CMP304 Artificial Intelligence

ML for Computer Vision Project Report

Joseph Roper - 1901881

Introduction

The aim of the project is to use computer vision to track a face and identify the age, gender and emotion of the person in the camera. This can be used at self-checkout terminals in stores to monitor customer satisfaction whilst using the machines. Age detection can be used with the challenge 25 program for underage purchasing, eliminating the need for an employee to stand by the self-checkouts within stores. The prediction can also be used to track purchases to groups for calculating product trends, which can be pushed by the store's marketing team for advertising.

The resources which will be used for the project are Python, Visual Studio Code, Open-CV, TensorFlow, NumPy, Pandas, Matplotlib and Keras.

Python is a programming language which is good for computer vision projects as most of the libraries are catered to it. Visual Studio Code is an interface which was used to program in Python. Open-CV is a library for Python which focuses on functions for computer vision. TensorFlow is a library focused on machine learning which is used for this project. NumPy, Pandas and Matplotlib are all subsidiary libraries which were used to help make the program.

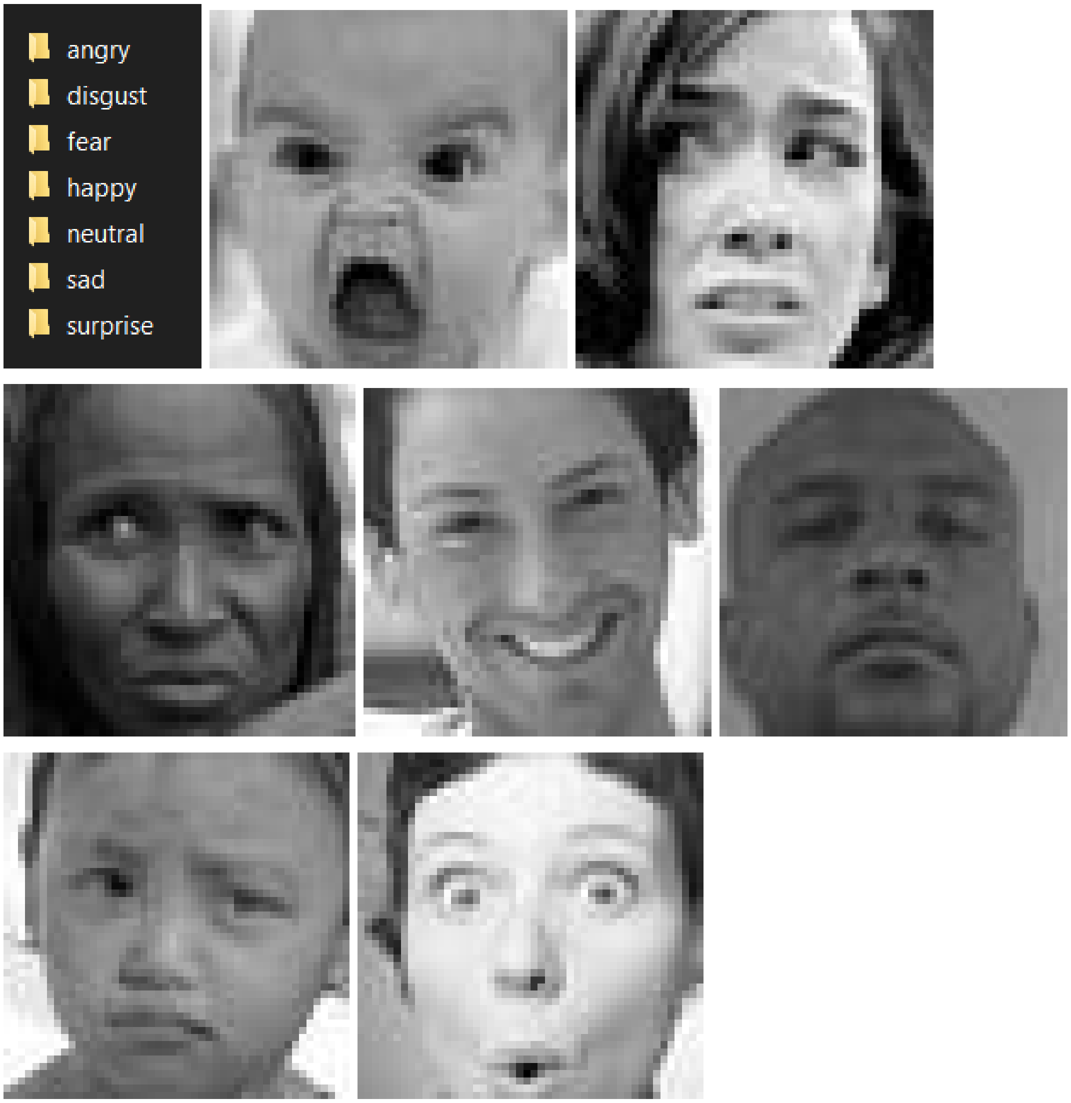
Data Specification

Two datasets were used for my application, one to train the emotion recognition model and the second to train the age and gender detection models.

Two datasets were used, as the UTK face dataset did not have images demonstrating the different emotions. Therefore, using this alone would not have allowed the emotion detection model to be trained properly. The FER-2013 dataset does not have a wide range of ages, making it unsuitable to train the age detection model. This means both datasets were necessary to train the different models.

FER-2013 Dataset

The dataset used to train the emotion recognition model was the FER-2013 dataset found on the website Kaggle. The dataset consists of roughly 35,000 images altogether. About 28,000 images for training and 7,000 images for testing. These images are 48 by 48 pixels in size, and greyscale and have been pre-labelled into 7 emotions: Angry, Disgust, Fear, Happy, Sad, Surprise and Neutral. The dataset is large, meaning the model will be more diverse and accurate when used in a program, as it will have a lot of data to justify its recognition.



(Images from Fer-2013 Dataset[1])

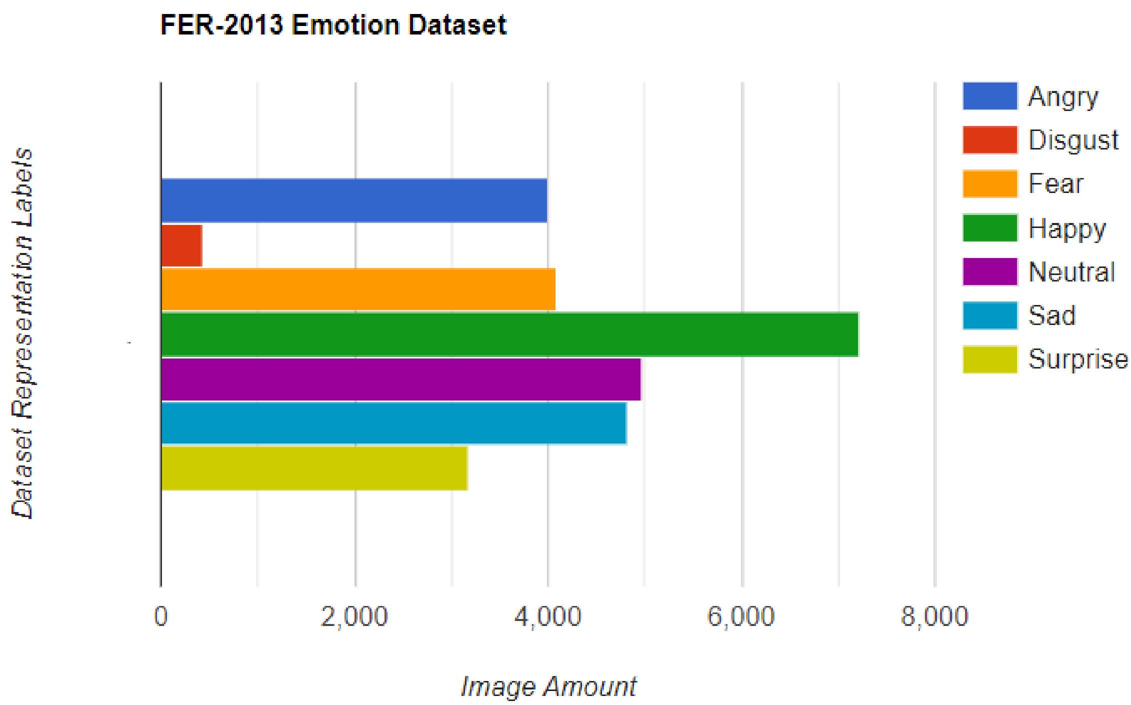
The dataset represents features of each human race, meaning it should be able to accurately judge the user's emotions regardless of their origin. However, there are substantially a lot more pictures of white people within the dataset, most likely resulting in the model's accuracy being higher with that race.

