

forked from [djdunc/casa0018](#)

Code

Pull requests

Actions

Projects

Wiki

Security

Insights

Settings

main ▼

...

[casa0018](#) / [Assessment](#) / [Report](#) / [Final_report.md](#)



JunzheGan Update Final_report.md



1 contributor

99 lines (57 sloc) | 9.58 KB

Human Facial emotion detection model

Introduction

The inspiration of this project is from PwC Game-based Assessment. One of their tasks is asking candidates to match human face showing leftwards with correct emotion showing on the right.



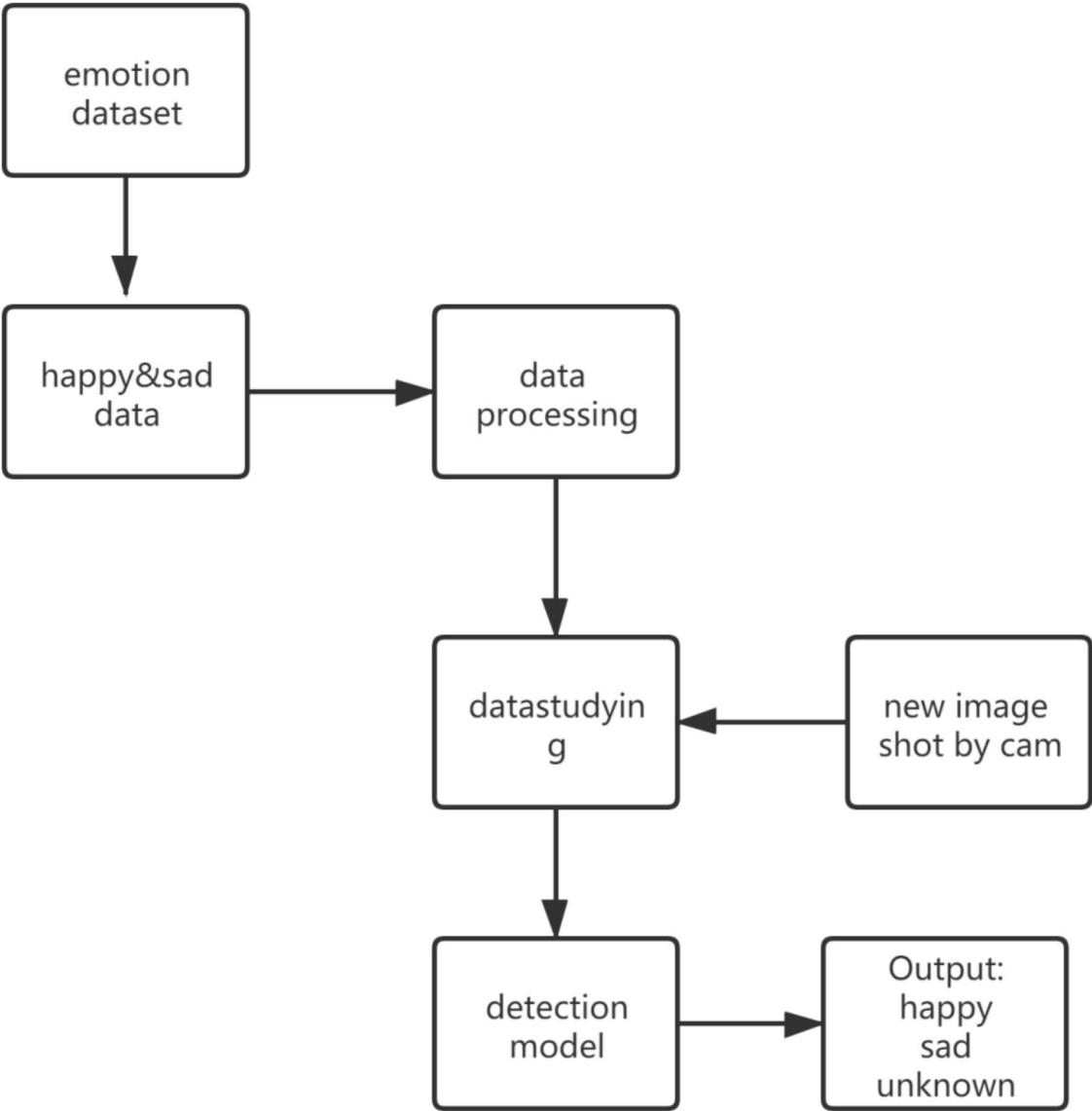
This has motivated me to use deep learning techniques building a model which can swiftly detect human facial emotions. This could be used on analysis of audience feeling as the online meeting via zoom and other platform has replaced the traditional face-to-face meeting during the pandemic of covid-19. Through research online, similar technique has been widely applied in retail business. As Vihar suggested, knowing campaign effectiveness through facial behavioural analytics with AI enabled camera on digital signage has enhanced customers' experience. Facial recognition with deep learning algorithms help retailers to understand different moods and expressions of shoppers in response to various discounts and promotional offers, so they can design better promotional campaigns. Nevertheless, Walmart has also adopted this technology, it classifies facial expressions of customers at checkout lines with AI-based algorithms measuring the service satisfaction index. Moreover, Iva Djukić, a marketing specialist has found that business spokesperson with positive emotions will make their brand more appealing to customers. Thus, it is very essential to use deep learning model to study human emotions.

