Deep Convolutional Neural Networks-Based Age And Gender Classification With Facial Images

G.Kaviya Priya[1], Dr. S. Miruna Joe Amali[2], Dr. M. Raghini[3], Dr. P. R. Vijayalakshmi[4]
[1, 2, 3]Dept of Computer Science Engineering
[4]Professor, Dept of Computer Science Engineering
[1, 2, 3, 4] K.L.N. College of Engineering

Abstract- *We build an age and gender classification system to classify age and gender with the help of deep learning framework. Automatic age and gender classification has become relevant to an increasing amount of applications, particularly since the rise of social platforms and social media. Nevertheless, performance of existing methods on real-world images is still significantly lacking, especially when com-pared to the tremendous leaps in performance recently re-ported for the related task of face recognition. Keypoint features and descriptor contains the visual description of the patch and is used to compare the similarity between image features. In this we show that by learning representations through the use of deep-convolutional neural networks (CNN-VGG16) and using resnet architecture algorithm for classification, a significant increase in performance can be obtained on these tasks. To this end, we propose a simple convolutional net architecture that can be used even when the amount of learning data is limited. We evaluate our method on the recent dataset for age and gender estimation and show it to dramatically outperform current state-of-the-art methods.*

Keywords- Facial Images,CNN-VGG16, resnet architecture, Deep Convolutional Neural Networks.

I. INTRODUCTION

Age and gender classification is extremely realistic. It is widely used in gathering demographic information, controlling entrance, searching video and image, improving recognition speed and accuracy, etc. Besides, age and gender classification shows important potential in the business field. The log records acquired from the system ultimately input to statistical analysis about consumers in order to accomplish real-time personalized recommendations.

The study of gender classification that began in the 1980s is a two-stage process. At the first stage, study researched on how to distinguish between sexes psychologically taking ANN methods mainly. Golomb trained a 2-layer fully connected network. The average error rate of our design was about 8.1%. Cottrell trained a Back Propagation neural network based on the principal component analysis to the samples. Valentin and Edelman adopted eigenface and linear neuron respectively. After a phase, automatic gender classification attracted more and more attention with the rising demand of intelligent visual monitoring. Gutta combined radial basis function neural network with C4.5 decision tree into a hybrid classifier, getting a high averaged rate of recognition of 96%. Subsequent methods such as support vector machine and Adaboost delivered quite good reports to the gender classification.

II. DOMAIN EXPLANATION

Image Processing

2.1 What is an image?

An image is an array, or a matrix, of square pixels (picture elements) arranged in columns and rows. In a (8-bit) greyscale image each picture element has an assigned intensity that ranges from 0 to 255. A grey scale image is what people normally call a black and white image, but the name emphasizes that such an image will also include many shades of grey.

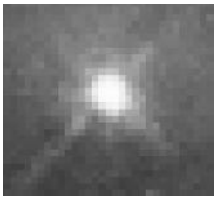


Fig: An image - an array or a matrix of pixels arranged in columns and rows.

A normal grayscale image has 8 bit colour depth = 256 grayscales. A “true colour” image has 24 bit colour depth = 8 x 8 x 8 bits = 256 x 256 x 256 colours = ~16 million colours.

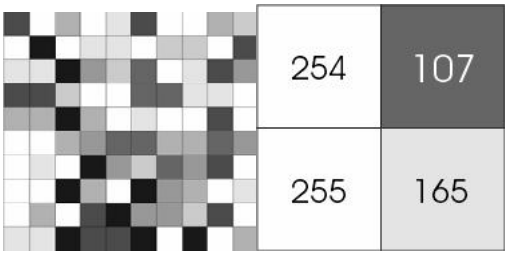


Fig. Each pixel has a value from 0 (black) to 255 (white). The possible range of the pixel values depend on the colour depth of the image, here 8 bit = 256 tones or greyscales.

Fig: A true-colour image assembled from three grayscale images coloured red, green and blue. Such an image may contain up to 16 million different colours.

Some grayscale images have more grayscales, for instance 16 bit = 65536 grayscales. In principle three grayscale images can be combined to form an image with 281,474,976,710,656 grayscales.