Emotion recognition is a classification problem, as the model will be viewing an image and sorting it into a category. Due to this, supervised training will be used for the

[1] (SAMBARE, 2013)

emotion detection model. The FER-2013 dataset has its features pre-labelled into the different emotions, which will supplement the project's workload, as the data will not need to be sorted by hand.

When analysing the graph below it is noticeable that the dataset is lacking in images under the disgust label, this will cause problems when training the model and may result in the program not being able to clearly detect the emotion.



Pre Processing

The image pre-processing which will be performed on this dataset is not extensive as there is already some applied, including greyscale. The images will be rescaled to be smaller to reduce training times for the model.

UTK Face Dataset

The UTK Face dataset will be used for training and testing the age and gender detection models. The dataset consists of 23,708 pictures of human faces which are coloured. The size of each image is 200 by 200 pixels.

The data is pre categorized into 4 labels which are represented as numbers on each image name. The first label is the age of the people within the dataset and ranges from age 1 to 116. The second label is the gender of the images, which is categorized into male and female, assigned 0 and 1 respectively. The third label is the race of the person in the image ranging from 0 to 4, the 5 categories are, White, Black, Asian, Indian and other such as Hispanic, Latino and Middle Eastern. The last label is the date/time of when the picture was collected for the dataset. The race and date & time labels will not be utilized within the program, as they are not necessary for training an age and gender recognition model. However, the dataset does have a good representation of different races, which will result in a more accurate program for people from all backgrounds.

```
[age]_[gender]_[race]_[date&time].jpg
```

(Images from UTKFace[2])

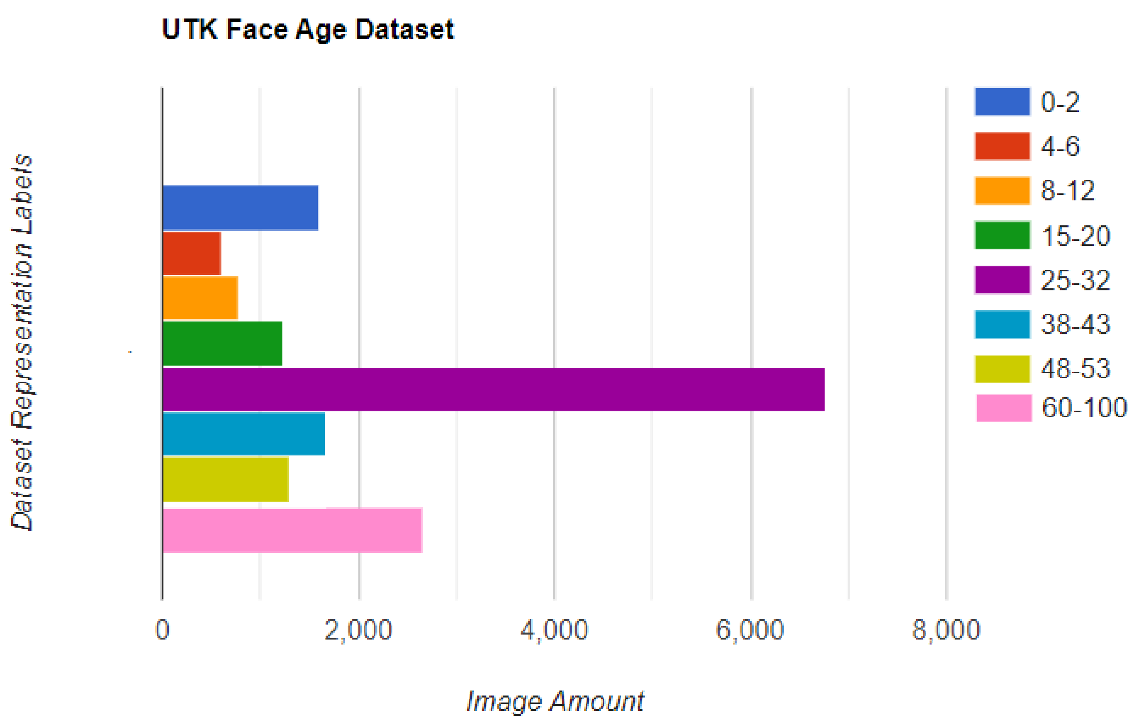
The images are JPEGS, making the file size a lot smaller, resulting in faster training times. However, the quality is also worse when compared to PNG, which could cause the model's accuracy to decrease.

As the program has to verify if the user is of legal age to purchase age-restricted goods, it would have been more useful to have the age label as categories such as (0,17) and

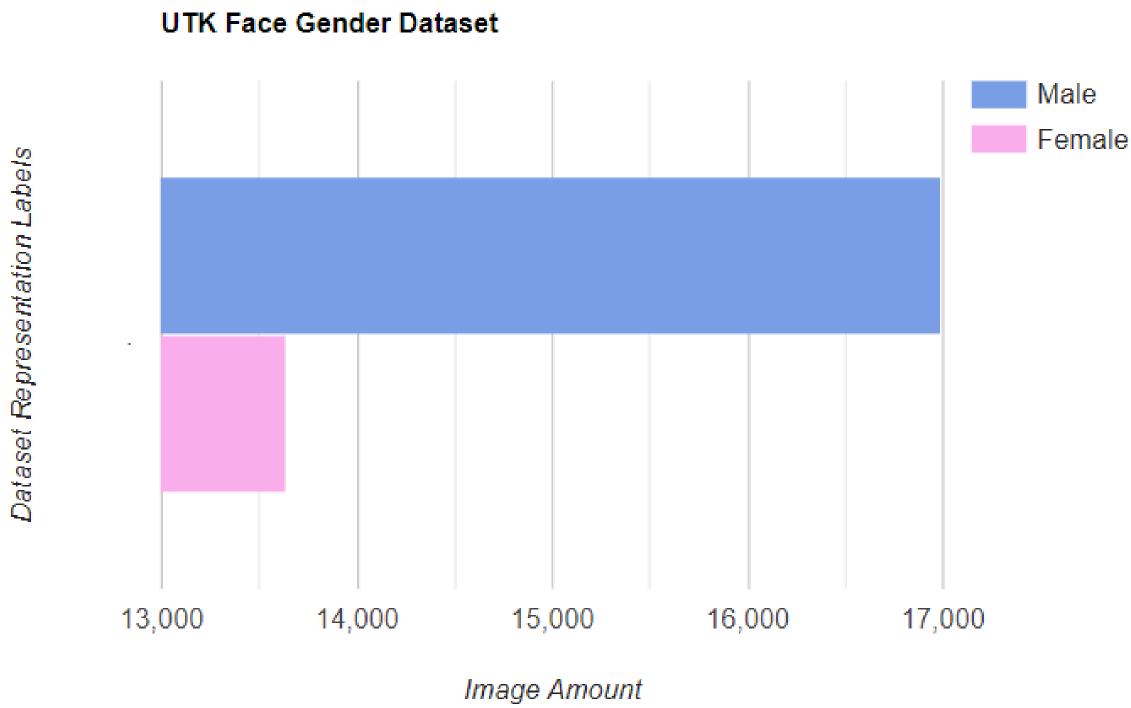
[2] (UTKFace, n.d.)

(18 & above), This would have resulted in more examples for each category. When reviewing the dataset, it was found that the age label is the individual's age, resulting in few examples for each age. This is very specific to the problem, as individual age is not needed. For the problem, it would have been more beneficial if the dataset was categorized into age ranges such as (0,17) and (18 & above), as the program only has to verify if the user is of legal age to purchase age-restricted goods.

If a new dataset was to be created for the age detection model, the data would be labeled this way. However, using a pre-made dataset would be more beneficial to the outcome of the project, as it would be difficult to get the same amount of reliable data independently using websites such as Google to collect images.



This graph shows that the dataset has a lot more images for people aged 25 to 32. Due to the problem relating to the challenge 25 program to improve the data set in the future, more images could be added. These would be especially for the age range of 15 to 20, as 15 to 17-year-olds are most likely to want to buy alcoholic drinks.



From this bar chart of the dataset, it is clear that there are more images containing the male gender. The bar chart above shows clearly that there are more images containing the male gender; it can be inferred therefore that the age detection model will have discrepancies when detecting the female gender.

As the UTK Face dataset did not categorise the images by folder and instead labelled the file names of the images, Regex was used to sort the images into categories , so the graphs could be made.

Pre Processing

Extensive preprocessing on this dataset is needed, as the file sizes are large, which would result in the model's training times being far too great. First, the images will be converted to greyscale, as colour is not needed for age and gender recognition.

