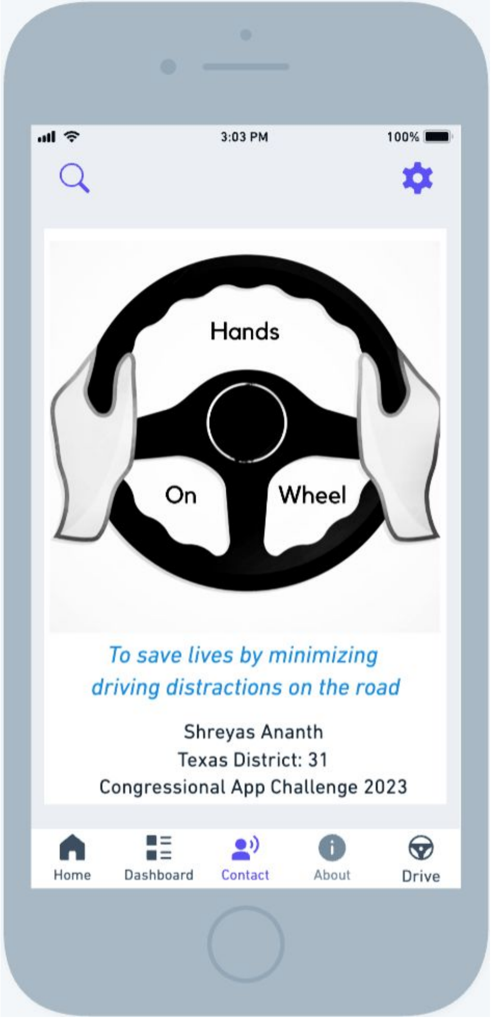


To save lives by minimizing driving distractions on the road

Shreyas Ananth
Texas District: 31
Congressional App Challenge 2023

Problem with Distracted Driving

Fatalities	1.35 million*	3,144**	226***
Injuries	50 million	421,000	53,874

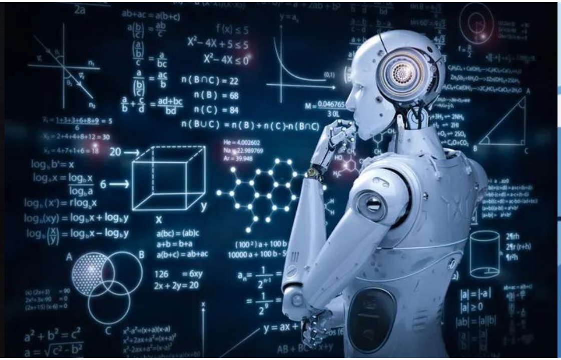
2019 * World Health Organization, ** National Highway Traffic Safety Administration, *** Texas Department of Transportation (TxDOT)

Major Driving Distractions

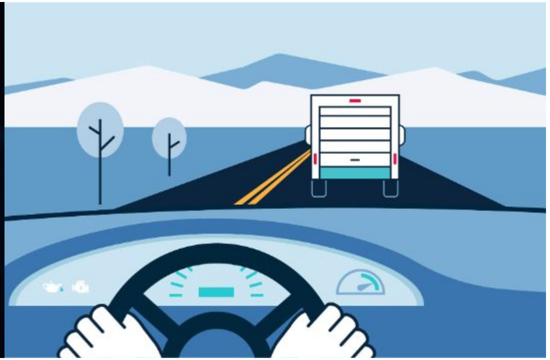
- Texting
- Calling
- Emotional Conversations
- Eating
- Drinking
- Putting on makeup

