COMP 1804
Applied Machine Learning Report

Khassan Nazhem
001151725
Dr. Dimitrios Kollias
April 16, 2021

Table of Contents

List of Tables

List of Figures

A Face Annotations Study

***Abstract* – This report focuses on designing a machine learning solution to detect human facial attributes. The solution tackles several aspects of the problem of facial attributes such as data collection and processing to ensure clarity of images and annotations data, the suitable methodology used to achieve the best accuracy, and the evaluation among other states of the artwork on related tasks. To achieve this, two machine learning algorithms were applied: Convolutional Neural Network (CNN) and ResNet-50. Running such algorithms led to different results and conclusions where CNN showed more efficiency than ResNet-50 due to technical difficulties and lack of images in the dataset. Further study is required to fulfil existing gaps through applying higher level algorithms and datasets.**

***Index Terms* – Machine learning, CNN, Deep Learning, Face Recognition, Facial Features, TensorFlow, Python.**

1. Introduction

A. Motivation

In the biometric recognition field, facial attribute recognition is an on-growing topic. The use of face recognition is rapidly growing and is being applied to many means. Currently, this technology is used in various fields like identity registration, security passes for smart devices, and artificial intelligence. Hence, extensive research on this task is required through implementing multiple Machine Learning – Deep Learning – solutions for facial attribute recognition and classification. Many algorithms are based on Deep Learning, some with high performance and average accuracy, and others offering high accuracy with average performance. However, the priority will be for Convolutional Neural Network (CNN), a Deep Learning-based algorithm, which is used in this case offering several designs to achieve successful facial attributes detection and classification (Kattenborn *et al.*, 2021).

B. Goals

The proposed solution will provide high accuracy results on detecting specific face attributes such as the presence of wrinkles, freckles, glasses (eyeglasses and sunglasses), the color of hair, and the length of the hair. To achieve such outcomes, the program follows several stages. The initial stage is data collection, accomplished by using an annotation tool to derive facial attributes information from the human image dataset. Next is preprocessing and partitioning, this stage handles the preparation of data through cleaning and organizing. Such preprocessing allows for more suitable appliances for building and training Machine Learning models by splitting it into training and testing datasets. This is followed by applying two Deep Learning algorithms: CNN and ResNet-50. The main advantage of these algorithms is that they support unsupervised training focusing on automatic detection of the important features (Kattenborn *et al.*, 2021). After training the algorithm, it is tested on a different dataset known as the validation dataset, where the algorithm should be able to automatically classify the mentioned facial attributes. Finally, the conclusion will be derived by comparing the used algorithms with the available ones and evaluating their performance through discussing various accuracy metrics.

2. Related Work

The development of Deep Learning architectures to upgrade the feature recognition model has been the focus in recent years. This led to the creation of many famous architectures like VGG, ResNet, AlexNet and so on which are state-of-the-art for achieving tasks like feature detection. A recent study was done by Li and Lima which focused on enhancing the gradient for loss function in facial recognition using ResNet-50 (Li and Lima, 2021).

Residual Network is an architecture that provides a solution for the vanishing gradient problem. This usually occurs in deep networks due to the inability to find the proper weights provided by the loss function. ResNet relies on skipping some layers and the activation function of previous layers (Jampour, Abbaasi and Javidi, 2021).

In the paper by Li and Lima, the goal was to use the deep residual network (ResNet-50) since it pools convolutional neural networks for feature detection. Starting with preprocessing the collected images, a set of 700 images which include different people expressing a variety of emotions like a neutral face, happiness, sadness, anger, astonishment, fright, and disgust were used. Next, the features were extracted and classified with a convolutional neural network. This network provides great performance in filtering the image features and detecting edges through two layers. The lower layer can detect edges and angles in the images whilst the higher layer can detect more complicated features to enhance the image classification (Li and Lima, 2021). The authors proceed with applying two pooling layers (Average Pooling and Max Pooling) which can help in preventing overfitting and integrating the detected feature map.

