Google I/O Extended

MediaPipe with a bit of Bard

Martin Andrews - Singapore ML GDE

About Me

RED DRAGON AI

- 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

Google Developers Experts

RED DRAGON AI

About Red Dragon AI

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

Overview

MediaPipe with a bit of Bard

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

MediaPipe

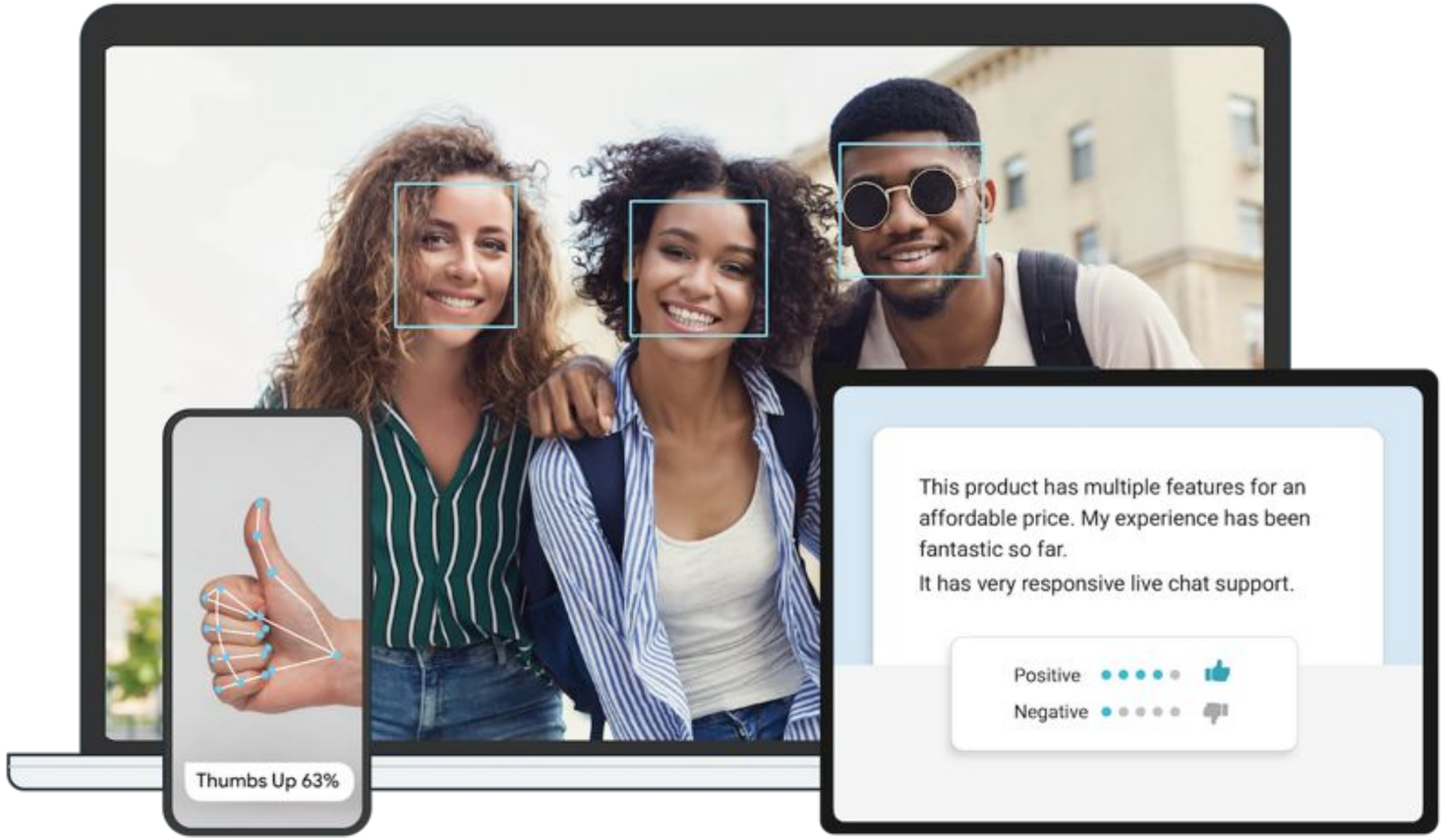
What is MediaPipe?

The ecosystem is coming together...