Combining the power of Computer Vision and AI

Computer Vision

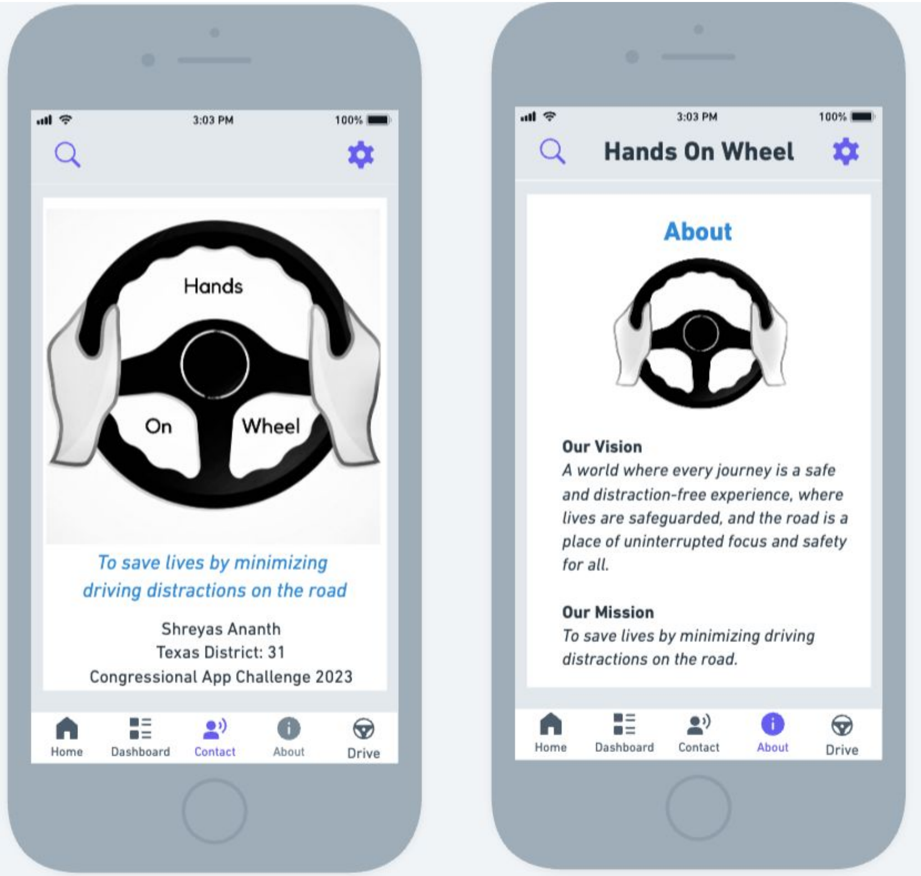


Machine Learning



User-Friendly Mobile App

Introducing Hands On Wheel



To save lives by minimizing driving distractions on the road

Shreyas Ananth
Texas District: 31
Congressional App Challenge 2023

Hands On Wheel

About

Our Vision

A world where every journey is a safe and distraction-free experience, where lives are safeguarded, and the road is a place of uninterrupted focus and safety for all.

Our Mission

To save lives by minimizing driving distractions on the road.

Vision

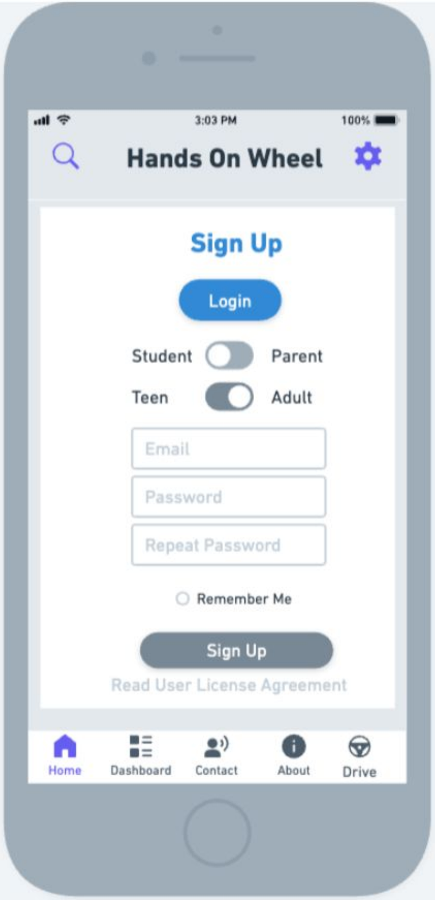
A world where every journey is a safe and distraction-free experience, where lives are safeguarded, and the road is a place of uninterrupted focus and safety for all.

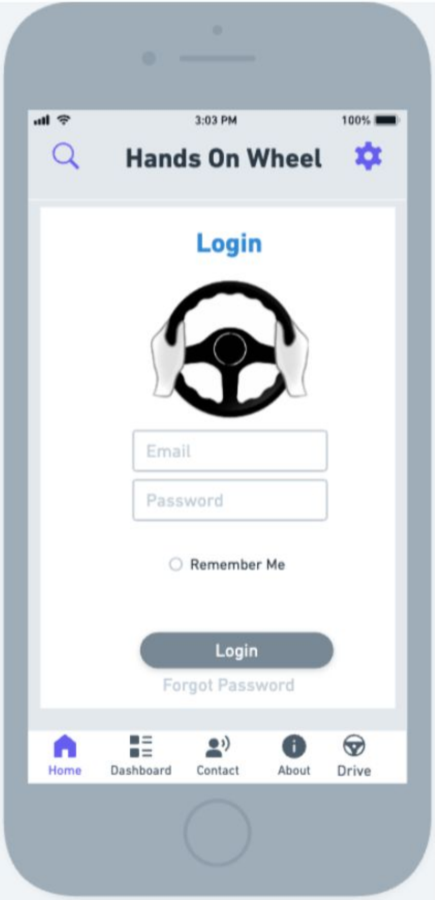
Mission

To save lives by minimizing driving distractions on the road.

Free app supported by iOS and Android

Registration & Activation





Signup

Register as a Parent or Student.

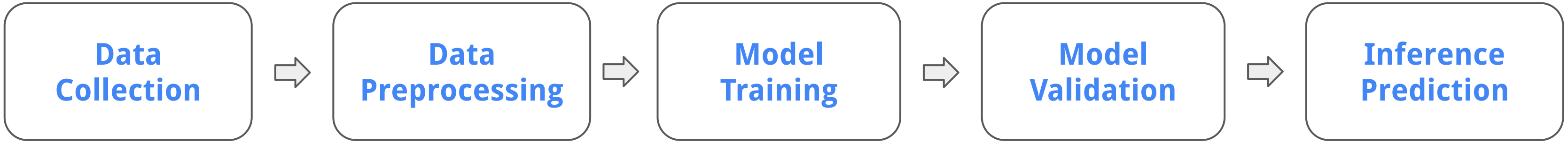
While we encourage Teens to sign-up, we also monitor Adult driving focus.