The output of the used methodology provided two conclusions. One of the conclusions was that the main problem - the decreasing gradient - occurred due to the increasing number of layers in the network, this was solved when applied the ReLU activation function and using batch normalization. The second conclusion is that usually, the increase in the depth of the network leads to better performance, however, the study found that with the increase in the depth of the network, the accuracy was decreasing (Li and Lima, 2021).With that being said, the work in this paper will focus on applying a custom CNN and the ResNet-50 model to get an on-hands evaluation metrics and identify the limitations.

3. Methodology

A. Data Collection

The data collection process consists mainly of annotating the human images dataset. Data annotation is very important since it provides the labels that will be fed for the machine learning algorithm. Using the data annotation tool provided by the instructor, the 1,990 images were successfully annotated and saved into "annotations.csv" according to the following structure: image name, wrinkles, freckles, glasses, hair color, hair top, and human. The values of the mentioned annotations include:

- image name (text)
- wrinkles (binary: has/does not have), class 0: does not have, class 1: has
- freckles (binary: has/ does not have), class 0: does not have, class 1: has
- glasses (3 values: do not wear/wear normal/wear sunglasses), class 0: does not wear, class 1: wear normal, class 2: wear sunglasses
- hair color (9 values: brown/black/gray/blond/red/white/mixed/other), class 0: brown, class 1: black, class 2: gray, class 3: blond, class 4: red, class 5: white, class 6: mixed, class 7: other, class 8: not visible
- hair top (4 values: bald or shaved, has few hairs, has thick hair), class 0: bald or shaved, class 1: has few hair, class 2: has thick hair, class 3: not visible
- not human (binary is human/is not human), class 0: is human, class 1: is not human

B. Preprocessing and Partitioning

The preprocessing and partitioning consist of several steps. Initially, the "annotations.csv" file is loaded and saved into a dataframe that will be handy on a later stage. The dataframe is sorted according to the name of the image. Upon loading the dataframe, all rows where the not human attribute is equal to 1 are dropped since the image does not represent human features.

	image	wrinkles	freckles	glasses	hair_color	hair_top	not_human
0	33.jpg	0	0	1	8	3	0
1	172.jpg	1	0	0	8	3	0
2	265.jpg	1	0	0	2	1	0
3	370.jpg	1	0	1	2	1	0
4	390.jpg	1	0	0	3	2	0

Figure 1. Data Labels

This is proceeded by loading the 1,990 images from the data file and saving them sorted as well according to the file name. While the images are being loaded, the algorithm checks whether the image is human or not - if not, it is not loaded. By implementing such functions to remove non-human images, the number of images decreased to 1,823. Moreover, the images are all resized to a scale that matches all of the existing images; width and height of 128. Due to the low number of images, some annotations have a low count especially for freckles, wrinkles and presence of glasses. So, to solve this issue, each image that presents freckles, wrinkles or glasses is added twice into the array. These images are also slightly augmented with rotation or flipping and then added into the array of images. This led to a huge increase in the image samples to 2,495 for wrinkles, 1,869 for freckles, 2,275 for glasses, 3,371 for hair color and 2,415 for hair top.

Figure 2. Before and After Augmentation

After loading the labels dataframe and all the images in a usable format, they are transformed into a NumPy array of "uint8" type so that they can be split and partitioned. Using the train split function, the two dataframes are split into training and testing images from the images set, training and testing labels from the labels dataframe. Giving it a test size of 0.01-0.1 so the numbers are split accordingly: 20-200 images and labels for validating and the rest for training according to the feature: wrinkles – 2,470, freckles – 1,850, glasses – 2,252, hair color – 3,340 and hair top – 3,211. This will help in maintaining the algorithm and avoid overfitting on a later stage.