There are two general groups of 'images': vector graphics (or line art) and bitmaps (pixel-based or 'images'). Some of the most common file formats are:

GIF — an 8-bit (256 colour), non-destructively compressed bitmap format. Mostly used for web. Has several sub-standards one of which is the animated GIF.

JPEG — a very efficient (i.e. much information per byte) destructively compressed 24 bit (16 million colours) bitmap format. Widely used, especially for web and Internet (bandwidth-limited).

TIFF — the standard 24 bit publication bitmap format. Compresses non-destructively with, for instance, Lempel-Ziv-Welch (LZW) compression.

PS — Postscript, a standard vector format. Has numerous sub-standards and can be difficult to transport across platforms and operating systems.

PSD – a dedicated Photoshop format that keeps all the information in an image including all the layers.

Pictures are the most common and convenient means of conveying or transmitting information. A picture is worth a thousand words. Pictures concisely convey information about positions, sizes and inter relationships between objects. They portray spatial information that we can recognize as objects. Human beings are good at deriving information from such images, because of our innate visual and mental abilities. About 75% of the information received by human is in pictorial form. An image is digitized to convert it to a form which can be stored in a computer's memory or on some form of storage media such as a hard disk or CD-ROM. This digitization procedure can be done by a scanner, or by a video camera connected to a frame grabber board in a computer. Once the image has been digitized, it can be operated upon by various image processing operations.

Image processing operations can be roughly divided into three major categories, Image Compression, Image Enhancement and Restoration, and Measurement Extraction. It involves reducing the amount of memory needed to store a digital image. Image defects which could be caused by the digitization process or by faults in the imaging set-up (for example, bad lighting) can be corrected using Image Enhancement techniques. Once the image is in good condition, the Measurement Extraction operations can be used to obtain useful information from the image. Some examples of Image Enhancement and Measurement Extraction are given below. The examples shown all operate on 256 grey-scale images. This means that each pixel in the image is stored as a number between 0 to 255, where 0 represents a black pixel, 255 represents a white pixel and values in-between represent shades of grey. These operations can be extended to operate on color images. The examples below represent only a few of the many techniques available for operating on images. Details about the inner workings of the operations have not been given, but some references to books containing this information are given at the end for the interested reader.

Images and pictures

As we mentioned in the preface, human beings are predominantly visual creatures: we rely heavily on our vision to make sense of the world around us. We not only look at things to identify and classify them, but we can scan for differences, and obtain an overall rough feeling for a scene with a quick glance. Humans have evolved very precise visual skills: we can identify a face in an instant; we can differentiate colors; we can process a large amount of visual information very quickly.

However, the world is in constant motion: stare at something for long enough and it will change in some way.

Even a large solid structure, like a building or a mountain, will change its appearance depending on the time of day (day or night); amount of sunlight (clear or cloudy), or various shadows falling upon it. We are concerned with single images: snapshots, if you like, of a visual scene. Although image processing can deal with changing scenes, we shall not discuss it in any detail in this text. For our purposes, an image is a single picture which represents something. It may be a picture of a person, of people or animals, or of an outdoor scene, or a microphotograph of an electronic component, or the result of medical imaging. Even if the picture is not immediately recognizable, it will not be just a random blur.

Image processing involves changing the nature of an image in order to either

1. Improve its pictorial information for human interpretation,
2. Render it more suitable for autonomous machine perception.

We shall be concerned with digital image processing, which involves using a computer to change the nature of a digital image. It is necessary to realize that these two aspects represent two separate but equally important aspects of image processing. A procedure which satisfies condition, a procedure which makes an image look better may be the very worst procedure for satisfying condition. Humans like their images to be sharp, clear and detailed; machines prefer their images to be simple and uncluttered.

Images and digital images

Suppose we take an image, a photo, say. For the moment, lets make things easy and suppose the photo is black and white (that is, lots of shades of grey), so no colour. We may consider this image as being a two dimensional function, where the function values give the brightness of the image at any given point. We may assume that in such an image brightness values can be any real numbers in the range (black) (white).

