Dear A, B, and C, Wow, what a project! You chose a demanding learning task, which you matched with a hyper-powerful (and efficient) network architecture near the state of the art in computer vision. You handled the challenges of this project very well and ended up with good performance. But you don't stop there - you still point out important caveats and warn about your model's shortcomings. And you don't stop there either - you do a saliency analysis of the neural network for different classes to peek inside of the (massive) neural network to see what information is most important for its classification. A very impressive semester project (not to forget about the experimental trial you ran to establish a human baseline performance level). I'm excited to see you rocket-start your neural networks journey and I am looking forward to seeing what neural networks you will develop in the future. Some details are missing from the report, they are not essential since you also provided the code - but they should nevertheless be documented in the report. There were also some larger conceptual mixups regarding cross validation, see inline comments. I liked your ideas of connecting computer vision with pareidolia in humans. This would need more thinking to work out the connection in more detail, but perhaps you would enjoy reading more about the concept of "inductive biases" in machine learning to connect your ideas to (see, for example, arXiv:1806.01261v3). Cheers, Steve

Dear A, B, and C,

this is an extraordinary project - original choice of task and data, and using advanced deep learning methods. Thus like Steven I am duly impressed. You really got yourselves involved here!

While you use advanced models and methods, there are however still some basics that I feel you didn't yet get right. Your treatment of "10-fold cross validation" is either totally mis-explained, or what you did wasn't cross-validation. Also your interpretation of "overfitting" looks like a misinterpretation of the term. But that hardly diminishes the overall impressive and ambitious work that you did.

For displaying our comments I recommend using Adobe Acrobat - other pdf readers may not display all comments properly. - You find the grading breakdown on the next page.

Best regards, Herbert Jaeger (July 25, 2022)

Evaluation form for semester project report, Neural Networks (AI), Spring 2022

Name of students: A, B and C
Name of grader: Steven Abreu, Herbert Jaeger
Date: July 9th / July 15, 2022

Fill in violet fields with grades on scale 0-10. If not applicable, put weight to zero. For bonus, fill in green field (as grade addition); this bonus will be added to the report grade ("Report Grade", capped at 10.0)
For Latex use, enter a 1 in the red field when Latex was used, else 0.

	grade	weight %	total	criteria
Presentation (40%)				
Figures, tables, pseudocode	9.00	5	0.45	good quality figs? Captions? Symbols explained in caption? Figs etc. referenced in text? Sources of imported graphics given? Legible in B/W printout? Figs informative (not redundant)? Pseudocode transparent?
Technical writing	8.00	35	2.8	clear formulations? Concise formulation? Mastership of technical formulations / math formulations? Good text flow? Appropriate language (not sloppy in technical sections, not too dry in motivation sections?) Clear flow of argumentation? Contents well structured? Correct English, typos? All used external sources (literature, code) referenced? Reference list uniformly and correctly formatted? are experiments/implementations clearly, completely, succintly described? Can experiments be checked/reproduced? results clearly, completely, succintly documented? Technical descriptions done in clear English and/or formulas and/or pseudocoe, without relying on toolbox terminology?
Latex	1.00		0	was Latex used? If yes leave the 1.00, else write 0 in the red field
Technical quality (60%)				
Penetration of subject	9.00	20	1.8	Challenges of data / task well recognized? Used data transparently described and inherent challenges discerned? Clear perception of goals for the prooject? Good task statement? Challenges inherent in dataset perceived and adequately addressed? Appropriate choice of model type? choice of preprocessing, feature extraction, learning algorithm motivated? Motivational examples?
Technical work	9.00	30	2.7	Results connected back to starting question? Statistical basics done properly where applicable (error bars!)? Maths correct? Reasonable choice of model type? Appropriate pre-processing of raw data? Implementations / experiments reasonably thought out? Cross-validation / early stopping appropriately used? Planned professionally (modular, efficient, transparent, documented)? Realistic assessment of qualities / deficiencies of results?
Exhaustiveness	10.00	10	1	Is the amount of work done adequate for an extended "lab exercise"? (if ok, full score here - for extra effort give some bonus below)
Bonus	0.50	100	0.5	extraordinary achievements, e.g. much extra work, very independent work, very difficult topic, interdisciplinary connections, original thinking. Max 1.5 bonus grade points possible; typically fractions of 1.0
Report Grade			***9.25***	

It's All in the Palm of Your Hand

Biometric Classification Using Various CNN Architectures

[student names and emails withheld]

Neural Networks Semester Project - Group ##

Department of Artificial Intelligence
University University of Groningen
9747 AG Groningen, The Netherlands
July 3, 2022

Abstract

Humans are able to collect and process biometric information – that is, unique physiological features defining a person – at impressive speeds. Recent developments in the field of deep learning have also shown that machine learning models have the potential to mimic human performance on certain tasks and, more often, even outperform humans doing the same tasks. Intrigued by these advancements we decided to tackle the challenge of inferring biometric data, specifically age, gender, and skin color, from dorsal and palmar images of hands. For that purpose, two different types of deep learning architectures were implemented, namely ResNet and MobileNet. The models were trained with cross-validation on the "11K Hands" dataset (Afifi, 2019), which includes dorsal and palmar images of hands. Evaluation of test performance was done through various measures, including F1 scores, R^2 scores as well as mean squared and mean absolute error. Furthermore, the decision-making process behind the algorithm was explored through gradient saliency mapping. Reported results show promising performance, although overfitting cannot be entirely ruled out.

Keywords: Convolutional Neural Networks; MobileNets; ResNet; Biometrics; Image Classification

Contents

Introduction

The phenomenon of pareidolia describes the tendency of humans to interpret meaningful patterns in ambiguous stimuli, without there being any pattern to speak of. Particularly common stimuli are faces of other humans - due to evolution, our brains have practically become hard-wired to facilitate this effect (Caruana & Seymour, 2022). Pareidolia is not restricted to faces, however. It can also include other vaguely perceived patterns (e.g., Figure 1). Pareidolia has also previously been implemented in computer vision (Chalup et al., 2010). In contrast to this phenomenon of fallibility, however, humans are capable of effortlessly collecting and classifying information about others simply from their looks. We were fascinated by the efficiency of humans in detection and categorisation, and wondered if this behaviour could be implemented with machine learning - as it has proven capable of outperforming humans before (Fiel & Sablatnig, 2011). Hence, we delved deeper into the topic.

After some additional research, we were inspired by the scientific field of biometrics: biometrics is the extraction and study of unique physiological human features, which can then be used to reveal further information about a given person (Dantcheva et al., 2015). It is a field in which convolutional neural networks (CNN), perhaps unsurprisingly, have become an increasingly common sight (Afifi, 2019); a fact that we believe serves as strong evidence in favour of our idea.

On that note, we will shortly outline the task we set ourselves: our goal was to design a CNN that could infer various details about a person based solely on a picture of their palm or the back of their hand. This single CNN would, ideally, be able to successfully identify a person's age, gender and skin colour by extracting the relevant visual features - these particular attributes were chosen simply because they struck us as interesting and because we had already chosen a sufficient number of prediction targets. Our network was trained on roughly 11,076 pictures of numerous different hands from the "11k Hands" datatset, the details of which will be discussed more thoroughly in the following section.