Login

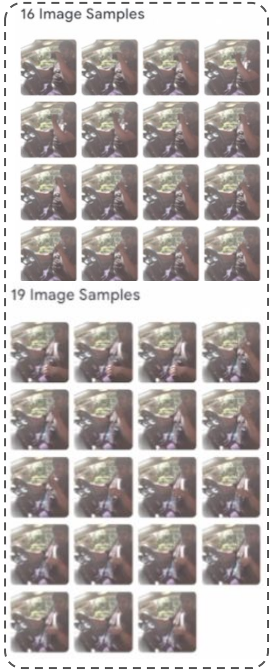
Once registered, you can login to activate the driving mode.

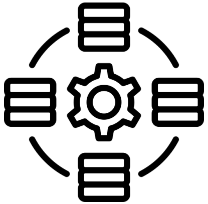
Clicking on Forgot Password will send an email to the registered, with a link to reset the password.

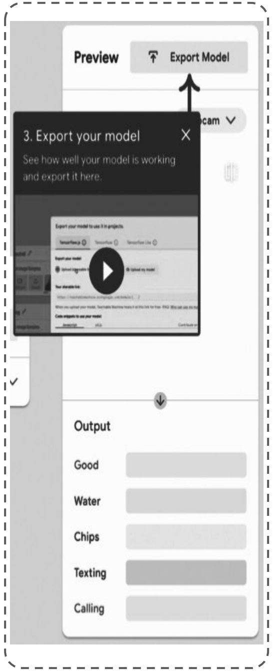
Machine Learning Workflow



Data Collection → Data Preprocessing → Model Training → Model Validation → Inference Prediction

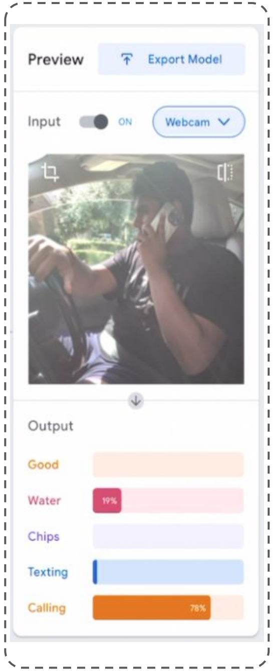






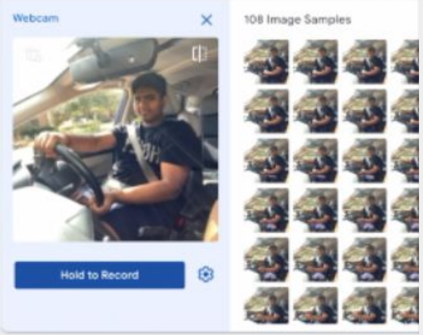
3. Export your model

See how well your model is working and export it here.

