Quantisation
- "I am building a text classifier. Please give me positive and negative one-sentence reviews of a toaster (10 examples each)."
- "Can you write 5 sarcastic negative reviews of a toaster?"

## BERT-classifier

Now let's train a text classifier based on the [MobileBERT](#) model.

```
supported_model = text_classifier.SupportedModels.MOBILEBERT_CLASSIFIER
hparams = text_classifier.HParams(epochs=2, batch_size=48, learning_rate=3e-5, export_dir="bert_exported_mo
options = text_classifier.TextClassifierOptions(supported_model=supported_model, hparams=hparams)
```

Create and train the text classifier like we did with the average word embedding-based classifier.

**Warning:** This can take ~25 minutes on a GPU runtime and nearly 7 hours on CPU. We strongly recommend using a GPU runtime.

```
bert_model = text_classifier.TextClassifier.create(train_data, validation_data, options)
```

Evaluate the model. Note the improved performance compared to the average word embedding-based classifier.

```
loss, acc = bert_model.evaluate(validation_data)
print(f'Test loss:{loss}, Test accuracy:{acc}')
```

The MobileBERT model is over 100MB so when we export the BERT-based classifier as a TFLite model, it will help to use quantization which can bring the TFLite model size down to 28MB.

```
from mediapipe_model_maker import quantization
quantization_config = quantization.QuantizationConfig.for_dynamic()
bert_model.export_model(quantization_config=quantization_config)
bert_model.export_labels(export_dir=options.hparams.export_dir)
```

# MediaPipe FTW!

- On-Device ML components finally starting to work together
- Bard is a productivity win
- Still early in the AI game...



# Machine Learning Singapore MeetUp

- Our MeetUp has over 5000 members now! : Monthly in-person meetings
- Go to : <https://www.meetup.com/machine-learning-singapore/>



# Thank you!



**Martin Andrews**

Head of AI : Red Dragon AI

Google Developer Expert Machine Learning & Deep Learning



@mdda123



@mdda