Artificial Intelligence Final Project Presentation

Team: 2

Team Members:
Ong Lee Lee, Teo Kai Heng,
Tan Gaik Neo Ivy & Seah Ghim Sin Steven

WHAT IS AI?

What is AI?

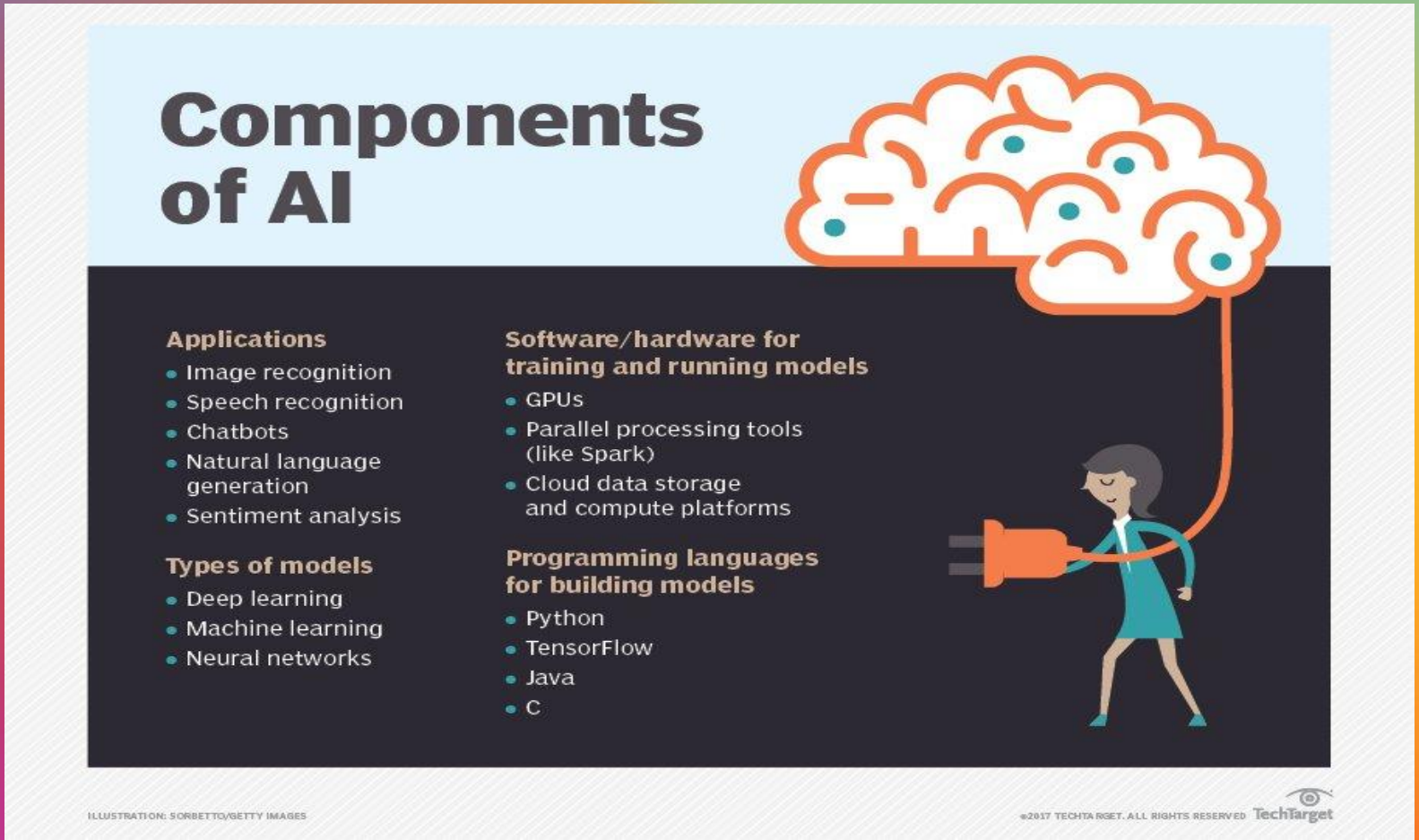
In the simplest terms, AI which stands for artificial intelligence refers to systems or machines that mimic human intelligence to perform tasks and can iteratively improve themselves based on the information they collect. AI manifests in a number of forms:

- Chatbots use AI to understand customer problems faster and provide more efficient answers
- Intelligent assistants use AI to parse critical information from large free-text datasets to improve scheduling

AI is much more about the process and the capability for superpowered thinking and data analysis than it is about any particular format or function.

AI is intended to significantly enhance human capabilities and contributions; not to replace human. That makes it a very valuable business asset.

KEY COMPONENTS OF AI



Project Background

Currently in the Market, there are varieties of smart home camera which can provide different solutions / features.

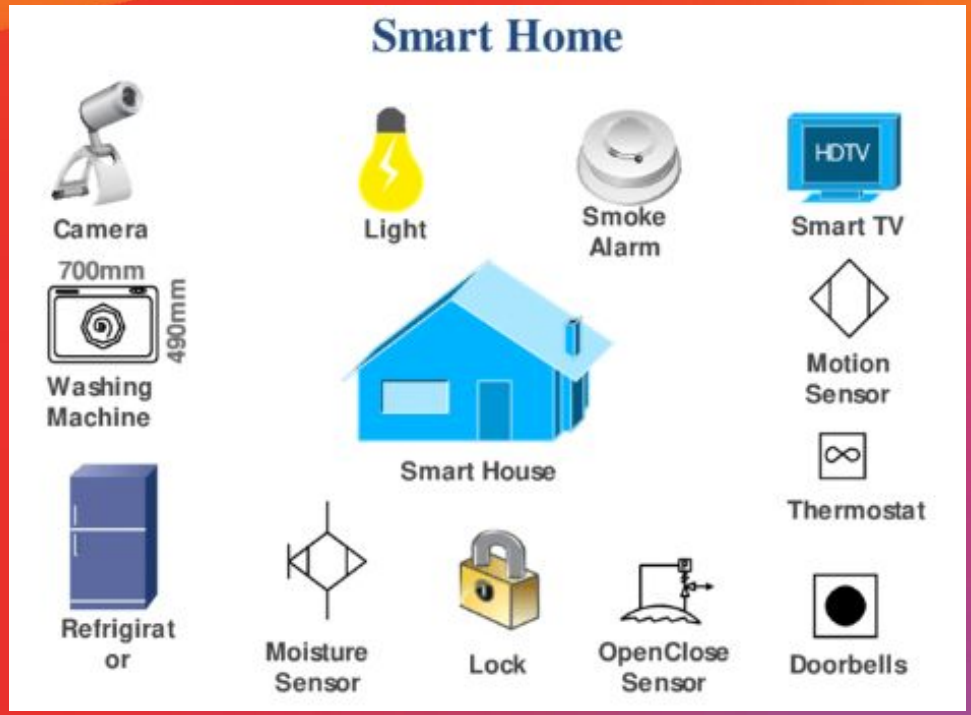
However, **most of them are designed for the purpose of home security and surveillance**, with dedicated features such as facial recognition.

Other useful specific features are only found in industrial surveillance camera applications.

Project Background

In addition, existing smart home solutions require smart home IoT gadgets to be integrated with many different kinds of sensors like smoke detector, motion sensor etc to achieve certain outcomes.

This requires buying and installing many different kinds sensors to be installed in several locations.



These are some of the **existing limitations of smart home cameras and smart IoT home solutions**.

However, smart home solutions have the potential to solve many pain points at home…

Pain Points to solve with smart home solutions

- Efficient energy use (ie. less energy wastages and consumption) will allow for a more sustainable path with regards to global resource consumption and can contribute towards tackling climate change. Also, the energy costs will be one key area where the conflict in Ukraine will affect Singapore, with pump prices for petrol and diesel here expected to rise, along with electricity rates for both businesses and households.

- According to SCDF FIRE INCIDENTS STATISTICS 2021, Fires of electrical origin was the leading type of fire in 2021, accounting for 588 cases (31.9%). Cooking activities was the next highest cause of fires with 408 cases (22.1%), followed by dropped light with 349 cases (18.9%). Fire incidences can result in damages / loss of property or even lives

Pain Points to solve with smart home solutions

- Falls are a main cause of mortality and disability in the elderly. More than one-third of people 65 years of age or older fall each year, and in half of such cases the falls are recurrent. The risk doubles or triples in the presence of cognitive impairment or history of previous falls.

- There have been more cases of elderly dying at home undetected. This is especially so due to the general trend of an aging society, in conjunction with fall in birth rates and marriages, resulting in more elderly living alone.

SP Group's electricity tariffs to go up by 8.1% in third quarter

This is the sixth consecutive quarter of increase. SP Group reviews the electricity tariffs every quarter based on guidelines set by the Energy Market Authority (EMA).

Fire in Bedok North flat claims 3 lives, including that of 3-year-old

Three people, including a three-year-old, have died in a fire that broke out in the living room of a flat in Bedok North on early Friday morning (May 13).

Five cases of seniors in Singapore who died alone at home

SINGAPORE - When they died, none of their family members may have known - or cared.

A 74-year-old, who lived alone with two cats, had left a light on in her Bedok North Avenue 2 flat. She had not been seen feeding stray cats in her neighbourhood or hanging out her laundry as she used to.

When worried neighbours opened her unlocked door on Sept 4, they were hit by a rotting smell. The old woman was found dead.

Falls are the main cause of injury in older people

SINGAPORE - Transport Minister Khaw Boon Wan, 66, said recently that he fractured his left arm when he hastily got out of bed to go to the toilet. The resulting fall led to a three-hour operation and a two-week stay at Singapore General Hospital.

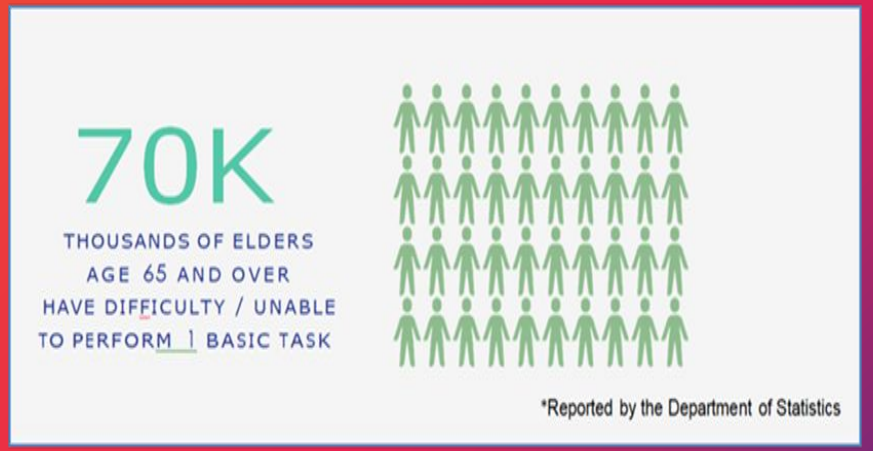
Published 31 Mar 2019, 4:30 pm SGT

Globally, about one in three older adults experience a fall each year. ST PHOTO: DESMOND WEE

Statistics of Falls in Singapore

https://www.worldlifeexpectancy.com/singapore-falls

Statistics of Singapore Aging Population



Based on Singapore census data :

- **70K Elderly, over 65 years, have difficulty with Basic Tasks – self- care, communicating , moving around.**
- **The Elderly population, is projected to hit 25% by 2030 and 35% by 2035**
- **Estimated 83,000 living alone by 2030.**

More Seniors prefer their independence and family members , caregivers are unable to monitor 24/7. Caring for their safety cannot be limited to just watching the cameras and security systems.

The top concerns are accidental falls, compromised mobility and stability, dementia care and safety (like unattended fires on kitchen stoves, 'strangers' calling at the door) , while alone.