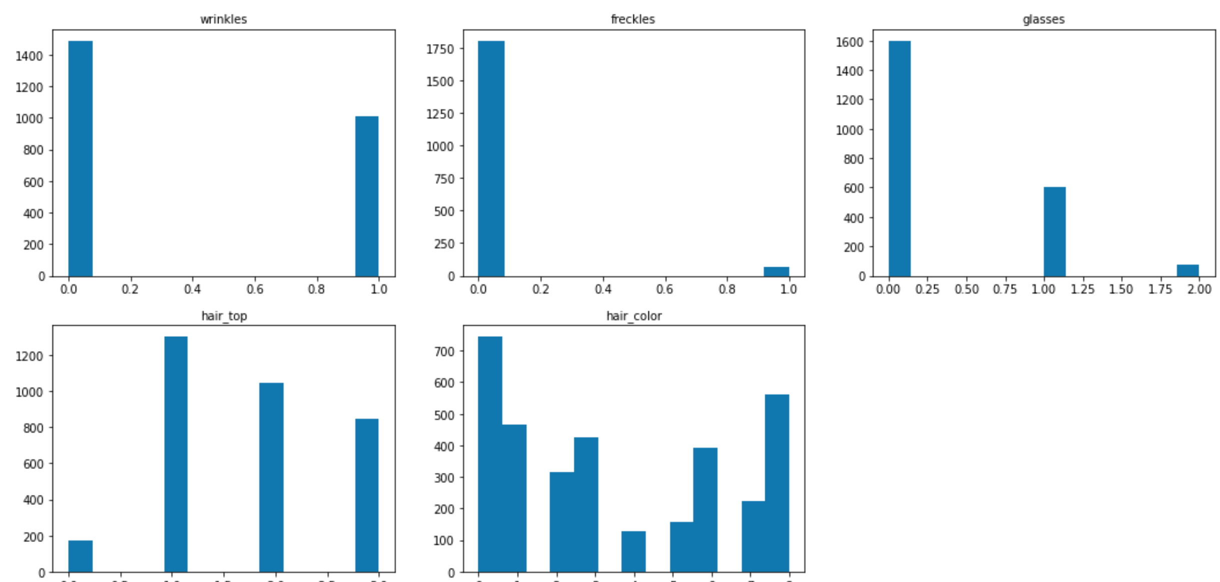

Figure 3. Feature Distributions

All the sets (training images and labels, testing images and labels) are normalized. Normalization is better for this algorithm since it is more suitable than standardization when it comes to neural networks, transforming all the values to a range of 0-1 (Yu *et al.*, 2020).

Finally, to make sure that the image dataset and the corresponding labels are loaded correctly, 16 images are printed with their labels to ensure that they are matching.

C. Machine Learning Algorithm

Before applying the custom CNN layers, an additional Keras data augmentation is applied on the dataset which includes randomly flipping, zooming in and rotating the image.

CNN is a deep neural network algorithm that is used for image recognition and feature detection due its high precision. The model is based on creating a network of interconnected neurons to process different layers using weights and proceeds with outputting the features (Memon *et al.*, 2021).

The first layer in a CNN network is the Convolutional Layer, which is responsible for applying filters or kernels on the input image. Four Convolutional Layer area applied in this solution. Typically, the layer's filters must be smaller than the input so that it can pass over all the images. In this case, a kernel of (3,3) is applied on a (128,128,3) input; the 3 is the width and height of filter (kernel) which is based on the RGB depth of image and 128 is the width and height of the input image. In addition, each image has an extra padding so that filter does not miss the start and ending rows and columns. The Activation Layer applies the ReLU (Rectified Linear Unit), in this algorithm it is applied to increase non-linearity in the model.

As for the Pooling Layer, it is applied after each Convolutional Layer to make the input image smaller, making them four pooling layers as well. This layer takes a pooling window of (2,2). Before the last Convolutional Layer, a drop out layer of value (0.3: drops 30%) is applied to ensure that the algorithm doesn't overfit by dropping some of the inputs since the number of images used is not very high. Finally, the Fully Connected Layer is added which involves flattening and transforming the feature map into a single array.

The output of the model applies an activation function such as SoftMax: multiple outputs; 3 for Glasses, 4 for Hair Top and 9 for Hair Color CNNs or Sigmoid: two outputs 1 or 0 for Wrinkles and Freckles CNNs. Moreover, the model uses Adam's optimizer of a learning rate 0.001 and 0.0001 depending on the target, Loss function of Sparse Categorical Cross Entropy, Epochs 10, 20 and 25 depending on the target and Batch Size of 64 and 128.