A digital image from a photo in that the values are all discrete. Usually they take on only integer values. The brightness values also ranging from 0 (black) to 255 (white). A digital image can be considered as a large array of discrete dots, each of which has a brightness associated with it. These dots are called picture elements, or more simply pixels. The pixels surrounding a given pixel constitute its neighborhood. A neighborhood can be characterized by its shape in the same way as a matrix: we can speak of a neighborhood. Except in very special circumstances, neighborhoods have odd numbers of rows and columns; this ensures that the current pixel is in the centre of the neighborhood.

Image Processing Fundamentals:

Pixel:

In order for any digital computer processing to be carried out on an image, it must first be stored within the computer in a suitable form that can be manipulated by a computer program. The most practical way of doing this is to divide the image up into a collection of discrete (and usually small) cells, which are known as *pixels*. Most commonly, the image is divided up into a rectangular grid of pixels, so that each pixel is itself a small rectangle. Once this has been done, each pixel is given a pixel value that represents the color of that pixel. It is assumed that the whole pixel is the same color, and so any color variation that did exist within the area of the pixel before the image was discretized is lost. However, if the area of each pixel is very small, then the discrete nature of the image is often not visible to the human eye. Other pixel shapes and formations can be used, most notably the hexagonal grid, in which each pixel is a small hexagon. This has some advantages in image processing, including the fact that pixel connectivity is less ambiguously defined than with a square grid, but hexagonal grids are not widely used. Part of the reason is that many image capture systems (*e.g.* most CCD cameras and scanners) intrinsically discretize the captured image into a rectangular grid in the first instance.

Pixel Connectivity

The notation of pixel connectivity describes a relation between two or more pixels. For two pixels to be connected they have to fulfill certain conditions on the pixel brightness and spatial adjacency.

First, in order for two pixels to be considered connected, their pixel values must both be from the same set of values V. For a grayscale image, V might be any range of graylevels, e.g.V={22,23,...40}, for a binary image we simple have V={1}.

To formulate the adjacency criterion for connectivity, we first introduce the notation of neighborhood. For a pixel p with the coordinates (x,y) the set of pixels given by:

$$N_4(p) = \{(x+1, y), (x-1, y), (x, y+1), (x, y-1)\}$$

is called its 4-neighbors. Its 8-neighbors are defined as

$$N_i(p) = N_4 \cup \{(x+1, y+1), (x+1, y-1), (x-1, y+1), (x-1, y-1)\}$$

From this we can infer the definition for 4- and 8-connectivity:
Two pixels p and q, both having values from a set V are 4-connected if q is from the set $N_4(p)$ and 8-connected if q is from $N_8(p)$.

General connectivity can either be based on 4- or 8-connectivity; for the following discussion we use 4-connectivity.

A pixel p is connected to a pixel q if p is 4-connected to q or if p is 4-connected to a third pixel which itself is connected to q. Or, in other words, two pixels q and p are connected if there is a path from p and q on which each pixel is 4-connected to the next one.

A set of pixels in an image which are all connected to each other is called a connected component. Finding all connected components in an image and marking each of them with a distinctive label is called connected component labeling.

An example of a binary image with two connected components which are based on 4-connectivity can be seen in Figure 1. If the connectivity were based on 8-neighbors, the two connected components would merge into one.

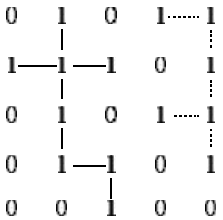


Fig:Two connected components based on 4-connectivity.

1. Pixel Values

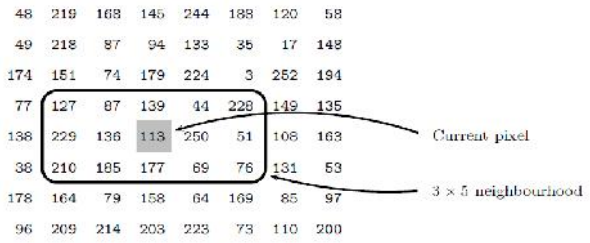
Each of the pixels that represents an image stored inside a computer has a *pixel value* which describes how bright that pixel is, and/or what color it should be. In the simplest case of binary images, the pixel value is a 1-bit number indicating either foreground or background. For a gray scale images, the pixel value is a single number that represents the brightness of the pixel. The most common *pixel format* is the *byte image*, where this number is stored as an 8-bit integer giving a range of possible values from 0 to 255. Typically zero is taken to be black, and 255 is taken to be white. Values in between make up the different shades of gray.