Proposed Smart Home Camera Solution

Using **solely the smart home camera** as a **one-size-fit-all comprehensive computer vision solution** to detect and trigger actions, when there are detections for the following scenarios:

1. **Detected that there is no presence of person in the room**
2. **Detected person in the room to be asleep**
3. **Detected fire when there is nobody at home or when the stove is left on accidentally**
4. **Detected the possible event of elderly falling down**
5. **Detected the possible death of person at home**

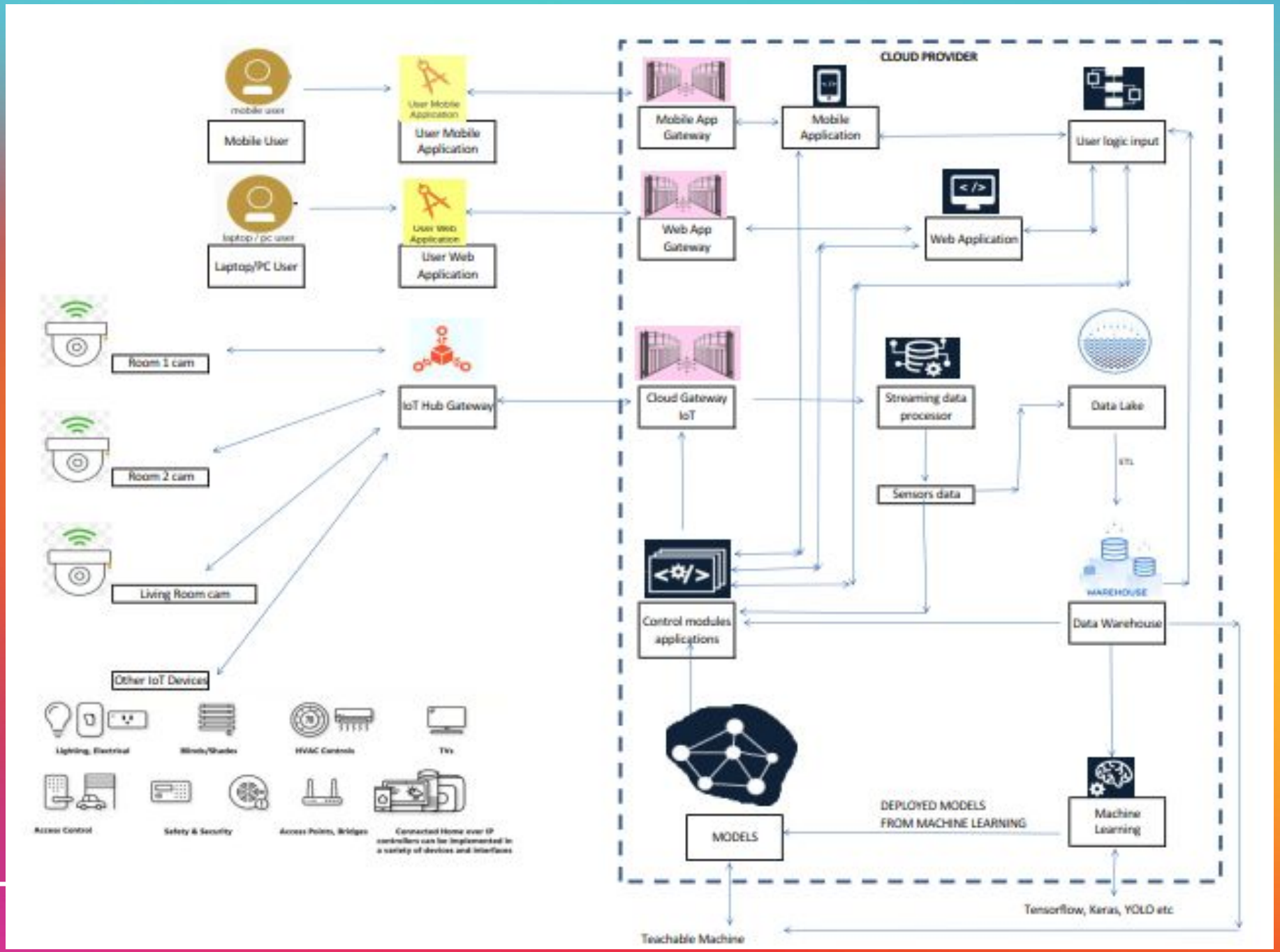
The above visual detections and follow-up triggered actions would form the smart home solutions with **desired design features** to **solve the aforementioned pain points**, and **without excessive sensors**.
The smart camera can be incorporated as part of the overall smart home systems setup.

Possible triggers/actions

- **Human presence detection** - Can trigger smart switches for electrical appliances such as lights, aircon, fans to be switched off if there is no person detection for some time. The triggered switches are for those residing within the room that the camera is located. This allows for energy savings.
- **Sleep detection** - Can trigger smart switches for certain electrical appliances such as lights, TV, music speakers to be switched off if it is detected that people in the room are asleep for some time. This will enable energy savings.
- **Fire detection** - Can trigger alarms via speakers at home, send alert messages to home owners' phones, or trigger fire sprinklers / fire suppression system if there is continuous fire detection for a brief amount of time. This will facilitate people to evacuate or automatically suppress the fire.
- **Fall detection** - Can send alert messages to loved ones' phones to alert possible fall has taken place for the concerned elderly for further action.
- **Possible death detection** - If a person is detected to be "sleeping" for an unusually long time, an alert message can be sent to home members' / loved ones' phones to notify of such scenario.
- Certain features can be tweaked and set by users based on their preferences, such as the specific appliances switches to be triggered, duration of detection before triggering of actions etc.

General High-level Architecture

This general architecture shows how the smart cameras and other IoT devices can fit into the overall smart home solutions.



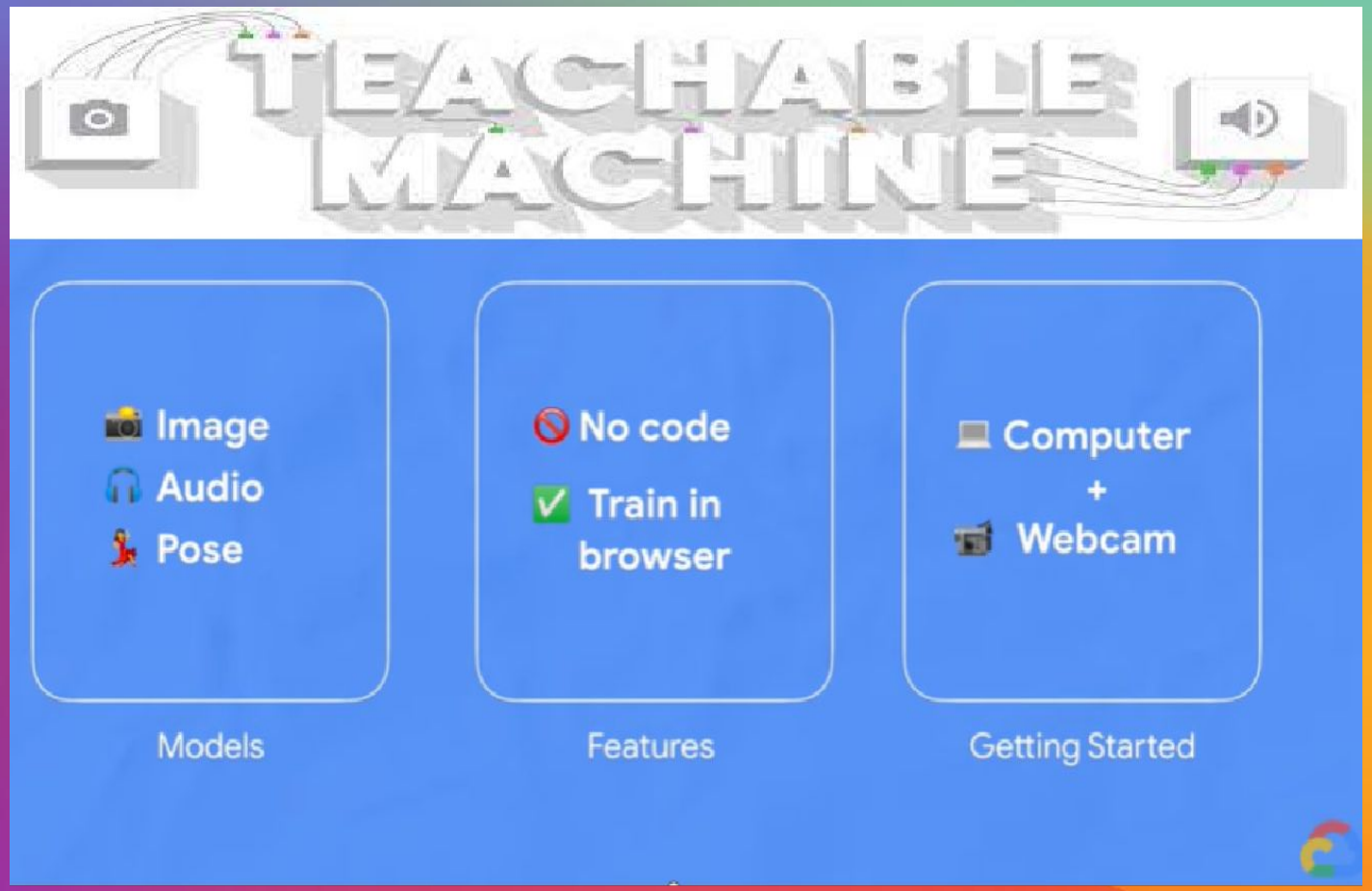
Project Scope and Tools

For the scope of the project, our team is focused on the part of computer vision and inference AI modeling in order to achieve performing models for the required visual detections.

Our team attempted modeling via 2 approaches as we would like to have an understanding of both approaches and their pros & cons:

- using low/no code tool approach
- using coding tool approach

1. Low/No-code Modeling tool: Teachable Machine (Tensorflow)
2. Coding modeling tool: Python and the relevant libraries
3. Integrated Development Environment: Colab
4. Pictures pre-processing Tool: Roboflow
5. Online metrics Calculator for Confusion Matrix



Source https://teachablemachine.withgoogle.com/

Both the Image and Pose models are learning off from the pre-trained MobileNet models (trained using ImageNet Database)

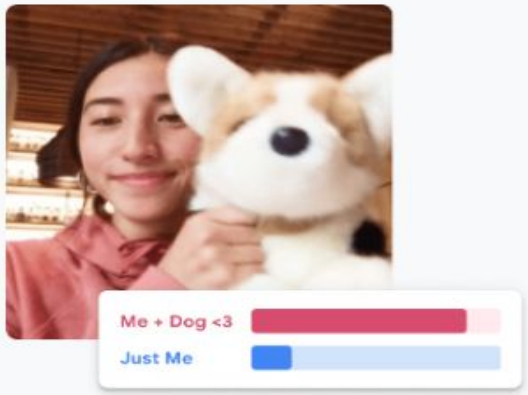
MobileNet - it's a Convolutional Neural Network designed for Mobile and embedded Vision applications.

Teachable Machine (No Code/ Low Code)

For this project, we are exploring TM Image and Poses pre-trained models.

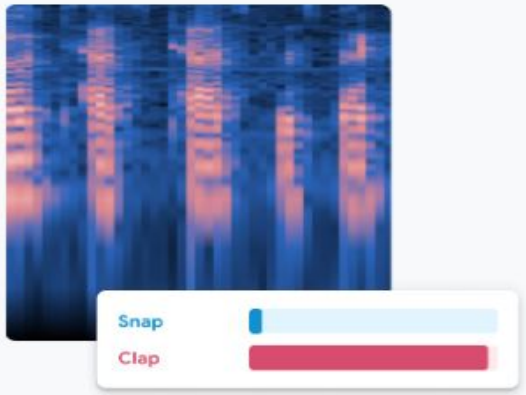
Image Classification - determining the type of object or scene is in a digital image.

Human Poses Classification — Based on Human daily activities, determining the Poses



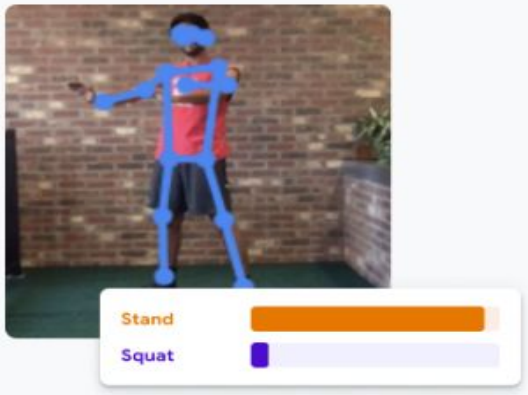
Images

Teach a model to classify images using files or your webcam.



Sounds

Teach a model to classify audio by recording short sound samples.