D. Evaluation

For experimental purposes, not only was a custom CNN applied, but also a ResNet-50 to compare and analyze the different outcomes. ResNet-50 provides a solution for the vanishing gradient problem in deep networks due to the inability to find the proper weights. To avoid this problem, ResNet skips skipping some layer and relies on the activation function of previous layers (Jampour, Abbaasi and Javidi, 2021).

However, due the size of the data and the imbalance of feature count, ResNet-50 was not able to apply its full potential, thus giving the custom CNN better performance. Even though the custom CNN had a better performance, it was still leading to low accuracy values. This can be interpreted as the model is overfitting since it does not carry sufficient data to learn about the varying features.

With that being said, the evaluation will focus on the custom CNN. Beginning with an overall advantages and disadvantages of using custom CNN on the dataset available. CNN can compile much faster than other algorithms especially when run on GPU since it is designed to run computations in parallel (Ting, 2020). Moreover, working on a set of images, CNN is encouraged since it fits best in feature detection and recognizing objects which include colors, shapes, and edges (Xiao, Song and Gupta, 2021). However, as mentioned before CNN runs best on computers with high quality hardware and super GPUs, thus adding a huge set of images would be inefficient to perform the algorithm locally. Besides, the images should always be accompanied with labels dataset; applying labels and images on 10,000 images is very time consuming. Finally, to achieve high accuracy, the dataset should be large enough to apply image detection on every available label.

The following pages detail the study of each label using custom CNN.

I. Wrinkles custom CNN

Applying CNN on the wrinkle's dataset was very successful since the images demonstrated a wide range for both wrinkles available and not available. Having at least 1,400 images with no wrinkles and more than 1,000 with wrinkles. Testing the output of the algorithm on the set below gave a successful prediction for 70% of them.

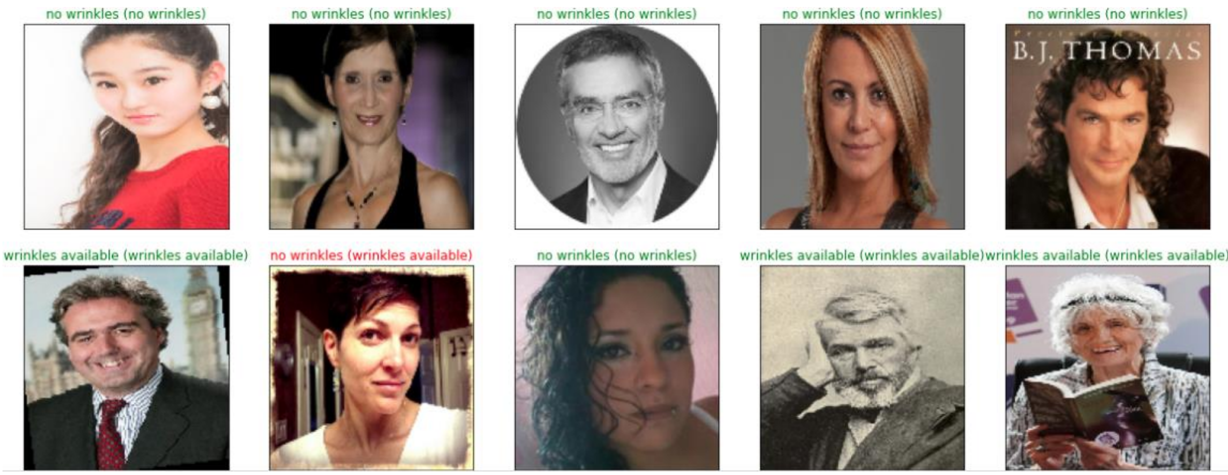

Figure 4. Wrinkles Predictions

	Precision	Recall	F1-Score	Feature Count
0	0.74	0.78	0.76	1400+
1	0.65	0.59	0.62	1000+
accuracy			0.70	
macro avg	0.69	0.69	0.69	
weighted avg	0.70	0.70	0.70	