CNNs can be defined as follows: they are, in essence, multi-layer perceptrons (MLP) containing several layers of "feature maps", in addition to the typical processing, input and output layers seen in standard MLPs. Feature maps, which take their inspiration from the visual mechanisms in organic brains, are collections of "feature-detecting" neurons. These neurons learn specific patterns which they can later identify within a given input, thereby making CNNs naturally suited for image recognition tasks. A more in-depth description is given in the methods section later in our report.

It was this propensity for image recognition, as well as the inherently feature-based nature of CNNs, that led us to consider their capabilities as feature extractors. Our hope was that a CNN, having been trained on a sufficiently large set of data, could learn to extract features from a particular image and make accurate classifications based on those features.

Figure 1: An example of pareidolia

Data

The dataset used for our project, the "Hands and palm images dataset", was found on kaggle. This dataset was, in turn, taken from Mahmoud Afifi - the original author of the paper which proposed the dataset (Afifi, 2019). The dataset is named "11k Hands" and can also be found online along with the original analyses, results and CNN structures built around it.

The dataset consists of 11,076 hand images coming from 190 different people, each yielding a resolution of 1600 x 1200 pixels. The subjects were in the age range of 18 to 75 years old, but we thought it prudent to remove some outliers. Consequentially, the age range shrank to 18 to 30 years while retaining about 99% of the data. To generate a variety of images, the participants were asked to randomly open or close their fingers, resulting in a multitude of hand images with spread fingers. During this process a video was taken, recording both the palmar and dorsal sides of the hands. That is, from the back and front of the hands. Both hands of the subjects were video taped. Structural similarity was investigated to determine suitably different frames to use as end-image results. The original authors removed oversaturated, cropped or blurry images from the dataset. After this evaluation, an average of 58 images per subject were obtained.

The raw data is accompanied by metadata. Namely, a subject ID (integer) was recorded along with the age (integer) of each person. Gender ("male"/"female"), skin colour information ("very fair"/"fair"/"medium"/"dark") as well as the aspect of hand ("dorsal right"/"dorsal left"/"palmar right"/"palmar left") were stored as strings. Truth values (integer values "0"/"1") are used to flag potential accessories (such as rings, watches etc.), nail polish or irregularities (such as bandages, missing parts of fingers or deformed nails). Naturally the images have file names ("Hand_0000062.jpg") which are, admittedly, less useful for our classification task.

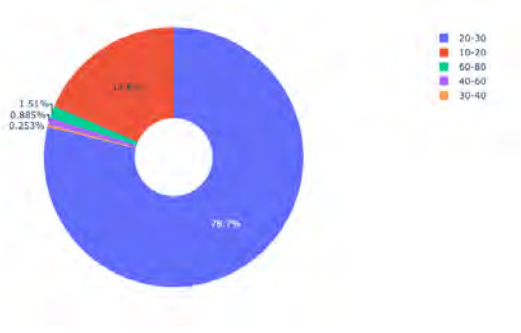

Figure 2: Distribution for Age

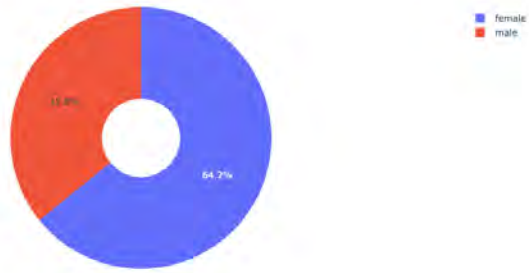

Figure 3: Distribution for Gender

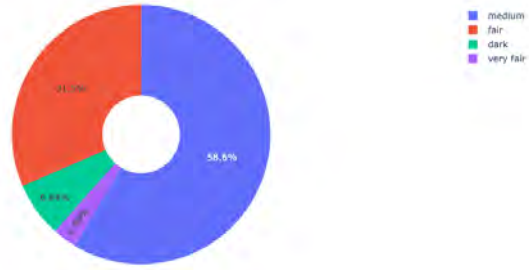

Figure 4: Distribution for Skin Colour

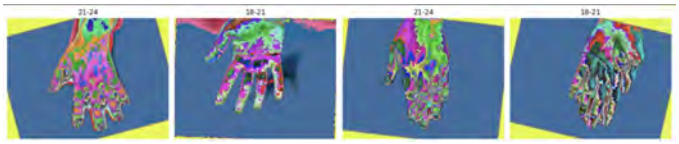

Figure 5: Batch visualisation after data augmentation. The second image shows an unaugmented original image

Methods & Architecture

In the following section we will discuss the various strategies we have used to tackle the uneven distributions in our dataset, build our models and the metrics with which we will be evaluating them. Specifically, we cover data pre-processing, the architectures and hyperparameters used and our chosen performance measures.

Data Pre-processing

Unfortunately, the dataset is riddled with skewed distributions (see Figure 2 to Figure 4 for a graphical overview of the distributions of interest of the data). We therefore had to use data augmentation methods to acquire additional versions of the less represented groups. We oversampled groups that are underrepresented in our dataset and undersampled the overrepresented ones in order to get a balanced training set within each iteration of our 10-fold cross-validation. Oversampling was done by horizontal and vertical flipping of images as well as rotation. Finally, we cropped the images back to the standard size.

Additionally, normalisation of pixel values was performed to bring all images into a closer range. The normalisation mapped all pixel values to floating point values in range [0, 1] and RGB scale. Figure 5 shows a batch visualisation after our augmentation and pre-processing has taken place. After the preparation of the data, the training of our various models could begin.

Cross-validation

Cross-validation is a process by which one can more easily detect, and reduce the risk of, over and under-fitting - hence our decision to use it in building our own model. It involves dividing an initial dataset into two subsets, T and V; the former is the ”reduced training set”, and is used (predictably) to train the model. The latter is the ”validation set”, and is used for testing within training.

Naturally, the first step in implementing cross-validation was to split our dataset into the previously defined subsets T and V. To do so accurately (rather than arbitrarily) we made use of 10-fold cross validation. The idea behind k-fold cross validation can be broadly summarised as follows: first, your initial dataset S is broken up into k number of subsets, all of which should be of roughly equal size. You then assemble every possible combination of these subsets iteratively (in other words, some are placed in the reduced training set and others in the validation set) and, for each one, examine the training error of your model given that combination. This process,

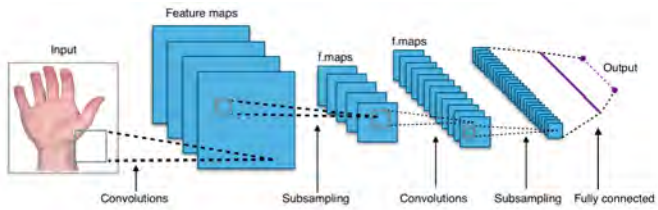

Figure 6: Typical sequence of the layers and operations within a CNN

although computationally expensive, typically gives a good indication of how your initial dataset should be divided up in order to minimize the validation risk.