To represent color images, separate red, green and blue components must be specified for each pixel (assuming an RGB color space), and so the pixel `value' is actually a vector of three numbers. Often the three different components are stored as three separate `grayscale' images known as *color planes* (one for each of red, green and blue), which have to be recombined when displaying or processing. Multispectral Images can contain even more than three components for each pixel, and by extension these are stored in the same kind of way, as a vector pixel value, or as separate color planes.

The actual grayscale or color component intensities for each pixel may not actually be stored explicitly. Often, all that is stored for each pixel is an index into a colour map in which the actual intensity or colors can be looked up.

Although simple 8-bit integers or vectors of 8-bit integers are the most common sorts of pixel values used, some image formats support different types of value, for instance 32-bit signed integers or floating point values. Such values are extremely useful in image processing as they allow processing to be carried out on the image where the resulting pixel values are not necessarily 8-bit integers. If this approach is used, then it is usually necessary to set up a color map which relates particular ranges of pixel values to particular displayed colors.

Pixels, with a neighborhood:



Color scale

The two main color spaces are **RGB** and **CMYK.**

RGB

The **RGB color model** is an additive color model in which red, green, and blue light are added together in various ways to reproduce a broad array of colors. RGB uses additive color mixing and is the basic color model used in television or any other medium that projects color with light. It is the basic color model used in computers and for web graphics, but it cannot be used for print production.

The secondary colors of RGB – cyan, magenta, and yellow – are formed by mixing two of the primary colors (**red, green** or **blue**) and excluding the third color. Red and green combine to make yellow, green and blue to make cyan, and blue and red form magenta. The combination of red, green, and blue in full intensity makes white.

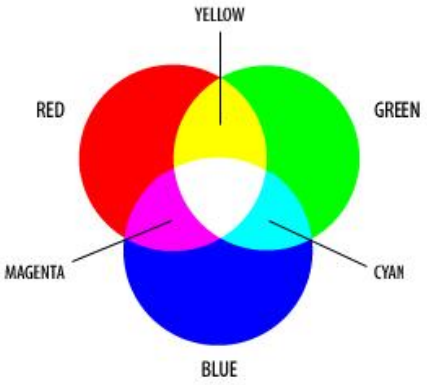
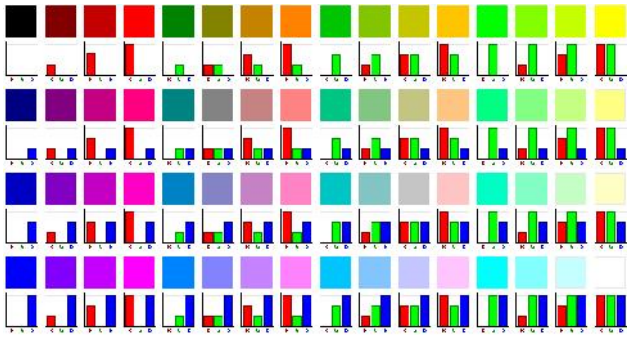


Fig: The additive model of RGB. Red, green, and blue are the primary stimuli for human color perception and are the primary additive colours.

To see how different RGB components combine together, here is a selected repertoire of colors and their respective relative intensities for each of the red, green, and blue components:



Some applications:

Image processing has an enormous range of applications; almost every area of science and technology can make use of image processing methods. Here is a short list just to give some indication of the range of image processing applications.

1. Medicine

- Inspection and interpretation of images obtained from X-rays, MRI or CAT scans,
- Analysis of cell images, of chromosome karyotypes.

2. Agriculture

- Satellite/aerial views of land, for example to determine how much land is being used for different purposes, or to investigate the suitability of different regions for different crops,
- Inspection of fruit and vegetables distinguishing good and fresh produce from old.