Table 1. CNN Wrinkles Metrics

Confusion Matrix

116	33
41	60

Accuracy vs Validation Accuracy

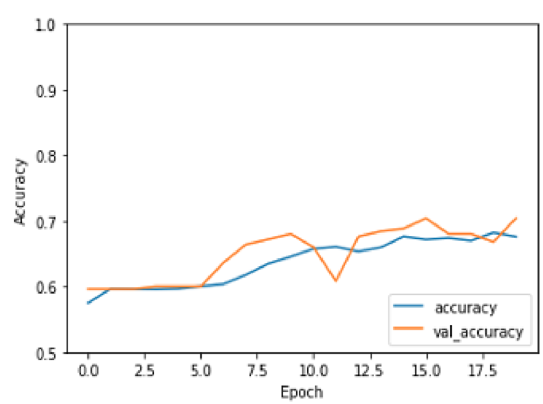

Figure 5. Wrinkles Metrics

II. Freckles custom CNN

Applying CNN on the freckles labels was unsuccessful regardless of all the tries to enhance the algorithm. One of the solutions for instance was to double the number of images with freckles from 70 to approximately 150 and downsizing the number of images that do not present freckles, however this application did not work as the dataset became less than 300 images. Another possible solution was trying hyperparameter tuning via grid search, yet this method was unsuccessful as well, leaving no other solution but to try adding more freckles images in future work. The output below shows how the algorithm is only predicting no freckles for all the images, meaning that it has no knowledge about freckles pattern.

Figure 6. Freckles Predictions

	Precision	Recall	F1-Score	Feature Count
0	0.97	1.00	0.98	1600+
1	0.00	0.00	0.00	< 150
accuracy			0.97	
macro avg	0.48	0.48	0.49	
weighted avg	0.94	0.97	0.95	

Table 2. CNN Freckles Metrics

Confusion Matrix

91	0
3	0

Accuracy vs Validation Accuracy

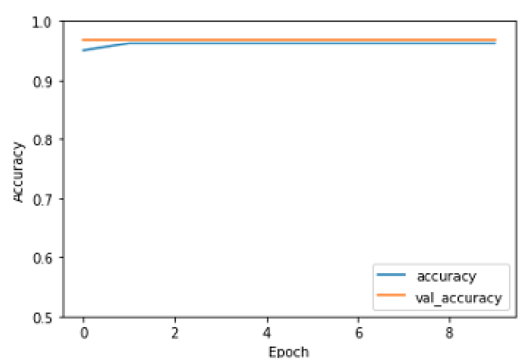

Figure 7. Freckles Metrics

III. Glasses custom CNN

Likewise for applying CNN on the glass's labels, it was unsuccessful. The same solutions were applied such doubling the number of images and downsizing the number of images that do not present glasses. The algorithm has good information about images with no glasses but very small information about the images with eyeglasses (class 1). Further, regarding the sunglasses, images (class 2) all the metrics were equal to 0. The output below shows how the algorithm is only predicting no glasses for all the images, meaning that it has no knowledge about sunglasses.

Figure 8. Glasses Predictions

	Precision	Recall	F1-Score	Feature Count
0	0.75	0.97	0.85	1400+
1	0.73	0.26	0.39	< 400
2	0.00	0.00	0.00	< 70
accuracy			0.75	
macro avg	0.49	0.41	0.41	
weighted avg	0.72	0.75	0.70	

Table 3. CNN Glasses Metrics

Confusion Matrix

158	2	0
51	10	0
6	1	0

Accuracy vs Validation Accuracy

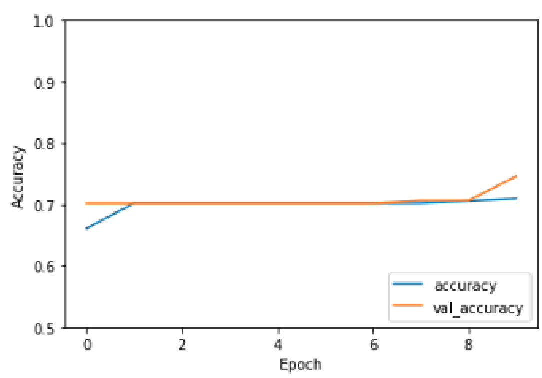

Figure 9. Glasses Metrics

IV. Top Hair custom CNN

Using CNN to detect hair length was a successful choice. Having an equilibrium for the image distribution resulted in great scores for all the features. Regardless, the f1-score for bald features could've been adjusted if the number of images was slightly increased since it has the lowest number out of all.