MediaPipe

On-Device Machine Learning

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



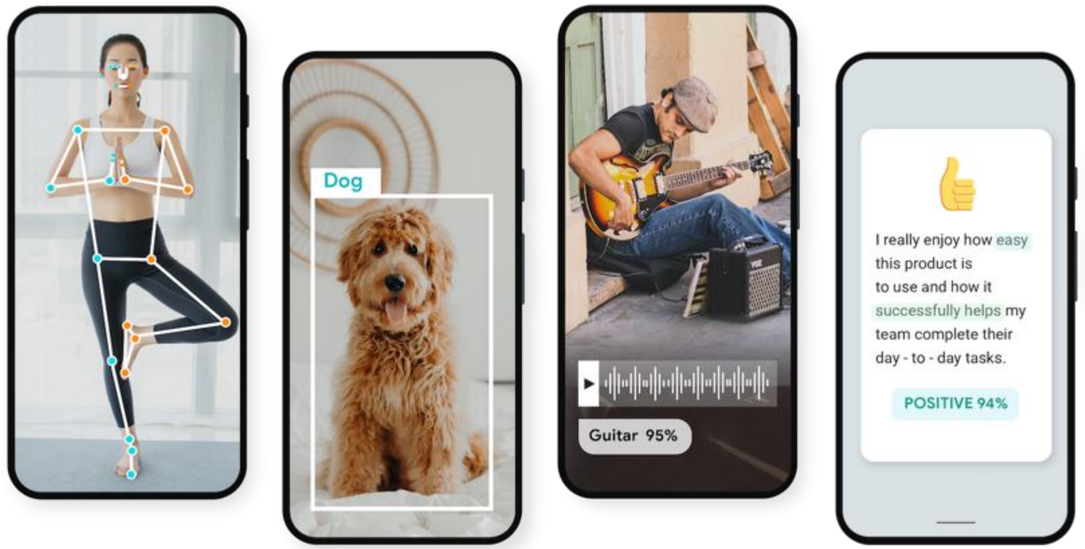
On-Device Benefits

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

