Machine Learning for Interpretable Age Estimation

Julius Rieser

Abstract—**Age estimation from facial images has been growing as a machine learning topic as it has many real-world applications. It can help with security control for minors, human-computer interaction based on age, and law enforcement concerning identity. These various problems could be solved by building and understanding a machine learning model that labels a facial image into an age range. This comes with its fair share of issues such as people ageing differently due to genetics, the environment, or the facial photo quality. The overall goal of this project is to develop a new genetic programming (GP) method to learn a regression model for age brackets estimation. The method will take advantage of the ability of GP to produce interpretable models and provide further insight into factors and contributors that lead to age prediction. The results of this project include how accurate the GP is in estimating a person's age compared to existing solutions, and a deep analysis on why the GP chose certain regions over others and how those regions contributed to the final age estimation.**

I. Introduction

THIS project aims to understand facial ageing patterns through machine learning. This is done by using GP and interpreting the model produced through various graphs and tables to see which facial features contributed to what part of the estimated age calculation. Through understanding this model, various conclusions and comparisons with existing papers on facial ageing patterns will be discussed to see how this model fares against proven research in this field.

A. Motivation

Ageing has been a process which most living beings are subject to. Humans, which are subject to ageing try to make the most of their limited time doing the things they love. But as time passes changes occur to the body internally and externally. These changes affect what humans can and can't do. Internal changes are not obvious to other people as they cannot perceive them, but external changes become apparent. These apparent changes are what this project focuses on as they allow humans to roughly estimate someone's age based on their facial features and this project aims to achieve something similar.

The main goal of this project is to use machine learning to interpret age estimation. This means creating and understanding the factors and contributors that lead to facial ageing. This information could be used in hopes of solving sustainability problems such as security control for minors, human-computer interaction based on age, and law enforcement concerning identity. Each one of these problems can cause issues to do with age fraud which could be dangerous for the parties involved due to legal rights. These are attempted to be solved by building and understanding a machine learning model that labels a facial image into an appropriate age range that a person could fall in.

The motivation behind this project is to be able to interpret which features lead to age estimation as current deep-learning methods for age estimation are usually black boxes. It is difficult to understand how the estimation comes about with standard methods involving neural networks, but with the symbolic nature of GP, it is possible to obtain some interpretable age estimation models.

As age has been the standard for determining how old a person is, naturally lots of research has been done on what causes ageing and its effects on the human body [1]. Even without reading these research papers children from a young age learn to understand how old a person is just by looking at their face. According to computer vision anything that humans perceive and understand a computer should be able to be trained to understand what humans understand or even go beyond human comprehension abilities. This project aims to apply this theory in to age estimation and interpret the results to what we know about facial ageing to see any differences or similarities to human understanding of age.

B. Goal and Objectives

The goal of this project is to develop a GP approach for age estimation which makes use of the symbolic representation of GP to interpret the final model output. GP is a machine learning algorithm that uses a population of individuals which evolve after every generation to reach a better accuracy.

The objective of this project to help realise the goal, is to run the model multiple times to find multiple different model solutions. As there will be lots of local optima where the best solution has been found within a local region, but we want to find the best model if possible. To help with this GP methods for image classification will be carried out. This is a process which starts with preprocessing steps to reduce noise and differences such as size and colour. These images are then used as the terminal set (variables) for the GP algorithm to use, where after the evolutionary process an equation is returned which shows the steps applied to an image to receive the final age estimation of a face. The equation created will be a new function set that works well for facial age estimation which uses region of interest (ROI), edge detection methods, mean, standard deviation and basic operators such as addition and subtraction to reach a final age estimation.

Key findings to take away from the solution include the accuracy achieved in terms of Mean Absolute Error (MAE), ROI, the final model and the steps it uses to reach the final age estimation, a comparison to existing Neural Network (NN) solutions and a baseline which only uses basic operators, and lastly a comparison to human logic and knowledge on age.

This project was supervised by Qi Chen (primary), and Bing Xue

By understanding and comparing the GP model created in this project to these criteria we can deduce the effectiveness of the model produced and provide further insight on how the ROI and edge detection methods could relate to the steps humans apply to estimate someones age.

II. Related Work

Plenty of researchers have been interested in seeing whether age can be determined through machine learning. This section will outline existing solutions and how previous research went about estimating age.

Many different takes on age estimation using machine learning techniques go through similar steps to achieve a decent model. They start with acquiring a dataset, in this case, facial images. Then a lot of preprocessing gets done, this varies slightly from each take, but all of them somehow have to find a person in the image and normalize each of the images so that they all roughly look the same. Then extraction of features or ROI gets done which will be used in training the model, these features will be treated as the inputs to estimate the output age.

A. Literature Review

A couple of existing papers on the overview of age estimation are [2] [3] [4]. These papers helped with understanding how age estimation can be achieved through facial images, and what to look out for when creating a model. This research helped by allowing us to focus on GP rather than the steps that lead up to it. This includes things like:

1) Age Estimation Methods: Existing methods on creating a model for facial image recognition/classification use hand-crafted pre-processing methods such as Histogram of Oriented Gradients (HOG) or Local Binary Pattern (LBP) which are used to extract edges and shapes from facial images [4]. Other than that deep learning methods such as convolutional neural networks (CNN) or Multilayer preceptors (MLP) could be used to extract useful features from the images. The advantages of using hand-crafted methods over deep learning methods include less computationally expensive and more efficient, but of course less accurate. However, using CNN or MLP we cannot understand what these methods do on the inside making it harder to interpret the age estimation model. Therefore it would be best to use hand-crafted methods as they are interpretable.

Other than using hand-crafted methods for facial feature extraction, normalization is required to acquire quality features. Otherwise, some external noise could disrupt the GP algorithm during training, for example, colour, image size, resolution, image contents etc. The methods used for dealing with this noise and normalising the data as much as possible include:

- face detection and exclusion of background: Images are usually taken with the scenery and the person could be anywhere in the image. So detecting the face in the image and then removing everything apart from the face is required. This reduces errors due to unnecessary information.
- resizing of each image: Every photo taken of a person has a different depth to it. This means the face could be from extremely close-up, to extremely far away. The face detection then detects this face and naturally, the facial images would have different sizes. So resizing each image after face detection reduces errors due to differences in the information given by each face.
- grey-scale all the images: As the images are going to be sent through an edge detection algorithm, the images need to provide the same conditions to avoid inconsistency in edge detection.