Figure 10. Top Hair Predictions

	Precision	Recall	F1-Score	Feature Count
0	0.63	0.35	0.45	200+
1	0.67	0.81	0.73	1000
2	0.73	0.57	0.64	800+
3	0.56	0.55	0.55	600+
accuracy			0.65	
macro avg	0.65	0.57	0.59	
weighted avg	0.66	0.65	0.65	

Table 4. CNN Hair Top Metrics

Confusion Matrix

5	14	8	7
0	207	19	55
2	56	102	39
1	40	13	107

Accuracy vs Validation Accuracy

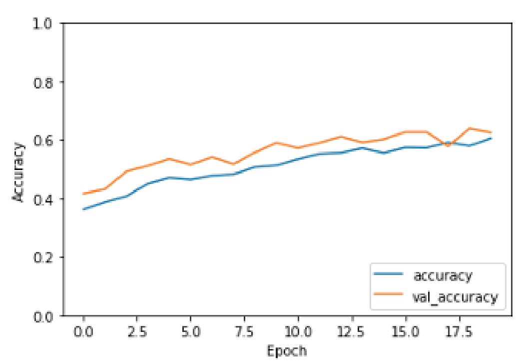

Figure 11. Top Hair Metrics

V. Hair Color custom CNN

The hair color labels included 9 different labels which are not equally distributed among approximately 3200 images, giving some images more value than the others. In this test result, the red color features seem to be insignificant amongst the others since it has the lowest number of images of less than 100. Moreover, the labels are not very precise since the hair color can be a matter of subjectivity and different opinions: different hair colors can be mistaken especially when there is a wide variety and mix of it. For example, blonde, brown, and mixed can be labeled differently according to the annotation tool. The figure below shows a test result of the CNN output which has a successful accuracy for black, white, brown, and grey, but confusion among mixed hair color.