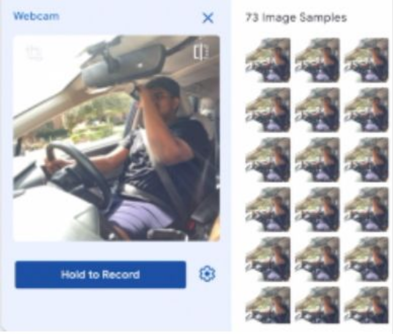


Sample Data Collection

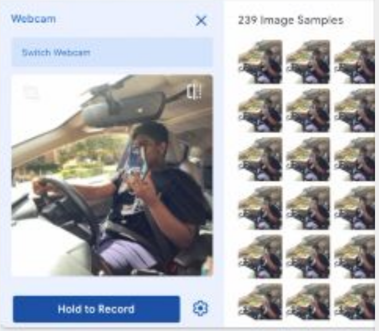
Good



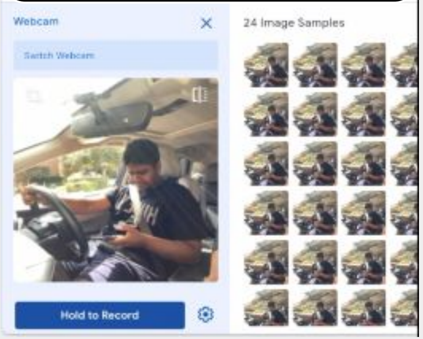
Water



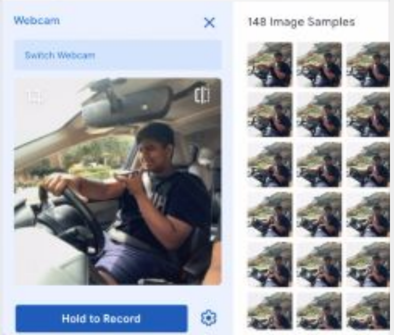
Chips



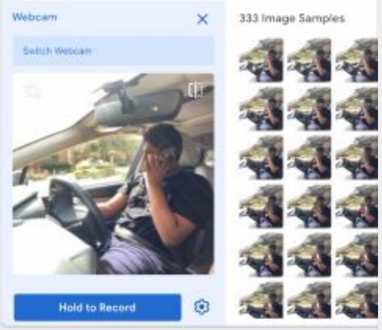
Texting



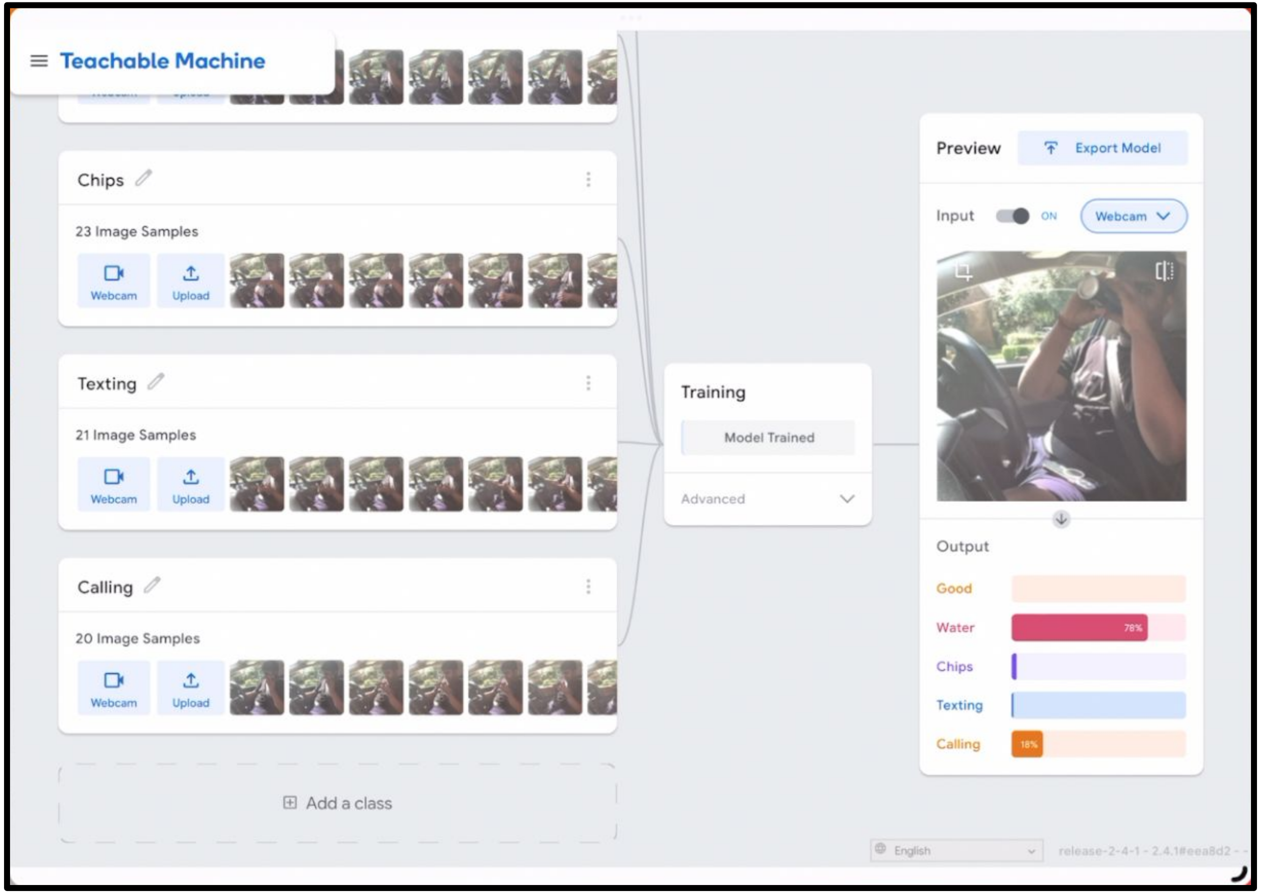
Calling



Calling



Machine Learning Model Training



Teachable Machine

Chips

23 Image Samples

Texting

21 Image Samples

Calling

20 Image Samples

Training

Model Trained

Preview

Input ON Webcam

Output

Good

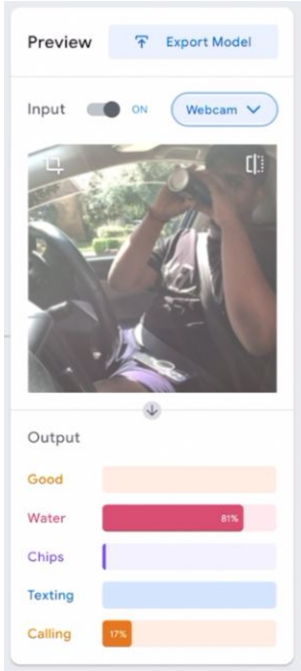
Water 78%

Chips

Texting

Calling 18%

Driver Focus Predictions

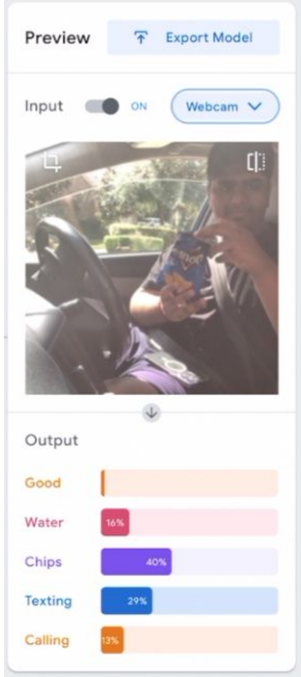


Input ON Webcam

Output

Good
Water 81%
Chips
Texting
Calling 17%

Drinking

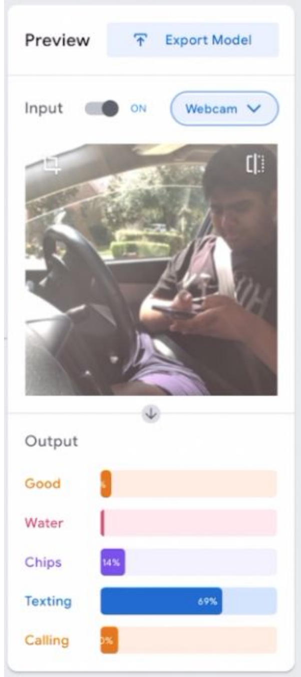


Input ON Webcam

Output

Good
Water 16%
Chips 40%
Texting 29%
Calling 13%

Eating

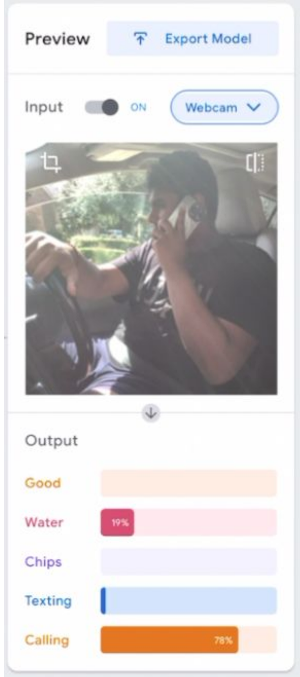


Input ON Webcam

Output

Good %
Water
Chips 14%
Texting 69%
Calling 0%

Texting

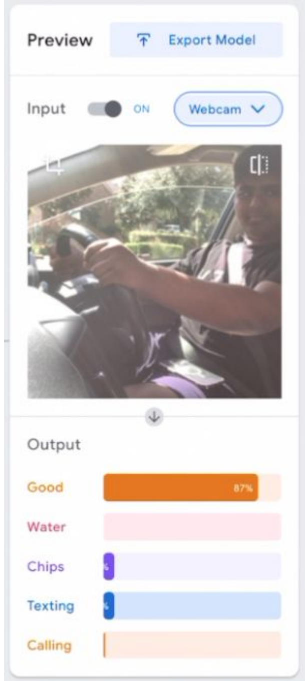


Input ON Webcam

Output

Good
Water 19%
Chips
Texting
Calling 78%

Calling



Input ON Webcam

Output

Good 87%
Water
Chips %
Texting %
Calling

Good

Machine Learning Model Code

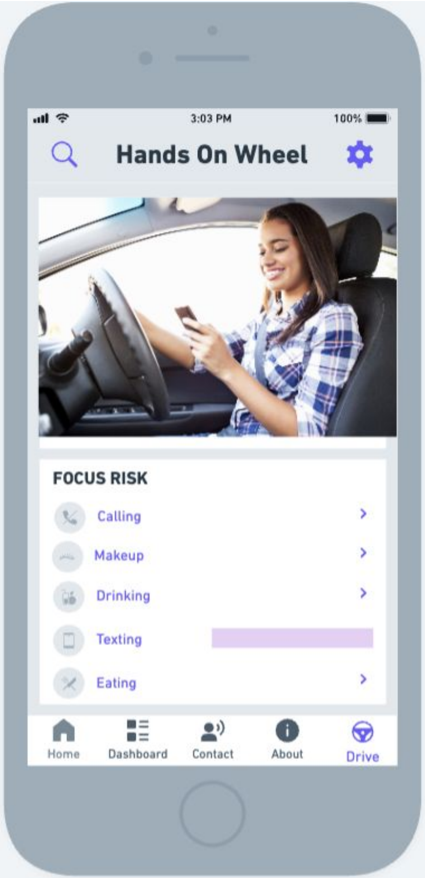
```python
from keras.models import load_model  # TensorFlow is required for Keras to work
from PIL import Image, ImageOps  # Install pillow instead of PIL
import numpy as np

# Disable scientific notation for clarity
np.set_printoptions(suppress=True)

# Load the model
model = load_model("keras_Model.h5", compile=False)

# Load the labels
class_names = open("labels.txt", "r").readlines()

# Create the array of the right shape to feed into the keras model
# The 'length' or number of images you can put into the array is
# determined by the first position in the shape tuple, in this case 1
data = np.ndarray(shape=(1, 224, 224, 3), dtype=np.float32)

# Replace this with the path to your image
image = Image.open("<IMAGE_PATH>").convert("RGB")

# resizing the image to be at least 224x224 and then cropping from the center
size = (224, 224)
image = ImageOps.fit(image, size, Image.Resampling.LANCZOS)

# turn the image into a numpy array
image_array = np.asarray(image)

# Normalize the image
normalized_image_array = (image_array.astype(np.float32) / 127.5) - 1

# Load the image into the array
data[0] = normalized_image_array

# Predicts the model
prediction = model.predict(data)
index = np.argmax(prediction)
class_name = class_names[index]
confidence_score = prediction[0][index]

# Print prediction and confidence score
print("Class:", class_name[2:], end="")
print("Confidence Score:", confidence_score)
```