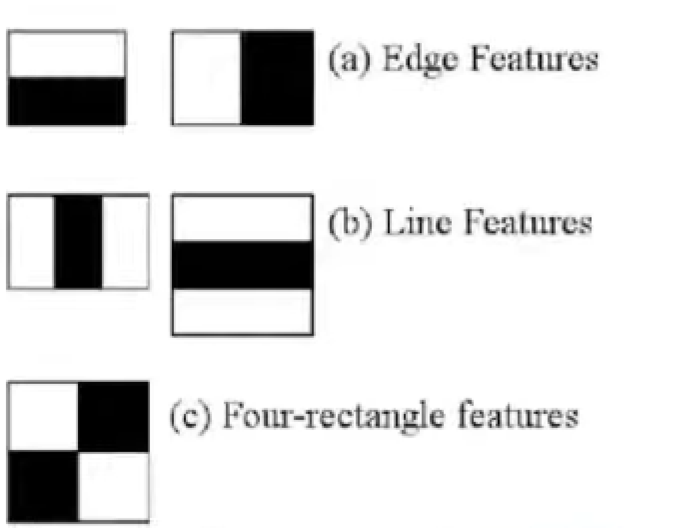
Then the image size will be decreased by half, which results in the training time being decreased by half as well. The image size could be reduced more, this would be at the detriment of the model's accuracy, as there would be less pixel data to train the model, making it more difficult for the convolution layer to locate patterns for the recognition.

Datasets Analysis

Using a pre-made dataset provides more trustworthy data as compared to Google search images which would provide a lot of invalid data and if missed would hurt the performance of the models. Two datasets were chosen, as neither one covered enough features with their data to be used for all 3 models' training datasets.

Methodology

The program uses OpenCV's pre-made haar cascades model to detect faces within pictures for the models to run prediction. This model detects features on the face similar to the convolution layer. Each feature is a single value obtained from subtracting the white pixel data prom the black pixel data to create an edge. These edges are then tested against a pre-existing edge detection applied onto a face to see if they match.



(Image showing haar cascade detection[3])

Emotion Model

Training

Because the emotion recognition problem requires one feature, the images to be imputed the sequential model was chosen as it is effective with linear topology. The model is created incrementally by adding multiple layers such as convolutional, pooling, dropout, flatten and dense layers.

The number of times data is given to the neural network for training are called Epochs. Epoch stands for the number of times data is given to the neural network for training. Multiple epochs are needed when training for a good model. The standard for model training is 40 epochs or more; the 3 models trained for this program use 50 epochs.

[3] (OpenCV: Cascade Classifier, n.d.)

Layers

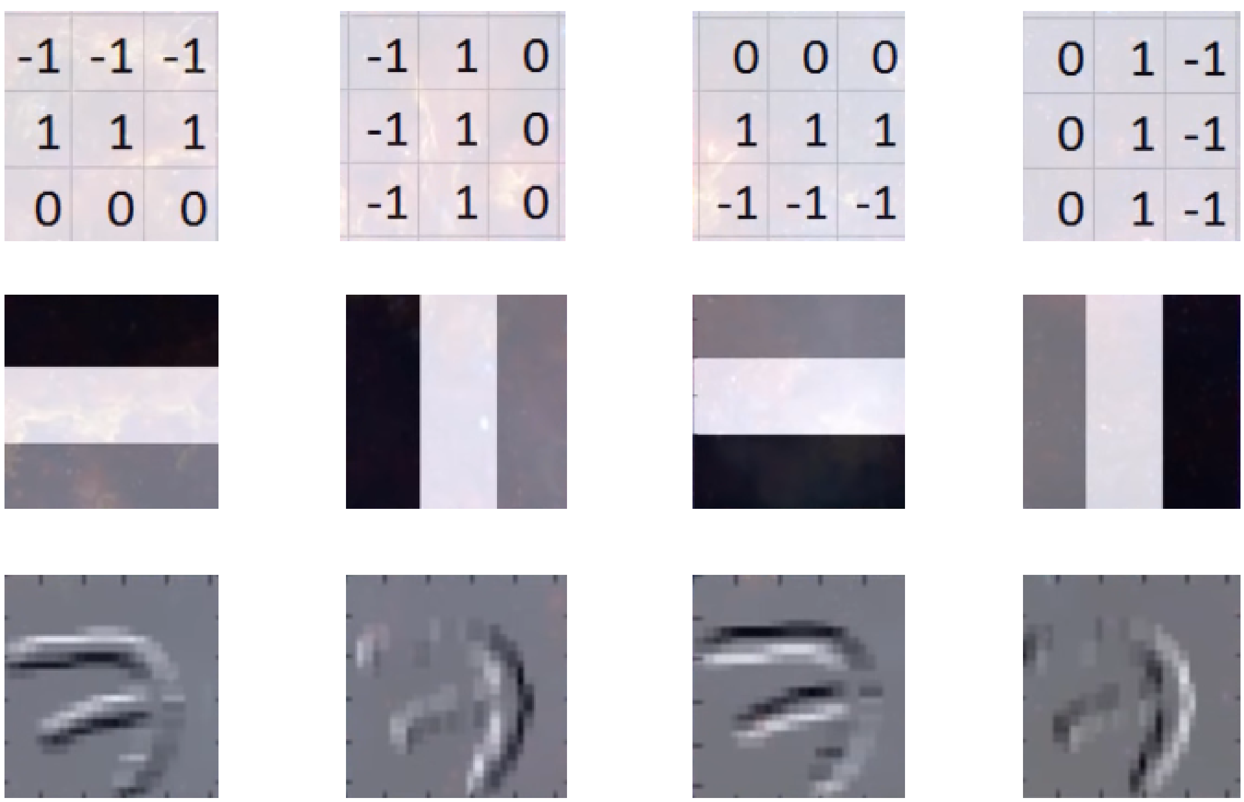
Convolutional

The convolutional layer detects patterns in an image's pixel data, the kernel size is the size of a filter made up of a small matrix. This filter passes across each 3 by 3 block of pixel data within the image and computes the dot product of the pixels within the cell.

Using multiple convolution layers increases the training time for the model, however it also increases the detail of image characteristics found in the training data per image which should increase the accuracy of the model.

The emotion detection model uses four, two-dimensional convolution layers, each with a kernel size of 3 by 3. The program starts with 32 filters in the first layer and multiplies this value by two for each convolutional layer in the sequence. This is done as convolutional layers capture patterns like edges and with each layer, these patterns combine creating bigger patterns which require more filters to catch these new larger combinations of patterns.

RELU is the activation hyperparameter chosen for this program. RELU ensures that all pixel data have positive values after the convolution layer calculation, which is then used for the calculation of the next layer.

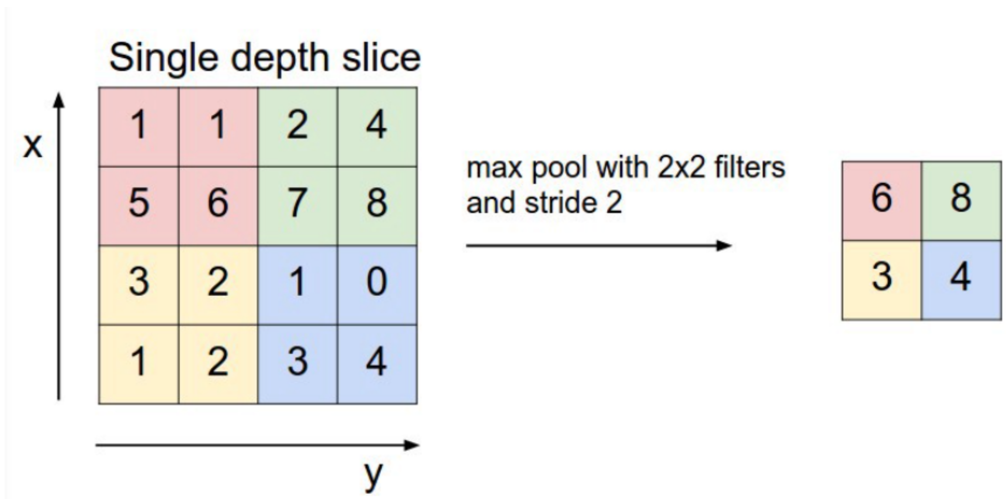


(Diagram showing how a convolution layer works[4])

Pooling

The pooling layer prepared the dataset for the next layer or process. The emotion model uses the pooling layer to reduce the size of the data set images by half in the x and y-direction. This is used to lower the processing time for the model's training.