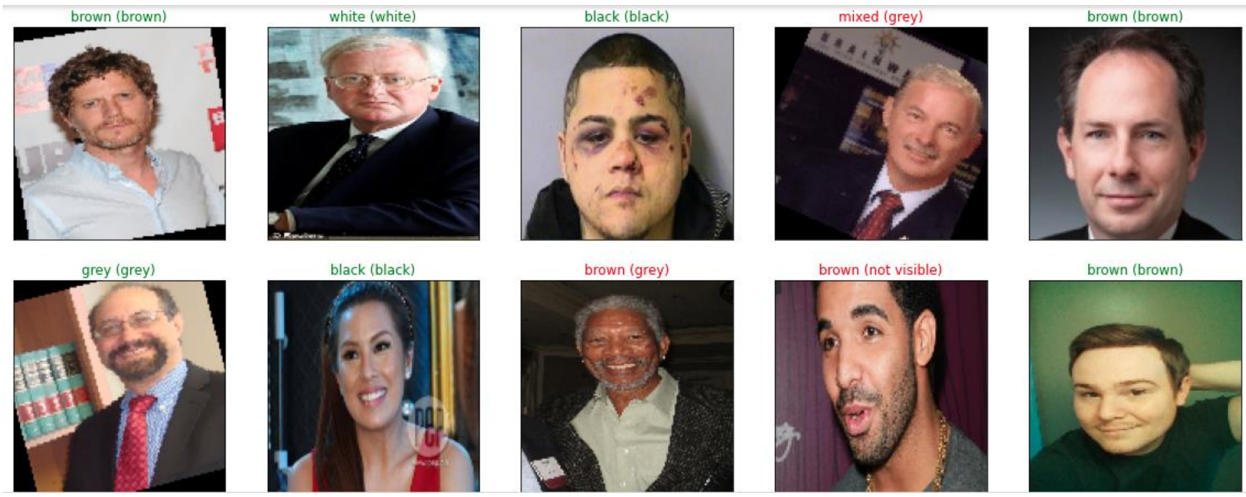

Figure 12. Hair Color Predictions

	Precision	Recall	F1-Score	Feature Count
0	0.28	0.36	0.31	700+
1	0.39	0.64	0.49	400+
2	0.23	0.16	0.19	300+
3	0.67	0.23	0.34	400+
4	0.00	0.00	0.00	< 100
5	0.75	0.33	0.46	100+
6	0.20	0.25	0.22	400+
7	0.12	0.15	0.14	200+
8	0.52	0.44	0.48	500+
accuracy			0.33	
macro avg	0.35	0.29	0.29	
weighted avg	0.37	0.33	0.3	

Table 5. CNN Hair Color Metrics

Confusion Matrix

27	5	0	5	1	1	1	1	4
10	11	0	0	1	1	2	1	2
4	3	3	3	0	1	2	2	1
5	0	2	13	1	1	0	2	2
3	0	0	1	0	1	0	1	2
0	0	0	3	0	3	0	0	3
14	2	0	2	0	1	4	0	1
8	0	0	2	4	0	0	0	1
7	4	2	1	0	0	1	2	17

Accuracy vs Validation Accuracy

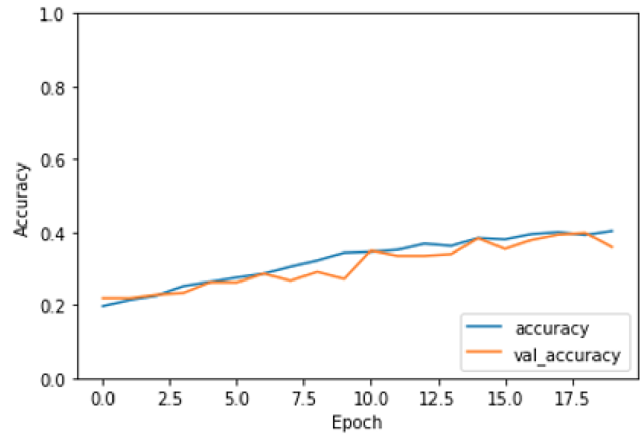

Figure 13. Hair Color Metrics

VI. Resnet-50 on Wrinkles

```
              precision    recall  f1-score   support

           0       0.60      1.00      0.75       149
           1       0.00      0.00      0.00       101

    accuracy                           0.60       250
   macro avg       0.30      0.50      0.37       250
weighted avg       0.36      0.60      0.45       250

CONFUSION MATRIX
[[149   0]
 [101   0]]
```

Figure 14. Resnet Wrinkles Metrics

VII. Resnet-50 on Freckles

```
              precision    recall  f1-score   support

           0       0.97      1.00      0.98        91
           1       0.00      0.00      0.00         3

    accuracy                           0.97        94
   macro avg       0.48      0.50      0.49        94
weighted avg       0.94      0.97      0.95        94

CONFUSION MATRIX
[[91  0]
 [ 3  0]]
```

Figure 15. Resnet Freckles Metrics

VIII. Resnet-50 on Glasses

```
              precision    recall  f1-score   support

           0       0.70      1.00      0.82       160
           1       0.00      0.00      0.00        61
           2       0.00      0.00      0.00         7

    accuracy                           0.70       228
   macro avg       0.23      0.33      0.27       228
weighted avg       0.49      0.70      0.58       228

CONFUSION MATRIX
[[160   0   0]
 [ 61   0   0]
 [  7   0   0]]
```

Figure 16. Resnet Glasses Metrics

IX. Resnet-50 Top Hair

```
              precision    recall  f1-score   support

           0       0.00      0.00      0.00        34
           1       0.41      0.96      0.58       281
           2       0.20      0.02      0.03       199
           3       0.00      0.00      0.00       161

    accuracy                           0.41       675
   macro avg       0.15      0.24      0.15       675
weighted avg       0.23      0.41      0.25       675

CONFUSION MATRIX
[[  0  33   1   0]
 [  0 271  10   0]
 [  0 196   3   0]
 [  0 160   1   0]]
```