Our own 10-fold cross validation, for instance, led us to distribute our initial training sample in the following manner: 80% (8860) of our images were placed in the reduced training set, 12% (1330) were placed in the validation set and the remaining 8% (886) were later used for testing.

Conventional Convolutional Neural Networks

Convolutional neural networks, as described briefly at the beginning of our report, are a particularly interesting kind of MLP. In between their input and output layers, CNNs are primarily comprised of a sequence of convolutional and subsampling (or pooling) layers - the length of which varies from network to network.

The convolutional layers are made up of a number of learnable filters, each of which has a predetermined width and height. These filters convolute an incoming input (such as an image or earlier feature map) into a new feature map: in other words, the filters calculate the dot product of a portion of the input and add that dot product to the feature map, repeating this process until the entirety of the original input has been convoluted. The limited scale of the filters means that, once learned, they will only respond to specific patterns (or features).

The subsampling layers are comparatively simple: they receive a number of feature maps, all of the same width and height, and send forward a "minimized" version of each one by combining clusters of neurons into single neurons. There are numerous reasons as to why subsampling layers are such an integral component of CNNs: for one thing they reduce the size of the input, which has the direct benefit of reducing computational cost. Furthermore, they place less of an emphasis on a feature's *exact* location and more of an emphasis on its general location with respect to other features. This is extremely useful in reducing the risk of overfitting.

All of our models were trained with similar parameters. Training was done for 25 epochs for our gender and skin colour models, and 10 epochs for our age model. All three models were trained with 256 batches, each containing 32 samples from the dataset. Additionally, the first layer in each of our models was frozen: this is because we used pre-trained

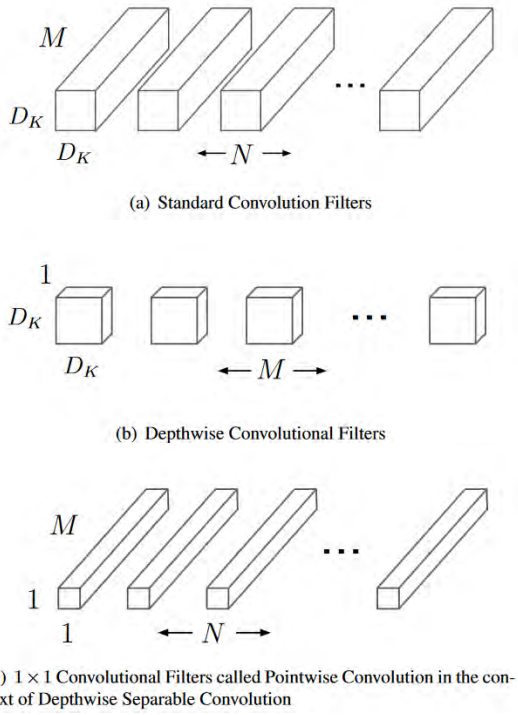

Figure 7: Standard convolutional filters (a) and the replacement through depthwise (b) and pointwise (c) convolution. Taken from Howard et al. (2017)

neural networks as the basis for our own CNNs, meaning that our models could already recognise extremely basic shapes and did not need to be retrained in that regard.

Gender and Skin colour classification

MobileNets Architecture MobileNets, developed by Google Inc. (Howard et al., 2017), are a class of models that aims to be lightweight whilst, simultaneously, not losing too much efficiency. The architecture makes use of depthwise convolution to build these. In a typical MobileNet model, the total number of parameters remains fairly low with 4.2 million. Nevertheless, this network architecture is still able to achieve some depth with 28 layers. Within a MobileNet an initial, standard (3x3) convolution is factorised into depthwise convolutions which are then further augmented by 1x1 pointwise convolutions (Howard et al., 2017; Simonyan & Zisserman, 2014). In total, this results in one standard, 13 depthwise and 13 pointwise convolution layers and one average pooling layer, making up the 28 total ones. This separation of convolution layers allows for the splitting of functions. Specifically, depthwise separable convolution creates a layer for filtering and one for combining. Filtering is done within the depthwise convolution layer, while the pointwise convolution enables the combination. The whole journey of splitting is undertaken to immensely reduce computational costs and model size. Figure 7 provides a graphical overview of this procedure. The final layer is fully connected and feeds into a softmax layer commonly used

for classification (see upcoming Activation function section). Pointwise convolutions make up for the majority of the network architecture with around 75% of the total parameters being part of them. The network also spends most of the computation time (95%) here.

The cost reduction happens because, rather than relying on a product of input and output channels as well as the kernel and feature map size, it splits these and computes a sum of smaller products. This approach achieves eight to nine times lower computation resources (see the original paper for more specific measurements and calculations).

Loss functions Gender and skin colour are classification, rather than a regression, tasks. Hence, we make use of cross-entropy. It is a method that measures how much information (in information theory: bits) is needed for identification of a class from a set, especially when a coding scheme has been optimised (here trained) for an estimated distribution rather than the true distribution. In machine learning it can be used as a loss function to measure the dissimilarity between an estimated distribution and the true distribution. For gender, we could use the binary cross-entropy (BCE) loss function which is defined as follows

$$-\sum_i p_i \log(q_i) = -(y \log(\hat{y}) + (1-y)\log(1-\hat{y})) \quad (1)$$

where p is the predicted classification, q is the true class and the corresponding labels $\hat{y}$ for estimated outcome and y for the true label, respectively.

As we have multiple skin colours, BCE no longer suffices. Instead, we calculate separate losses for each label per observation and sum them together resulting in the categorical cross-entropy loss function

$$-\sum_{c=1}^{M} y_{o,c} \log(p_{o,c}). \quad (2)$$

Here, M is the number of class labels, y the binary indicator if the class label c is correctly classified for observation o and p the predicted probability observation o is of class c.

Activation functions For the wrapper functions, we also made use of two different functions. Specifically, we used the *standard sigmoid function*

$$sigmoid(z) = \sigma(z) = \frac{1}{1+e^{-z}} \quad (3)$$

for our two-class binary classification of gender. To accommodate the multiclass logistic regression necessary for the skin colour classification, we used the *softmax* function

$$softmax(z_i) = \sigma(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \quad for\ i = 1, 2, \ldots, K. \quad (4)$$

Update rule Our networks update their weights with the help of the Adam optimiser. It is an algorithm developed by Kingma and Ba (2014). The Adam optimiser is an extensive procedure that exploits the benefits of other algorithms. Intuitively, it combines gradient descent with momentum to converge to the minimum faster, and root mean square propagation (RMSProp) to adapt the learning rate between iterations. Gradient descent with momentum is a technique used to build up momentum that pushes the updates into a certain direction by taking the moving average of past gradients into account. By using an exponential moving average, more weight can be given to recent gradients. There are some differences between the Adam optimiser and RMSProp with momentum, however: Adam updates are directly estimated and includes bias-correction terms. Without these corrections, the bias can lead to very large step-sizes and eventual divergence. The pseudocode provided in the original paper (see Figure 19 in the appendix) shows the procedure without the involvement of the various mathematical equations (which are described and explored in much depth in the original paper!).