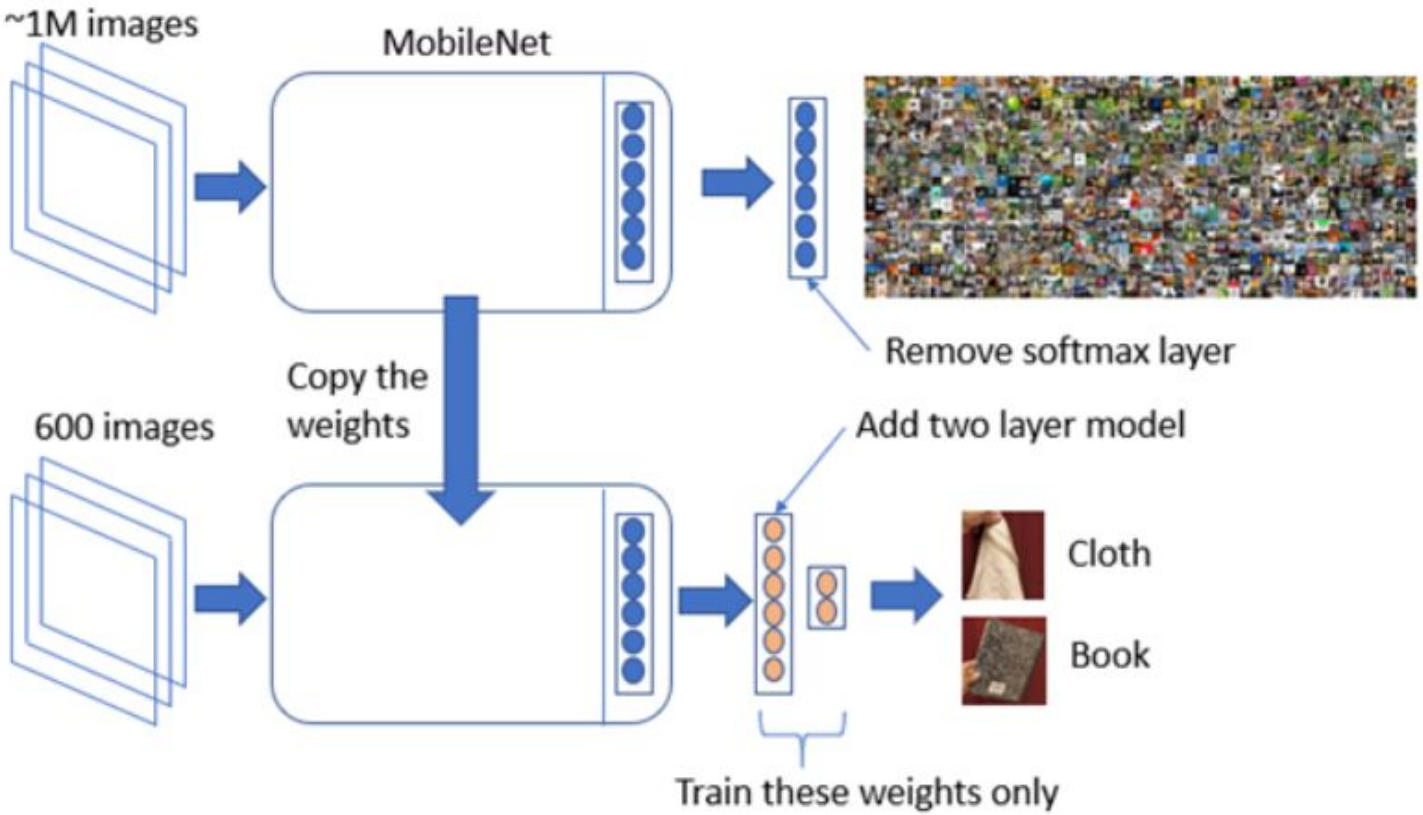


Poses

Teach a model to classify body positions using files or striking poses in your webcam.

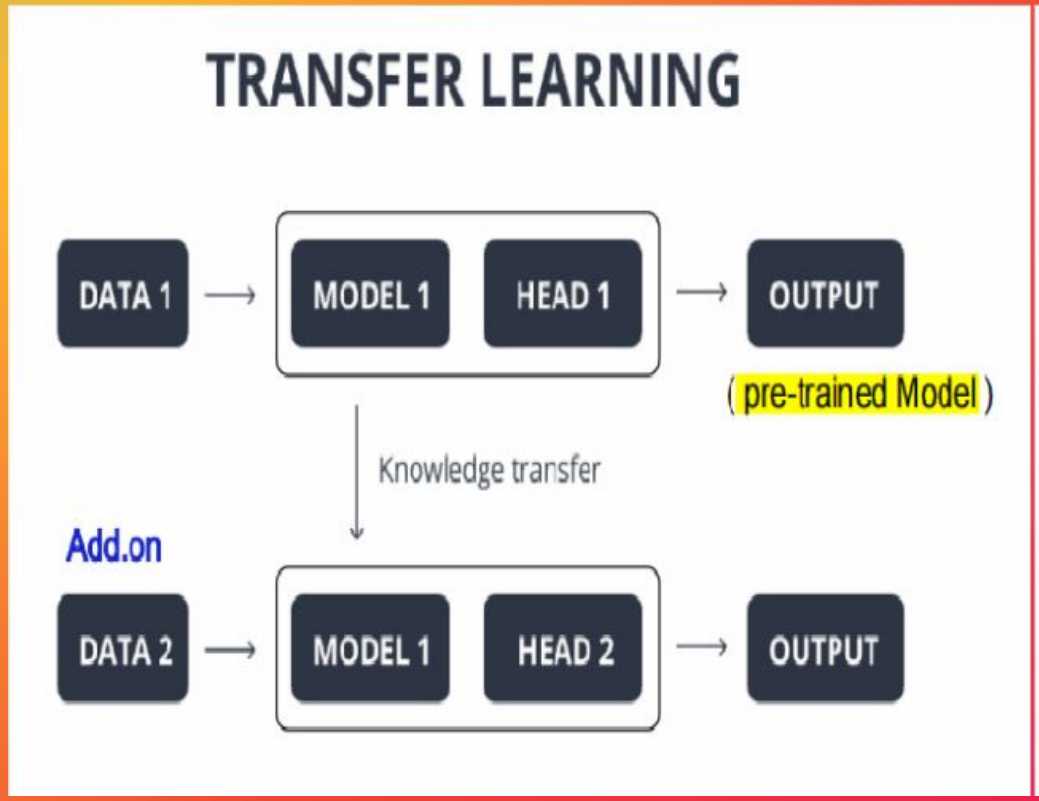
Source https://teachablemachine.withgoogle.com/

Architecture of pre-trained MobileNet



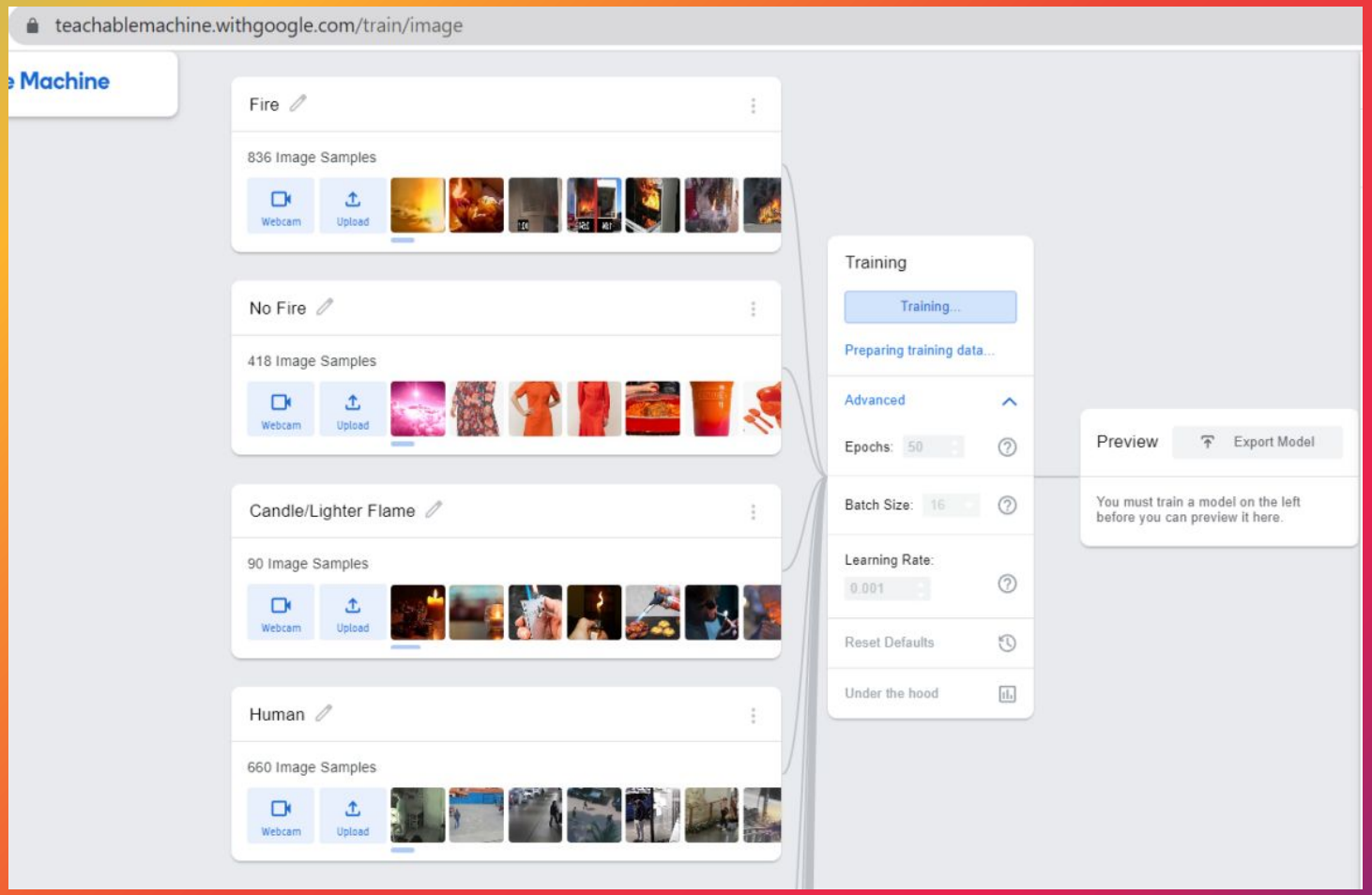
MobileNet paper and ImageNet Thumbnails

Transfer Learning Technique

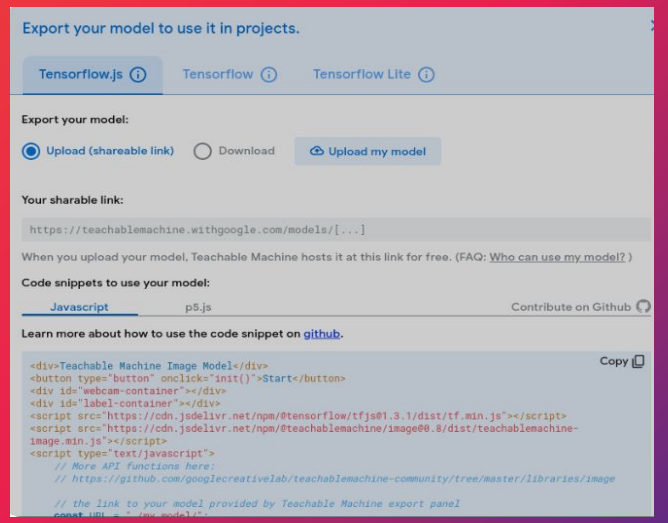


Source https://teachablemachine.withgoogle.com/

TM Modeling Process



https://teachablemachine.withgoogle.com/



Epochs

One epoch means that each and every sample in the training dataset has been fed through the training model at least once. If your epochs are set to 50, for example, it means that the model you are training will work through the entire training dataset 50 times. Generally the larger the number, the better your model will learn to predict the data.

You probably want to tweak (usually increase) this number until you get good predictive results with your model.

Batch Size:

A batch is a set of samples used in one iteration of training. For example, let's say that you have 80 images and you choose a batch size of 16. This means the data will be split into 80 / 16 = 5 batches. Once all 5 batches have been fed through the model, exactly one epoch will be complete.

You probably won't need to tweak this number to get good training results.