2) Common Challenges Faced: The most common challenges faced when using facial images of people for machine learning include head-pose and alignment, image Resolution and lack of Data. For age estimation, challenges such as lifestyle and health conditions, genetics, and facial modifications [4] could disrupt the learning process due to uncontrollable variance between each person. Some of these can be fixed to some extent like head pose and alignment, image resolution, and lack of data, but all of the other areas are nearly impossible to account for unless prior information is given. Of course, with prior information like facial modifications, this would require knowledge of the person which leaks into the facial recognition field which we do not want. As with recognition, we could find the person and directly figure out their age through the internet but that falls outside the scope of his project.

3) Benchmark Datasets: Free online available datasets: For a good age estimation model lots of data of varying age groups and people is required. As lack of data can be an issue, free online datasets have been created. These datasets have been created for the purpose of machine learning and creating models. Each image contains at least one person's face and their age, found either in the metadata or as part of the file name. Some of the readily available datasets include [4] IMDB-WIKI, Human and Object Interaction Processing (HOIP), The Asian Face Age Dataset (AFAD), Cross-age Celebrity Dataset (CACD), WebFace, MORPH, Specs on Face (SoF), MegaAge, Adience, UTKFace, Facial Recognition Technology (FERET), and FGNET.

Each of these datasets has its own uses. AFAD for example is a dataset on Asian faces, this however could cause bias in the model due to racial differences. Therefore, concerning this project, some of the above datasets are not fit for what this project is trying to accomplish. As previously stated diversity is key, of the above models only IMDB-WIKI, WebFace, MORPH, MegaAge, UTKFace fit this description. To further narrow these down the amount and quality of these images will be taken into account, and we would then come to IMDB-WIKI, UTKFace, and MORPH as the ideal datasets for this project.

4) Evaluation Metrics: There are a couple of evaluation metrics that can be used. For classification and regression there are Mean Squared Error (MSE) and MAE which both achieve the same task, big error value means bad, low error value means good. This will determine how the GP algorithm will update its next generations. There are also some other age estimation evaluation metrics listed such as cumulative score and one-off, but these concern classification problems rather than regression so a regression evaluation metric is required for this project.

With these models, it is hard to see whether or not there is a relationship between the inputs and the outputs. For this project, we definitely know there is a relation between the facial images of people and their age, but what we do not know is how well a machine learning algorithm does to estimate age. This is why finding existing solutions will help to see how well this model does against previous attempts. This gives useful information on how my model fared against other models, and whether or not there are any improvements that can be made to my model. Whether it be changing the preprocessing or the GP algorithm.

B. Neural Networks

CNN's are the most used and favoured image classifiers out of all the NN's out there [5]. This is because CNN's provide benefits such as reducing the number of parameters within the network, and obtaining local information rather than global which allows for using whole images as the inputs. These CNN models consist of input, hidden and output layers. The input layer would be the pixel values of each image, the hidden layers would apply some sort of filter on the images to find relevant features to determine the goal otherwise known as the output from the output layer. Where the output layer would be the number of different classes for classification, in relation to this project it would be either age brackets or exact ages.

Advantages:

- Not much knowledge is required for image classification: All that is required of the programmer is filling up the model with layers such as convolutional layers, pooling layers and a final output layer for categorization. This means that any person with previous NN knowledge would be able to create a model that achieves a high accuracy.
- Automatic Feature Extraction: This is because CNN's use raw pixel data from each image as the input and automatically discover features and characteristics of the images. Rather than needing to know much about image preprocessing and image extraction like the other two evolutionary algorithms mentioned.

Disadvantages:

- Lots of data is required: This is because, with a small dataset, the CNN would overfit causing unseen data to be classified incorrectly most of the time. There are some techniques which reduce overfitting such as dropout, batch normalisation and data augmentation. These techniques help by increasing noise, normalising data which in turn reduces computation time, and increasing the amount of data. These techniques however make the CNN more complex and generally increase the amount of time it takes to complete training.
- Black box: The main reason why this project will not use a CNN for interpretable age estimation is due to it being a black box. It is very hard to see what the CNN actually does to reach a final age estimation, this is because the filters applied within the hidden layer and the number of nodes and what they represent each having their own weights and biases consisting of numbers becomes too complex for human comprehension.
- Computationally expensive to train and run: As mentioned before lots of data is required to achieve good accuracy during training and on unseen data. Not only that but a larger number of layers or features and the complexity of the CNN also factor into how long it takes for the model to learn age estimation from facial images.

C. Genetic Programming

GP is an evolutionary algorithm created by John Koza in 1988 and published in 1992 [6]. GP is still widely used today to solve many machine learning problems [7], [8].

An existing implementation of GP which also uses facial images that could help with this project is [9]. This book contains lots of helpful GP techniques and code to help with getting started and implementing a GP algorithm. A couple of takeaways from the code include:

- The main algorithm which includes the function set used for image classification, various fitness functions used for classification, and how DEAP was effectively used to help with implementing a GP algorithm.
- recreating DEAP's implementation of the GP evolutionary process to accommodate a validation set and making a suitable offspring creation process for the problem based on observations made of previous runs in achieving an optimal model.
- feature function's used for the function set. This consists of multiple different methods in extracting edges from images which the model can use to classify facial images or with respect to this project estimate a person's age.
- recreating DEAP's implementation of generating programs/trees for each individual during initialization, reproduction, crossover and mutation. Due to image classification using lots of different data types such as images which are arrays, floats, ints, and each of the function sets requiring a specific input to then generate a corresponding output a strongly typed function set will be used with custom datatypes to ensure correct behaviour during the training process [7].

All of the aforementioned techniques will be used by this project but slightly changed to a regression-type implementation.

D. Useful tools for development

Other useful tools found from existing research Which could be used to aid in development:

1) Datasets: As previously mentioned, online datasets will be used for training and testing purposes. Specifically IMDB-WIKI, UTKFace, and/or MORPH as each of these datasets has a large sample size of facial images varying across a large age range. But also lots of existing solutions were created using these datasets, especially MORPH due to its constrained conditions and sample size. This will allow me to directly compare my solution to existing solutions without worrying about model differences due to differences in data.

- IMDB-WIKI which is the largest of the three uses 523,051 samples of 20,284 individuals from different time periods with an age range of 1-90 years old.
- UTKFace is a dataset that uses over 20k facial images that cover an age range of 0-116 with a large variety of positions, facial expressions, illumination, resolution, and so on. But after applying preprocessing steps these can be controlled to some extent.
- MORPH is a dataset with 55,134 samples of 13,618 individuals between 16-77 years old. Unlike the other two datasets, MORPH is constrained meaning the images are all taken identically in a controlled environment.

2) Programming Language: Python as a language: There are many programming languages which could fulfil this project's needs. These languages include MATLAB, Python, Java, C++, and Darwin. Due to previous experience and knowledge of existing libraries which can aid in the development process, Python was chosen. With its vast range of capabilities including parallel processing and multitudes of other libraries that can assist with GP and visualization. But most importantly due to the GP Python library DEAP, Python is arguably the most popular language for data scientists [10].