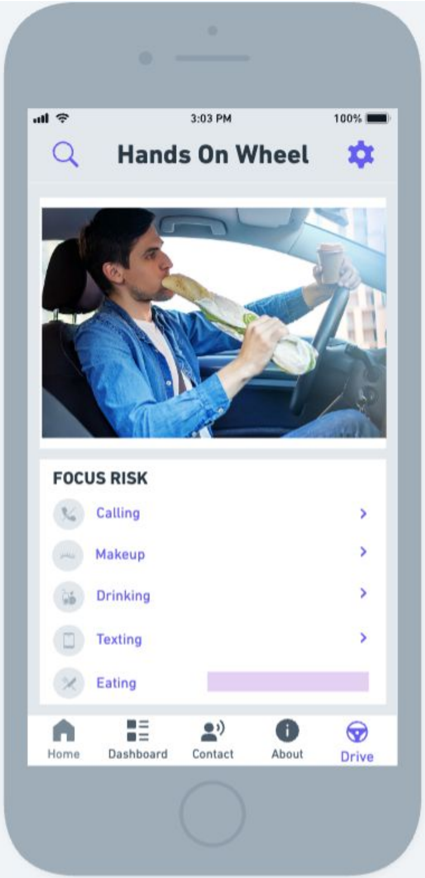
Mobile App Software

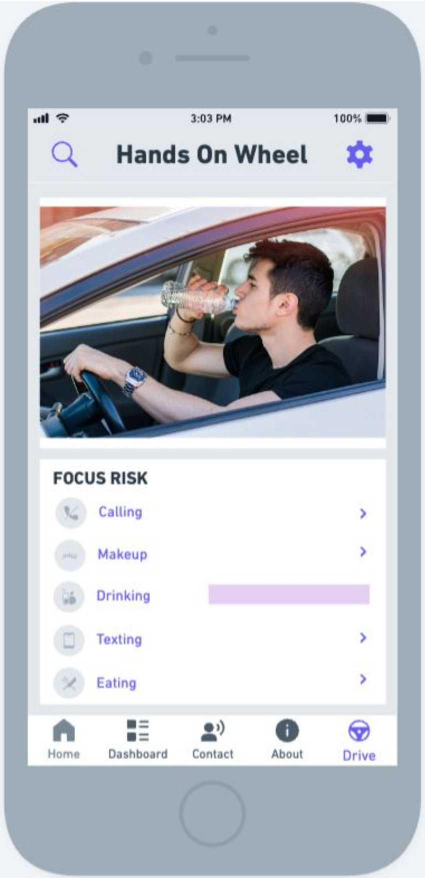
```
<div>Teachable Machine Image Model</div>
<button type="button" onclick="init()">Start</button>
<div id="webcam-container"></div>
<div id="label-container"></div>
<script src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs@latest/dist/tf.min.js"></script>
<script
src="https://cdn.jsdelivr.net/npm/@teachablemachine/image@latest/dist/teachablemachine-image.min.js
"></script>
<script type="text/javascript">
    // More API functions here:
    // https://github.com/googlecreativelab/teachablemachine-community/tree/master/libraries/image

    // the link to your model provided by Teachable Machine export panel
    const URL = "./my_model/";

    let model, webcam, labelContainer, maxPredictions;

    // Load the image model and setup the webcam
    async function init() {
        const modelURL = URL + "model.json";
        const metadataURL = URL + "metadata.json";

        // load the model and metadata
        // Refer to tmImage.loadFromFiles() in the API to support files from a file picker
        // or files from your local hard drive
        // Note: the pose library adds "tmImage" object to your window (window.tmImage)
        model = await tmImage.load(modelURL, metadataURL);
        maxPredictions = model.getTotalClasses();

        // Convenience function to setup a webcam
        const flip = true; // whether to flip the webcam
        webcam = new tmImage.Webcam(200, 200, flip); // width, height, flip
        await webcam.setup(); // request access to the webcam
        await webcam.play();
        window.requestAnimationFrame(loop);
```

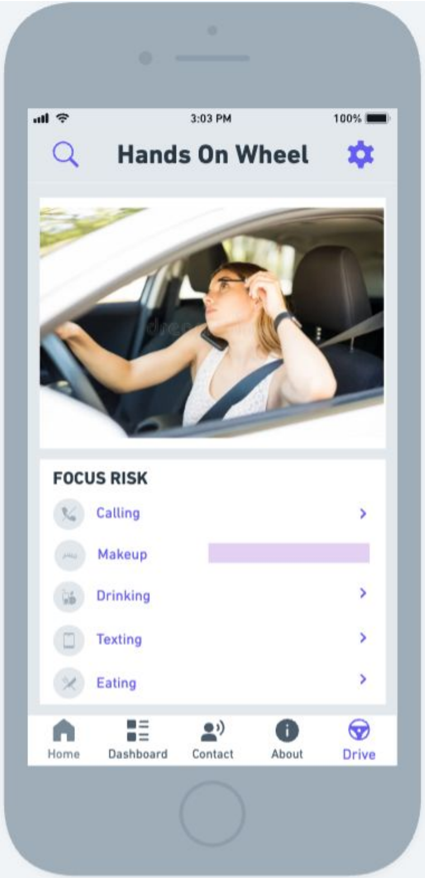
```
// append elements to the DOM

document.getElementById("webcam-container").appendChild(webcam.canvas);
        labelContainer = document.getElementById("label-container");
        for (let i = 0; i < maxPredictions; i++) { // and class labels
            labelContainer.appendChild(document.createElement("div"));
        }
    }

    async function loop() {
        webcam.update(); // update the webcam frame
        await predict();
        window.requestAnimationFrame(loop);
    }

    // run the webcam image through the image model
    async function predict() {
        // predict can take in an image, video or canvas html element
        const prediction = await model.predict(webcam.canvas);
        for (let i = 0; i < maxPredictions; i++) {
            const classPrediction =
                prediction[i].className + ": " +
prediction[i].probability.toFixed(2);
            labelContainer.childNodes[i].innerHTML = classPrediction;
        }
    }
</script>
```