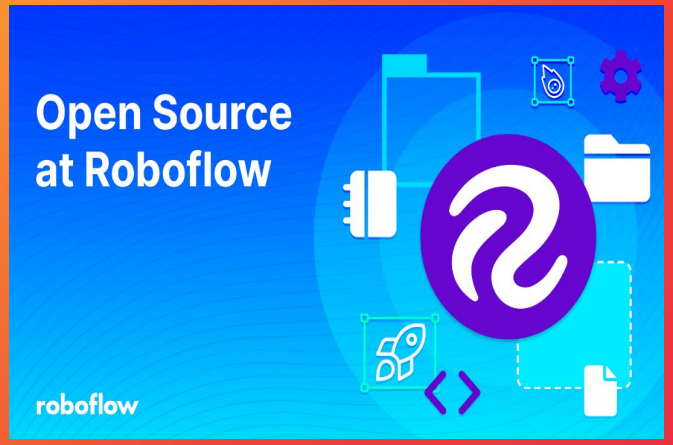
Learning Rate

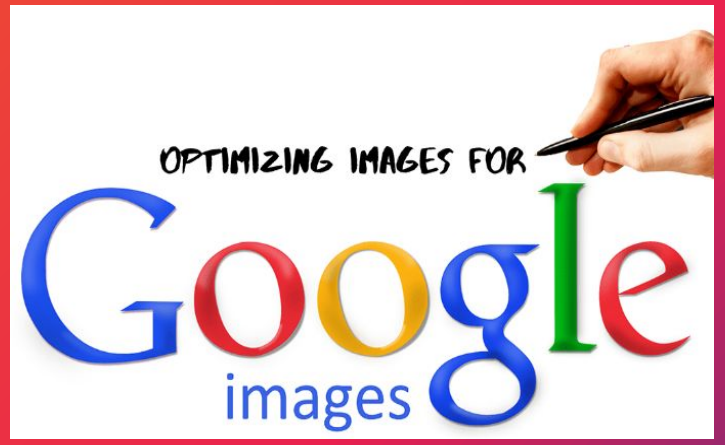
Be careful tweaking this number! Even small differences can have huge effects on how well your model learns.

Datasets

We are using datasets from:

1. https://universe.roboflow.com/
2. https://www.google.com/imghp
3. https://www.youtube.com/user/youtube/videos





Fire Training & Testing Samples

Fire Trained Images（836）

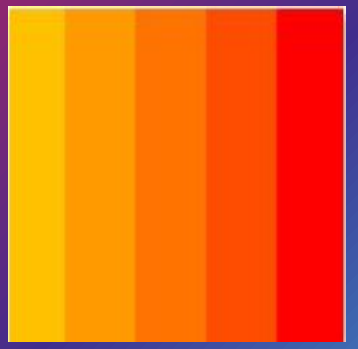





TEST Images（100 ）



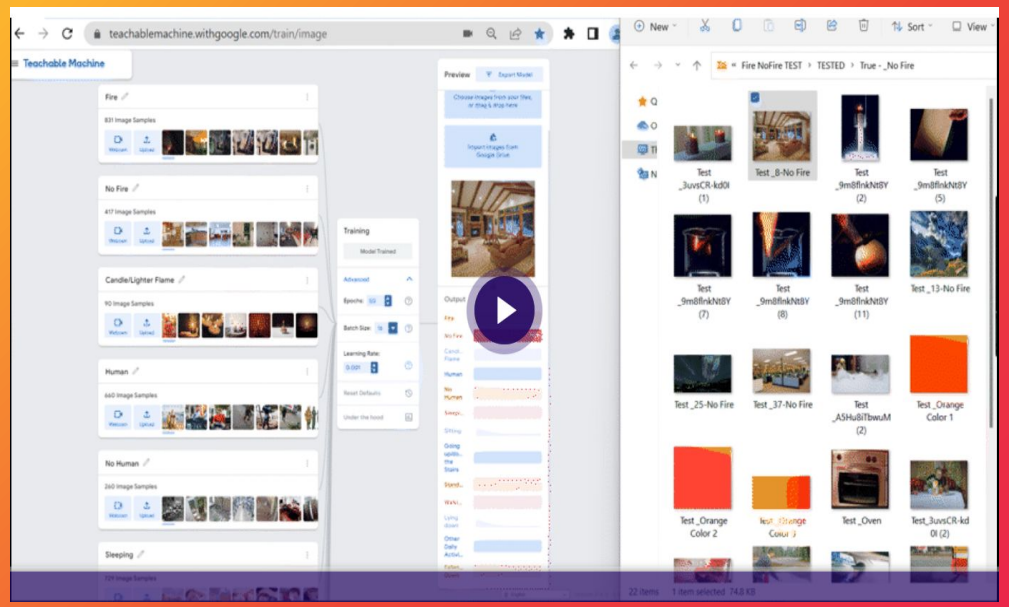


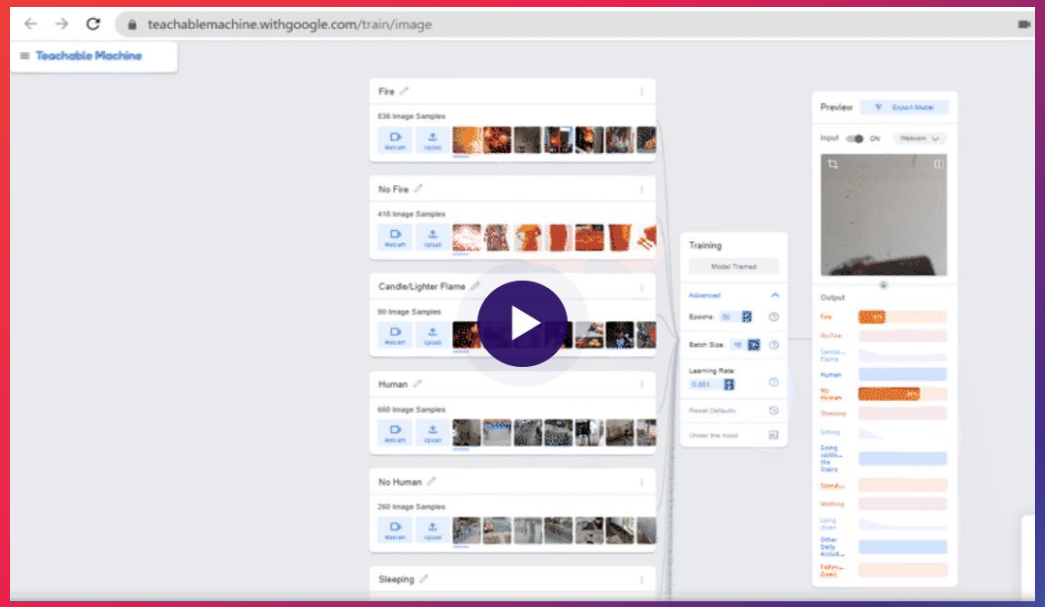
https://teachablemachine.withgoogle.com/models/kmGO-Dh2F/

Sample Fire Detection Outcome from TM

Test Sample Clips



Webcam Clips



Teachable Machine score for Fire

Confusion Matrix

Online Calculator

	True Positive	True Negative
Predicted Positive	74	1
Predicted Negative	5	21

Calculate

Measure	Value	Derivations
Sensitivity	0.9367	TPR = TP / (TP + FN)
Specificity	0.9545	SPC = TN / (FP + TN)
Precision	0.9867	PPV = TP / (TP + FP)
Negative Predictive Value	0.8077	NPV = TN / (TN + FN)
False Positive Rate	0.0455	FPR = FP / (FP + TN)
False Discovery Rate	0.0133	FDR = FP / (FP + TP)
False Negative Rate	0.0633	FNR = FN / (FN + TP)
Accuracy	0.9406	ACC = (TP + TN) / (P + N)
F1 Score	0.9610	F1 = 2TP / (2TP + FP + FN)
Matthews Correlation Coefficient	0.8414	TP*TN - FP*FN / sqrt((TP+FP)*(TP+FN)*(TN+FP)*(TN+FN))

Problems we face when doing this Project

Need to be cautious when it comes to using the color of a flame for fire detection. Because the impurities and other chemicals can cause a colored element of a flame that you might not have been expecting.

Typical reactions that cause specific flame colors are listed here, however, we must stress – this is not the complete list.

Strontium chloride/nitrate – can turn a flame RED.

Carbon and/or calcium chloride – can turn a flame ORANGE. (common reaction when you're making a fire outdoors for cooking or for warmth)

Sodium chloride (table salt), sodium carbonate and borax – can all turn flames YELLOW.

Copper and/or barium – may turn a flame GREEN

Copper chloride and totally consumed carbon – can result in a BLUE flame.

Indium – results in INDIGO flames.

Magnesium, Magnesium sulfate – can both lead to a bright WHITE flame.

Potassium nitrate, potassium sulfate together – burn with a VIOLET flame.

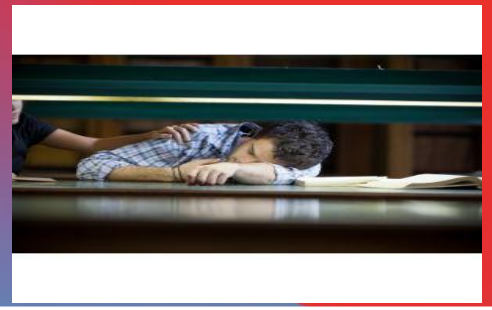
https://firefighterinsider.com/hottest-color-of-fire-flame/

Sleeping Training & Testing Samples

Sleeping Trained Images (729)





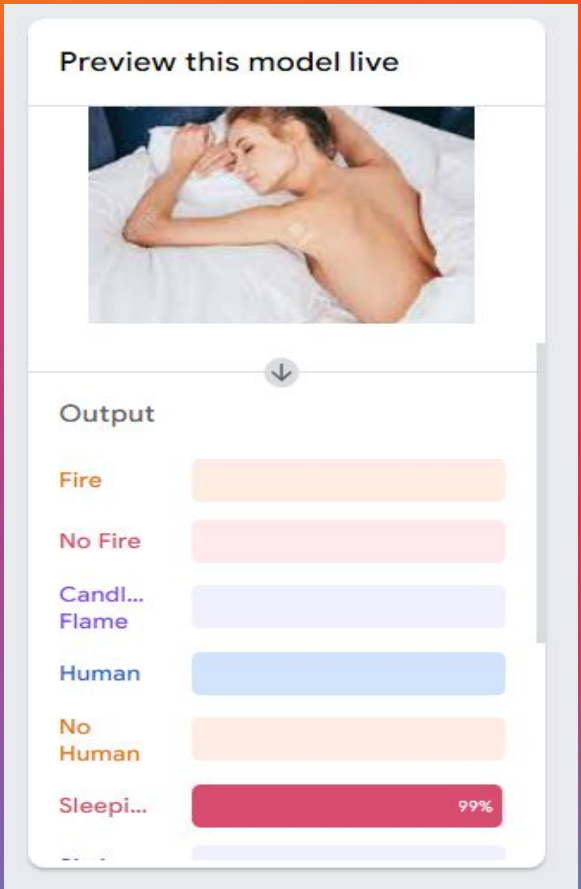



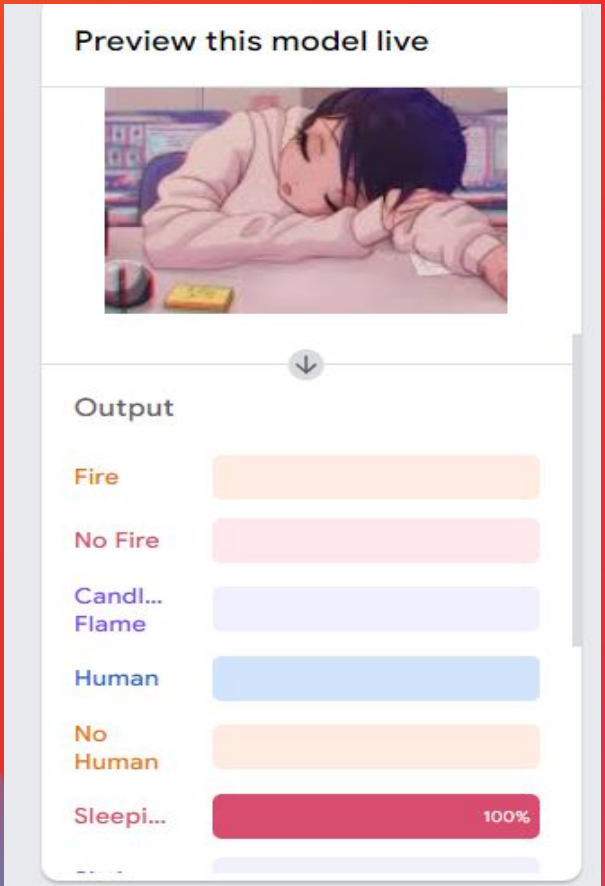
TEST Images (100)

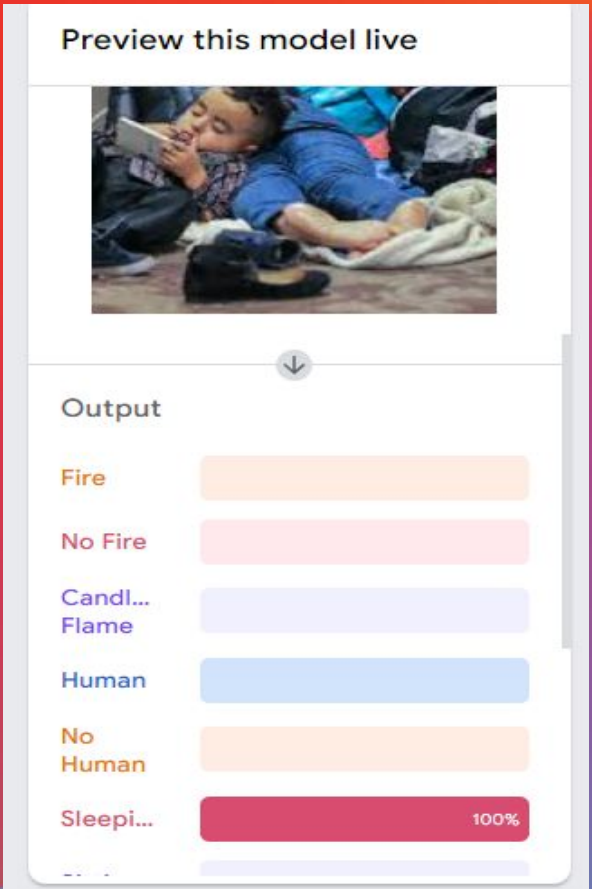





Result from TM for sleeping







Preview this model live

Output

Fire	
No Fire	
Candl... Flame	
Human	96%
No Human	

Preview this model live

Output

Fire	
No Fire	
Candl... Flame	
Human	
No Human	99%
Sleepi...	