As images are not always preprocessed to achieve the goal of this project. The most important step to normalization of images is to get rid of any background noise in an image. This will focus the image on only the face area rather than having trees or anything else mess up the estimation. Python libraries which provide face detection methods include cv2 and face_recognition.

- cv2: This uses the OpenCV library which is a Python library that handles images and videos and has lots of built-in functions to do all sorts of stuff with these images and videos. One of which is face detection. The face detection method used by OpenCV uses the Haar Cascade approach [11]. This approach makes use of multiple classifiers that detect various features in an image. In this case, it would be facial features such as eyes, jaw, nose, mouth, and ears. Once these features have been found the Haar Cascade algorithm decides whether or not it can detect a whole face from it. If the algorithm decides that it can see the whole face then a face has been detected, but if a face is only partially visible then it will not be detected.
- face_recognition: This is a pre-trained model which achieves a 99.38% accuracy on face recognition with labelled data. This project however is only concerned with the face detection part of it. Apart from not knowing how face detection works due to limited documentation. This library detects faces at the same rate as cv2, indicating the possibility that it uses the same Haar Cascade algorithm.

The only noticeable difference between the two face detection methods is that cv2 has a lot more options when finding faces. This does not mean it is better as it started finding a lot more false positives like having a neck as a face. This comes from the available parameters when deciding what is a face and what is not. cv2 provides parameters such as scaleFactor, minNeighbors, and minSize. scaleFactor is not important as it only changes image size, and minSize is not important either as it checks for faces of at least a particular size. minNeighbors allows for multiple or fewer faces to be found as haar cascade checks for faces through rectangles if there are enough rectangles close enough it will identify that as a face. So a higher minNeighbors value means it will become stricter and low means more false positives.

Python as a language also includes a GP library called DEAP. This library will be used to do all the GP parts of the project instead of writing the same algorithms used in the library from scratch. As GP can take a very long time the larger the dataset and the more depth in the model, training the model would take an extremely long time when only using the CPU. So concurrency will help in splitting the number of tasks evenly among multiple computers.

As genetic programs will take a very long time to run, concurrency is required to speed up the training process. This will be achieved by making use of ECS Grid jobs from jobs at Engineering Computer Science (ECS) in the Victoria University of Wellington (VUW). This will allow multiple runs of the genetic program to be completed simultaneously on different machines. This will make the whole training and development process a lot faster than if only one machine were used.

Visualization of the generated GP tree is important to help interpret the model and extract valuable information such as features used and how they contribute to age estimation. For this, pygraphviz will be used as a tree visualization library within Python to show what regions the age estimation model used. and whether or not it could identify the regions which we believe to show ageing.

For comparison purposes, Matplotlib will be used as a graphing and table creation library to evaluate and compare the final models produced. As not only interpreting the model but also evaluating the model is important to see how well GP does to estimate age. This indicates the confidence level that the model can actually estimate age using the facial features it selected.

III. Design

There are many different approaches to figuring out a person's age through their facial image. We will highlight a couple of the more prominent ones and then talk about why GP was used over the others and the technical aspects of GP

A. Evolutionary Algorithms

There are lots of different evolutionary algorithms, each with their own strengths and weaknesses in developing a solution to a problem which would take humans an extended period of time to reach the optimal solution. Not all evolutionary algorithms are ideal for image classification and regression problems so here are some worth considering for age estimation.

- Genetic Algorithm's can be used together with CNN's to help with simplifying and automating the CNN learning

process. For example, knowing exactly the number of layers, nodes/filters and any other domain knowledge to reach an optimal solution would be difficult to do manually. So a genetic algorithm can be used to automate the process and run a population of CNN's with different parameters and obtain an accuracy out of it to train the next generation which simplifies the process even more.
Advantages:
  - Automation of generating multiple models to systematically reach an optimal model solution.
  - Accuracy: even though it will take an extended period of time, at the end of the evolutionary process a highly accurate model can be expected.

  Disadvantages:
  - Time spent training to find an optimal solution: Running one CNN through a large dataset of images once takes an extremely long time due to the nature of NN's which have lots of layers and filters being applied.
  - model representation: One of the major goals of this project is creating a machine learning algorithm which is interpretable. Genetic algorithms however use a binary representation which is difficult to interpret especially when used in tandem with a CNN solution.
  - constructing the algorithm: as the representation is binary each 0 and 1 belongs to a single variable within the CNN. This could get complex rather quickly the more complex a model is required to achieve high accuracy.
- GP is an evolutionary algorithm that evolves computer programs rather than solutions. One major advantage of GP is that the representation of these computer programs is that they are tree-like, graph-like, linear, etc. making them all easily interpretable. These tree-like models consist of function nodes and terminal nodes. *Talk about regression over classification*
  Advantages:
  - Interpretability: As the final model created through GP takes the form of a tree-like structure, the model becomes interpretable through human eyes. Unlike NN's where you cannot directly look into what each layer does.
  - parallelism: As the fitness function uses each image to calculate some accuracy or error value it would take a very long time to do everything sequentially. This is why parallelism helps with calculating the fitness of each individual by calculating multiple in parallel.
  - probabilistic: Rather than manually figuring out the ROI within an image and calculating some output using some operators and metrics obtained from these ROI, GP automatically figure out these things and only becomes better due to the fitness function implementation and selection which takes the best individuals.

  Disadvantages:
  - The larger the model the longer time it takes: For example, if the tree consisted of one function node and two terminal nodes, evaluating this tree against every image would be quicker than a tree with a larger amount of both function and terminal nodes.
  - Depending on the genetic operators, fitness function, function set, and terminal set it can generate desired outputs: The most important aspect of GP is understanding the data and how to reach a desired output. But even with this knowledge, defining each of these aspects still requires trial and error using an iterative approach, by understanding what went wrong and what could be changed to make it better.
  - Using a large dataset and an evaluation like cross-evaluation on top of a classification model such as LBP can cause extremely long training times: As previously mentioned in the parallelisation section, a genetic program takes a very long time to train. Even with parallelisation helping out, depending on the size of the dataset and fitness function used it can even take multiple days to do one generation with parallelisation.

B. Facial Recognition

Facial recognition is a completely different take on facial image classification and regression problems. This is because it does not estimate age but rather you train some model to identify a facial image and grab their data from some database to then return their exact age. So facial recognition will not be used within this project as it is not suitable for age estimation

C. Classification and Regression

The difference between classification and regression is that the output of a classification problem is discrete whereas the output of a regression problem is continuous. For this project, both classification and regression can be applied. That is because even though age is related to time which is a continuous variable. Age can still be categorized into ranges within the lower limit of 0 and the upper limit of the oldest person ever lived.