MediaPipe

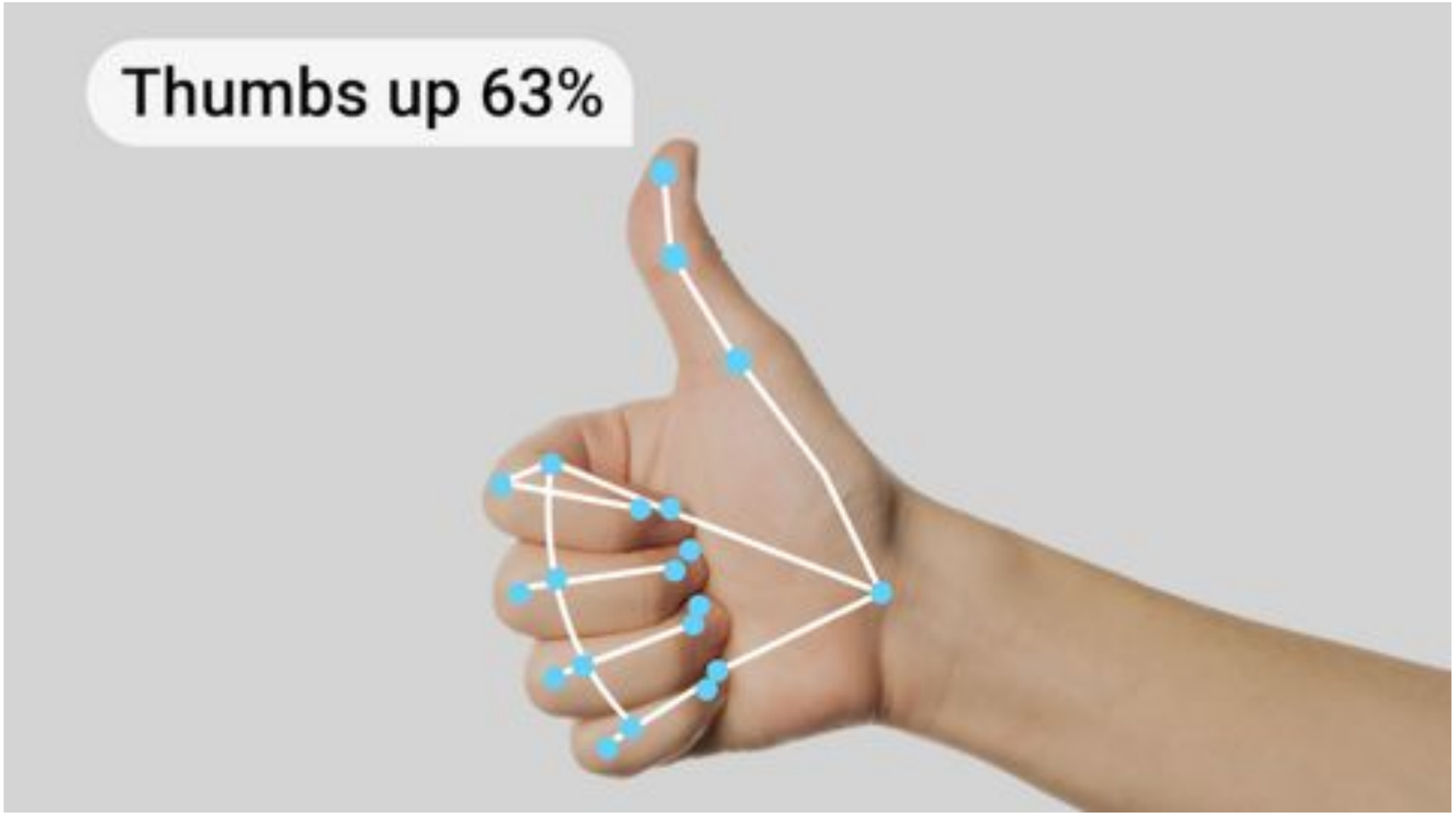
Big picture now coming together

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



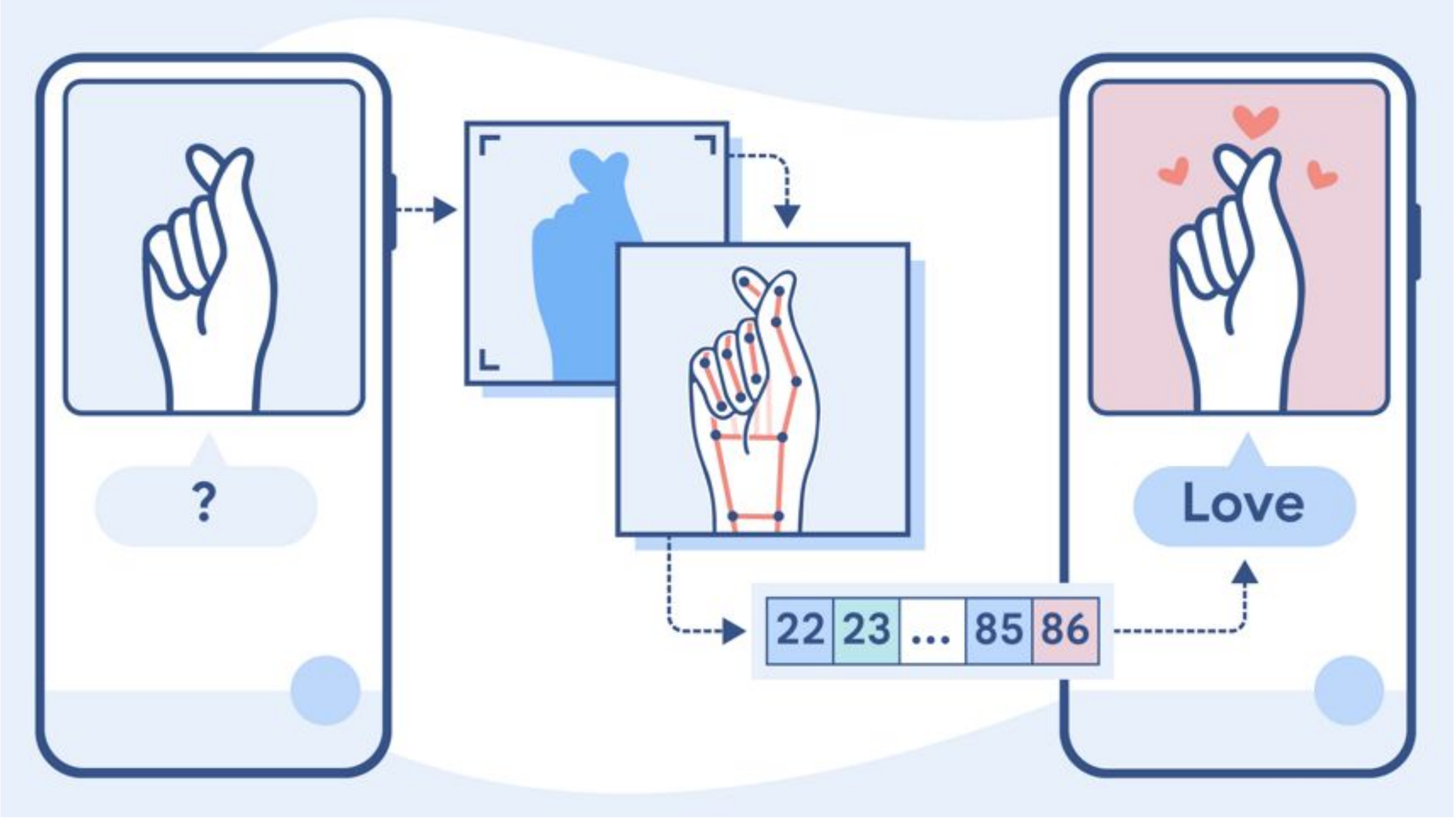