Figure 17. Resnet Top Hair Metrics

X. Resnet-50 Hair Color

	precision	recall	f1-score	support
0	0.12	0.02	0.04	45
1	0.25	0.14	0.18	28
2	0.00	0.00	0.00	19
3	0.17	0.12	0.14	26
4	0.00	0.00	0.00	8
5	0.00	0.00	0.00	9
6	0.00	0.00	0.00	24
7	0.00	0.00	0.00	13
8	0.17	0.82	0.28	34
accuracy			0.17	206
macro avg	0.08	0.12	0.07	206
weighted avg	0.11	0.17	0.10	206

```
CONFUSION MATRIX
[[ 1  3  0  4  0  0  0  0 37]
 [ 3  4  0  1  0  0  0  0 20]
 [ 0  4  0  3  0  0  0  0 12]
 [ 1  1  0  3  0  0  0  0 21]
 [ 0  0  0  0  0  0  0  0  8]
 [ 1  0  0  0  0  0  0  0  8]
 [ 1  0  0  0  0  0  0  0 23]
 [ 0  2  0  4  0  0  0  0  7]
 [ 1  2  0  3  0  0  0  0 28]]
```

Figure 18. Resnet Hair Color Metrics

4. Future Work

In the future work, it is important to provide a larger dataset for the algorithm, since in most of the performances the lack of images led to overfitting regardless of the efforts put to increase the persistence of images. Moreover, in the presence of super computers, it would be more feasible to try different algorithms that includes a great variety of hyperparameters and fine tuning using Random Search libraries from Keras Tuners. Such adjustments would highly increase the performance of the algorithm through fixing the weights and biases. Additionally, more optimization algorithms should be tested and include up to date state-of-the-art features detection tasks with an empowered GPU. Finally, feature detection and classifications would be more precise when applying pretrained data and transfer learning especially in cases where the image includes multiple features to be predicted.

5. References

Jampour, M., Abbaasi, S. and Javidi, M. (2021) CapsNet Regularization and its Conjugation with ResNet for Signature Identification, Pattern Recognition, Elsevier BV, p. 107851, [online] Available at: http://dx.doi.org/10.1016/j.patcog.2021.107851.

Kattenborn, T., Leitloff, J., Schiefer, F. and Hinz, S. (2021) Review on Convolutional Neural Networks (CNN) in vegetation remote sensing, ISPRS Journal of Photogrammetry and Remote Sensing, Elsevier BV, 173, pp. 24–49, [online] Available at: http://dx.doi.org/10.1016/j.isprsjprs.2020.12.010.

Li, B. and Lima, D. (2021) Facial expression recognition via ResNet-50, International Journal of Cognitive Computing in Engineering, Elsevier BV, 2, pp. 57–64, [online] Available at: http://dx.doi.org/10.1016/j.ijcce.2021.02.002.

Memon, N., Parikh, H., Patel, S. B., Patel, D. and Patel, V. D. (2021) Automatic land cover classification of multi-resolution dualpol data using convolutional neural network (CNN), Remote Sensing Applications: Society and Environment, Elsevier BV, 22, p. 100491, [online] Available at: http://dx.doi.org/10.1016/j.rsase.2021.100491.

Ting, Z. (2020) A parallel computing method of CNN network convolution and network value optimization solution, Microprocessors and Microsystems, Elsevier BV, p. 103456, [online] Available at: http://dx.doi.org/10.1016/j.micpro.2020.103456.

Xiao, Z., Song, K.-Y. and Gupta, M. M. (2021) Development of a CNN edge detection model of noised X-ray images for enhanced performance of non-destructive testing, Measurement, Elsevier BV, 174, p. 109012, [online] Available at: http://dx.doi.org/10.1016/j.measurement.2021.109012.

Yu, C., Li, Z., Yang, Z., Chen, X. and Su, M. (2020) A feedforward neural network based on normalization and error correction for predicting water resources carrying capacity of a city, Ecological Indicators, Elsevier BV, 118, p. 106724, [online] Available at: http://dx.doi.org/10.1016/j.ecolind.2020.106724.