The advantages of using classification over regression include the use of classification algorithms such as decision trees, random-forest classifiers and k-nearest neighbour among many others. These algorithms can take in lots of features and figure out patterns within them to then classify data. Whereas with regression, exact value output calculations of any input variable even outliers for example an age above the maximum could be found.

This is the main reason for using a regression-type fitness function over a classification one as people these days live a lot longer than in the past and this could continue into the future. Another reason for regression over classification is due to the classifiers taking an extended period of time to converge and classify. So we chose to create a regression-type fitness function using a MAE value during training to get lots of results.

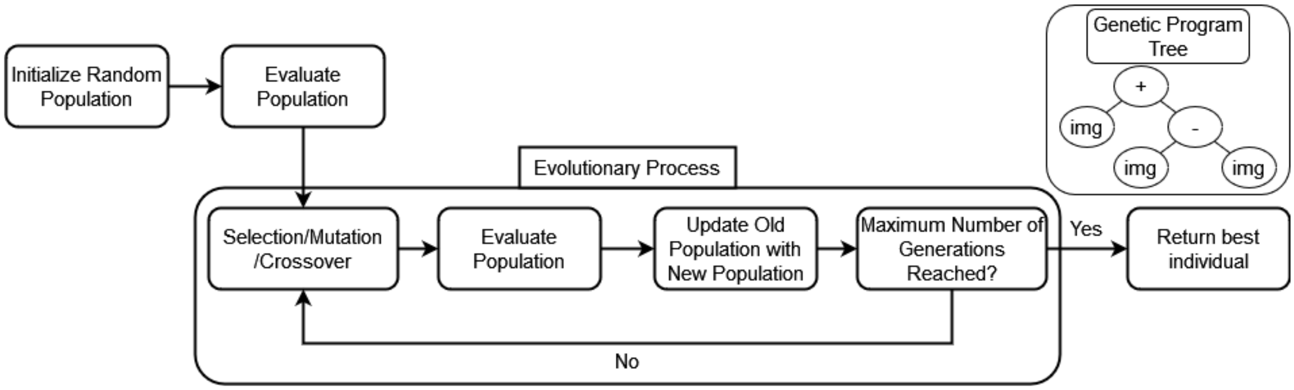

Initialize Random Population
Evaluate Population
Evolutionary Process
Selection/Mutation /Crossover
Evaluate Population
Update Old Population with New Population
Maximum Number of Generations Reached?
Yes
No
Return best individual
Genetic Program Tree
+
img
-
img
img

Fig. 1. Genetic Program Evolutionary Process

D. Technical Solution

The evolutionary algorithm this project uses is GP. The GP we have developed uses regression to estimate the exact age a facial image of a person falls in. Fig. 1 is an example of a genetic program which initializes a random population of individuals, evaluates them, and then starts the evolutionary process. This process continues for x generation times, generating offspring from the current population by using genetic operations such as mutation, crossover and elitism, then evaluating and updating the old population with the new population. Once x generations have passed the best individual is returned in the form of a tree consisting of function and terminal nodes. This tree is used in estimating an age value for each facial image. A little more information on the process is explained below:

Steps to evolve a GP:

- Generating an Initial Random Population: A population consists of a number of individuals where an individual is a model solution to a problem. These models consist of function nodes and terminal nodes which generate some sort of output that a regression/classification fitness function uses for evaluation.
- Evaluation: Either a regression or classification method is used to output some value to differentiate each individual from another and know how well each of them performs by trying to find either a maximum or minimum value.
- Selection Method: During the next generation, the evaluated individuals will then be used during offspring creation by applying a selection method. This randomly grabs the best-performing individuals by applying genetic operators such as crossover and mutation to generate a new offspring population.
- Updating old population with new population: This either only grabs the new offspring which requires some elitism to keep the best individual from the previous generation, or the new population will consist of both the best offspring and current population individuals narrowed down to population size.
- Hall of Fame: Once the genetic program has finished training the best model is then returned as a model solution to the problem. This can be used to interpret ROI/important features found, and evaluate the accuracy of this model to other solutions and what is considered a reasonable margin of error for this kind of problem.

IV. Implementation

As stated within the design, we have used a GP approach for interpretable age estimation. As outlined in Fig. 1 there are multiple components in a genetic program and we will break down each section and how we implemented them.

A. Preprocessing

For accurate age estimation, the facial images require some processing to reduce differences between them. This includes:

- Depending on the dataset used, face detection: For my current program, we do not use face detection as the free online dataset UTKFace already has some preprocessing done on the images which includes finding and aligning every face. However, this is a very important step for facial image classification and regression problems.
- Resizing: To ensure every image uses the same dimensions we resized all the images to 200x200. This helps with the learning process as the location of each facial image will be in roughly the same spot on the image. This means that if a particular region is used as a feature the chances that another image will have the same feature in the same position will be greatly increased.
- Gray-scale: The main reason for doing this is to reduce the dimensionality of the pixels from RGB to a single value. This allows the edge detection algorithms to find edges between pixels to help with finding facial features that could contribute to a person's age.

B. Parameter Settings

- training dataset size: Due to time constraints, a smaller dataset was used to produce lots of results to talk about during the evaluation. For this reason, we used 4700 facial images randomly selected from the UTKFace dataset as the training images.
- Validation dataset size: The validation set is used during the training process to ensure no overfitting occurs. This is done by evaluating a validation set and seeing whether or not the training accuracy increases but the validation

accuracy decreases. For this, we used 200 random images from the same dataset to make sure the training process doesn't overfit on the training data which leads to falsely classifying unseen data such as the test data.