MediaPipe

Example Task = Gesture Recognition



MediaPipe

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

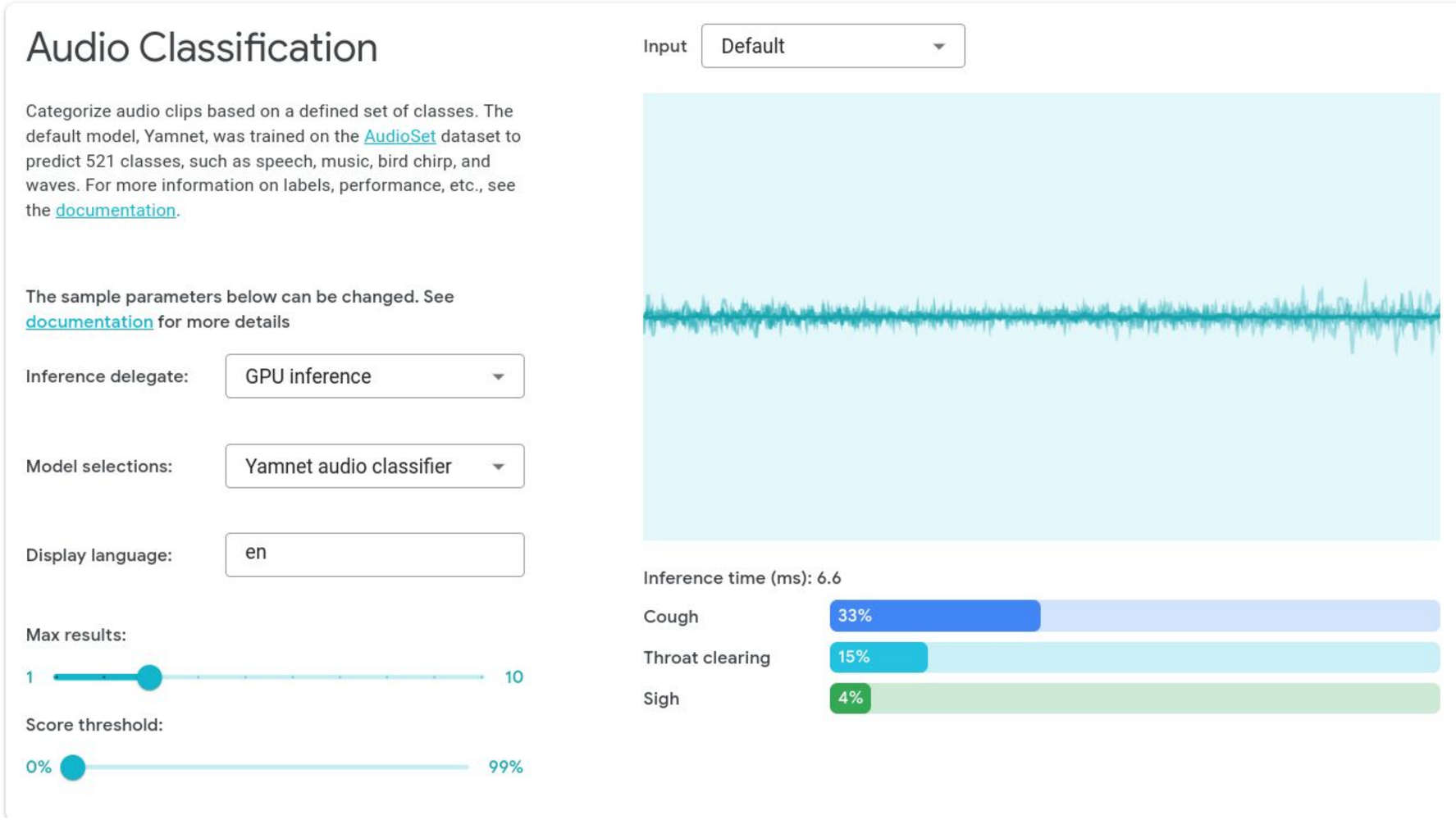
Demo Time!