Driver Focus Predictions

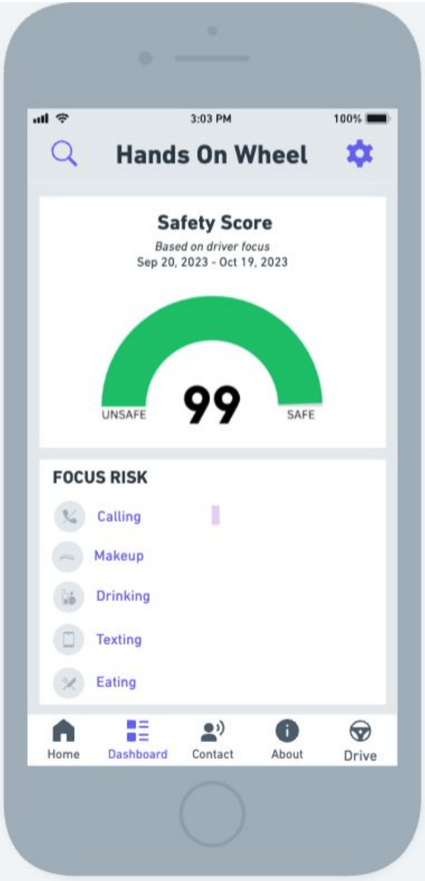








Driver Focus Dashboard



3:03 PM 100%

Hands On Wheel

Safety Score

Based on driver focus

Sep 20, 2023 - Oct 19, 2023

UNSAFE 99 SAFE

FOCUS RISK

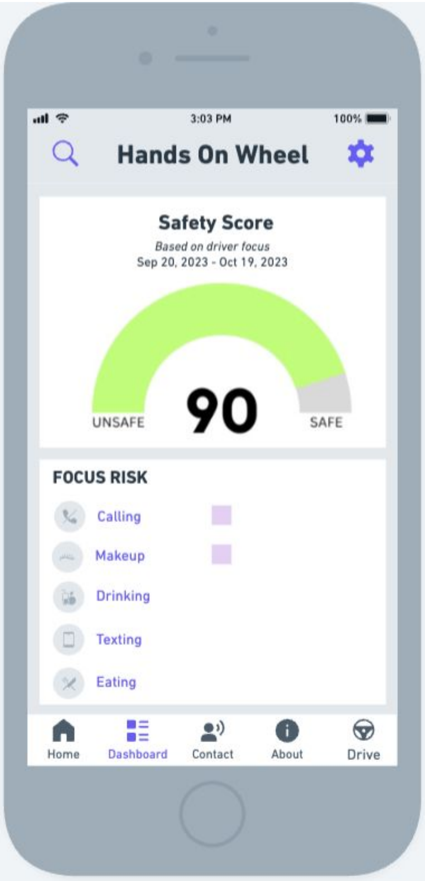
Calling

Makeup

Drinking

Texting

Eating



3:03 PM 100%

Hands On Wheel

Safety Score

Based on driver focus

Sep 20, 2023 - Oct 19, 2023

UNSAFE 90 SAFE

FOCUS RISK

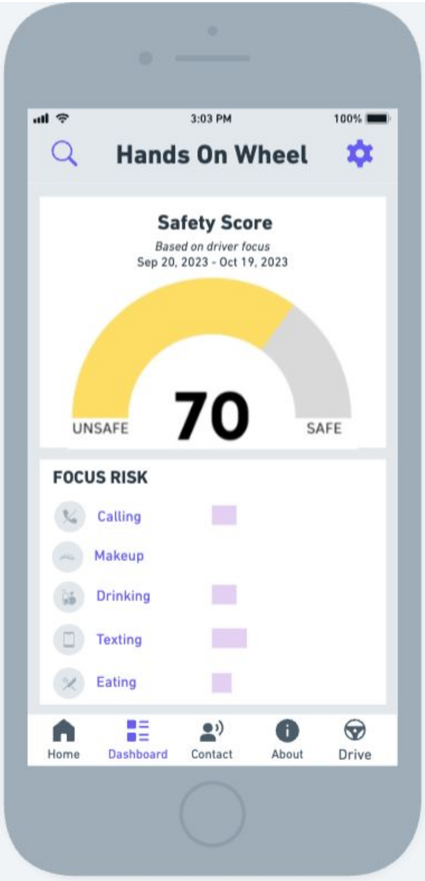
Calling

Makeup

Drinking

Texting

Eating



3:03 PM 100%

Hands On Wheel

Safety Score

Based on driver focus

Sep 20, 2023 - Oct 19, 2023

UNSAFE 70 SAFE

FOCUS RISK

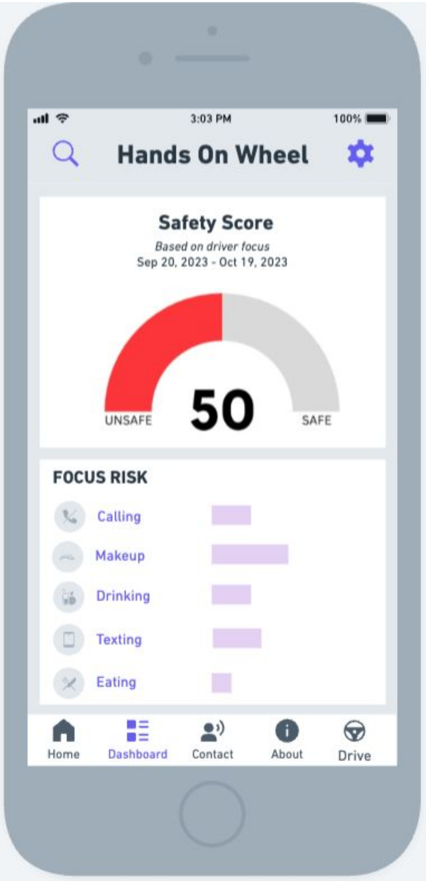
Calling

Makeup

Drinking

Texting

Eating



3:03 PM 100%

Hands On Wheel

Safety Score

Based on driver focus

Sep 20, 2023 - Oct 19, 2023

UNSAFE 50 SAFE

FOCUS RISK

Calling

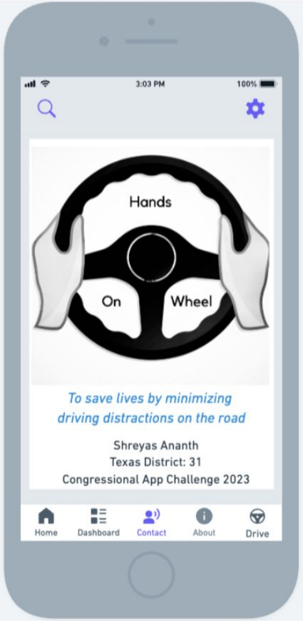
Makeup

Drinking

Texting

Eating

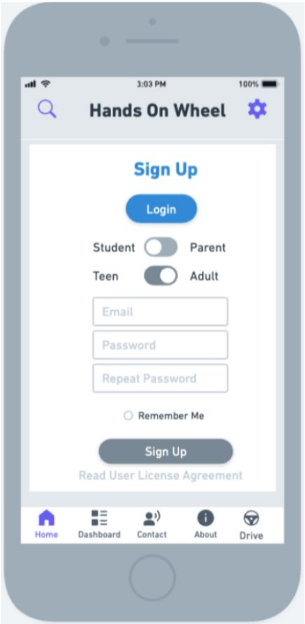