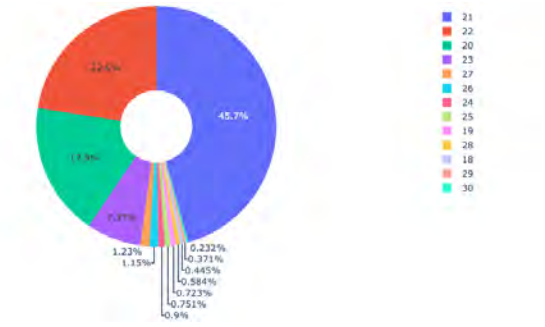

Figure 8: The age distribution without outliers and transformed to a linear scale instead of age groups

Age classification

Instead of classification with limited classes, we implemented an age classification based on linear regression so as to estimate age in years rather than age groups. This changed our distribution, as can be seen in Figure 8

ResNet Architecture We used ResNet-50 as the basis for our age classification CNN. ResNets, or "residual networks", are made up of several so-called "residual blocks" (He et al., 2016). Residual blocks are groups of layers, within which the output of one layer is added to another layer deeper within the block - crucially, however, this output skips over the layers in between the sender and the receiver. ResNets were designed as a response to CNNs with too many layers: although additional layers offered the possibility of more powerful and more varied problem-solving capabilities, they also introduced a slew of unwelcome issues. For one thing, added layers increase the risk of the vanishing and exploding gradient problems. These, in turn, cause the network's accuracy to decrease. In other words, extending the depth of a neural network can have a wholly adverse affect on its performance -

but the "skip connections" utilized by ResNets allow them to achieve a greater number of layers than their standard CNN counterparts.

ResNet-34 was the first neural network of its kind. As the name might imply, it was composed of 34 different layers: aside from the input and output, ResNet-34 contains 16 residual blocks made up of 2 layers (see Figure 9). ResNet-50, by comparison, boasts 16 residual blocks of 3 layers. These blocks are sometimes referred to as "bottleneck residual blocks", owing to their use of 1x1 convolutions at the block's beginning and end (see Figure 10). They allow ResNet-50 to remain efficient and avoid degrading, while still extending its depth to an impressive 50 layers.

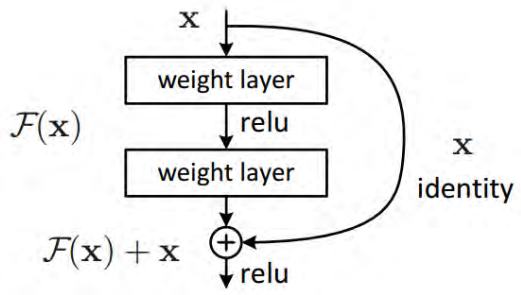

Figure 9: Visual representation of a standard residual block

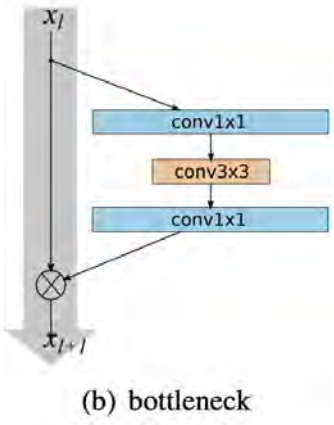

Figure 10: "Bottleneck residual block" as seen in ResNet-50

Loss function As opposed to our gender and ethnicity classification CNNs, this network features the widely-used mean squared error (MSE) as its loss function. As a reminder, MSE is given by:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{5}$$

where y_i is the ith observed value, $\hat{y}_i$ is the corresponding predicted value and N is the number of observations.

The motivation behind our decision to use MSE is relatively straightforward: both of our previous two CNNs, having received an image to classify, attempt to correctly sort it into one of a small number of clearly distinguishable, pre-defined categories. Our age classification CNN, by contrast, attempts to correctly guess the exact age of a given palm's "owner" - it should be obvious, therefore, that expecting consistent, total accuracy in this case is simply unrealistic. By using MSE, we were instead able to measure how close our CNN's guess was to the correct answer/age, rather than simply dismiss its output as incorrect should its guess be off by a few years.

It is worth mentioning that we also looked at the mean absolute error (MAE) when evaluating this particular model. The MAE provides us with a clear, more intuitive (or at least more readable) representation of the network's performance: an MAE of 2, for instance, tells us that our CNN's predictions were, on average, 2 years higher or lower than the correct age.

We will now briefly outline the hyperparameters used for this particular CNN, some of which remain unchanged from our gender and ethnicity classifiers:

Activation Our age classification CNN uses a straightforward linear activation function. Sometimes also referred to as the "identity function", this ensures that the weighted sum of the input is not changed in any way.

$$f(x) = x \tag{6}$$

Optimizer This CNN uses the Adam optimizer, as was also the case for our previous two CNNs.

Performance measures

Next, we will define the metrics we have used to evaluate the goodness of our models.

Mean Squared Error (see Equation 5). MSE is an objective measure that quantifies the difference between the true and predicted value, squaring the average difference across the dataset.

Mean Absolute Error is another objective measure, but it can be a little more intuitive for various dependent variables. MAE represents the difference between the original and predicted values extracted. It uses the averaged absolute difference over the data set. In the case of age, it represents how far off from the truth a model is in years. MAE can be computed as follows:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{7}$$

where N represents the number of observations y which represent true values and $\hat{y}$ which represent predicted values.

R^2 is the coefficient of determination. It is a very helpful metric to measure goodness of fit of correlational regression models. R^2 represents the coefficient of how well the values fit compared to the original values; the higher the value, the better the model. It is usually in the range of 1 (perfect fit) to 0 (a constant line fit) and the values are interpreted as percentages. However, negative values are also possible. They indicate a worse model than a straight baseline (e.g., $x = 1$).

$$R^2 = 1 - \frac{\sum(y_i - \hat{y})^2}{\sum(y_i - \bar{y})^2} \tag{8}$$

where $\bar{y}$ represents the mean value of true distribution of y. It is important to note that R^2 cannot be used for nonlinear regression models, such as logistic regression.

Precision represents the number of true positives among the total number of "guessed" positives. In other words, it is a measure of how many positive guesses were correct.

$$Precision = \frac{TP}{TP + FP} \tag{9}$$

where TP are true positives; correctly identified "true values. FP are false positives; incorrectly identified "true" values.

Recall is, essentially, a measure of how many of the total number of true positives were successfully detected - this is because false negatives can be interpreted as true positives that were misidentified/missed.

$$Recall = \frac{TP}{TP + FN} \tag{10}$$

where FN are false negatives; incorrectly identified "false" values.

F1 scores are weighted combinations of precision and recall. F1 scores are a useful metric to show the performance of classification in one glance.

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} = \frac{2 * TP}{2 * TP + FP + FN} \tag{11}$$

Accuracy represents how many correct predictions a model has produced across all test data.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{12}$$

where TN are true negatives; correctly identified "false" values.

Results

This section will detail the results of our three models, in accordance with the previously described performance measures.