Research Question

Can I build a facial emotion detection model to determine whether the target is happy or sad?

Application Overview

The project is built through three main steps. The first step is data gathering. The initial data was sourced online with much more content than this project needs. Data selection has been carried out to ensure the image quality and the general pattern among "happy" and "sad" are distinctive. The second step is imported data into Edge Impulse, which is a very convenient online source for training and developing deep learning models. The image data was divided into training and testing based on purpose as well as "happy" and "sad" based on properties. The data was then studied and trained by a pre-trained image classification model. The final model is derived by selecting the best performing model through the initial study and verify with the validation dataset(which just being separated from the training data). With the newly built model, live classification through web cam can be conducted to see how will the model perform with unfamiliar data. After the model studying new image, it will provide its identification of the image, and the output should fall within "happy", "sad", or "unknown".



Data

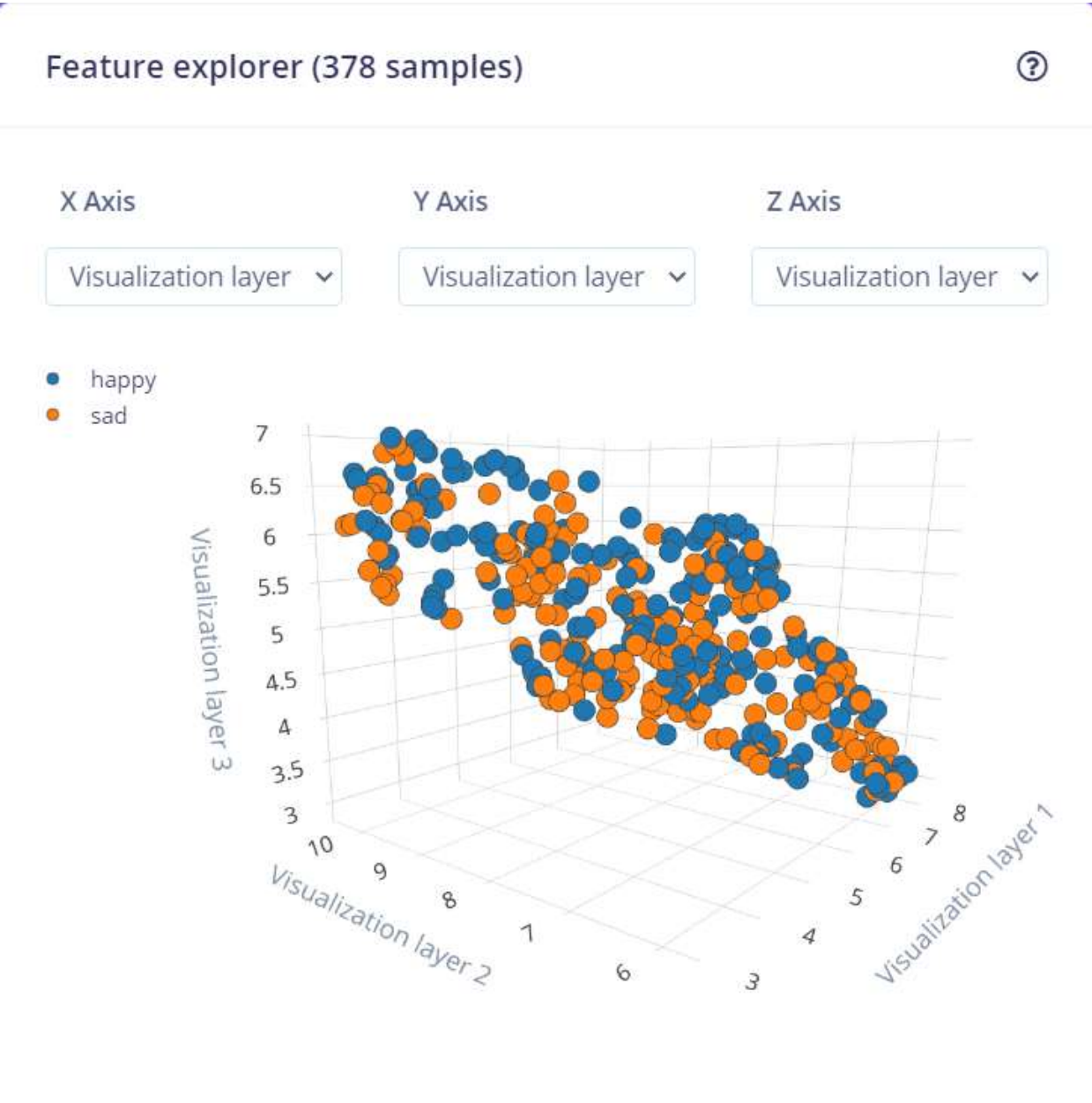
The data is derived from Kaggle Emotion Detection dataset. The dataset originally have images categorised into seven feelings group, for simplicity, only data showing happy and sad emotions are used. Moreover, 189 pictures containing people with different race, age and gender are carefully selected from thousands of happy images to use as training data for "happy" category and 24 pictures are used as testing data. The same happened for choosing training and test data for "sad". The dataset used is showing below:



These data was then imported to Edge-impulse with labelled "happy" or "sad". The image data was further squashed to a size of 96 * 96.

Model

The model is completely build using basic functions on edge impulse. The Transfer Learning block is selected at create impulse stage as it is recommended for image classification purpose, and the output features will be the two categories, "happy" and "sad". For the image processing, grayscale is selected to avoid the light issue, followed by the generation of features of 378 training data. As the figure shown below, the "happy" data and "sad" data are quite separated from each other.



When it comes to setup the neural networks, 50 number of training cycles with learning rate at 0.006 are set. The minimum confidence rating is set at 55%. As for the Neural network architecture, MobileNetV2 0.35 with 10 neurons as fianl layer and the dropout rate being set at 0.2 are chosen to ensure performance quality and prevent overfitting.

Neural Network settings



Training settings

Number of training cycles ?

50

Learning rate ?

0.006

Data augmentation ?



Minimum confidence rating ?

0.55

Neural network architecture

Input layer (9,216 features)



MobileNetV2 0.35 (final layer: 10 neurons, 0.2 dropout)

Choose a different model

Output layer (2 features)

Experiments

Three experiments has been taken out to find the relationship between accuracy, loss rate and parameters about neural networking.

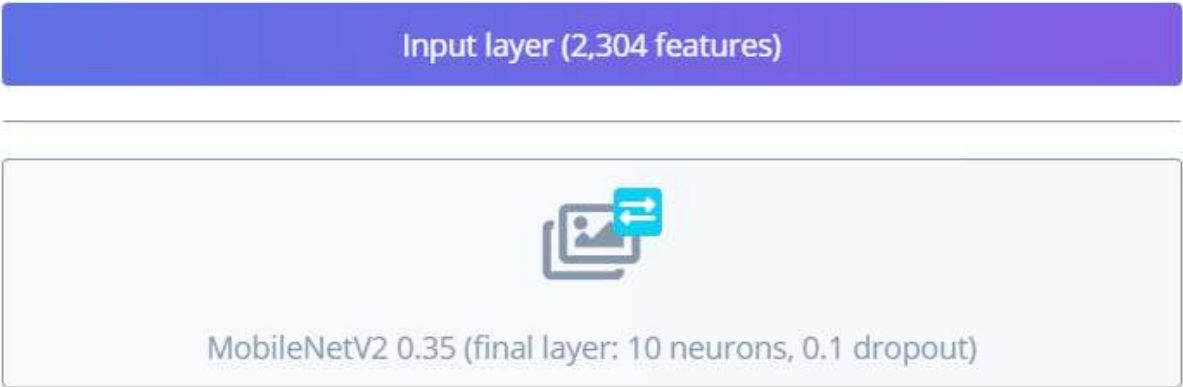
Neural Network settings



Training settings

| | |
|-----------------------------|------------------------------------|
| Number of training cycles ? | <input type="text" value="20"/> |
| Learning rate ? | <input type="text" value="0.001"/> |
| Data augmentation ? | <input type="checkbox"/> |
| Minimum confidence rating ? | <input type="text" value="0.60"/> |

Neural network architecture