The program uses the pooling layer three times while training the model, each time reducing the image size by half.



(Pooling layer diagram[5])

Dropout

To prevent overfitting while training the program, the dropout layer was introduced. This will reset a random detected edge or facet of the image found in the convolutional layers at a probability set in the program.

The program uses the drop out layer 4 times after each convolution layer at a probability of 10% for the dropout rate.

---

[4] (Convolutional Neural Networks (CNNs) explained, 2022)
[5] (Stanford University, n.d.)

Flatten

The flatten layer takes the 2D data representation set at the convolution layer and performs dimensional reduction to make the data representation 1D for the dense layer to read.

The program uses the flatten layer once before the last dense layers computations.

Dense

The program uses the dense layer when training the model to define how many outputs the model will have based on the unit hyperparameters value. For example, the model will recognize 7 emotions, so the model must have 7 outputs, this is achieved using the dense layer. The dense layer in this model has its activation set to softmax which allows for the model to have multiple outputs.

Age Model

Training

The age detection model is trained similarly to the emotion detection model ass the only feature imputed into the model are the images. Because of this, the sequential training model was used.

Layers

Convolutional

The age detection model uses four convolutional layers, each with a kernel size of 3 by 3. The amount of filters per layer goes in the following order: 128, 128, 256 and 512. As age is harder to detect when compared to emotion, more filters were needed per convolution layer, so more patterns could be recognized, for example edges found because of wrinkles on the face. Less convolution layers were used compared to the emotion model, as to not cause overfitting for the model.

Pooling

The program uses four pooling layers for the age detection model. The pooling layers are used to reduce the size of the training images. For the age detection model, the pooling layers are set to 3.

Dropout

The program uses one dropout layer when training the model before the last two dense layers. The probability of this is set to 20%. Only one dropout layer is used as it is hard to detect patterns between the images, so any that are found do not necessarily want to be discarded.

Flatten

One flatten layer is used when training the age detection model to prepare the data for the last two dense layers.

Dense

The final dense layer for the age detection model has one output layer with the activation type set to linear, as the model detects one node, age which is just one number, this is not a classification problem.

Gender Model

The gender detection model is trained similarly to the emotion detection model. As the only feature imputed into the model are the images, because of this the sequential training model was used.

Layers

Convolutional

The gender detection model uses five convolutional layers, each with a kernel size of 3 by 3. The program starts with 32 filters in the first layer and multiplies this value by two for each convolutional layer in the sequence. RELU is the activation hyperparameter chosen for the convolution layers in this model.

Pooling

The program uses five pooling layers for the age detection model. The pooling layers are used to reduce the size of the training images. For the age detection model, the pooling layers are set to 3. This was needed as the image sizes within the UTK Face dataset are bigger compared to the FER-2013 dataset used for the emotion model. This lowers the training time for the model.

Dropout

The program uses one dropout layer when training the model before the last two dense layers. The probability of this is set to 20%. Only one dropout layer is used, as any more than that started to lower the model's accuracy.

Flatten

One flatten layer is used when training the age detection model to prepare the data for the last two dense layers.

Dense

The final dense layer for the age detection model has one output layer with the activation type set to binary cross-entropy, as the output will only have two states: Male or Female.

Results

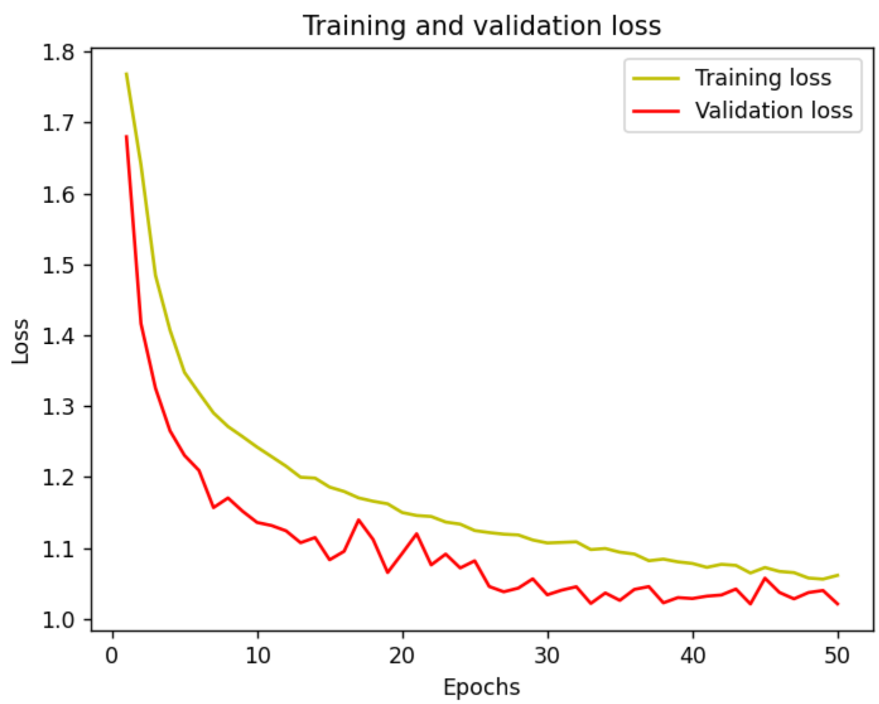
Emotion Model

Training Charts

These charts were made after the model was trained.

Training & Validation Loss

The training and validation loss curves both go down after each epoch which is desirable, however, the validation loss is lower than the training loss. This means that the training data is harder to model than the evaluation.



Training & Validation Accuracy

The training and validation accuracy graph shows that the model recognition ability is quite good, as the accuracy sits at around 60% on the graph. In testing, In testing, this has gone up about 70%. There is a slight case of under fitting as shown in the graph below as the validation accuracy is higher than the training data accuracy. To resolve this, the model could have a higher probability of dropout while training, resulting in a decrease in regularization in the model. The model could also be trained for longer by introducing more epochs.

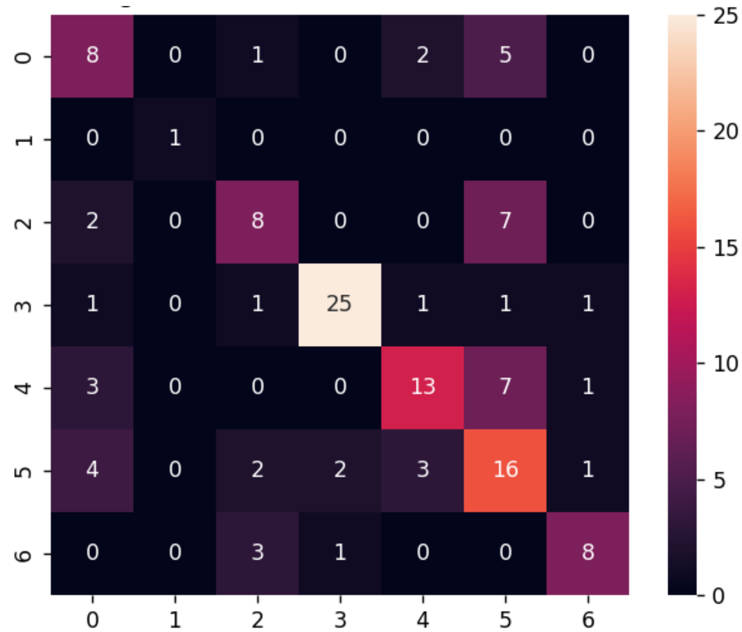
Heatmaps

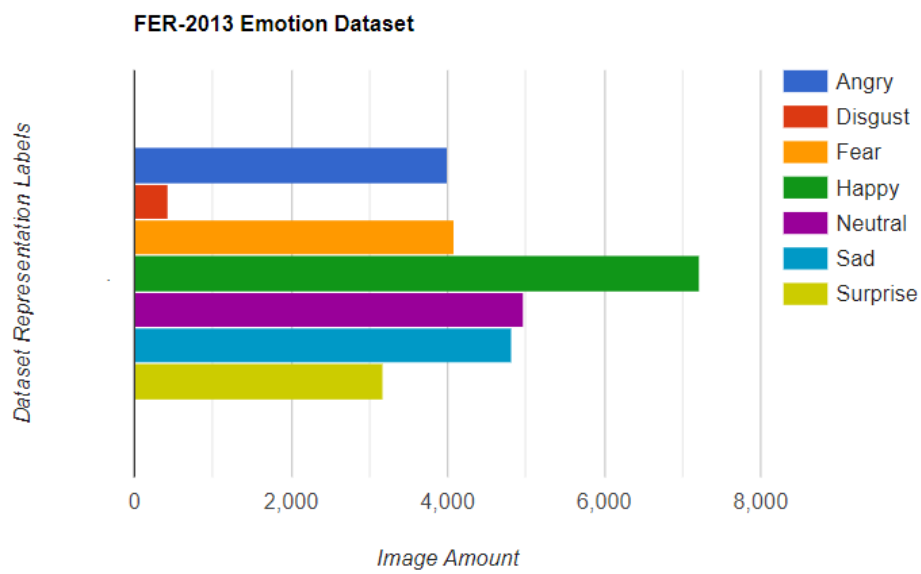
The confusion matrix was tested, the confusion matrix mapped onto a heat map in batches of images. The images below are tested with a batch size of 128 Images. The axis values represent the emotion classes.