	Precision	Recall	F1-score	Accuracy
Female	0.99	0.83	0.90	
Male	0.76	0.98	0.86	
				0.88

Table 1: Gender classification model performance

	Precision	Recall	F1-score	Accuracy
Very Fair	1.00	0.87	0.93	
Fair	0.97	0.95	0.96	
Medium	0.97	0.99	0.98	
Dark	0.86	1.00	0.92	
				0.97

Table 2: Skin colour classification model performance

In order to have a set of "human results" with which to compare our model, we conducted the following simple experiment: two participants, one of whom was partly aware of the distribution of our data, were shown 32 palmar and dorsal images (drawn randomly from the original 11k Hands dataset) and asked to guess the age, gender and skin colour of the person being shown. The answers of each participant were kept hidden so as not to influence the other. Our participants performed with the following accuracy, on average: with regards skin colour their accuracy was 0.5938, for gender it was 0.5469. In terms of age, our participants performed with an MAE of 9.7656. See Figures 20 and 21 in the appendix for the corresponding confusion matrices. Also consult Figure 22 in the appendix for a general overview of human accuracy.

The above experiment, although extremely rudimentary, provided us with a rough approximation of how well humans perform in these kinds of identification and classification tasks.

Our initial intention, as outlined during the introduction, was to implement a single, multi-output CNN. Unfortunately we were unable to successfully implement such a model, and were instead forced to design and train three separate, single-output CNNs - as described in detail in the previous section. As such, we have no results to show for our multi-output approach. Nevertheless, we lay out the results of our single-output CNNs below.

Gender classification

The results obtained from our gender classifier can be broken down as follows: its final precision (9) value was 0.99 for "female palms" and 0.76 for "male palms". Its recall (10) values were 0.83 and 0.98 for female and male palms, respectively. Its female palm f1-score (11) was 0.90 and its male palm f1-score was 0.86 - as such, the model's total accuracy (12) was 0.88. These results can be seen more clearly in Table 1. Figure 11 shows the corresponding confusion matrix and Figure 12 plots accuracy and mean loss over epochs.

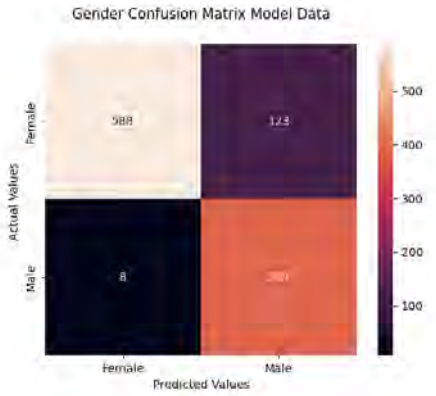

Figure 11: Confusion matrix for our ”Gender” classification model

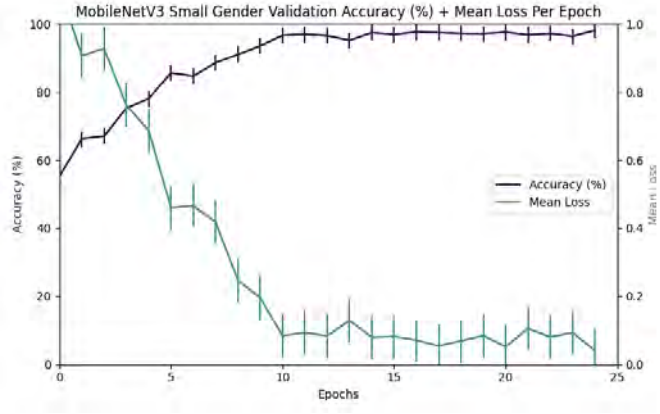

Figure 12: Accuracy and mean loss of our gender classification model, per epoch. Error bars indicate ±1SE for the respective scales

Skin colour classification

Our skin colour classification model performed with an overall accuracy of 0.97. The exact details (its precision, recall, etc.) are shown in Table 2. Again, the confusion matrix of interest can be seen in Figure 13 whereas Figure 14 shows accuracy and mean loss over epochs.

Age classification

Our age classification model produced an R^2 of -0.2169. It also achieved an MAE of 1.2031, as shown in Figure 15.

Discussion

Here, we are going to reflect on what we have learned, explore reasons as to why some ideas have not worked out and unfold what future steps could be taken to improve our current work.

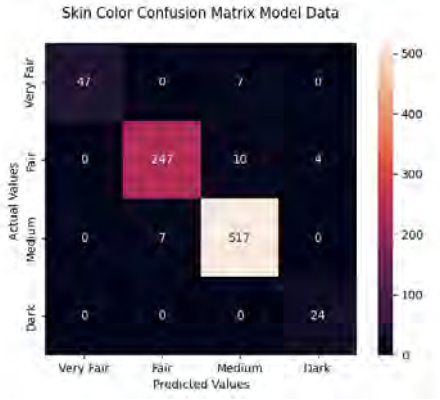

Figure 13: Confusion matrix for ”Skin colour” classification our model

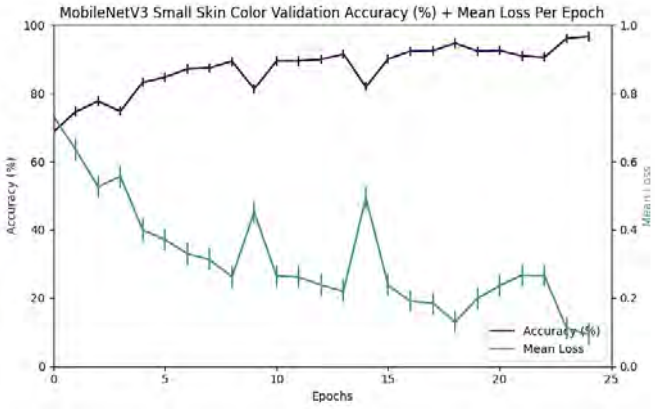

Figure 14: Accuracy and mean loss of our skin colour classification model, per epoch. Error bars indicate ±1SE for the respective scales

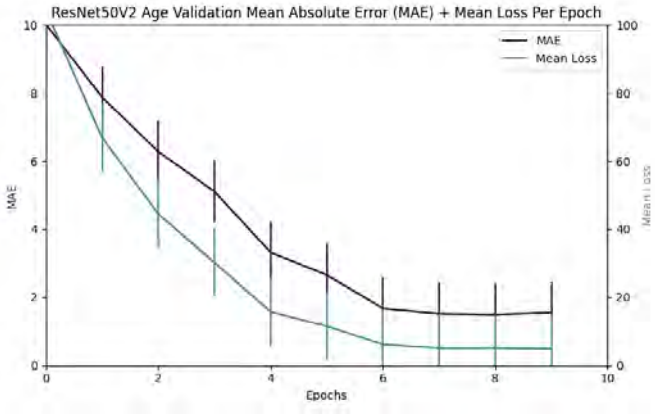

Figure 15: MAE and mean loss of our age classification model, per epoch. Error bars indicate ±1SE for the respective scales