Hands on Wheel can minimize distractions and save lives!



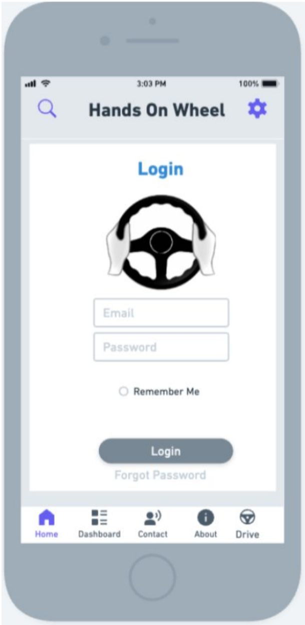
To save lives by minimizing driving distractions on the road

Shreyas Ananth
Texas District: 31
Congressional App Challenge 2023

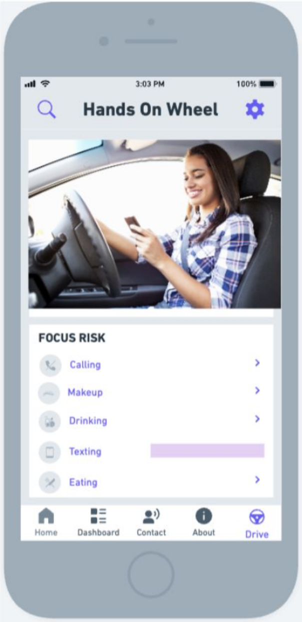
Home



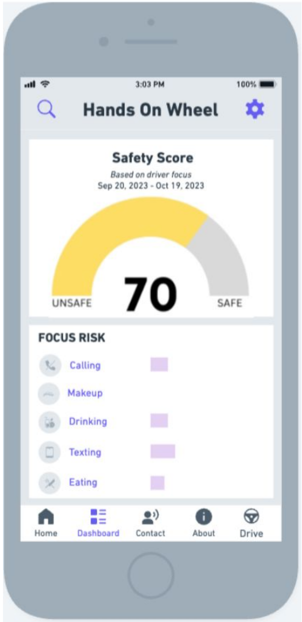
Signup



Login



Drive Mode



Safety Score

Next Steps

- Share the App with friends & family
- Launch the App on Apple Store and Google Play
- Integrate Focus Risk Score with Insurance companies
- Improve Model inference prediction accuracy
- Invest in long-term behavior analysis

Benefits

- Minimal distractions while driving, especially for Teens
- No additional equipment needed, works on Apple iOS and Android phones
- Safer roads, lower fatalities and injuries
- Opportunity for parents to monitor Teen driving behaviors
- Lower insurance costs

Acknowledgements

- Parents
- Teachers
- Mentors
- Google Teachable Machines [ML model development]
- Whimsical [App design]
- Bubble.io [App development]