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

Demo

Audio Classification

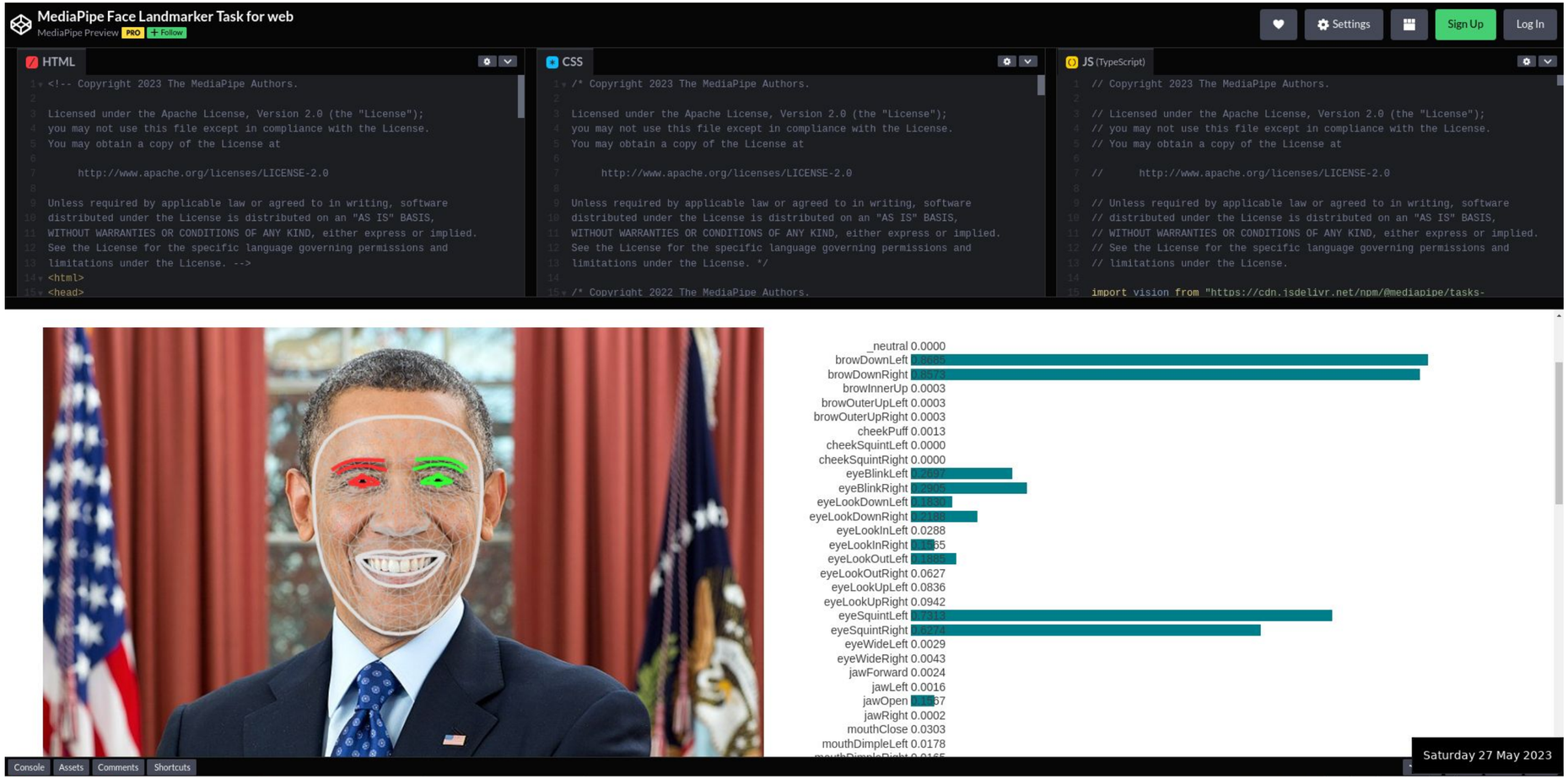
- (Using the UI)