Emotion:	Angry	Disgust	Fear	Happy	Neutral	Sad	Surprise
Value:	0	1	2	3	4	5	6

The heatmap below shows that the labels are getting accurately predicted, besides the disgust emotion, however, this directly corresponds with how much data is represented within each emotion class in the FER-2013 dataset. The lowest accuracy is the disgust label which has the least amount of images provided, and the highest accuracy is with the Happy label which has the most data.

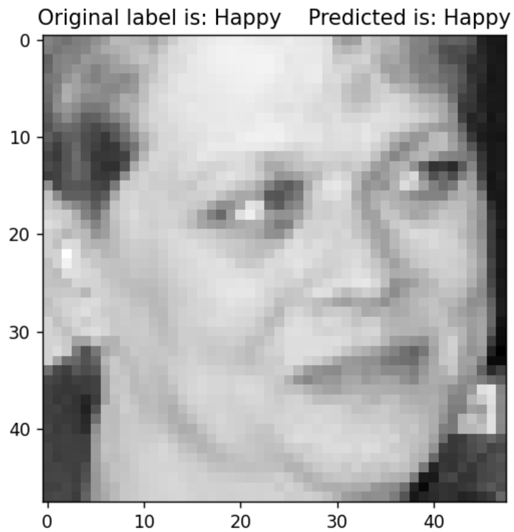
There are some discrepancies shown within the heatmap such as the model continuously predicting sad as other labels mainly; angry, fear and neutral. To fix this, more images will have to be added to all of these labels. However, it is understandable why the model does this as all of those emotions can involve a straight or frowned mouth, making them hard to distinguish.

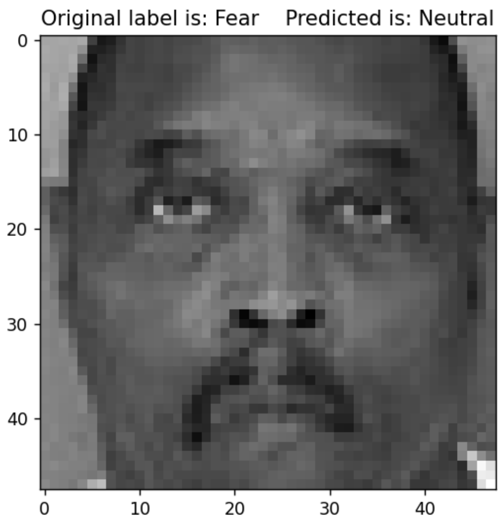


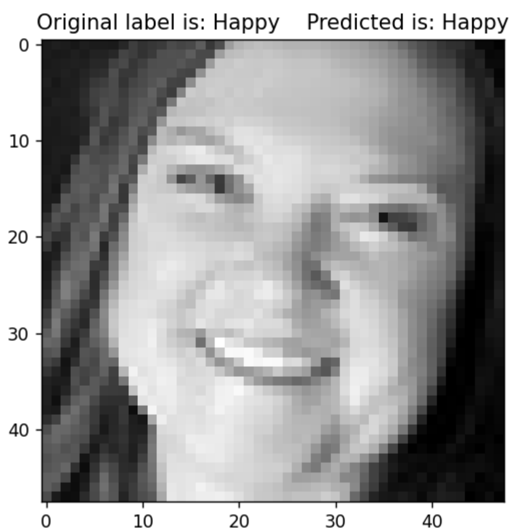


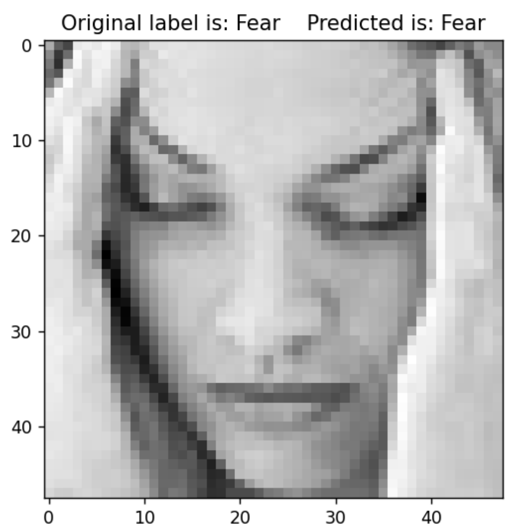
Validation

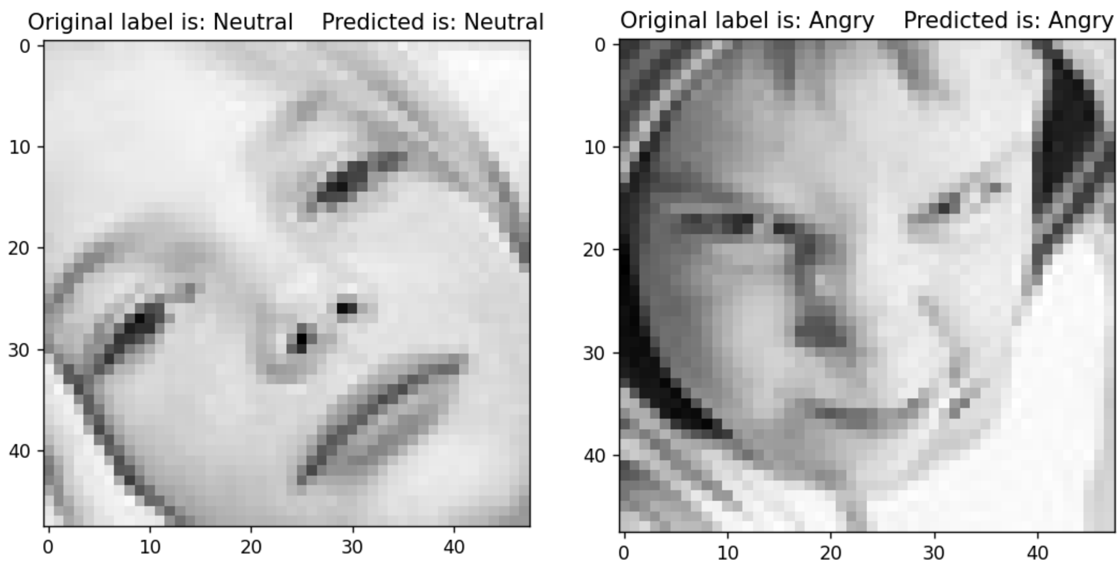
As you can see from the images below, the model accurately predicts emotions when they are clearly shown on the face. However, it struggles with abstract emotions like fear and disgust, which are even hard to tell in real life as they are not commonly shown. The second image does not accurately represent the emotion of fear, it is ambiguous. Therefore, the image should be removed from the dataset as it is an anomaly.











(Images from FER-2013 Dataset[6])

Discussion

The model is tested in batches of 128 images which lowers the accuracy of the testing, however, due to the lack of testing data, this will have to suffice. The accuracy ranges from 65% to 70%, and with more testing data this could be increased.