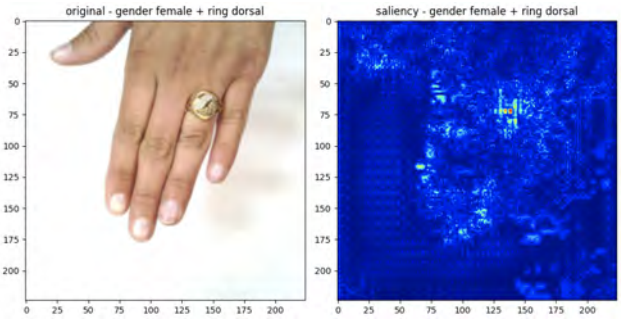

Figure 16: Saliency map showing a clear bias towards the "Female" label when a ring is worn

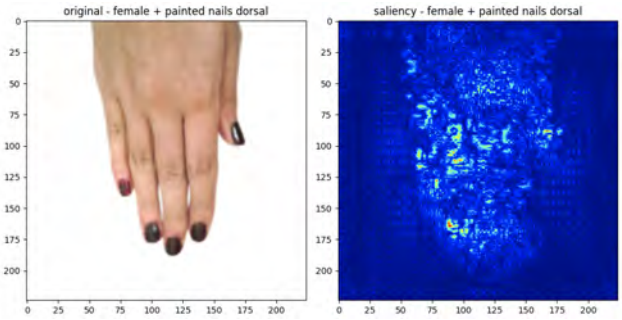

Figure 17: Saliency map showing a clear bias towards the "Female" label when nails are painted

Outcomes

Our "human results", although certainly not representative of the broader population, provide us with a basic grasp of how well our models perform in comparison to the average person: in all three areas (gender, skin colour and age) our models vastly outperformed our human participants. With regards to gender classification, for instance, our model's judgements are clearly more accurate (and therefore reasoned, as indicated by Figure 16) than the near-random judgements being made by our participants.

Saliency maps are able to give some insight into the decision-making process of the various networks by visualising what the algorithm pays most attention to. The saliency maps for age, gender, and skin color were obtained by back-propagating the gradient through the CNN to obtain the output derivative and plotting the max values of each output pixel. In Figure 16 we can see that the defining feature for the taken decision ("Female" class) is a ring. This could be connected to the fact that accessory wearing is much more prominent amongst women within the data set. The same phenomenon can be seen in Figure 17; painted nails are a basis of bias toward the "Female" class for the model. The saliency map for age classification (which is very similar to Figure 16) also seems to pay close attention to any ring presence, and usually is inclined to classify persons as older when a ring is present. Human interpretation behind that inclination would be the notion that marital status is usually attained later in life, and the model learns that implicitly.

However, salience maps do not tell the whole story. Comparing Figure 16 with one from the skin colour predictor (e.g., Figure 18), we remain uncertain of what is really going on in the black box of neural networks. The focal points here are much less intuitive than the highlighted ring for gender classification.

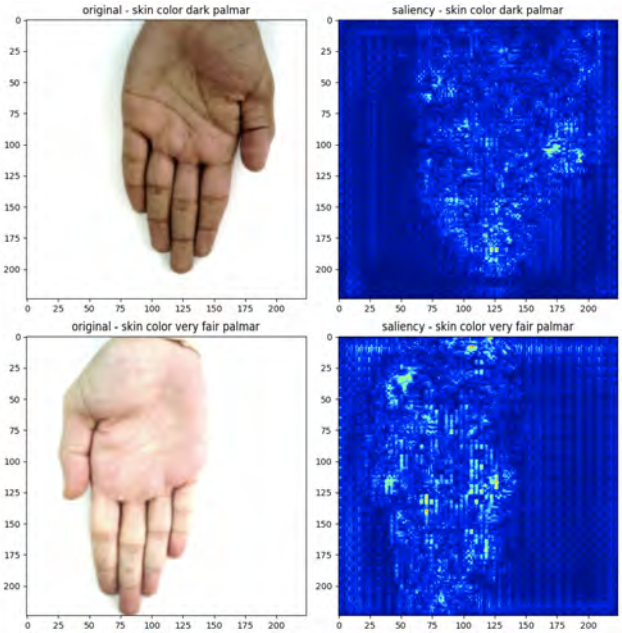

Figure 18: Saliency map for skin colour prediction that gives little insight into the black box of neural networks

General Discussion

Irrespective of how they compare to humans, the performance of all three of our models is very encouraging. There are, however, a handful of flaws and potential problems that should be highlighted, particularly with regards to our age classification model. Firstly, it proved to be extremely slow to train: its 10 epoch training cycle took upwards of 6 hours to complete. In comparison, our other two models both completed their 25 epoch training cycles in roughly the same time. We are confident that the age classification model's architecture (ResNet) is the cause of this particular issue: MobileNets, as mentioned previously, are exceptionally good at reducing computational cost. Their design significantly reduces the number of involved parameters, thereby allowing MobileNet models to be trained relatively quickly on large sets of data. The problem, therefore, is that ResNets clearly do not allow for this same level of efficiency. Although residual networks offer many benefits, our results seem to suggest that they are outclassed by MobileNets in terms of parameter and computational cost reduction.

Another, separate issue with our age classification model is that of overfitting. Although the model's MAE is quite low, its negative R^2 value is a clear indication that it may be over-

fitted to the unbalanced data distribution. Our skin colour classification model, which performs with near-perfect accuracy, must also be considered a potential case of overfitting for the simple reason that our dataset's skin colour distribution is, also, significantly skewed. To be specific, 58.6% of our palm images are of the "Medium skin colour" variety. At the very least, this figure tells us that we should treat the accuracy of our skin colour classifier with caution. It should be noted that our gender classification model - which also performs with a high degree of accuracy - shows little indication of having been overfitted. Nevertheless, it is important to keep in mind that 25 epochs may be too many and introduce some risk of overfitting (Ying, 2019) - a point that applies to both of our MobileNet models. There remain parameters we can tune a little more, but there is only so much we can do. To ensure a lower risk of overfitting across all categories a larger, more balanced dataset is necessary.

There are, however, several positive aspects of our implementation that bear mentioning. For one thing, augmenting within folds (and only after splitting our data) allowed us to confidently rule out the possibility of a data leakage. Additionally, our decision to remove outliers from the original dataset and focus on palms representing ages between 10 and 30 led to an overall much more intuitive age classifier: originally, our age classification model sorted palms into one of several "age range categories", in much the same fashion as our other two models. This was our first attempt at tackling the issue of outliers, as forcing our model to try and guess a palm's exact age, when the range of possible answers was from 18 to 75, resulted in a largely unsatisfactory performance. We eventually decided that the best approach was simply to remove the outliers altogether. As a result of the drastically reduced range of "possible answers", our age classification model was able to accurately guess the exact age of a palm's owner and avoid being tied to the same categorisation system as its fellows.

Reflection and Conclusion

Our original goal was to develop a single, multi-output CNN capable of extracting the relevant features from a palm picture and determining a person's age, gender and skin colour. As we soon discovered, however, this was a somewhat overambitious goal: we were instead forced to implement multiple, single-output CNNs, with each one being designed to examine a single feature (i.e.: age, gender or skin colour).