Teachable Machine score for Sleeping

Confusion Matrix

Online Calculator

	True Positive	True Negative
Predicted Positive	64	6
Predicted Negative	19	11

Measure	Value	Derivations
Sensitivity	0.7711	TPR = TP / (TP + FN)
Specificity	0.6471	SPC = TN / (FP + TN)
Precision	0.9143	PPV = TP / (TP + FP)
Negative Predictive Value	0.3667	NPV = TN / (TN + FN)
False Positive Rate	0.3529	FPR = FP / (FP + TN)
False Discovery Rate	0.0857	FDR = FP / (FP + TP)
False Negative Rate	0.2289	FNR = FN / (FN + TP)
Accuracy	0.7500	ACC = (TP + TN) / (P + N)
F1 Score	0.8366	F1 = 2TP / (2TP + FP + FN)
Matthews Correlation Coefficient	0.3428	TP*TN - FP*FN / sqrt((TP+FP)*(TP+FN)*(TN+FP)*(TN+FN))

Sleeping detection Conclusion

TM demonstrates that it can recognised sleeping quite accurately if image of the person is on the bed. However when we use image of person sleeping in the car, lying on the table. It could not detect the person is sleeping . Sometimes it might even detect as “No human”.

We think that happen because we lump all classification together hence causing the TM to have too many model to go through.

In the future update, we will separated the classification into each individual as to avoid the TM from having too many classification and add in more image to better the score.

Humans Training & Testing Samples

Humans Trained images (899)





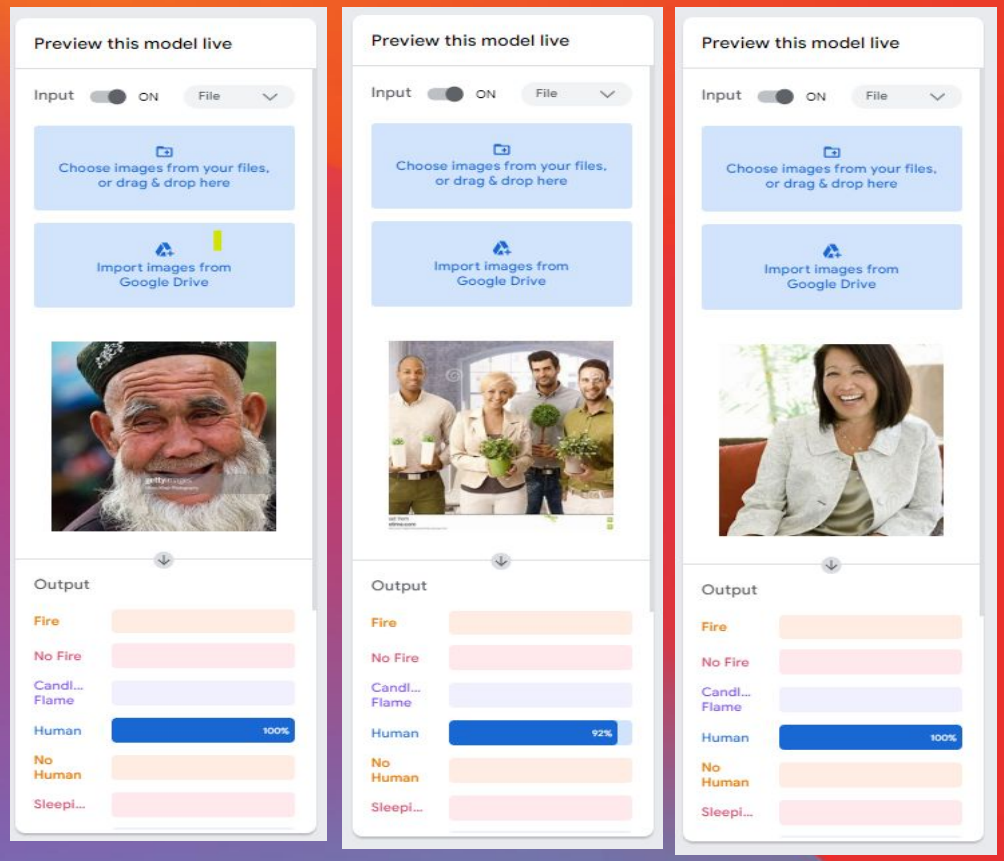


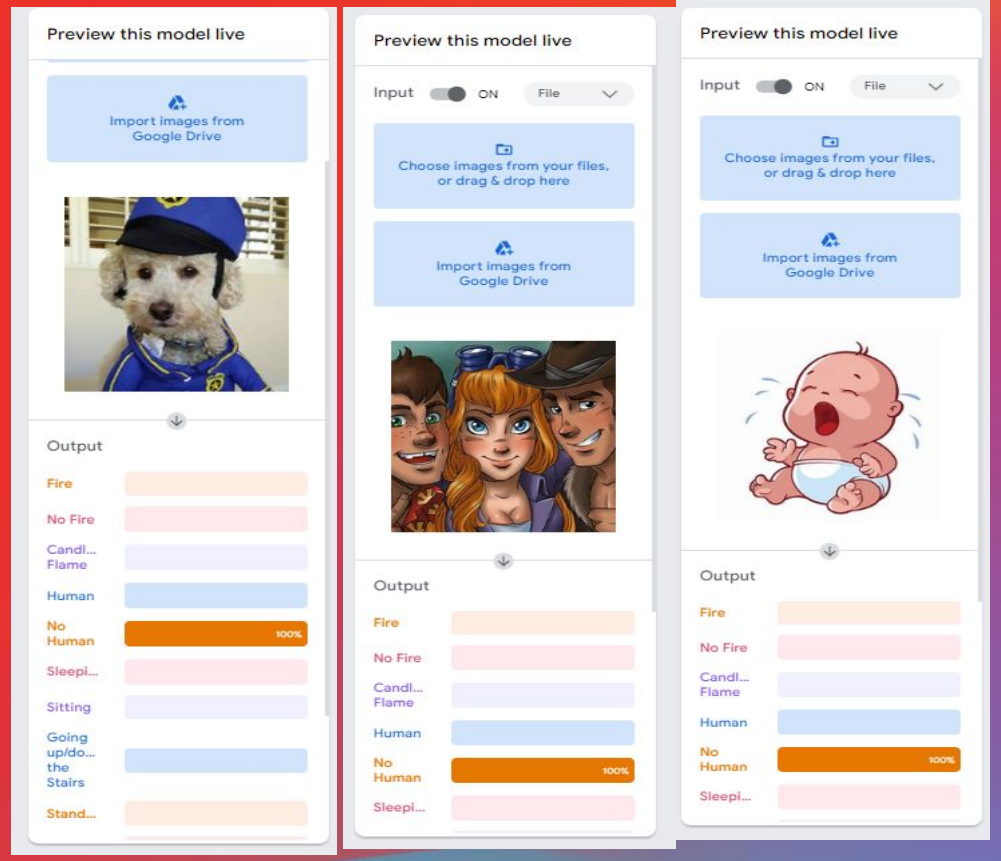


Humans Testing (100)




Results from TM for Humans (NonHumans)





Teachable Machine score for Humans

Confusion Matrix

Online Calculator

	True Positive	True Negative
Predicted Positive	70	5
Predicted Negative	15	10

Measure	Value	Derivations
Sensitivity	0.8235	TPR = TP / (TP + FN)
Specificity	0.6667	SPC = TN / (FP + TN)
Precision	0.9333	PPV = TP / (TP + FP)
Negative Predictive Value	0.4000	NPV = TN / (TN + FN)
False Positive Rate	0.3333	FPR = FP / (FP + TN)
False Discovery Rate	0.0667	FDR = FP / (FP + TP)
False Negative Rate	0.1765	FNR = FN / (FN + TP)
Accuracy	0.8000	ACC = (TP + TN) / (P + N)
F1 Score	0.8750	F1 = 2TP / (2TP + FP + FN)
Matthews Correlation Coefficient	0.4042	TP*TN - FP*FN / sqrt((TP+FP)*(TP+FN)*(TN+FP)*(TN+FN))

Conclusion of TM for Humans

Our TM could **recognize Humans vs Non-Humans** more accurately after **adding Training Data images** of : (**concept of Transfer Learning)**

* Humans with beards; elderly with wrinkled faces, humans dressed in Traditional costumes (faces visible)
* Under Non-Humans – images of 'animals' (e.g dogs with clothes); cartoon characters which look human

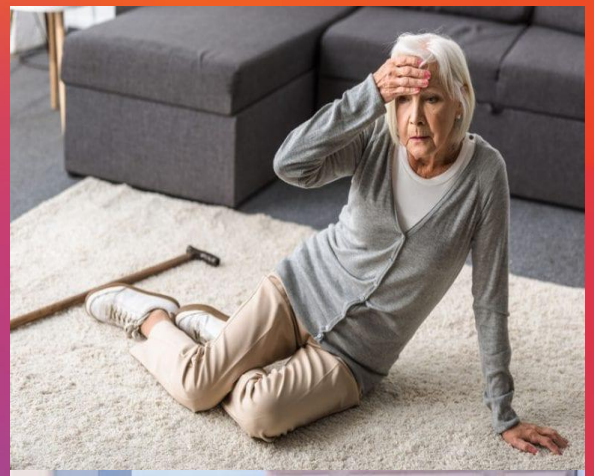
However, if a picture has **'multiple' objects** in it, the **TM response** may be **'inaccurate'** as it is **unable to determine the 'intended' object focus in the picture.** The tool currently does not have the feature to pre-select what are the 'main objects' into 'bounding boxes' (if option is required).