- Test dataset size: To see how well my model performs against unseen data that wasn't used during the training process, a test dataset is required. This test dataset serves as an accurate measurement of how well my model would perform against any other facial images that have gone through the preprocessing steps. For testing we have used 100 images also from UTKFace, a larger number of testing images would serve as a better test but we have chosen 100 purely because after preprocessing images they require a lot more space for some reason, so we took a small portion from the training data instead.
- Population size: We have used a population size of 200 individuals. A general rule of thumb for GP is to use a larger population size to generation ratio. This is because it allows for more crossover and mutation to occur on a larger variety of individuals. So a more exploration rather than exploitative approach was used to try and find an optimal solution in a large search space.
- Generations: For the reason mentioned before, we used a generation size of 100. This allows for plenty of learning to be done during training while also not taking too long that we would run out of time. During evaluation we will show the learning curve to further explain my choice of 100 generations.
- Crossover Rate: we used a crossover rate of 0.7. Crossover rate decides the probability of selecting two parent individuals to undergo crossover to generate two new offspring individuals. This is done by mixing some aspects of the parent trees to generate new trees that consist of nodes taken from both parents.
- Mutation Rate: we used a mutation rate of 0.29. The reason for having a lower mutation rate compared to crossover is because there is a higher chance that crossover will generate a better individual for the next generation than mutation. This is because during mutation anything can happen and it is not really controlled, compared to crossover which takes two individuals with nodes that somehow did well to be selected for the current population and reuse these nodes but interchange them with both individuals.
- reproduction rate: reproduction rate is the probability of neither crossover nor mutation happening so 0.01. The reason for having such a low number is that reproduction doesn't apply any change to an individual and just adds it to the offspring. This is used mainly for variety within the population, unlike crossover and mutation the fitness of this individual will not be better than the best individual in the new population.
- max depth: we have used 15 as the maximum depth of the GP tree. Ideally, a smaller depth would be better for interpretation but a larger depth allows the GP to experiment on a larger scale by using lots of function nodes and terminal nodes within one tree.

C. Technical details on each component of the proposed GP method

The overall framework outlined in Fig. 1 shows the basics of GP, this section will outline the technical details of how we implemented each section of the evolutionary process.

Starting with initializing a random population which creates 200 randomly generated individuals that contain anywhere between 2-8 nodes. As shown in Fig. 2 (a) which shows the structure of an individual, the nodes within a tree consist of terminal nodes such as the input images or some random integer values to do with region creation, or function nodes such as basic operators, mean and std calculations, region detection, or feature extractors such as edge detection algorithms or histograms.

1) Function set and terminal set used:

- Feature Concatenation: Joins multiple features together by applying a basic operator such as addition or subtraction, where the final output is the estimated age of the person in the facial image.
- mean/std calculations: The standard way of reducing an array (image/region) down to a single value is by calculating the mean or standard deviation of each number within the array.
- feature extractors [12]: Each of the following applies some form of filter to further break down an image into something a computer can understand.
  - Histogram Equalization: Adjusts the contrast of an image to enhance the image contents such as edges. This helps with edge detection algorithms and helps with mean/std calculations by either greatly increasing their values due to lots of contrast or vice versa.
  - Gaussian: used for smoothing, and reducing noise. Due to images having different resolutions and applying preprocessing steps such as resizing there is bound to be some noise within the image.
  - Laplace: is an edge detection algorithm that measures the second derivative which is the rate at which the first derivative changes. In other words, it calculates the change between adjacent pixels and determines whether or not that is an edge or not.
  - Gaussian Laplace: applies a Gaussian filter on an image and then detects edges using the Laplacian filter. The reason for including separate filters as well as a combination of each, is that we as humans do not know how a computer determines age based on a facial image. This is why we train a GP to apply whichever filters it deems necessary rather than using trial and error.
  - Gaussian Gradient Magnitude: applies Gaussian principles and computes the magnitude of the gradients within the image. As these images are grey-scale it computes the gradient from black to white.
  - Sobel x/y: Sobel looks for strong changes between pixels within a 3x3 area. Sobel x looks for changes between pixels on the horizontal level whereas Sobel y looks for vertical changes. These can be used in

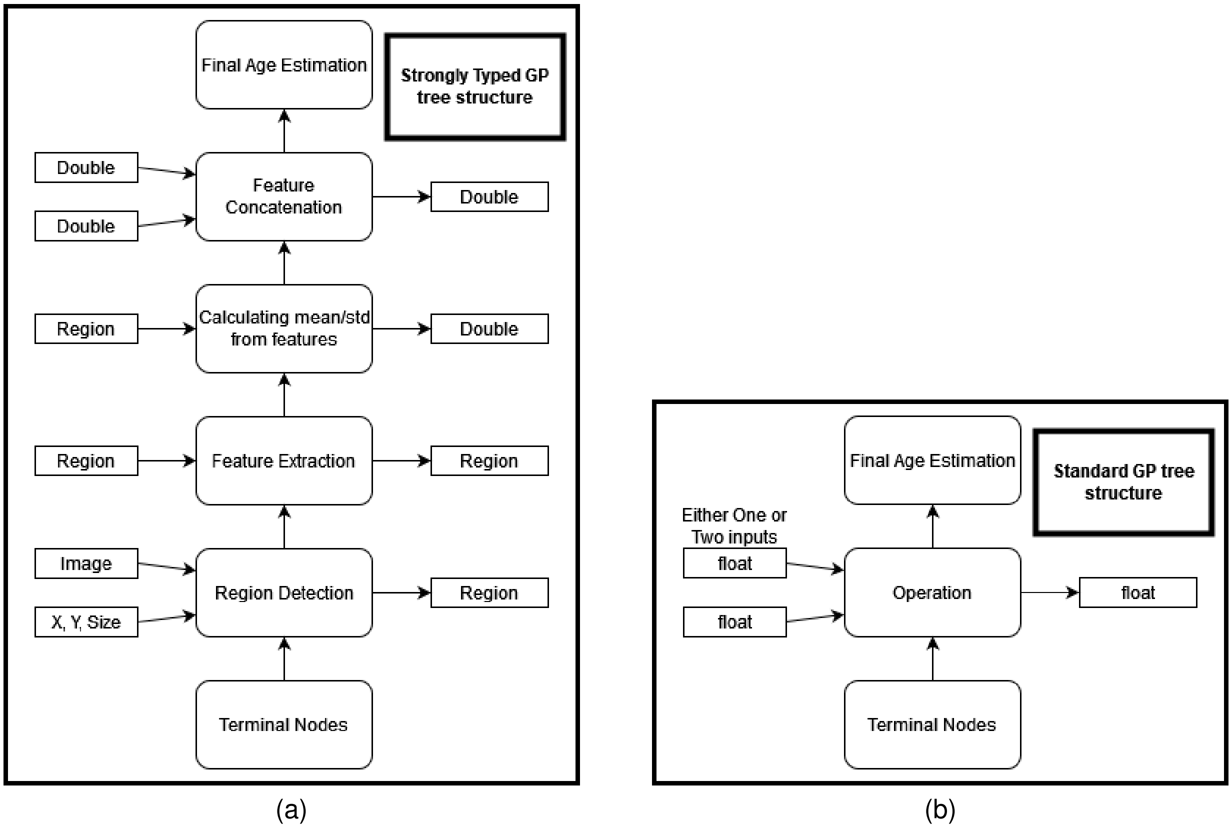


Fig. 2. The program structure of a strongly typed GP tree used in this work (a), and a standard GP tree (b)

  tandem which will act similarly to a laplacian filter so only x and y have been used for this project.
  - LBP: mainly used for image classification as it generates a feature vector by comparing a pixel with its neighbours and using 0s and 1s to determine whether that pixel value is a higher or lower value than its neighbours. So unlike the previous filters, this does not detect edges but rather finds textures within the image. Textures could be valuable for age estimation so we let the GP decide whether it wants to use LBP or not.
  - Histogram of Oriented Gradients: Same as LBP, this is mainly used for classification as it detects objects within an image. This is because it is not as powerful as the other filters when applied on a smaller region as there aren't any objects to be identified. But samefrom as before, we let the GP decide if it wants to use this feature extractor.
- region detection: This uses terminal nodes that consist of an image, an X value a y value, and a size. These terminal nodes take a region from an image to be used as a feature. The x and y values decide the top left position of the region and the size value decides how big the region should be. Either one or two size values can be used to create a square or rectangular region.