Although partly disappointing, this initial hurdle did open our eyes to the various difficulties surrounding multi-output regression tasks, as well as some areas in which our original design could be improved. For one thing, the sheer computational power demanded by multi-output CNNs may very well make them fundamentally unsuited to projects of our scale. It is also possible that our original network was trained in too few epochs, or that its depth/overall size was insufficient - but again, these are problems that are difficult to approach given our limited timeframe and budget.

Ultimately, despite the fact that our single multi-output CNN never came to fruition, we are confident that our three, single-output CNNs have fully accomplished the task we initially set ourselves: their performance would seem to indicate that neural networks are indeed capable of the same kind of classification that humans routinely exhibit, and potentially to a significantly higher level of accuracy. One possible way to further improve upon the classification capabilities of neural networks could be to treat pareidolia as a heuristic that facilitates information processing of crucial information, rather than an undesirable human flaw. We are social animals; we thrive through cooperation, and quickly detecting whether someone is hostile or friendly is of great importance, even to this day. What if we were to exploit pareidolia and apply it to computer vision, such that we bias our CNNs in favour of patterns that we deem important for the task at hand? Chalup et al. (2010) have taken some first steps in replicating pareidolia in computer vision. They, however, focused on aesthetics rather than utility. If we were to bias surveillance cameras with image classifiers towards stimuli of importance (weapons, for example) even faster processing times do not seem out of reach. Future research is necessary to see if pareidolia, usually seen as a human flaw, can be used to our advantage when tailored to the task at hand.

References

Afifi, M. (2019). 11k hands: Gender recognition and biometric identification using a large dataset of hand images. *Multimedia Tools and Applications*. https://doi.org/10.1007/s11042-019-7424-8

Caruana, N., & Seymour, K. (2022). Objects that induce face pareidolia are prioritized by the visual system. *British Journal of Psychology*, *113*(2), 496–507. http://search.ebscohost.com.proxy-ub.rug.nl/login.aspx?direct=true&db=psyh&AN=2022-16705-001&site=ehost-live&scope=site

Chalup, S., Hong, K., & Ostwald, M. (2010). Simulating pareidolia of faces for architectural image analysis. *2*, 262–278.

Dantcheva, A., Elia, P., & Ross, A. (2015). What else does your biometric data reveal? a survey on soft biometrics. *IEEE Transactions on Information Forensics and Security*, *11*(3), 441–467.

Fiel, S., & Sablatnig, R. (2011). Automated identification of tree species from images of the bark, leaves and needles. *Proceedings of the 16th Computer Vision Winter Workshop*, *16*, pp. 67–74.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. https://doi.org/10.48550/ARXIV.1704.04861

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. https://doi.org/10.48550/ARXIV.1412.6980

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. https://doi.org/10.48550/ARXIV.1409.1556

Ying, X. (2019). An overview of overfitting and its solutions. *Journal of Physics: Conference Series*, *1168*, 022022. https://doi.org/10.1088/1742-6596/1168/2/022022

Appendix

Algorithm 1: *Adam*, our proposed algorithm for stochastic optimization. See section 2 for details, and for a slightly more efficient (but less clear) order of computation. g_t^2 indicates the elementwise square $g_t \odot g_t$. Good default settings for the tested machine learning problems are $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$. All operations on vectors are element-wise. With β_1^t and β_2^t we denote β_1 and β_2 to the power t.

Require: α: Stepsize
Require: $\beta_1, \beta_2 \in [0, 1)$: Exponential decay rates for the moment estimates
Require: $f(\theta)$: Stochastic objective function with parameters θ
Require: θ_0: Initial parameter vector
  $m_0 \leftarrow 0$ (Initialize 1st moment vector)
  $v_0 \leftarrow 0$ (Initialize 2nd moment vector)
  $t \leftarrow 0$ (Initialize timestep)
  while θ_t not converged **do**
    $t \leftarrow t + 1$
    $g_t \leftarrow \nabla_\theta f_t(\theta_{t-1})$ (Get gradients w.r.t. stochastic objective at timestep t)
    $m_t \leftarrow \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$ (Update biased first moment estimate)
    $v_t \leftarrow \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2$ (Update biased second raw moment estimate)
    $\widehat{m}_t \leftarrow m_t / (1 - \beta_1^t)$ (Compute bias-corrected first moment estimate)
    $\widehat{v}_t \leftarrow v_t / (1 - \beta_2^t)$ (Compute bias-corrected second raw moment estimate)
    $\theta_t \leftarrow \theta_{t-1} - \alpha \cdot \widehat{m}_t / (\sqrt{\widehat{v}_t} + \epsilon)$ (Update parameters)
  end while
  return θ_t (Resulting parameters)

Figure 19: Pseudocode of the Adam algorithm. Taken from Kingma and Ba (2014)

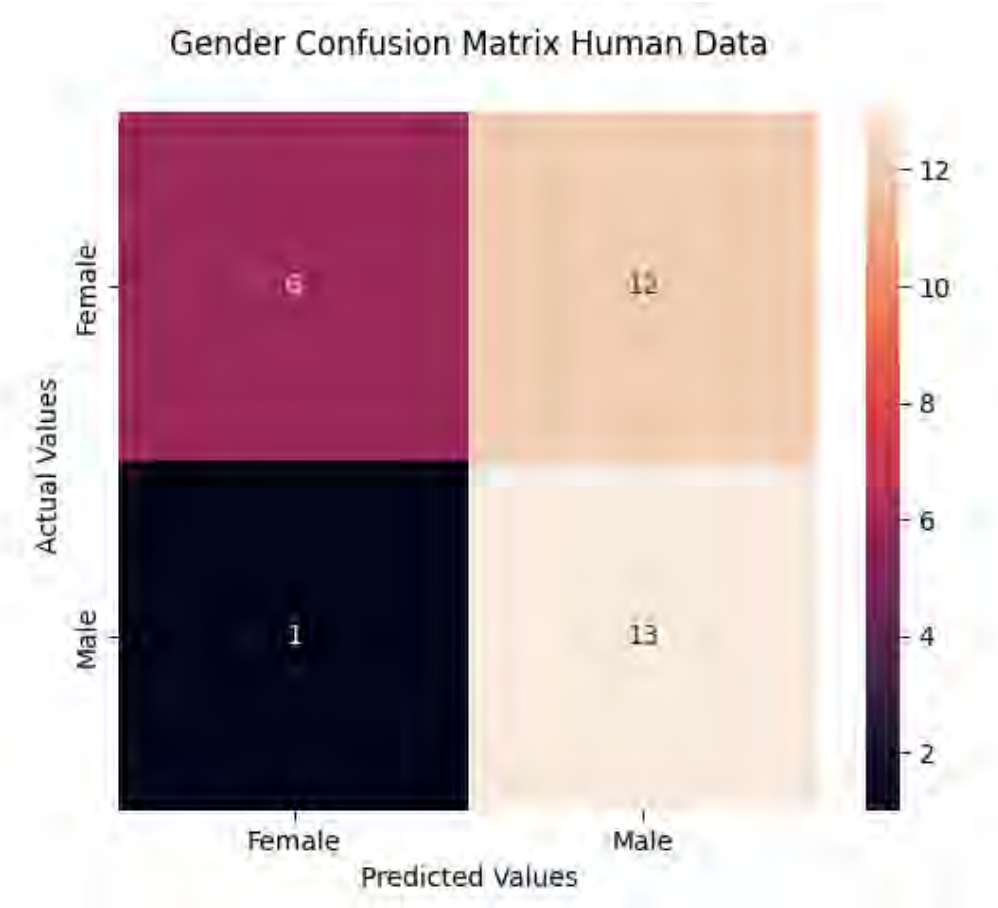

Figure 20: Gender confusion matrix for one of our participants who performed with an accuracy of 0.59