In our Project, we had 'Trained' our TM model with
* Images from other Classifications (e.g 'Walking', 'Sitting', 'Standing', 'Sleeping' , 'Other Daily Activities' ;' Up/Down the Stairs') which depicts other human 'actions' (may not be "mutually exclusive) .
E.g. An image of a human(s) ' exercising' in a picture , may be depicted as ' Walking'

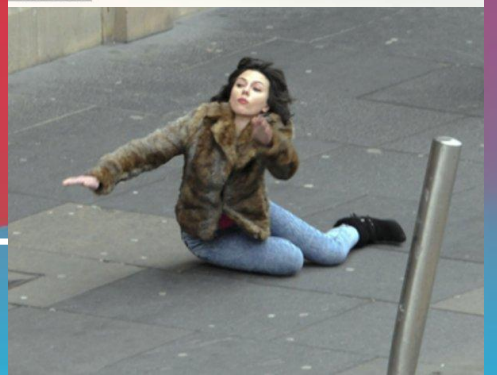
TM is a very easy to learn and quick to deploy Low Code/No Code Modeling Tool. It meets the objective of identifying the Human Classification fairly accurately , however, **it requires lots of relevant ' cleansed' data to Train the model .**

Fallen Down Training & Testing Samples

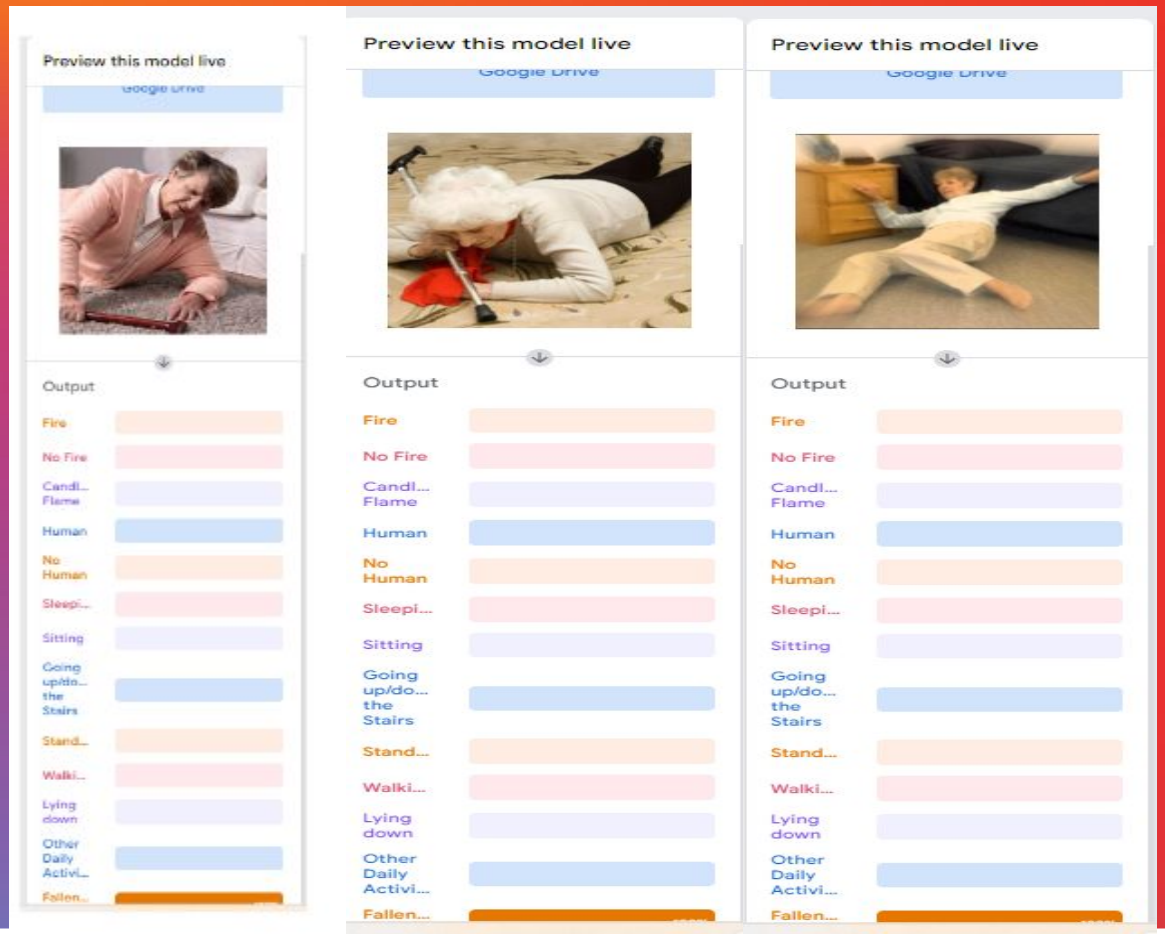
Fallen Down Trained images

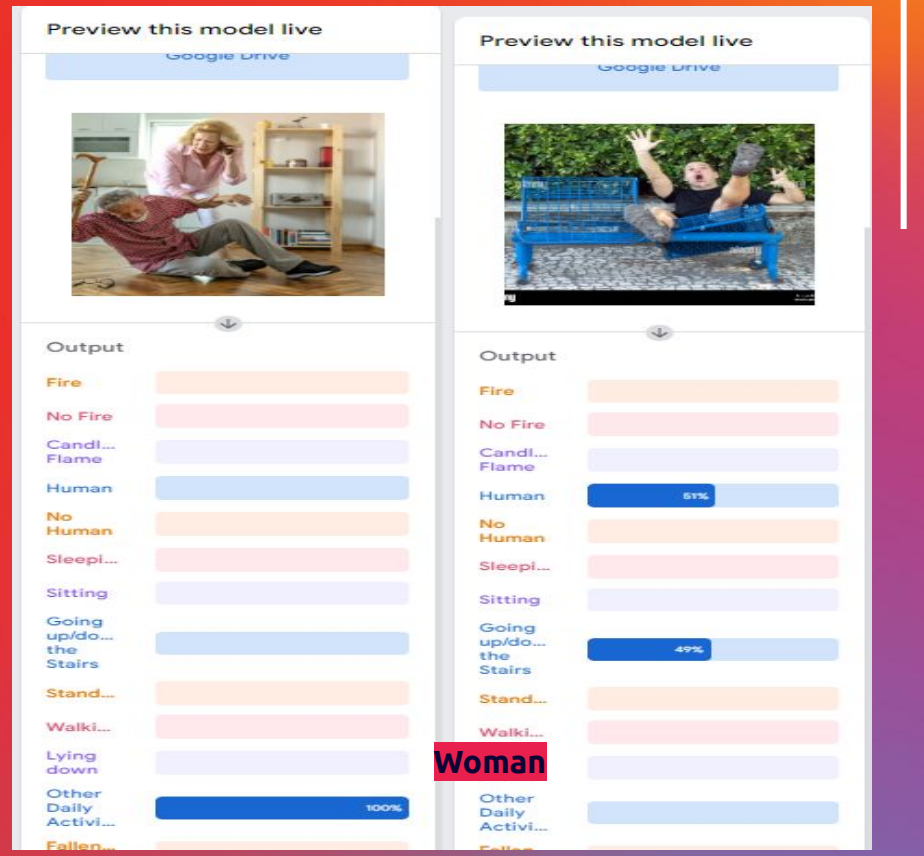



Fallen Down Testing Images


Results from TM for Fallen Down Images





Woman

1.. Woman helping Man in ‘ Other Active Daily Activities” ?
2. Man ‘falling down stairs” ?

Teachable Machine score for Falls

Confusion Matrix

Online Calculator

	True Positive	True Negative
Predicted Positive	72	1
Predicted Negative	26	1

Measure	Value	De
Sensitivity	0.7347	TPR =
Specificity	0.5000	SPC =
Precision	0.9863	PPV =
Negative Predictive Value	0.0370	NPV =
False Positive Rate	0.5000	FPR =
False Discovery Rate	0.0137	FDR =
False Negative Rate	0.2653	FNR =
Accuracy	0.7300	ACC = (T
F1 Score	0.8421	F1 = 2TP
Matthews Correlation Coefficient	0.0740	TP*TN - FP*FN / sqrt (

Conclusion of TM for Fallen Down

The TM is able to recognize **Fallen Down images** fairly accurately which included Static images of :

* **Humans that have fallen and are lying down; recover from a fall; stage of falling down the stairs**

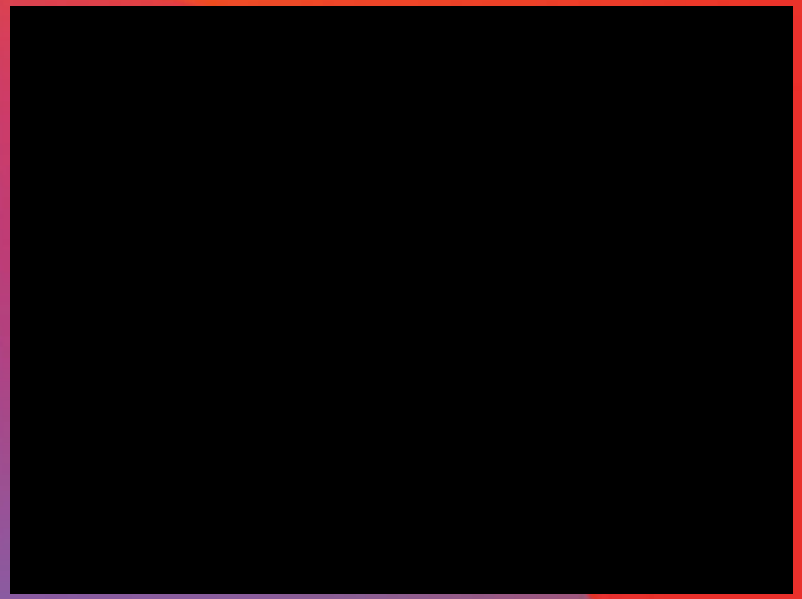
* **Inaccurate results do occur when surrounding 'objects' in the images are 'confusing' or 'difficult to detect'**

TM detects 1. Man seated on 'broken wooden chair' ,—> Humans and 'Going up/down the stairs'
2. Woman assisting fallen down Man → Other Active Daily Activities

* Picture Images shown to the TM , ideally should not contain too many 'objects' within the picture. (as this can be confusing as to which 'object' is the main focus (since there is no feature of 'bounding boxes')

The Tool 's current limitation is **NOT being able to capture moving 'videos' of images - accurately.**

Results of Fallen Down images from TM Sample Video Clips



Images of People Fallen Down or Falling

Video 1 - -50s (17Aug))

Video 3 - - 3min (17Aug))

Video 2 - 2.5min (17Aug))

Summary of TM - Benefits and Feedback

Benefits:

* It requires significantly **less Computing power, a smaller dataset with no Coding required.**

* Entire process of gathering data, training, re-training the model is done inside the Browser
* In term of **Data privacy, all data is created locally via Webcam or Microphone**
  the model itself allow <Download> locally,
  as User - we have the choice and rest assured that our data is safe, available only in our local server.
* **TM Provides support for JavaScript (using Tensorflow.json), Python projects (using Tensorflow) and**
  Android applications (using Tensorflow Lite)

  We also have abundance of options to use them in IoT Projects

Feedback (for new features)

* The only limit to what we can do with TM is our Imagination, what could make it even better..:

  > Datasets with video images

  > Option to define (in pictures with multiple items), the 'main objects' into 'bounding boxes'

Computer Vision with Python

- Programming using Python to work with the Computer Vision libraries in Colab (which also provides GPU for processing).
- For human / sleep / fire detection cases, PyTorch/YOLO model was used for object detection
- For fall detection case, Keras/Tensorflow/OpenCV was used for action inference
- Other common libraries include numpy, matplotlib etc.

Computer Vision with Python

- Leveraged on transfer learning based on pre-trained model from YOLO for human detection
- Perform training/modeling for sleep detection, fire detection and fall detection scenarios