3. Industry

- Automatic inspection of items on a production line,
- Inspection of paper samples.

4. Law enforcement

- Fingerprint analysis,
- Sharpening or de-blurring of speed-camera images.

Aspects of image processing:

It is convenient to subdivide different image processing algorithms into broad subclasses. There are different algorithms for different tasks and problems, and often we would like to distinguish the nature of the task at hand.

- **Image enhancement**: This refers to processing an image so that the result is more suitable for a particular application.

Example include:

- sharpening or de-blurring an out of focus image,
- highlighting edges,
- improving image contrast, or brightening an image,
- Removing noise.

Image restoration. This may be considered as reversing the damage done to an image by a known cause, for example:

- removing of blur caused by linear motion,
- removal of optical distortions,
- Removing periodic interference.

Image segmentation. This involves subdividing an image into constituent parts, or isolating certain aspects of an image:

- circles, or particular shapes in an image,
- In an aerial photograph, identifying cars, trees, buildings, or roads.

These classes are not disjoint; a given algorithm may be used for both image enhancement or for image restoration. However, we should be able to decide what it is that we are trying to do with our image: simply make it look better (enhancement), or removing damage (restoration).

An image processing task

We will look in some detail at a particular real-world task, and see how the above classes may be used to describe the various stages in performing this task. The job is to obtain, by an automatic process, the postcodes from envelopes. Here is how this may be accomplished:

- **Acquiring the image**: First we need to produce a digital image from a paper envelope. This can be done using either a CCD camera, or a scanner.
- **Preprocessing:** This is the step taken before the major image processing task. The problem here is to perform some basic tasks in order to render the resulting image more suitable for the job to follow. In this case it may involve enhancing the contrast, removing noise, or identifying regions likely to contain the postcode.
- **Segmentation**: Here is where we actually get the postcode; in other words, we extract from the image that part of it which contains just the postcode.
- **Representation and description** These terms refer to extracting the particular features which allow us to differentiate between objects. Here we will be looking for curves, holes and corners which allow us to distinguish the different digits which constitute a postcode.

Recognition and interpretation: This means assigning labels to objects based on their descriptors (from the previous step), and assigning meanings to those labels. So we identify particular digits, and we interpret a string of four digits at the end of the address as the postcode.

III. PROPOSED SYSTEM

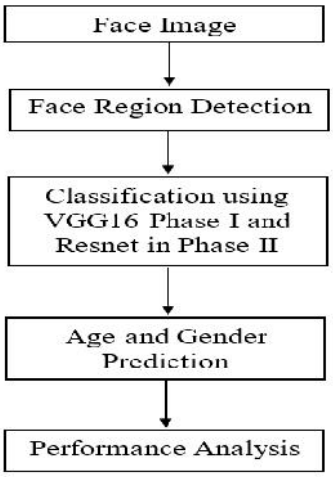
In Proposed System, architecture is used throughout our experiments for both age and gender classification. Prediction running times can conceivably be substantially improved by running the net-work on image batches. We test the accuracy of our CNN (VGG 16 in phase I and Resnet in Phase II) design using the recently released dataset designed for age and gender classification. We emphasize that the same network architecture is used for all test folds of the benchmark and in fact, for both gen-der and age classification tasks. This is performed in order to ensure the validity of our results across folds, but also to demonstrate the generality of the network design proposed here; the same architecture performs well across different, related problems. For age classification, we measure and compare both the accuracy when the algorithm gives the exact age-group classification and when the algorithm is off by one adjacent age-group (i.e., the subject belongs to the group immediately older or immediately younger than the predicted group). This follows others who have done so in the past, and reflects the uncertainty inherent to the task – facial features often change very little between oldest faces in one age class and the youngest faces of the subsequent class. Which used the same gender classification pipeline of applied to more effective alignment of the faces; faces in their tests were synthetically modified to appear facing forward.