[6] (SAMBARE, 2013)

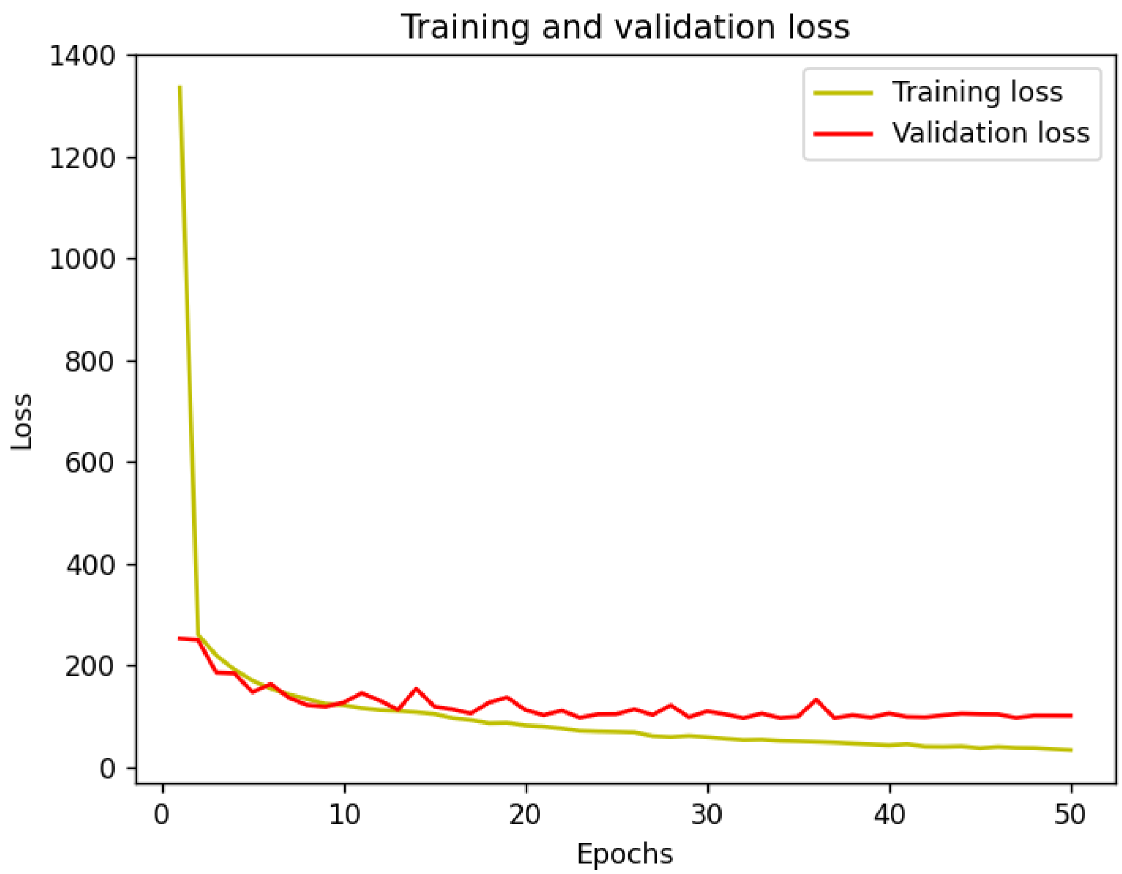
Age Model

Training Charts

These charts were made after the model was trained.

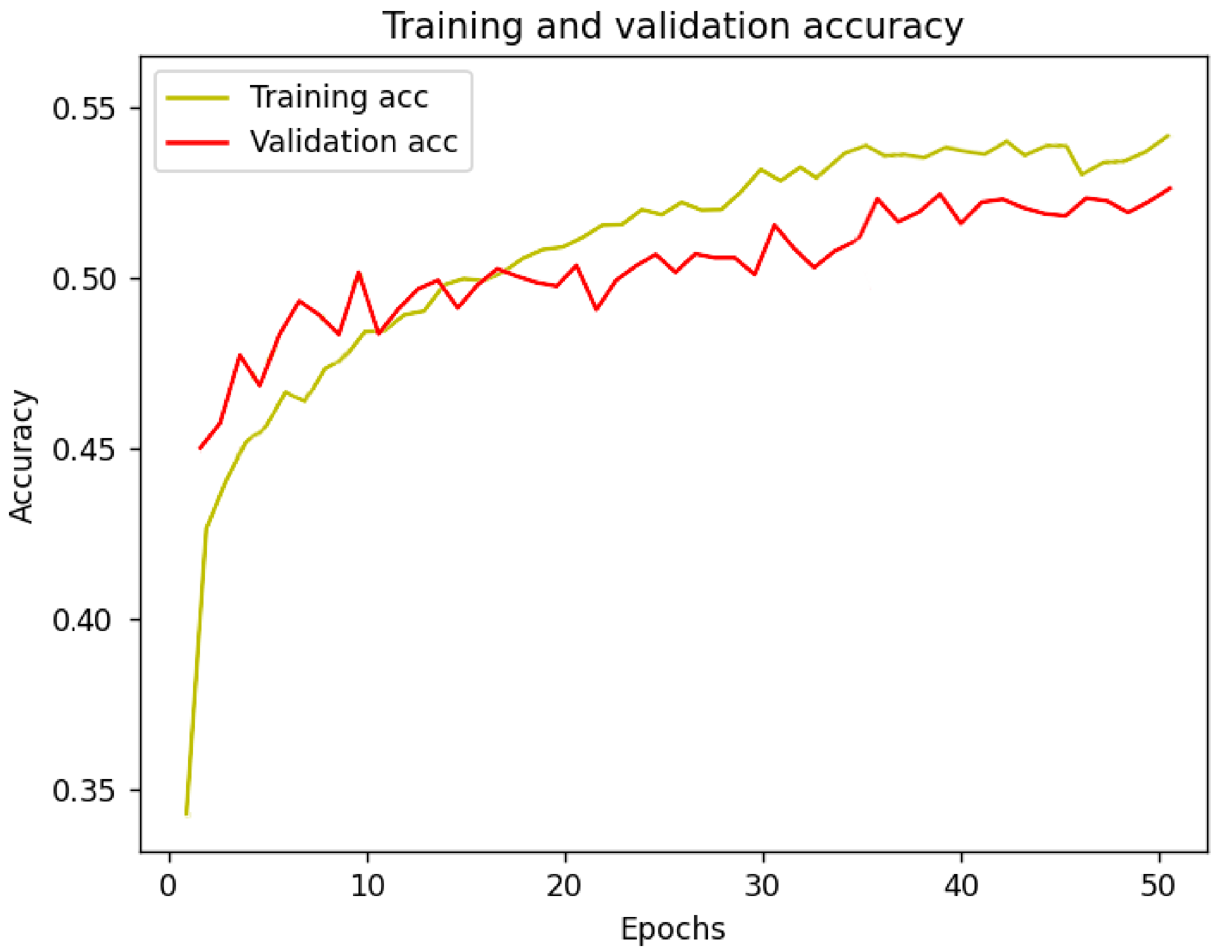
Training & Validation Loss

This graph shows a slight case of overfitting, as the training line falls a bit too far below the validation loss.



Training & Validation Accuracy

The accuracy of this program is fairly low compared to the other models although it still does perform well. The graph at 50 epochs shows that the accuracies are still going up and have not dropped off yet. The accuracy of the model could be improved by increasing the amount of epochs to 100, this would increase training time even more but the results would be improved.



A confusion matrix/heatmap could not be provided for this model, as the program's age detection is not classified into select categories. If a confusion matrix was made, there would be a value for every age on the axis, making the matrix too big and hard to read.

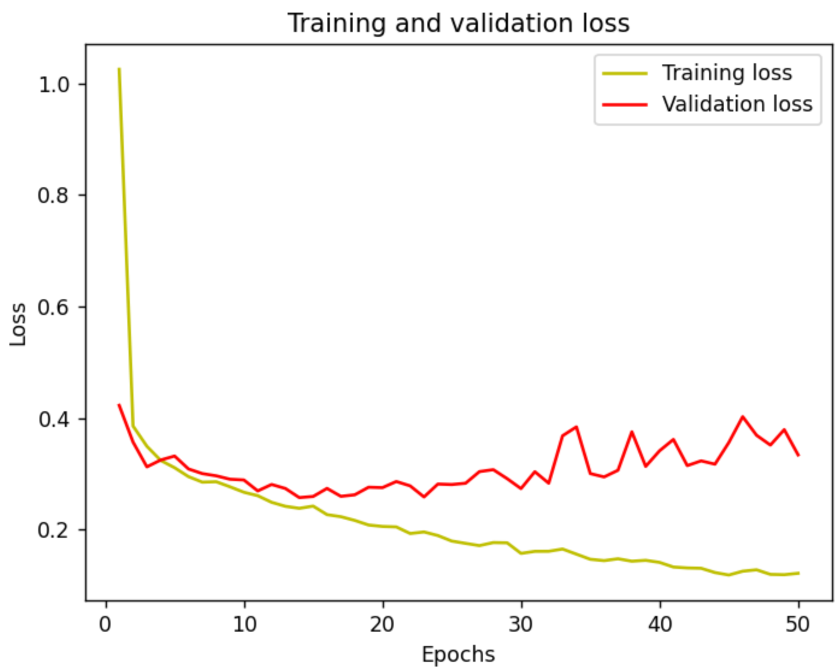
Gender Model

Training Charts

These charts were made after the model was trained.