For the first set, there are only 20 training cycles with learning rate at 0.001.The minimum confidence rating is kept at 60% as default. The result is not very satisfying since the model's accuracy rate is only 66.3%, with a loss of 0.69.

Model

Model version: ? Quantized (int8)

Last training performance (validation set)

%

ACCURACY

66.3%

LOSS

0.69

Confusion matrix (validation set)

| | HAPPY | SAD |
|----------|-------|-------|
| HAPPY | 71.1% | 28.9% |
| SAD | 40% | 60% |
| F1 SCORE | 0.70 | 0.61 |

Feature explorer (full training set) ?

happy - correct

sad - correct

happy - incorrect

sad - incorrect

Visualization layer 3

Visualization layer 2

Visualization layer 1

im151.png 249q1siv

For the second set, training cycles are increased to 50 and learning rate also rises to 0.006. The minimum confidence rating is decreased to 55% to reduce the "unknown" result.

https://github.com/JunzheGan/casa0018/blob/main/Assessment/Report/Final_report.md

8/14

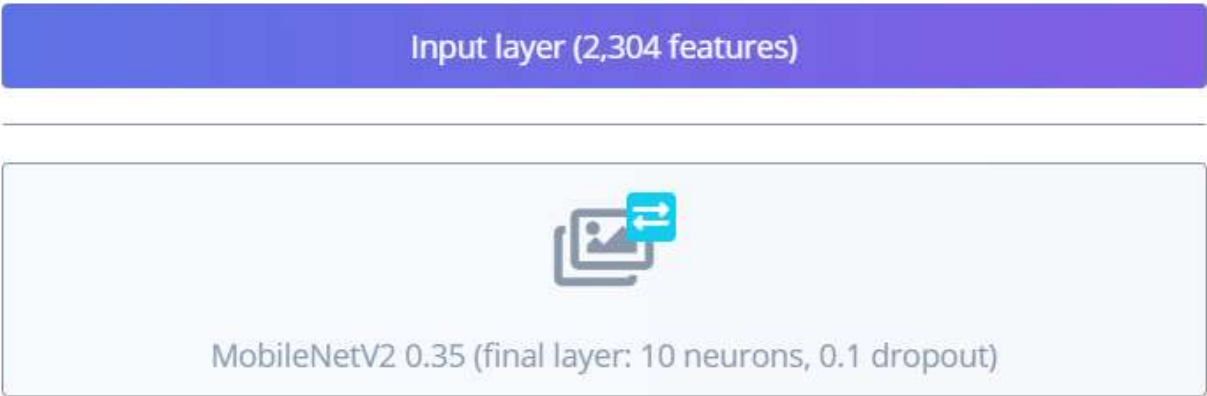
Neural Network settings



Training settings

| | |
|-----------------------------|-------------------------------------|
| Number of training cycles ? | <input type="text" value="50"/> |
| Learning rate ? | <input type="text" value="0.006"/> |
| Data augmentation ? | <input checked="" type="checkbox"/> |
| Minimum confidence rating ? | <input type="text" value="0.55"/> |

Neural network architecture



As expected, the accuracy rate improved significantly to 76.3%, with a loss of 0.98. This proves that increasing learning cycles and learning rate will improve the accuracy of the model. However, this can only be true under the assumption that the model is not overfitting.

Model

Model version: Quantized (int8)

Last training performance (validation set)

%

ACCURACY

76.3%

LOSS

0.98

Confusion matrix (validation set)

| | HAPPY | SAD |
|----------|-------|-------|
| HAPPY | 71.1% | 28.9% |
| SAD | 18.4% | 81.6% |
| F1 SCORE | 0.75 | 0.78 |

Feature explorer (full training set)

No samples were available to display in the feature explorer. Please contact support.