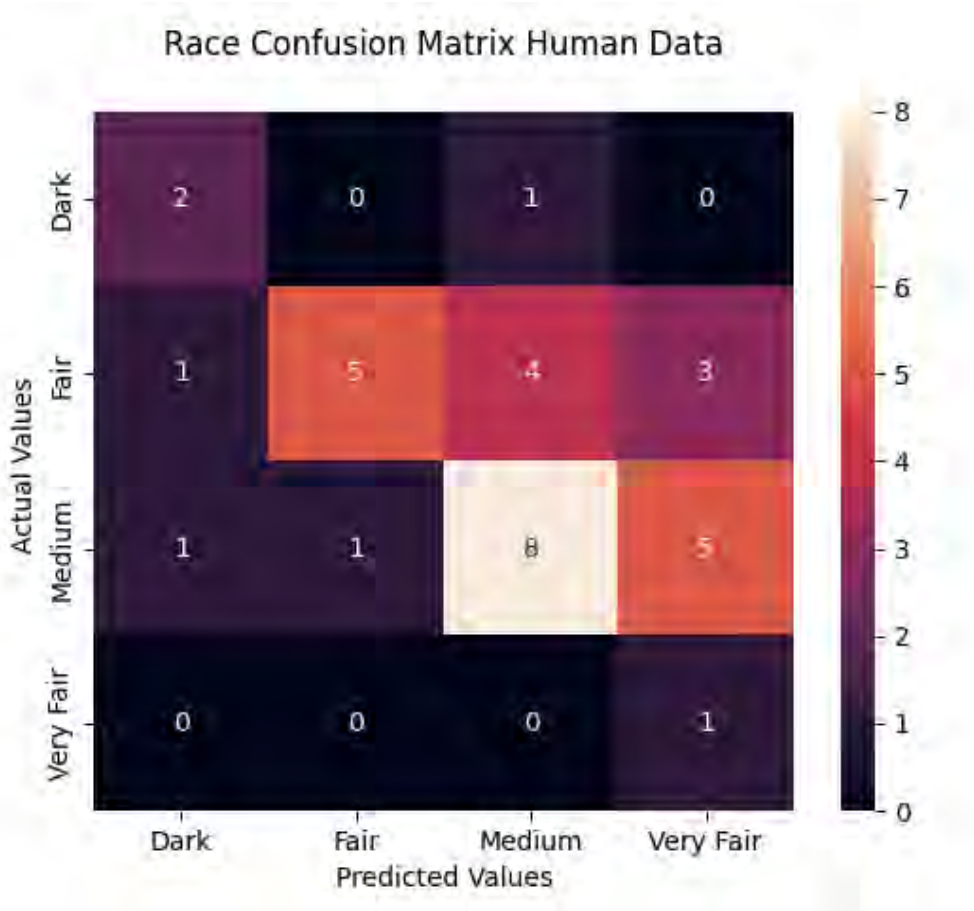

Figure 21: Skin Colour confusion matrix for our other participant who performed with an accuracy of 0.5

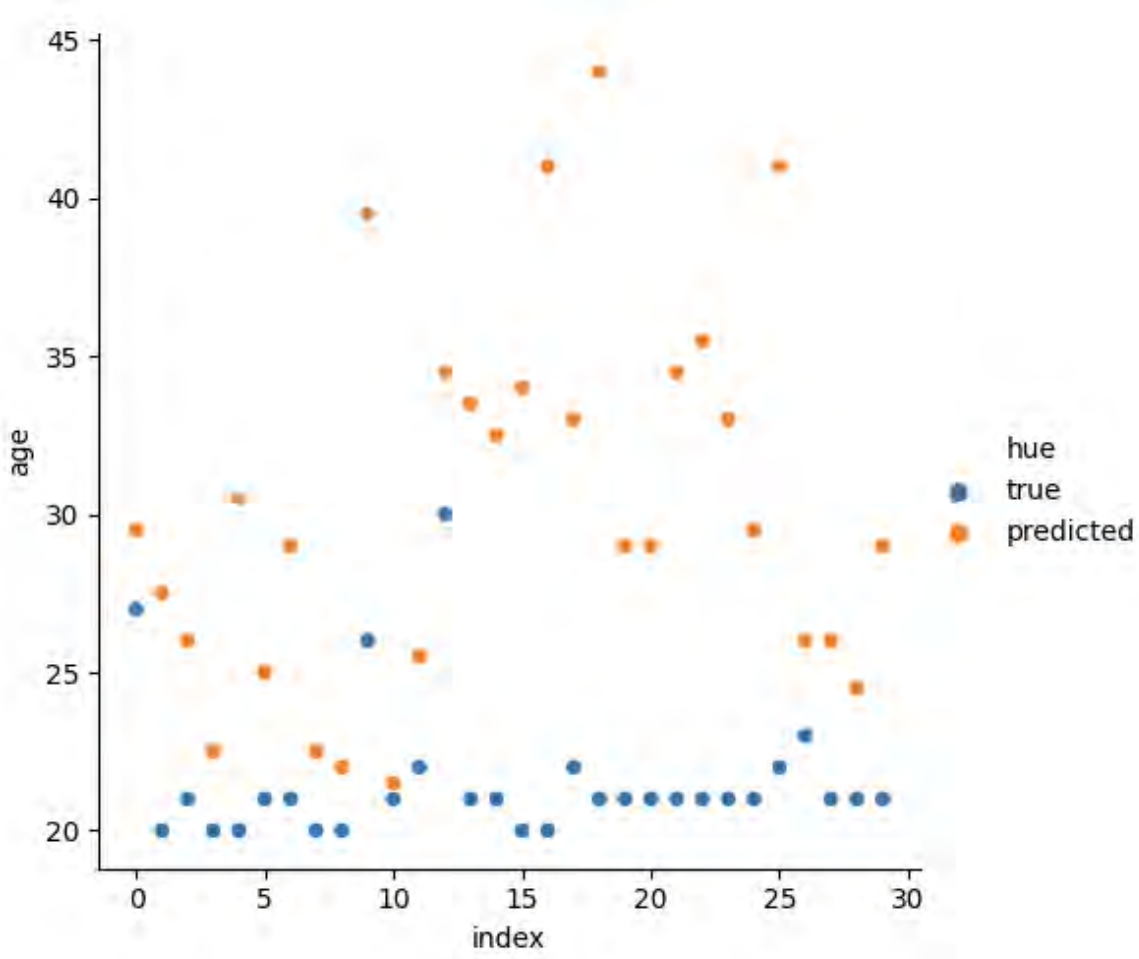

Figure 22: Age scatterplot showing the accuracy of our participants in contrast to the true values