2) Fitness function: The evaluation of an individual otherwise known as the fitness function, calculates some value which should be either minimized or maximized depending on the problem. As my GP uses regression we calculate a MAE error value which we want to minimize. The MAE takes the absolute value from the predicted age and subtracts that from the actual age of every facial image and gives the error as the mean of each absolute value. This error value should be minimized as we want the model to estimate the age of a person's facial image. This MAE value gets used during the selection process by selecting the individuals with the lowest values to be used for generating a new offspring population.

3) Genetic operators: Genetic operators are operations a GP uses to generate new individuals by crossover and mutation. There are a couple of different crossover and mutation methods when generating offspring so we will outline the method we used and how mutation can generate new trees that adhere to the program structure mentioned in Fig. 2 (a). Other than crossover and mutation, reproduction and elitism can be used to reuse individuals from the current population and add them into the offspring.

The offspring generated uses individuals from the current population to create a new population of individuals. As mentioned in the GP parameters section, 70% undergo crossover, 29% undergo mutation, and 1% undergo reproduction. To ensure that the GP error value keeps decreasing the best individual from the current population gets added to the offspring through elitism.

Here are the technical details of each of these genetic operators and elitism:

- Crossover: The crossover method used is called ”cxOnePoint”. This method takes two parent individuals and randomly selects a point within each individual and exchanges everything that is part of that point within the tree and returns the two newly generated individuals.

- Mutation: The mutation method used is called ”mutUniform”. This method takes one parent individual and randomly selects a point within the tree and mutates the whole subtree to generate a new individual.
- Reproduction: Takes one parent individual and keeps it for the next generation of individuals.
- Elitism: Selects the best individual from the current generation and keeps it for the next generation to ensure that the model is learning curve is only getting better.

D. *Other models used for comparisons*

The goal of this project is to develop an accurate interpretable age estimation model. For this reason, multiple different models were used to compare how well the new GP does against a baseline GP model and against an existing solution that uses a NN architecture.

1) Baseline: The baseline developed for comparison uses a standard symbolic regression method to estimate age. This consists of a function set made up of standard GP operators such as addition, multiplication, sine, etc. where the tree structure looks like Fig. 2 (b). The main difference between Fig. 2 (a) and Fig. 2 (b) is that (a) has a strict structure it follows. unlike (a) which uses images as the terminal set, (b) uses floats which restrict the functions that can be applied to the terminal nodes to mathematical operations.

As the baseline is a more basic version of the new GP algorithm that uses a strongly typed function set, it is expected to perform worse due to applying no feature extraction methods such as edge detection methods. The baseline will serve as an indicator for the new GP model to see whether or not it produces better results.

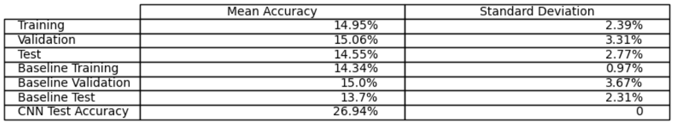
2) Existing CNN solution: As mentioned in the related work section, this is not the first age estimation model produced with machine learning techniques. There have been lots of attempts at facial image age classification which use CNN’s to estimate age. But due to the nature of CNN’s they are a black box algorithm so interpreting how the model estimates age is impossible. But these kinds of models can still be used for validation purposes to see how well the model outlined within the implementation section does against existing CNN models. We have used a model found within [13] for this purpose.

This CNN model is made up of five convolutional layers and pooling layers, then 2 fully connected (FC) layers with a final fully connected softmax layer which generates 101 output numbers that each represent one year from 0-100 years old. We do not know exactly what each of the convolutional and pooling layers do as the CNN representation is a black box, but we do know that lots of filters are being applied within each layer. After applying these filters, FC layers apply dropout to reduce the number of features to 101 for the last layer to use for age estimation.

V. Evaluation

The goal of this project was to accurately estimate age from facial images using a GP approach as the symbolic representation of GP allows us to interpret the final model produced. To achieve this, project objectives were carried out to create and compare the GP model produced to a baseline solution and an existing CNN solution. But also interpret the model produced against what is known about facial ageing to see whether or not the model accurately identified features such as wrinkles shown in prominent regions such as the forehead.

A. *Results and Analysis*



	Mean Accuracy	Standard Deviation
Training	14.95%	2.39%
Validation	15.06%	3.31%
Test	14.55%	2.77%
Baseline Training	14.34%	0.97%
Baseline Validation	15.0%	3.67%
Baseline Test	13.7%	2.31%
CNN Test Accuracy	26.94%	0

Fig. 3. Age estimation accuracy of each model to within 3 years of actual age

1) Discussion on the Confidence intervals between the baseline and the proposed solution:

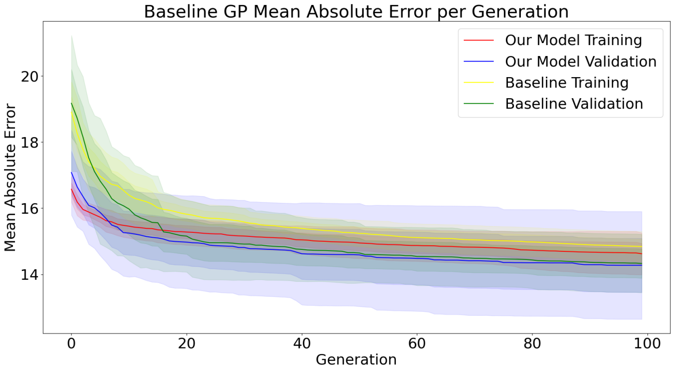


Fig. 4. Learning curve of my model with confidence level from 30 different runs of the GP

2) Our new model: In Fig. 4, the results found from running our GP 30 times according to the implementation in terms of MAE were sublime. However, as seen within the figure, there is a constant downward trend within the figure for both training and validation. This suggests that the model has not converged yet and as the validation set is also on a downward trend the model is not overfitting on the training set. However, not much change has happened from generations 20 - 100 therefore, if a larger generation number were used then the MAE would decrease but not much learning would have been achieved at the cost of time and resources.