On-device performance

INFERENCING ...

251 ms.

PEAK RAM USA...

132.8K

ROM USAGE

577.4K

https://github.com/JunzheGan/casa0018/blob/main/Assessment/Report/Final_report.md

10/14

The third test takes exactly the same setting as the second one despite it is the fourth time retraining the model. This explains why there are 9216 features for input layer in test 3 but only 2304 features in test 2.

Neural Network settings



Training settings

Number of training cycles ?

Learning rate ?

Data augmentation ?



Minimum confidence rating ?

Neural network architecture

Input layer (9,216 features)



MobileNetV2 0.35 (final layer: 10 neurons, 0.2 dropout)

As the result below shows, the ultimate model has the highest accuracy rate (84.2%) and the lowest loss (0.41) among the three. This again proves that increasing learning cycles and rate will generally improve the model accuracy. However, we need to be very caution to avoid the overfitting problem when varying the setting for neural networks.

Last training performance (validation set)



ACCURACY

84.2%



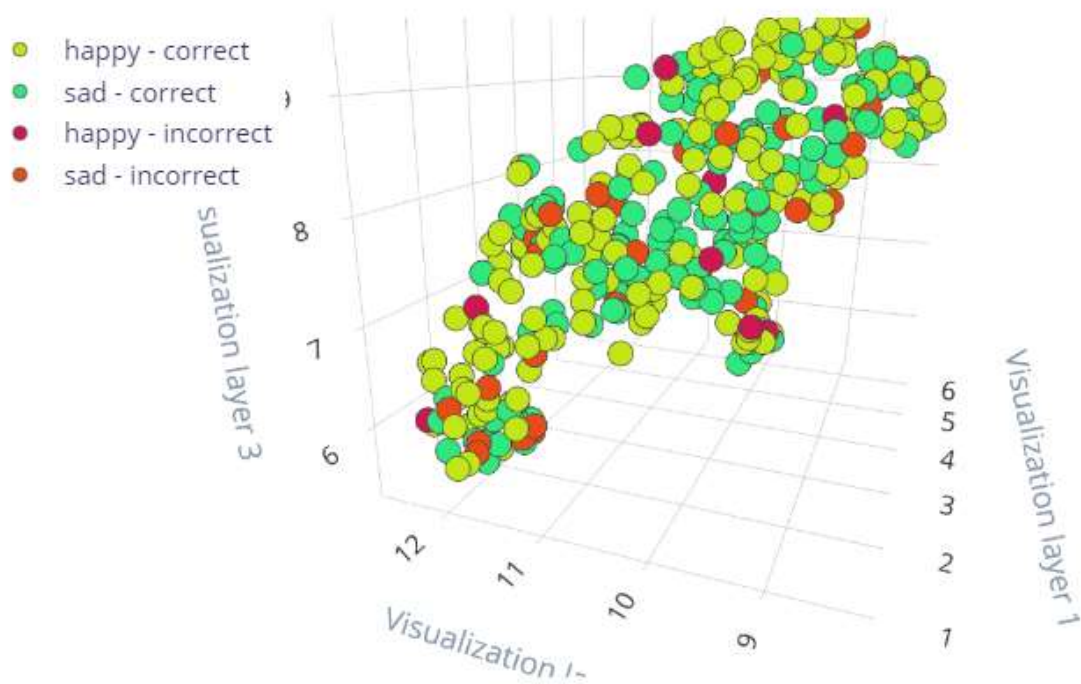
LOSS

0.41

Confusion matrix (validation set)

| | HAPPY | SAD |
|----------|-------|-------|
| HAPPY | 89.5% | 10.5% |
| SAD | 21.1% | 78.9% |
| F1 SCORE | 0.85 | 0.83 |

Feature explorer (full training set) ?



Results and Observations

The final validation result is shown below. The accuray rete for 48 test data is 83.33%, very close to 84.2% induced in the third experiment.