Advantages:

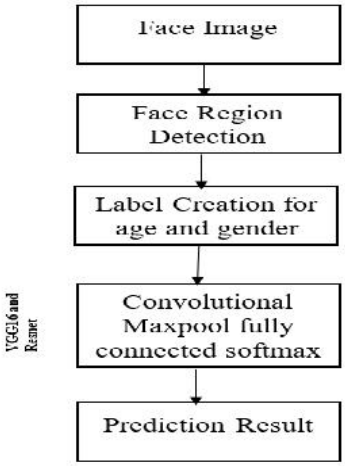
- The background will be extracted separately.

The Classification/ Recognition accuracy will be more.

System Architecture



Flow Diagram



IV. TESTING OF PRODUCT

System testing is the stage of implementation, which aimed at ensuring that system works accurately and efficiently before the live operation commence. Testing is the process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding an error. A successful test is one that answers a yet undiscovered error.

Testing is vital to the success of the system. System testing makes a logical assumption that if all parts of the system are correct, the goal will be successfully achieved. The candidate system is subject to variety of tests-on-line response, Volume Street, recovery and security and usability test. A series of tests are performed before the system is ready for the user acceptance testing. Any engineered product can be tested in one of the following ways. Knowing the specified function that a product has been designed to from, test can be conducted to demonstrate each function is fully operational. Knowing the internal working of a product, tests can be conducted to ensure that "al gears mesh", that is the internal operation of the product performs according to the specification and all internal components have been adequately exercised.

UNIT TESTING:

Unit testing is the testing of each module and the integration of the overall system is done. Unit testing becomes verification efforts on the smallest unit of software design in the module. This is also known as 'module testing'. The modules of the system are tested separately. This testing is carried out during the programming itself. In this testing step, each model is found to be working satisfactorily as regard to the expected output from the module. There are some validation checks for the fields. For example, the validation check is done for verifying the data given by the user where both format and validity of the data entered is included. It is very easy to find error and debug the system.

INTEGRATION TESTING:

Data can be lost across an interface, one module can have an adverse effect on the other sub function, when combined, may not produce the desired major function. Integrated testing is systematic testing that can be done with sample data. The need for the integrated test is to find the overall system performance. There are two types of integration testing. They are:

i) Top-down integration testing.
ii) Bottom-up integration testing.

WHITE BOX TESTING:

White Box testing is a test case design method that uses the control structure of the procedural design to drive cases. Using the white box testing methods, we derived test cases that guarantee that all independent paths within a module have been exercised at least once.

BLACK BOX TESTING:

- Black box testing is done to find incorrect or missing function
- Interface error
- Errors in external database access
- Performance errors
- Initialization and termination errors

In 'functional testing', is performed to validate an application conforms to its specifications of correctly performs all its required functions. So this testing is also called 'black box testing'. It tests the external behavior of the system. Here the engineered product can be tested knowing the specified function that a product has been designed to perform, tests can be conducted to demonstrate that each function is fully operational.

VALIDATION TESTING:

After the culmination of black box testing, software is completed assembly as a package, interfacing errors have been uncovered and corrected and final series of software validation tests begin validation testing can be defined as many, but a single definition is that validation succeeds when the software functions in a manner that can be reasonably expected by the customer.

- **USER ACCEPTANCE TESTING:**

User acceptance of the system is the key factor for the success of the system. The system under consideration is tested for user acceptance by constantly keeping in touch with prospective system at the time of developing changes whenever required.

- **OUTPUT TESTING:**

After performing the validation testing, the next step is output asking the user about the format required testing of the proposed system, since no system could be useful if it does not produce the required output in the specific format. The output displayed or generated by the system under consideration. Here the output format is considered in two ways. One is screen and the other is printed format. The output format on the screen is found to be correct as the format was designed in the system phase according to the user needs. For the hard copy also output comes out as the specified requirements by the user. Hence the output testing does not result in any connection in the system.

System Implementation:

Implementation of software refers to the final installation of the package in its real environment, to the satisfaction of the intended users and the operation of the system. The people are not sure that the software is meant to make their job easier.

The active user must be aware of the benefits of using the system

Their confidence in the software built up Proper guidance is impaired to the user so that he is comfortable in using the application Before going ahead and viewing the system, the user must know that for viewing the result, the server program should be running in the server. If the server object is not running on the server, the actual processes will not take place.