The mean MAE of each run after 100 generations for training was just below 15.0 with a confidence interval of 0.3. Whereas the validation had an MAE of 14.5 with a confidence interval of 1.0. As there is a lot of variation within the validation set between the different runs, this suggests one of three things, either overfitting is happening within different runs which cannot be seen within the figure, there is a difference between the training and validation sets which could cause this, or the amount of data used within validation was too small. As overfitting was ruled out due to there being a downward trend and there being 200 images used for validation, that indicates that there is lots of variation between the training and validation sets used. This means that the training dataset had ages more concentrated within a certain age range compared to validation and the feature extractors largely disproportioned the predicted ages because of it.

This is also represented in the table shown in Fig. 3, where the training, validation and test results had accuracies within the 14.55 to 15.06 range. As they are roughly the same as our model's training and validation shown in Fig. 4, this shows that the mean MAE for both training and validation is close. This is roughly what we expect as an MAE value of 14.5 and accuracy where the predicted age is within 3 years of the actual age, which is 3/14.5=0.2 or a 20% accuracy of correctly predicted age groups. As training, validation and test accuracies were lower than 20%, this suggests that a larger proportion of predicted age groups were within a lower year difference than 3 with a larger proportion having more than 3 years of difference.

3) Baseline model: In Fig. 4, the trend of the baseline MAE results per generation is the same as our model but more exponential. As it is exponentially decreasing, this suggests that the baseline has almost converged and cannot find a better solution.

This is supported by the results shown in terms of MAE which show that the training mean MAE has an error value of 15.1 with a confidence interval of 0.3, whereas the validation set had a mean MAE of 14.6 and a confidence interval of 0.5. This shows that the baseline was more confident in predicting age on the validation set than our model which means that each run found a similar solution after 100 generations as the MAE of each run is almost the same.

The accuracy achieved between our model and baseline shows better accuracies across the board. The difference however is negligible as it is very small. However, the test set results have a difference of 0.85% which indicates that our model performs better on unseen data than the baseline.

4) Comparisons between the two previously discussed models and the CNN solution:

As the CNN model used as a comparison for this project has been pre-trained, there are no training nor validation results available to show within Fig. 3. The place where we downloaded it from [13] does not include their results during training either. However, it is still relevant to include this model for analysis, as the difference between our new GP model and the CNN model tells us whether improvements can be made.

CNN models generally perform better at image classification than GP models due to being more complex. This is shown in Fig. 3 where the existing solution has an accuracy of 25.86% which is over 10% more than the baseline and our model. This means that there is still lots of room for improvement in generating an accurate GP model for age estimation.

5) Interpretation of the model compared to existing knowledge of facial ageing:

The second part of this project's goal is to interpret and compare the trees produced to understand what is happening within the tree and how that relates to our knowledge of facial ageing. For this purpose, two different trees from the 30 runs were selected to show which features the model thought were ROI by showing the tree structure with facial images representing what happened within each step towards facial age estimation.

We will specifically look at the regions the GP model has found and the main giveaway of facial ageing which is the presence of wrinkles [1]. As shown in [14], there are lots of areas on the face which start to form wrinkles from an earlier age. The most common areas are around the forehead, eyes and mouth area. So we would expect that the model used some or all of these ROI to estimate age. We would also expect that the models produced used regions which affect the elderly more than younger people such as the cheek, nose and forehead areas [1].

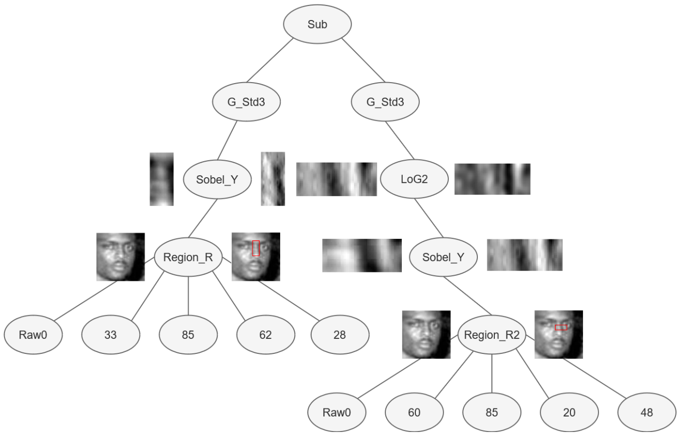


Fig. 5. First Model Solution with images showing what's happening

As shown in Fig. 5 there are two ROI, in relation to this image they are the nose and nose-to-eye area. This suggests that there is some relation between those regions and age. The feature extraction methods used were Sobel_Y which looks for vertical change between pixels and LoG2 which applies a Gaussian Laplacian filter which gets rid of noise and then tries to find changes between pixels. For both ROI the model applies a standard deviation calculation and subtracts the first region from the second.

This set of operations suggests that there is a lot of variance between the pixel values of the first region compared to the second. As shown within the last set of feature extractors, the left-hand side Sobel_Y operation produces an image with half-white and half-black pixels so a large variance compared to the right-hand side which consists of largely darker colours. In terms of this image and facial ageing, this could mean that the more wrinkles on the nose and eye area the more dark areas that are produced when in contact with light. This causes a larger variance on the left image in comparison to the right which when subtracted causes a larger number to be produced.

Comparing this to the knowledge outlined in [1] [14], our model did successfully find the eye area where bunny lines and tear troughs form which affects young and old, the model also chose the nose area which affects older people. This suggests that the model is calculating some sort of difference between them to estimate age.

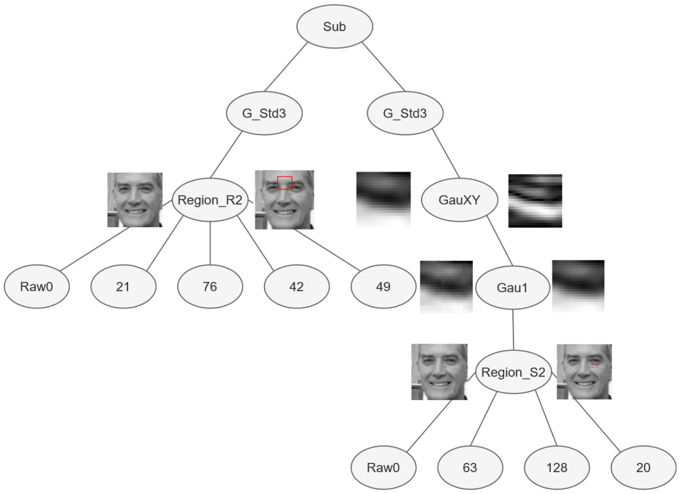


Fig. 6. Second Model Solution with images showing what's happening

The ROI found within Fig. 6 are very close to Fig. 5 but slightly shifted towards the forehead or between the eyebrows, and under the eye regions. This indicates that the identified ROI between the models are very similar and there is a definite correlation between them and facial ageing. The tree structure is also very similar compared to Fig. 5 with slightly different feature extractors being applied, with the left region not even using any but identifying frown lines [14]. However, as can be seen with the right region after applying Gaussian and Gaussian Gradient Magnitude there are clear under-eye wrinkles as well as tear troughs shown within the example image.