Deep learning

Typical CNN overview

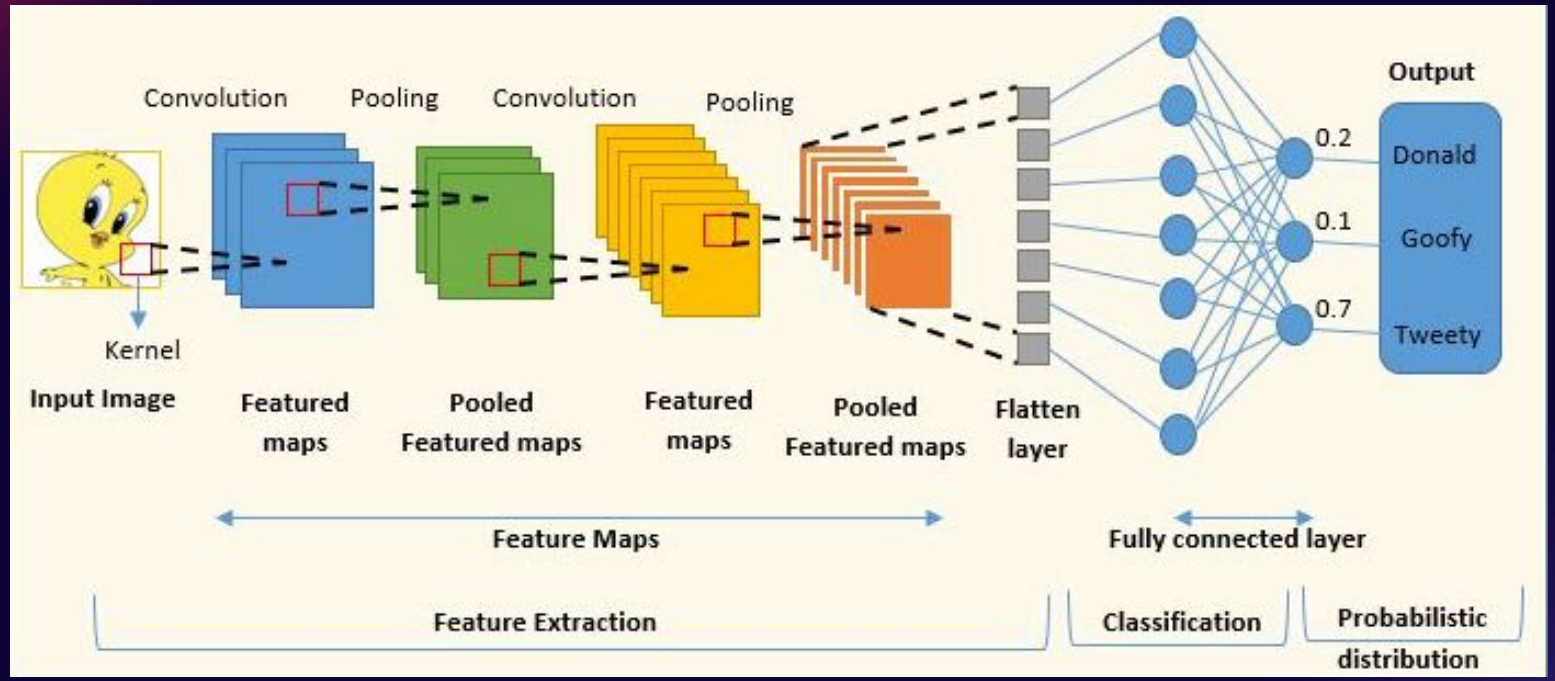


image courtesy of www.analyticsvidhya.com

In our case, we are adopting supervised learning

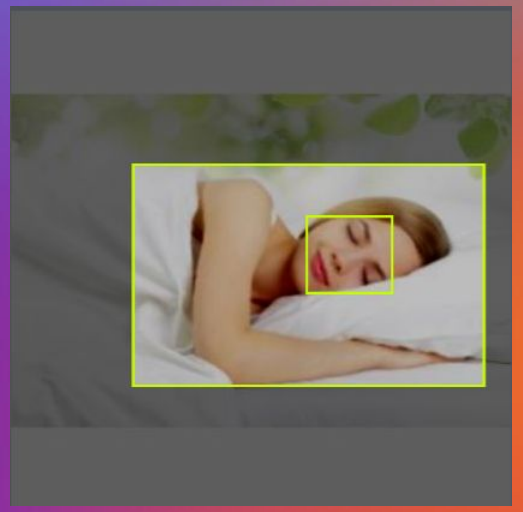
Dataset

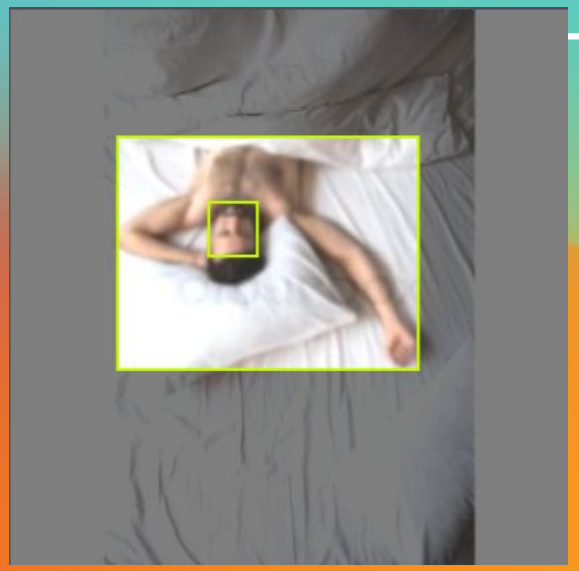
- Source of data images:
  - Kaggle - human activity detection dataset (https://www.kaggle.com/datasets/jithinnambiarj/human-activity-detection-dataset)
  - Internet (Google images)
- Source of data videos:
  - Fall detection dataset (http://fenix.univ.rzeszow.pl/~mkepski/ds/uf.html)

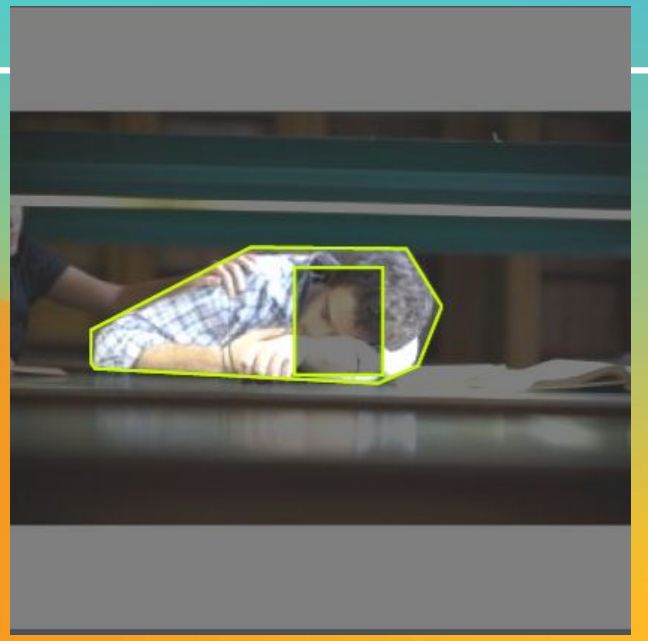
Dataset

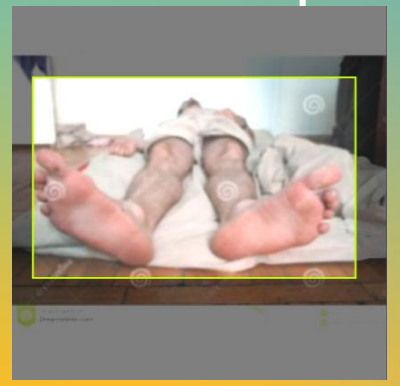
- Using Roboflow for preprocessing of images which includes:
  - Bounding
  - Annotating / Classifying
  - Resizing
  - Image augmentation - such as blurring, brightness augmentation, hue augmentation, cutouts
  - Separation into training/validation/test sets