Validation results

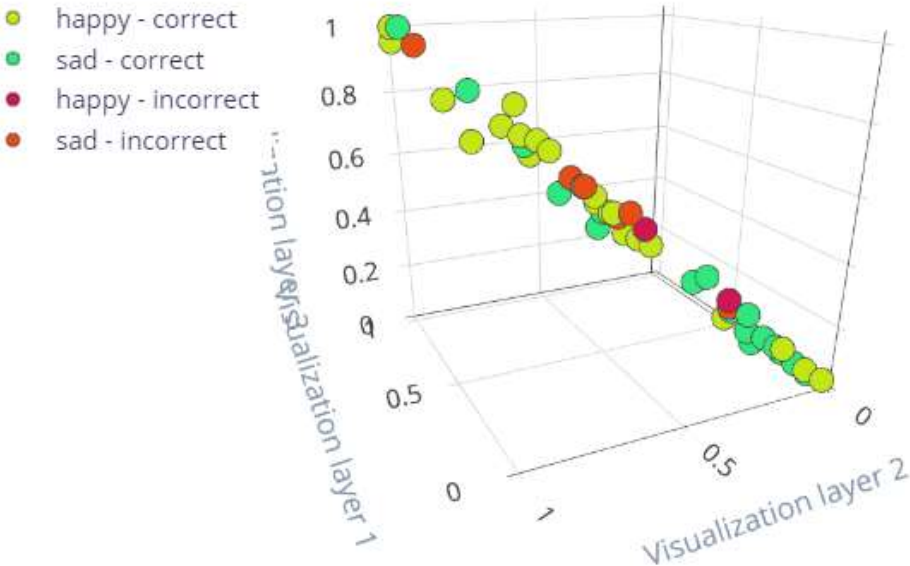
ACCURACY

83.33%

%

| | HAPPY | SAD | UNCERTAIN |
|-------|-------|------|-----------|
| HAPPY | 91.7% | 8.3% | 0% |
| SAD | 25% | 75% | 0% |

Feature explorer ?



By looking at the validation results table, one interesting thing is that accuracy rate for detecting "happy" face is higher than detecting "sad" face. The model is more likely to interpret sad emotion as happy as the result suggesting there are 25% chance mis-identify the sad emotion. This could be some sad pictures contain their hands covering the face. The model will interpret hands as abnormal pattern and regard that as happy. Alternatively, some people express their sad feeling by crying. The opened mouth was meant to be crying but the model may accidentally recongnise that as laughing. Sometimes, even the eyes will confuse the model. Generally, when people are happy or smiling, their eyes curved like a "n" shape, whereas when they are sad, eyes usually curved like an "u" shape. However, some people eyes are flattened even when they are sad, making the model very hard to decide.



Since the edge impulse currently does not support deployment to Arducam, the model is tested using webcam on smart phone and laptop. There are two videos, one is using my face as experimental data, the other one is using public photographs on unsplash. The model performs quite well in both scenarios. Hence, it is feasible to create a simple facial emotion detection model.

<https://github.com/JunzheGan/casa0018/blob/main/Assessment/Report/Videos/1.mp4>

<https://github.com/JunzheGan/casa0018/blob/main/Assessment/Report/Videos/2.mp4>

Bibliography

1. Iva, Djukić. (July 21, 2020). Using artificial intelligence and emotion recognition for PR City published. available at: <https://www.sotrender.com/blog/2020/07/ai-emotion-recognition/>
2. Vihar, Soni. (Aug 8, 2020). Facial Recognition in Retail - Enhance In-store Customer experience and Improve Retailer Operations. available at: <https://www.einfochips.com/blog/facial-recognition-in-retail-enhance-in-store-customer-experience-and-improve-retailer-operations/>
3. F. Agrafioti, D. Hatzinakos and A. K. Anderson, (March, 2012). "ECG Pattern Analysis for Emotion Detection," in IEEE Transactions on Affective Computing, vol. 3, no. 1, pp.102-115, doi: 10.1109/T-AFFC.2011.28.

Declaration of Authorship

I, Junzhe Gan, confirm that the work presented in this assessment is my own. Where information has been derived from other sources, I confirm that this has been indicated in the work.

Junzhe Gan

29-April-2021

Appendix

Edge Impulse Project-the latest public version is version 7, available here:

<https://studio.edgeimpulse.com/public/21886/latest>

The code and data used can all be found at github repository:

<https://github.com/JunzheGan/casa0018/tree/main/Assessment/Projects/Final%20Project>