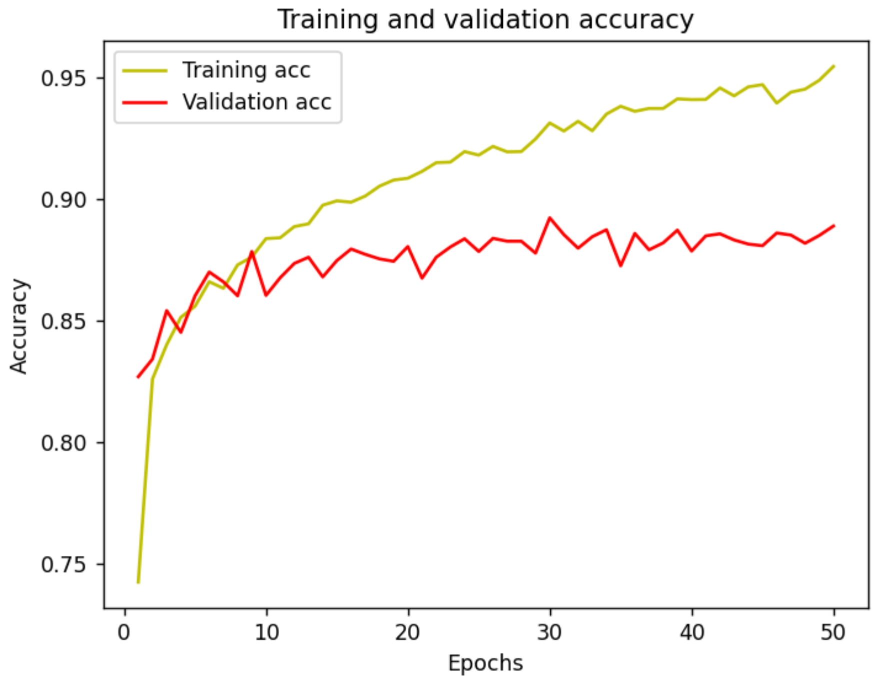
Training & Validation Loss

The validation loss increases too much compared to the training loss. This means the model is overfitted to the training data. To fix this regularization could be performed by adding dropout, to match the training loss closer to the validation, making the model more generalized.



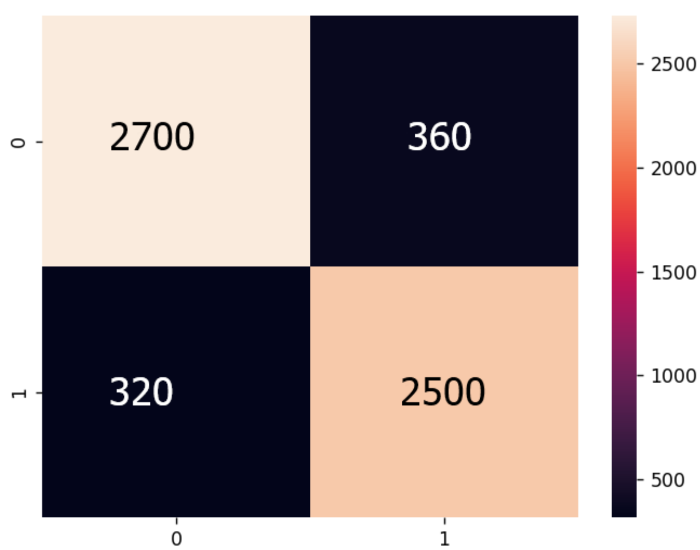
Training & Validation Accuracy

Again this graph shows a case of overfitting within the model, however other than that the accuracy is fairly high at the validation. However, the training accuracy needs to be brought down as currently, it is unrealistic.



Heatmap

The heat map for the gender detection model is very accurate, as the number of false positives and negatives are low, especially when compared to the true positives and negatives. Both classifications are similar in accuracy, although to make them match, more images of the female gender could be added to the dataset.



Discussion

Emotion Model

I found when testing the model it only detects emotions when they are exaggerated on the face, this is due to the FER-2013 dataset having exaggerated pictures for the training data. To resolve this, I would have to get images which show emotion without overemphasized expressions. However, I believe this would be difficult to find on the internet as if you search up sad faces on Google the majority of pictures would have exaggerated expressions. To get an accurate dataset to solve this problem, a new dataset would have to be created by hand taking new pictures to reflect the subtle expressions that someone would make when at a self-checkout.

I am happy with the emotion model detection and believe it is fairly accurate, when testing it is shown that some emotions aren't detected as well as others for this I would add more image data into those categories. The model seems to slightly overfit with the training data, to resolve this I would add more convolution layers or drop layers. To improve the model further, more emotion classifiers could be added, such as annoyed, as I believe that is a common feeling shared at self-checkout terminals.

I believe emotion detection can be useful at store self-checkouts to improve the customer experience while shopping. For example, the model could detect that the user gets angry when the tills voice assistant continuously speaks. If this is recognized as being a frequent experience with customers, the stores' company could come up with a solution to resolve this issue, such as by delaying the time between the till's voice assistant operations.

Whilst testing the model, I found disgust to be a hard emotion to portray, as each person usually has a different face to demonstrate this emotion. Therefore, I believe this emotion is redundant to try and detect within the application.

Gender Model

The gender detection is only completely accurate when the person it is detecting matches the characteristics of the traditional male or female gender. If the program is tested with a man having long hair, the detection will say they are female. In the future, to resolve this, more images of people who don't fit the classic gender stereotype will need to be added to the dataset.

I believe gender detection can be useful at store self-checkouts to see if specific genders are buying items which the other gender doesn't, other than the obvious purchases such as female sanitary products. These items can then be pushed with advertising on the store's app by giving coupons.

Age Model

The age detection model is the least accurate when performing out of the three models. However, it's also the hardest to detect and confirm in real life with people as depending on how a person has lived they can look older such as if they smoke or drink alcohol. Makeup can also affect how old someone looks. Therefore, I do not believe the model can be trusted with age verification at stores for the purchasing of age-restricted products such as; energy drinks, alcohol and painkillers, just yet.

I believe with more data and longer training times for the model, the accuracy can go up and be used for accurate customer age tracking. However, I believe the regulations to adopt this with the 'Challenge 25' program for the purchasing of age-restricted products will not pass.

The brightness of the input image greatly affects the age model's prediction, as within the dataset, the younger images are brighter due to baby photos usually being taken in photo studios and because children usually have paler skin. Whereas, the adult images are darker as the pictures were taken from less lit areas. This results in brightness heavily affecting the age detection, as the darker the image the older it will think you are and the brighter the image the younger it will think you are. To resolve this for future use, images for the dataset will have to be taken at all levels of brightness or just one set brightness level and make the store it is used in match that brightness.

Whilst testing the model I realised that because the dataset has more old people wearing glasses, if you wear glasses the model will predict that you are older. To resolve this, more images of young people with glasses would have to be added to the dataset to balance it out.

Ethics

The morals of my program would be put into question if it was to be applied at self-checkout terminals, as data collection should be monitored and the customers would have to agree to the use of their data for store and advertising improvement.

As the gender model makes predictions based on traditional gender stereotypes if the program was used to promote products as previously discussed. It would result in some men getting recommended female products and vice versa, this could insult some customers causing the shop's reputation to be scorned.

Sources of Error

During the creation of this project, a lot of time was spent on resolving installation errors, to get the libraries and software used working with the project. All the sources used to help fix these errors are referenced in the bibliography section of the report.

If I was to do this project again I would use an online program such as Google collab or Kaggle to program as this would have saved a lot of time on bug fixing which could have been spent on tweaking the models to make them more accurate.

The age detection for my program is not very accurate, I believe this is due to the complex nature of the problem.

Future improvements for this model would be to turn it into a classification problem with age groups so that the result can be more lenient, this would also fit the nature of implementation with self-checkout terminals more.

Final Conclusion

Overall, I do believe the application to be useful and would provide good data for stores if implemented at a self-checkout terminal. The models can be further improved, mainly more data can be added to the datasets. The other methods on improving the models were discussed previously in the report.

A lot was learned in this module with how machine learning works and applying it to computer vision. Computer vision is very interesting, especially the applications of it which can be used to improve people's way of life.

Learning about how images are analysed within the model layers provided a great understanding into the image processing pipeline.

Bibliography and References

Datasets

UTKFace. n.d. UTKFace. [online] Available at: <https://susanqq.github.io/UTKFace/> [Accessed 15 May 2022].