Compared to [1] and [14], Fig. 5 and Fig. 6 follow the same principles where the left region identifies ageing patterns found within young and old, and the right region identifies ageing patterns found at an earlier stage during the ageing process.

B. Further Discussions

After comparing and analyzing the models produced against existing knowledge of facial ageing patterns, we would have expected to see more regions being used for age estimation. For example, from both of the models studied, none of them used regions around the mouth. One possible explanation for this could be due to males having facial hair which could disrupt feature extraction.

One possible explanation for this could be that the mean and standard deviation calculations are very large. As RGB colours are from 0-255 when converted to grey-scale they are a single vector where 0 becomes black and 255 becomes white. Now if the pixels were evenly split you would expect mean values within the 127.5 area to be produced after feature extraction methods are applied. Due to the output of the model being the predicted age, these very high values need to somehow be reduced to within a human's age range which is why standard deviation calculations and subtraction were used over mean and addition as those help with reaching a smaller number. This also explains why the MAE of all the different runs could not achieve an error value below 14 due to the high variance calculations made by each model. Even small differences found within the images could greatly affect the estimated age.

A solution to fix this problem could be to Normalise/classify the facial images. This is because the mean and standard deviation calculations on images are quite high so normalization could be used to decrease the range between the values. This however would be better used with classification than regression as classification models can use the normalized data a lot better than a regression-type model. This is because min-max normalization would decrease values to within the 0-1 range which is not ideal for the output of our model for estimating age, classifiers however can use the normalized data for more precise estimation.

VI. Conclusions and Future Work

A. Conclusion

The aim of this project was to create an interpretable age estimation machine learning algorithm. Unlike existing solutions which use NN's we used GP as its symbolic representation allows deep analysis on features found by the model and how they contribute towards age estimation. We have met the project goals set at the beginning of this project as we have developed a new GP algorithm which has successfully found some facial ageing features but not all of them due to large number generation from mean and standard deviation calculations which were unsuitable for age. This also resulted in highly varied results as shown in the confidence graphs with their high MAE values. We believe that a classification approach would produce better results and that future research on interpretable age estimation using GP should prioritise the classification methods outlined within this report to help in achieving more accurate age predictions.

B. Future Work

As this project focussed more on regression rather than classification, classification was only briefly considered during our development process. The main reason for not delving too much deeper is that using a Python classification library such as Sci-kit Learn will take a lot longer to fully train a model than a simple regression fitness function. This is because classifiers take in inputs of lots of different features and they take a very long time to sort out that information and create an optimal age estimation using them. However, as mentioned within the evaluation, the regression-type fitness function struggled due to mean and standard deviation calculations being a lot higher than a reasonable age that humans could reach.

For future work, we propose that classification could be the key to reaching a higher accuracy for predicting the correct age of facial images. This work would include adding some feature concatenators which concatenate feature vectors together rather than using basic operators on mean and standard deviation calculation of image vectors. These feature vectors would be created using feature extractors or regions, which will then be used as the input for classifiers such as random forest or Naive Bayes. After training these classification-type models, evaluation and analysis comparing those models against ours and existing CNN solutions can be conducted.

VII. References

[1] healthandaesthetics. (2023) What happens to my face when i age? [Online]. Available: https://www.healthandaesthetics.co.uk/advice/what-happens-to-my-face-when-i-age-2/

[2] A. S. Al-Shannaq and L. A. Elrefaei, "Comprehensive analysis of the literature for age estimation from facial images," *IEEE Access*, vol. 7, pp. 93 229–93 249, 2019.

[3] H. Han, C. Otto, and A. K. Jain, "Age estimation from face images: Human vs. machine performance," *2013 International Conference on Biometrics (ICB)*, pp. 1—-8, 2013.

[4] K. ELKarazle, V. Raman, and P. Then, "Facial age estimation using machine learning techniques: An overview," *Big Data and Cognitive Computing*, vol. 6, no. 4, pp. 128–, 2022.

[5] P. P. Mohit Sewak, Md. Rezaul Karim, *Practical Convolutional Neural Networks*. Packt Publishing, 2018.

[6] J. R. Koza, *Genetic programming : on the programming of computers by means of natural selection*. Cambridge, Mass: MIT Press, 1992.

[7] A. U. Viktor Manahov, "The efficiency of bitcoin: A strongly typed genetic programming approach to smart electronic bitcoin markets," *International review of financial analysis*, vol. 73, pp. 101 629–, 2021.

[8] Z. V. Gisele Pappa, Mario Giacobini, *Genetic Programming: 26th European Conference, EuroGP 2023, Held As Part of EvoStar 2023, Brno, Czech Republic, April 12-14, 2023, Proceedings*. Cham: Springer, 2023, vol. 13986.

[9] Y. Bi, B. Xue, and M. Zhang, *Genetic Programming for Image Classification: An Automated Approach to Feature Learning*. Springer International Publishing, 2021.

[10] S. Y. Kim, Jinhan, "Software review: Deap (distributed evolutionary algorithm in python) library." *Genetic Programming and Evolvable Machines*, vol. 20, no. 1, pp. 139—-142, 2019.

[11] A. B. Shetty *et al.*, "Facial recognition using haar cascade and lbp classifiers," *Global Transitions Proceedings*, vol. 2, no. 2, pp. 330—-335, 2021.

[12] A. S. A. Mark S. Nixon, *Feature extraction & image processing for computer vision, 3rd ed*. Oxford: Academic Press, 2012.

[13] R. Rothe, R. Timofte, and L. V. Gool, "Deep expectation of real and apparent age from a single image without facial landmarks," *International Journal of Computer Vision*, vol. 126, no. 2-4, pp. 144–157, 2018.

[14] S. Matika. (2020) Common types of facial wrinkles (rhytides). [Online]. Available: https://www.westlakedermatology.com/blog/common-types-of-facial-wrinkles/