Demo

Facial Landmarks

- (Using the Web)



MediaPipe & Bard

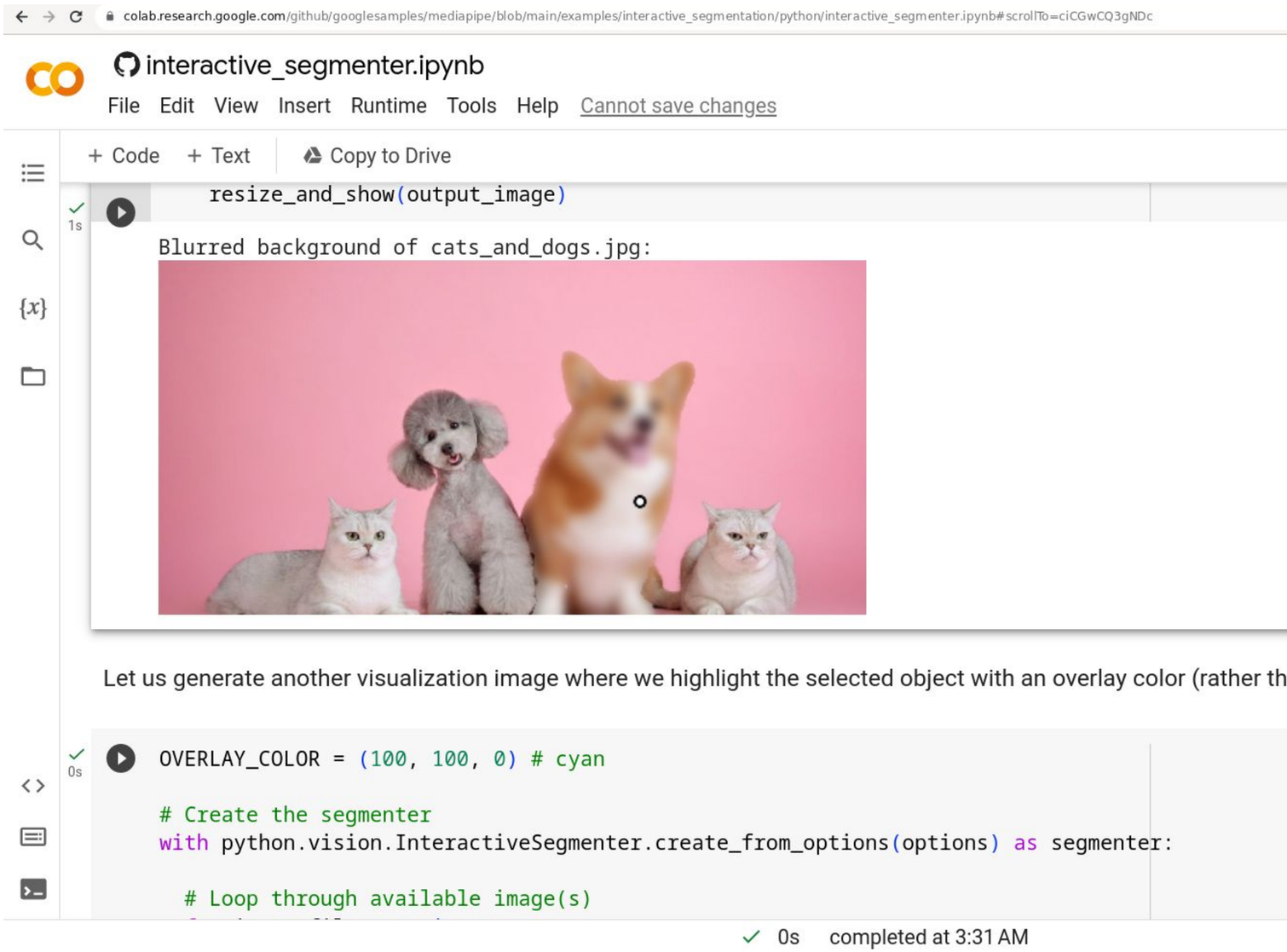
Let's get Bard involved!