SAMBARE, M., 2013. FER-2013. [online] Kaggle.com. Available at: <https://www.kaggle.com/datasets/msambare/fer2013> [Accessed 15 May 2022]

Pictures Used

Stanford University, n.d. *Pooling Layer Diagram*. [image] Available at: <https://cs231n.github.io/assets/cnn/maxpool.jpeg> [Accessed 15 May 2022].

Convolutional Neural Networks (CNNs) explained. 2022. [DVD] https://www.youtube.com/watch?v=YRhxdVk_sIs: DeepLizard.

Docs.opencv.org. n.d. *OpenCV: Cascade Classifier*. [online] Available at: <https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html> [Accessed 17 May 2022].

Resources/Programs Used

Rapidtables.com. (2019). Bar graph maker | Create a bar chart online. [online] Available at: https://www.rapidtables.com/tools/bar-graph.html.

Firas Dib (2019). Regex101 - online regex editor and debugger. [online] @regex101. Available at: https://regex101.com/.

Python (2019). Welcome to Python.org. [online] Python.org. Available at: https://www.python.org/.

Keras (2019). Home - Keras Documentation. [online] Keras.io. Available at: https://keras.io/.

Numpy (2009). NumPy — NumPy. [online] Numpy.org. Available at: https://numpy.org/.

Matplotlib (2012). Matplotlib: Python plotting — Matplotlib 3.1.1 documentation. [online] Matplotlib.org. Available at: https://matplotlib.org/.

Pandas (2018). Python Data Analysis Library — pandas: Python Data Analysis Library. [online] Pydata.org. Available at: https://pandas.pydata.org/.

PyPI. (2019). opencv-python. [online] Available at: https://pypi.org/project/opencv-python/.

TensorFlow (2019). TensorFlow. [online] TensorFlow. Available at: https://www.tensorflow.org/.
GitHub. (n.d.). opencv/opencv. [online] Available at: https://github.com/opencv/opencv/tree/master/data/haarcascades.

Research

Delua, J. (2021). Supervised vs. Unsupervised Learning: What's the Difference? [online] www.ibm.com. Available at: https://www.ibm.com/cloud/blog/supervised-vs-unsupervised-learning.

Super User. (n.d.). windows - How do I copy all file names in a folder to notepad? [online] Available at: https://superuser.com/questions/490306/how-do-i-copy-all-file-names-in-a-folder-to-notepad [Accessed 15 May 2022].

TensorFlow. (n.d.). The Sequential model | TensorFlow Core. [online] Available at: https://www.tensorflow.org/guide/keras/sequential_model#:~:text=A%20Sequential%20model%20is%20appropriate [Accessed 15 May 2022].

www.youtube.com. (n.d.). Sequential Model - TensorFlow Essentials #1 - YouTube. [online] Available at: https://www.youtube.com/watch?v=MQD1yMnZ_jk [Accessed 15 May 2022].

Github.io. (2012). CS231n Convolutional Neural Networks for Visual Recognition. [online] Available at: http://cs231n.github.io/convolutional-networks/#pool.

www.youtube.com. (n.d.). [TensorFlow 2 Deep Learning] Dense Layer. [online] Available at: https://www.youtube.com/watch?v=lor2LnEVn8M.

Verma, Y. (2021). A Complete Understanding of Dense Layers in Neural Networks. [online] Analytics India Magazine. Available at: https://analyticsindiamag.com/a-complete-understanding-of-dense-layers-in-neural-networks/.

www.ibm.com. (n.d.). What is Underfitting? [online] Available at: https://www.ibm.com/cloud/learn/underfitting.

Allport, T. (2019). Why You Should Use Convolutions in Your Next Neural Net — Using TensorFlow. [online] Medium. Available at: https://betterprogramming.pub/why-you-should-use-convolutions-in-your-next-neural-net-using-tensorflow-37d347544454 [Accessed 15 May 2022].

www.youtube.com. (n.d.). Object Detection OpenCV Python | Easy and Fast (2020) - YouTube. [online] Available at: https://youtu.be/HXDD7-EnGBY [Accessed 15 May 2022].

Zaremba, M. (2020). matzar/face_emotion_recognition. [online] GitHub. Available at: https://github.com/matzar/face_emotion_recognition [Accessed 15 May 2022].

valohai.com. (n.d.). What is a Machine Learning Pipeline? [online] Available at: https://valohai.com/machine-learning-pipeline/.

www.youtube.com. (n.d.). Real Time Age And Gender Recognition Using Pre-Trained Caffe models lPython Opencv|KNOWLEDGE DOCTOR| - YouTube. [online] Available at: https://www.youtube.com/watch?v=V0qeBb8F8XY&list=PLdkgl0WszxzEXs1SBE9IlciJAJ_8C2KGy&index=6 [Accessed 16 May 2022].

www.youtube.com. (n.d.). What is Image Preprocessing? - YouTube. [online] Available at: https://www.youtube.com/watch?v=VMeDbP8gX80&list=PLdkgl0WszxzEXs1SBE9IlciJAJ_8C2KGy&index=5 [Accessed 16 May 2022].

www.youtube.com. (n.d.). Advanced Computer Vision with Python - Full Course - YouTube. [online] Available at: https://www.youtube.com/watch?v=01sAkU_NvOY&list=PLdkgl0WszxzEXs1SBE9IlciJAJ_8C2KGy&index=6&t=354s [Accessed 16 May 2022].

www.youtube.com. (n.d.). FACE RECOGNITION + ATTENDANCE PROJECT | OpenCV Python | Computer Vision - YouTube. [online] Available at: https://www.youtube.com/watch?v=sz25xxF_AVE&list=PLdkgl0WszxzEXs1SBE9IlciJAJ_8C2KGy&index=7&t=32s [Accessed 16 May 2022].

www.youtube.com. (n.d.). OpenCV Python TUTORIAL #4 for Face Recognition and Identification - YouTube. [online] Available at: https://www.youtube.com/watch?v=PmZ29Vta7Vc&list=PLdkgl0WszxzEXs1SBE9IlciJAJ_8C2KGy&index=10 [Accessed 16 May 2022].

www.youtube.com. (n.d.). Computer Vision: Crash Course Computer Science #35 - YouTube. [online] Available at: https://www.youtube.com/watch?v=-4E2-0sxVUM&list=PLdkgl0WszxzEXs1SBE9IlciJAJ_8C2KGy&index=3&t=1s [Accessed 16 May 2022].

www.youtube.com. (n.d.). Epochs, Iterations and Batch Size | Deep Learning Basics - YouTube. [online] Available at: https://www.youtube.com/watch?v=SftOqbMrGfE [Accessed 16 May 2022].

Maklin, C. (2022). Dropout Neural Network Layer In Keras Explained. [online] Medium. Available at: https://towardsdatascience.com/machine-learning-part-20-dropout-keras-layers-explained-8c9f6dc4c9ab?gi=1c6345e9c770 [Accessed 16 May 2022].

The Wine and Spirit Trade Association. (n.d.). Challenge 25. [online] Available at: https://www.wsta.co.uk/challenge-25.

Installation Error Fixes

Coding for Entrepreneurs. (n.d.). Install Tensorflow GPU on Windows using CUDA and cuDNN // Blog // Coding for Entrepreneurs. [online] Available at: https://www.codingforentrepreneurs.com/blog/install-tensorflow-gpu-windows-cuda-cudnn/ [Accessed 15 May 2022].

Techs Magic. (2022). Fix Could not load dynamic library ‘cudart64_110.dll’ error [Steps]. [online] Available at: https://techsmagic.com/fix-could-not-load-dynamic-library-cudart64_110-dll-error-steps [Accessed 15 May 2022].

Stack Overflow. (n.d.). Error importing Seaborn module in Python. [online] Available at: https://stackoverflow.com/questions/28828917/error-importing-seaborn-module-in-python [Accessed 15 May 2022].

Stack Overflow. (n.d.). python - ImportError: No module named pandas. [online] Available at: https://stackoverflow.com/questions/33481974/importerror-no-module-named-pandas [Accessed 15 May 2022].

TechVidvan. (2020). SciPy Installation - Install SciPy for SciPy Programming. [online] Available at: https://techvidvan.com/tutorials/install-scipy/ [Accessed 15 May 2022].

Stack Overflow. (n.d.). python - ImportError: No module named matplotlib.pyplot. [online] Available at: https://stackoverflow.com/questions/18176591/importerror-no-module-named-matplotlib-pyplot [Accessed 15 May 2022].

says, A.D. (2020). Keras Installation and Environment Setup. [online] TechVidvan. Available at: https://techvidvan.com/tutorials/keras-installation/ [Accessed 15 May 2022].