Samples

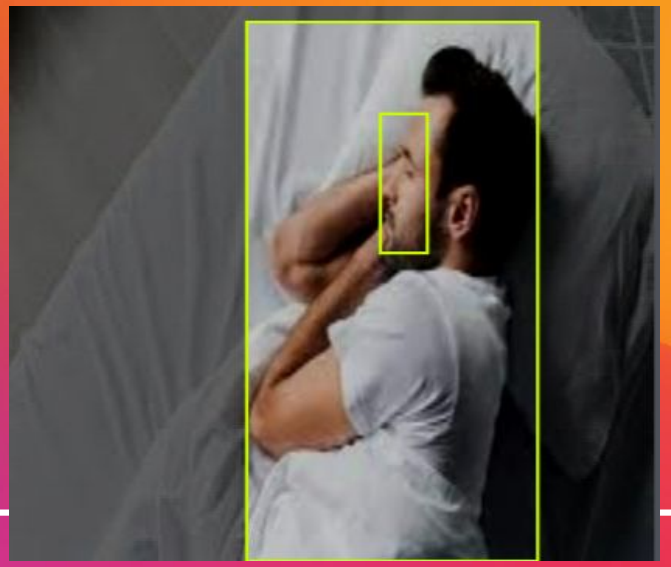


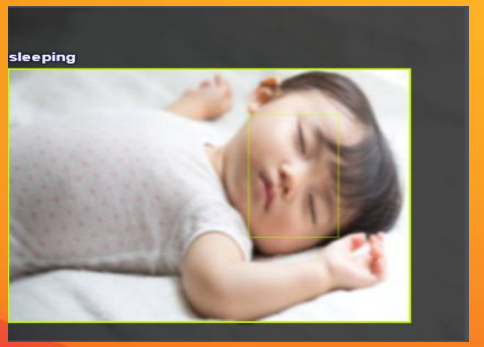


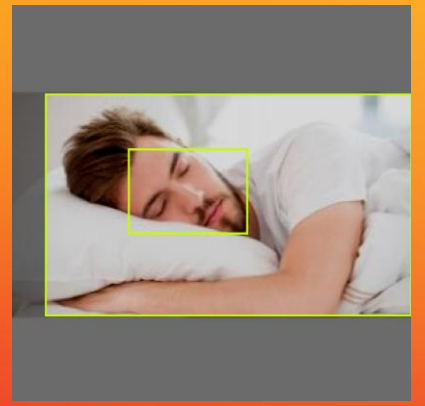




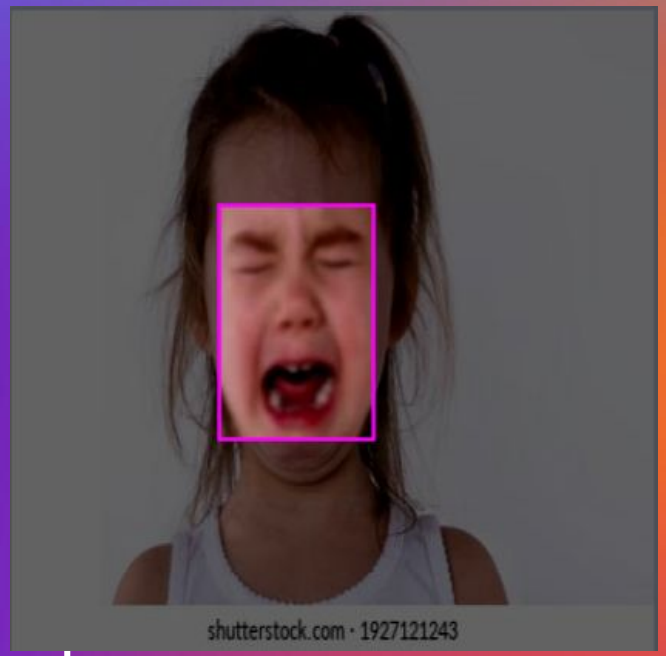
zzzZZ

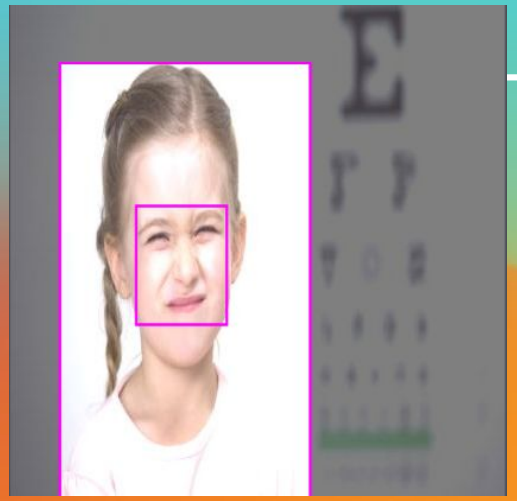


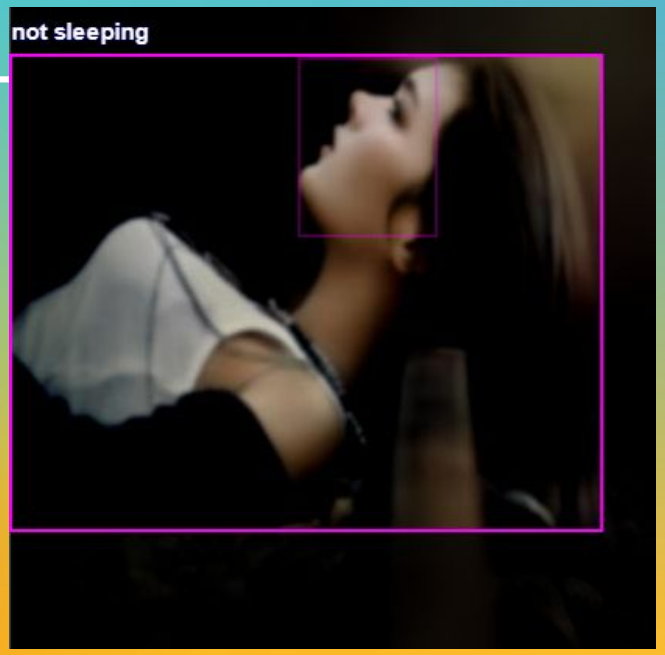


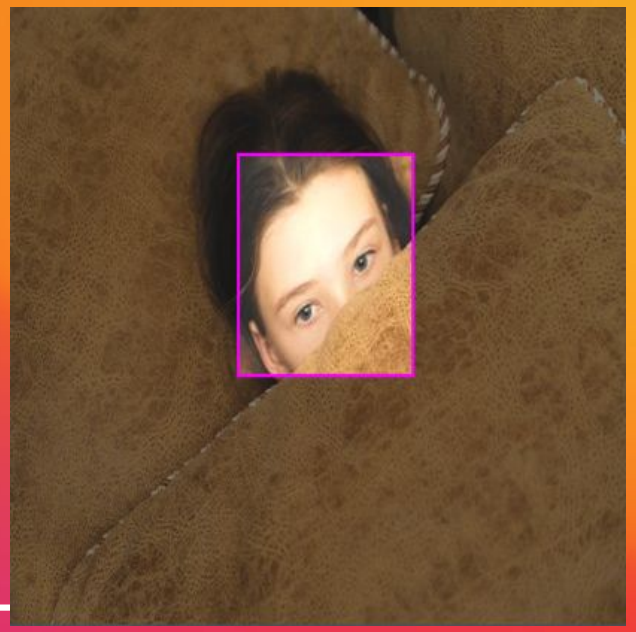


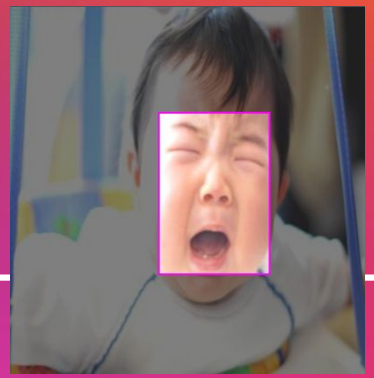
Samples











Nope.. I’m not asleep

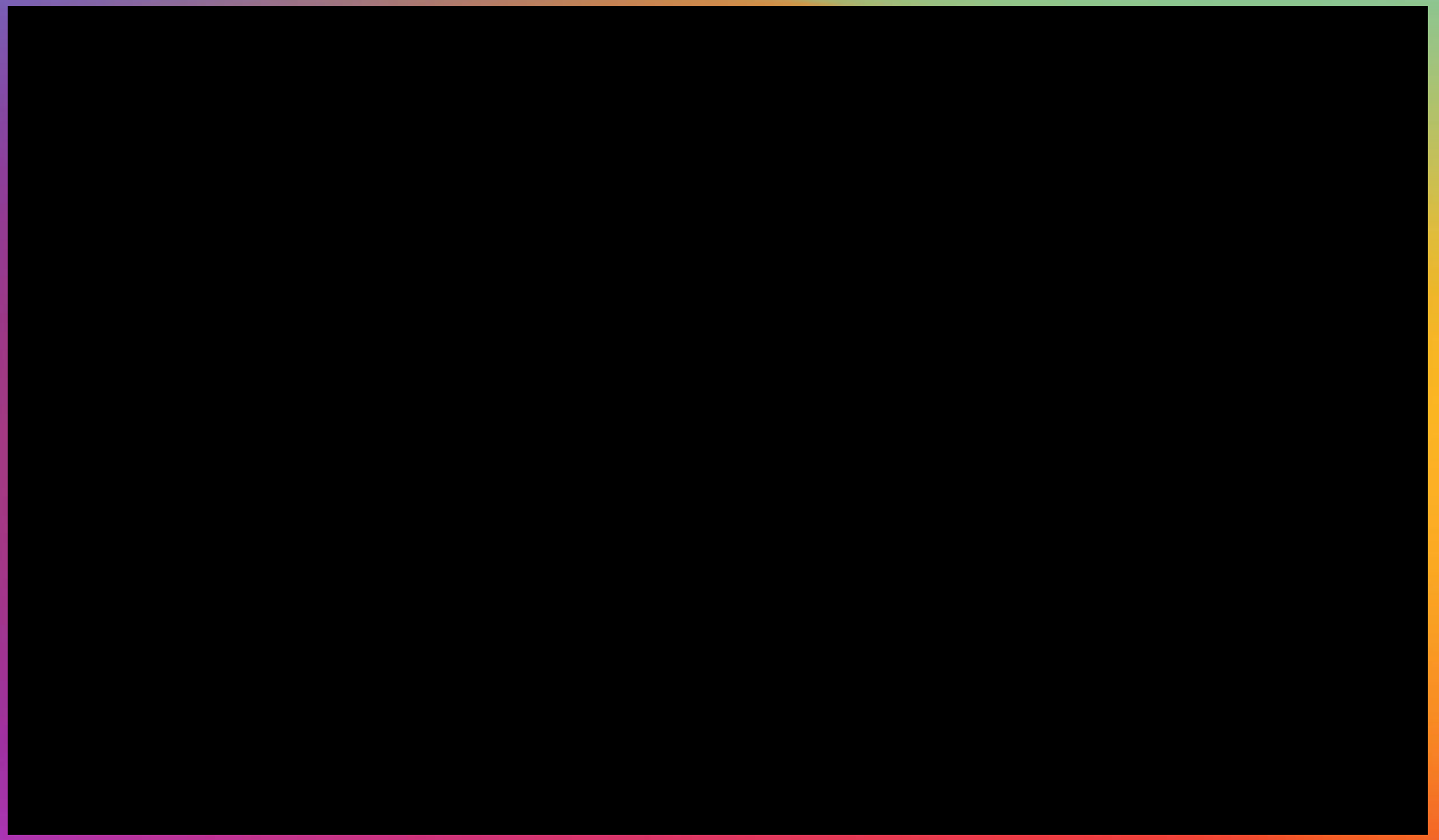
Samples

Fiery!!!

Samples

All good...

Samples



Positive fall training video

Samples

Negative fall training video

Dataset

- Quantity of dataset images used for training:
  - Sleep detection - Total 317 images. Sleeping: 190, Not sleeping 127. **With augmentation, total images = 536**.
  - Fire detection - Total 302 images. Fire: 174, Null 128. **With augmentation, total images = 512**
  - Fall detection - **Total 60 videos.** Fall: 30, Not Fall: 30.
- Human / person detection - Training dataset not required

Modeling process flow (obj detection)

For fire/sleep detections:

1. Coding via Python in Colab
2. Import annotated images sets (train/validation/test) from Roboflow (70/20/10)
3. Train via YOLO model (batch size 2, epoch 50, other hyp parameters default)
4. Test the model with test set
5. Real time detection via video test

For human detections, same as above but skipped steps 2&3 as it is based on pretrained model.

Modeling process flow (act detection)

For fall detections:

1. Coding via Python in Colab
2. Import video data and convert to array capturing the sequences keypoints
3. Split into train/test set in Python (90/10)
4. Train via Keras model (lr 0.001, epoch 50, Adam optimizer)
5. Test the model with test set
6. Real time detection via video test

Metrics for trained models

- Fire detection model metrics on validated set:

```
Model summary: 213 layers, 7012822 parameters, 0 gradients, 15.8 GFLOPs
               Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 15/15 [00:00<00:00, 16.43it/s]
                 all         59         57      0.766      0.439      0.514      0.277
```

- Sleep detection model metrics on validated set:

```
Model summary: 213 layers, 7015519 parameters, 0 gradients, 15.8 GFLOPs
               Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 15/15 [00:01<00:00, 11.24it/s]
                 all         59        107      0.715      0.692      0.736      0.424
        not sleeping         59         48      0.794      0.875       0.92      0.572
             sleeping         59         59      0.635      0.508      0.553      0.276
```

- Sleep detection model metrics on test set:

	True Positive	True Negative
Predicted Positive	14	1
Predicted Negative	1	23

Measure	Value	Derivations
Sensitivity	0.9333	TPR = TP / (TP + FN)
Specificity	0.9583	SPC = TN / (FP + TN)
Precision	0.9333	PPV = TP / (TP + FP)
Negative Predictive Value	0.9583	NPV = TN / (TN + FN)
False Positive Rate	0.0417	FPR = FP / (FP + TN)
False Discovery Rate	0.0667	FDR = FP / (FP + TP)
False Negative Rate	0.0667	FNR = FN / (FN + TP)
Accuracy	0.9487	ACC = (TP + TN) / (P + N)
F1 Score	0.9333	F1 = 2TP / (2TP + FP + FN)
Matthews Correlation Coefficient	0.8917	TP*TN - FP*FN / sqrt((TP+FP)*(TP+FN)*(TN+FP)*(TN+FN))

- Fire detection model metrics on test set:

	True Positive	True Negative
Predicted Positive	19	3
Predicted Negative	1	10

Measure	Value	Derivations
Sensitivity	0.9500	TPR = TP / (TP + FN)
Specificity	0.7692	SPC = TN / (FP + TN)
Precision	0.8636	PPV = TP / (TP + FP)
Negative Predictive Value	0.9091	NPV = TN / (TN + FN)
False Positive Rate	0.2308	FPR = FP / (FP + TN)
False Discovery Rate	0.1364	FDR = FP / (FP + TP)
False Negative Rate	0.0500	FNR = FN / (FN + TP)
Accuracy	0.8788	ACC = (TP + TN) / (P + N)
F1 Score	0.9048	F1 = 2TP / (2TP + FP + FN)
Matthews Correlation Coefficient	0.7455	TP*TN - FP*FN / sqrt((TP+FP)*(TP+FN)*(TN+FP)*(TN+FN))

- Fall detection model metrics on test set:

```
array([[[2, 1],
        [0, 3]],
```

=

	Predicted False	Predicted True
Actual False	2	1
Actual True	0	3

Measure	Value	Derivations
Sensitivity	1.0000	TPR = TP / (TP + FN)
Specificity	0.6667	SPC = TN / (FP + TN)
Precision	0.7500	PPV = TP / (TP + FP)
Negative Predictive Value	1.0000	NPV = TN / (TN + FN)
False Positive Rate	0.3333	FPR = FP / (FP + TN)
False Discovery Rate	0.2500	FDR = FP / (FP + TP)
False Negative Rate	0.0000	FNR = FN / (FN + TP)
Accuracy	0.8333	ACC = (TP + TN) / (P + N)
F1 Score	0.8571	F1 = 2TP / (2TP + FP + FN)
Matthews Correlation Coefficient	0.7071	TP*TN - FP*FN / sqrt((TP+FP)*(TP+FN)*(TN+FP)*(TN+FN))

```
accuracy_score(ytrue, yhat)
```

```
0.8333333333333334
```

Demo

People detection video 1

Demo

People detection video 2

Demo

Sleeping / Not sleeping detection video

Demo

Fire detection video 1

Demo

Fire detection video 2

Demo

Fall detection video 1

Demo

Fall detection video 2

Demo

Fall detection video 3

Pros vs Cons of using Computer Vision with Python

Pros

- Allows for tweaking of more hyperparameters
- Shows bounding boxes, together with any misdetection or wrongly located bounding boxes (localisation+classification)
- Allows for more types of use cases other than object/pose classification, such as action recognition, face recognition, user tracking, multi-objects detection etc
- Allows for continuous real time detection (as evidenced from the videos)

Cons

- Much more time-consuming compared to low/no code
- Requires coding
- In certain use cases, low/no code model may already be sufficient

Next steps

- Performance at this stage can still be improved for both platforms(TM and Python).
- Either of the platform models can potentially be implemented so long as they meet the business needs/required outcomes.
- Discussion and performance review will need to be performed with stakeholders. When the computer vision model is approved for its performance, it could then be incorporated into smart home integrated solutions before deployment to the cloud.

Other points to note

- Improvements that can be made to the modelling:
  - **Increase in data quantity.** Due to time (2-3 weeks) and resource constraints (4 members), the team have not been able to collect and feed more data to the model (ideally thousands, tens of thousands or even million of images). Similarly, a total of only 60 videos are used for training the fall detection modeling.
  - **Quality of data.** In some cases the quality of the data may not be comprehensive enough (eg. for fire case, many images showing fire in dark background), or may also require further refinements to reduce false positives /improve on the performance (eg. for fall detection training videos, may need to isolate only the falling sequence.)
  - Test under **other challenging test scenarios / environments** to improve model performance (eg performance under different lighting conditions)
  - **Greater diversity in training dataset** (eg. include a good diverse set of ethnicity to reduce model biasness and improve performance)

Other points to note

- Possible challenges / resistance in smart camera for home installations:
  - Privacy issues
  - Smart home solutions are not matured & uptake is still relatively low, at least in Singapore. Although this means that there is potential for higher uptake, however there may be many who are currently averse to installation of smart cameras at home
  - CyberSecurity issues

- Possible added features that can be incorporated in the future into the smart camera solutions:
  - Blurring of facial / body features of the camera feed for privacy
  - Facial recognition to recognize home owners vs strangers
  - User tracking
  - Area zoning
  - Smoke detection