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

Demo with Bard

Interactive Segmentation

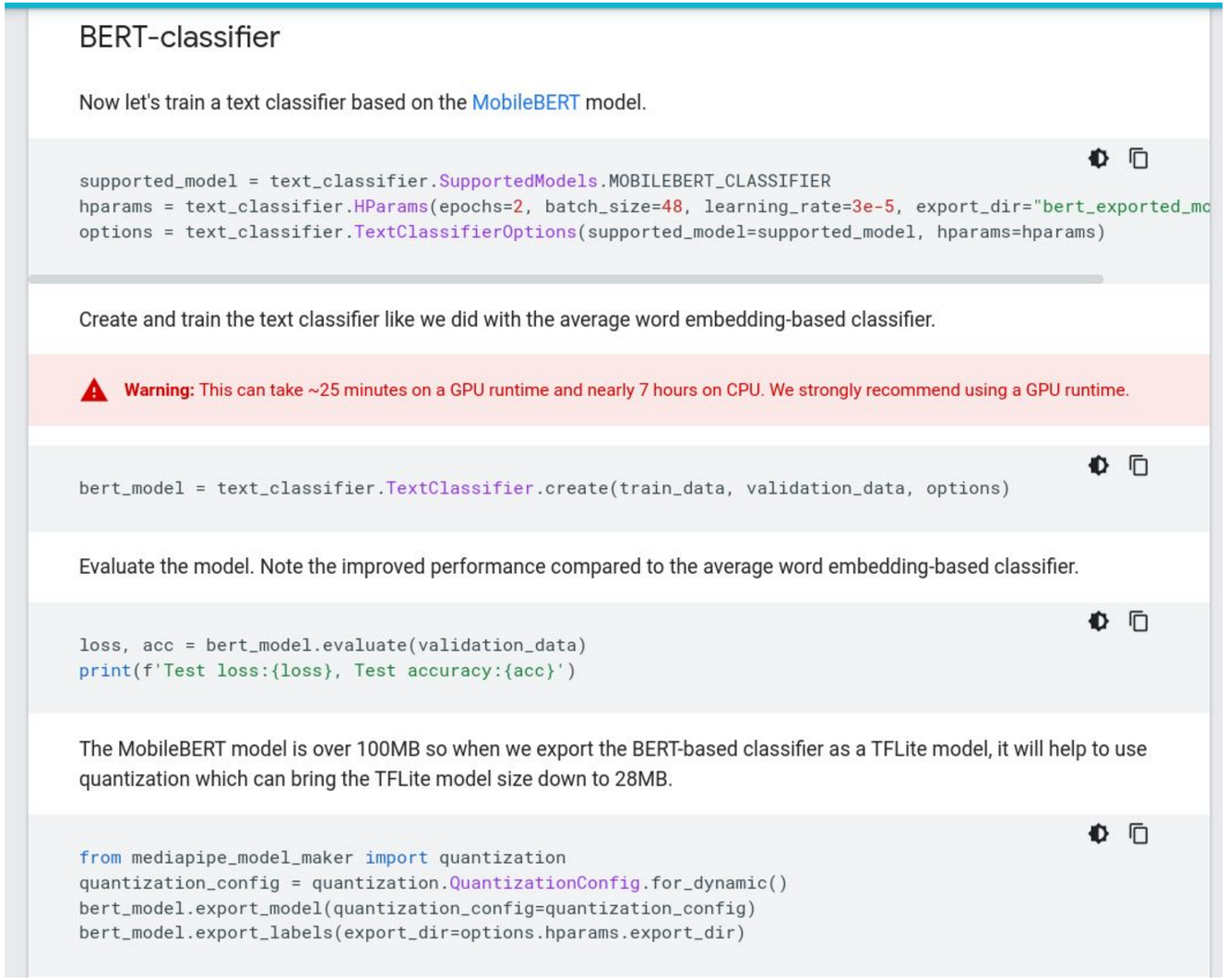
- Get Python Notebook
- Move to Colab
- Bard fix
- "I'm using mediapipe to do image analysis, and the following code blurs the foreground rather than the background : How should I change the code? "



Demo with Bard

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)
```

Wrap-Up

MediaPipe FTW!

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

ML Singapore MeetUp

Machine Learning Singapore MeetUp

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

Google I/O Extended

Thank you!

Martin Andrews

Head of AI : Red Dragon AI
Google Developer Expert Machine Learning & Deep Learning

@mdda123 @mdda