User Training:

To achieve the objectives and benefits expected from the proposed system it is essential for the people who will be involved to be confident of their role in the new system. As system becomes more complex, the need for education and training is more and more important.

Education is complementary to training. It brings life to formal training by explaining the background to the resources for them. Education involves creating the right atmosphere and motivating user staff. Education information can make training more interesting and more understandable.

Training on the Application Software:

After providing the necessary basic training on the computer awareness, the users will have to be trained on the new application software. This will give the underlying philosophy of the use of the new system such as the screen flow, screen design, type of help on the screen, type of errors while entering the data, the corresponding validation check at each entry and the ways to correct the data entered. This training may be different across different user groups and across different levels of hierarchy.

Operational Documentation:

Once the implementation plan is decided, it is essential that the user of the system is made familiar and comfortable with the environment. A documentation providing the whole operations of the system is being developed. Useful tips and guidance is given inside the application itself to the user. The system is developed user friendly so that the user can work the system from the tips given in the application itself.

System Maintenance:

The maintenance phase of the software cycle is the time in which software performs useful work. After a system is successfully implemented, it should be maintained in a proper manner. System maintenance is an important aspect in the software development life cycle. The need for system maintenance is to make adaptable to the changes in the system environment. There may be social, technical and other environmental changes, which affect a system which is being implemented. Software product enhancements may involve providing new functional capabilities, improving user displays and mode of interaction, upgrading the performance characteristics of the system. So only thru proper system maintenance procedures, the system can be adapted to cope up with these changes. Software maintenance is of course, far more than “finding mistakes”.

Corrective Maintenance:

The first maintenance activity occurs because it is unreasonable to assume that software testing will uncover all latent errors in a large software system. During the use of any large program, errors will occur and be reported to the developer. The process that includes the diagnosis and

correction of one or more errors is called Corrective Maintenance.

Adaptive Maintenance:

The second activity that contributes to a definition of maintenance occurs because of the rapid change that is encountered in every aspect of computing. Therefore, Adaptive maintenance termed as an activity that modifies software to properly interfere with a changing environment is both necessary and commonplace.

Perceptive Maintenance:

The third activity that may be applied to a definition of maintenance occurs when a software package is successful. As the software is used, recommendations for new capabilities, modifications to existing functions, and general enhancement are received from users. To satisfy requests in this category, Perceptive maintenance is performed. This activity accounts for the majority of all efforts expended on software maintenance.

Preventive Maintenance:

The fourth maintenance activity occurs when software is changed to improve future maintainability or reliability, or to provide a better basis for future enhancements. Often called preventive maintenance, this activity is characterized by reverse engineering and re-engineering techniques.

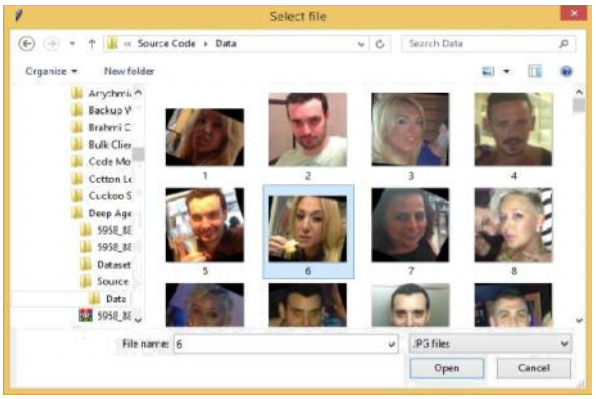
Modules:

- ✓ Input Image
- ✓ Face Region Detection
- ✓ Key-point Features
- ✓ Classification
- ✓ Performance Analysis

V. SCREENSHOTS

Step 1: Select the file in package

Step 2: Select the face images and destroyed images



Resnet:

Layer (type)	Output Shape	Param #
block1_conv1 (Conv2D)	(None, 32, 32, 64)	1792
activation_20 (Activation)	(None, 32, 32, 64)	0
block1_conv2 (Conv2D)	(None, 32, 32, 64)	36928
activation_21 (Activation)	(None, 32, 32, 64)	0
block1_pool (MaxPooling2D)	(None, 16, 16, 64)	0
block2_conv1 (Conv2D)	(None, 16, 16, 128)	73856
activation_22 (Activation)	(None, 16, 16, 128)	0
block2_conv2 (Conv2D)	(None, 16, 16, 128)	147584
activation_23 (Activation)	(None, 16, 16, 128)	0
block2_pool (MaxPooling2D)	(None, 8, 8, 128)	0
block3_conv1 (Conv2D)	(None, 8, 8, 256)	295168
activation_24 (Activation)	(None, 8, 8, 256)	0

block2_pool (MaxPooling2D)	(None, 8, 8, 128)	0
block3_conv1 (Conv2D)	(None, 8, 8, 256)	295168
activation_24 (Activation)	(None, 8, 8, 256)	0
block3_conv2 (Conv2D)	(None, 8, 8, 256)	590080
activation_25 (Activation)	(None, 8, 8, 256)	0
block3_conv3 (Conv2D)	(None, 8, 8, 256)	590080
activation_26 (Activation)	(None, 8, 8, 256)	0
block3_conv4 (Conv2D)	(None, 8, 8, 256)	590080
activation_27 (Activation)	(None, 8, 8, 256)	0
block3_pool (MaxPooling2D)	(None, 4, 4, 256)	0
block4_conv1 (Conv2D)	(None, 4, 4, 512)	1180160
activation_28 (Activation)	(None, 4, 4, 512)	0
block4_conv2 (Conv2D)	(None, 4, 4, 512)	2359808

```
    OPERATION          DATA DIMENSIONS   WEIGHTS(N)   WEIGHTS(%)

        Input   #####     32   32    3
       Conv2D    \|/  -------------------     1792     0.0%
         relu   #####     32   32   64
       Conv2D    \|/  -------------------    36928     0.1%
         relu   #####     32   32   64
  MaxPooling2D  Y max -------------------        0     0.0%
                #####     16   16   64
       Conv2D    \|/  -------------------    73856     0.2%
         relu   #####     16   16  128
       Conv2D    \|/  -------------------   147584     0.3%
         relu   #####     16   16  128
  MaxPooling2D  Y max -------------------        0     0.0%
                #####      8    8  128
       Conv2D    \|/  -------------------   295168     0.7%
         relu   #####      8    8  256
       Conv2D    \|/  -------------------   590080     1.3%
         relu   #####      8    8  256
       Conv2D    \|/  -------------------   590080     1.3%
         relu   #####      8    8  256
       Conv2D    \|/  -------------------   590080     1.3%
         relu   #####      8    8  256
  MaxPooling2D  Y max -------------------        0     0.0%
                #####      4    4  256
       Conv2D    \|/  -------------------  1180160     2.6%
         relu   #####      4    4  512
       Conv2D    \|/                       2359808     5.2%
         relu   #####      4    4  512
       Conv2D    \|/  -------------------  2359808     5.2%
         relu   #####      4    4  512
       Conv2D    \|/  -------------------  2359808     5.2%
         relu   #####      4    4  512
  MaxPooling2D  Y max -------------------        0     0.0%
                #####      2    2  512
       Conv2D    \|/  -------------------  2359808     5.2%
         relu   #####      2    2  512
       Conv2D    \|/  -------------------  2359808     5.2%
         relu   #####      2    2  512
       Conv2D    \|/  -------------------  2359808     5.2%
         relu   #####      2    2  512
       Conv2D    \|/                       2359808     5.2%
         relu   #####      2    2  512
      Flatten   |||| -------------------        0     0.0%
                #####           2048
        Dense   XXXXX -------------------  8392704    18.6%
         relu   #####           4096
      Dropout    || -------------------        0     0.0%
                #####           4096
        Dense   XXXXX -------------------  16781312   37.1%
         relu   #####           4096
      Dropout    || -------------------        0     0.0%
```

```
Query image Age: (25, 32)
Query image Gender: m
TP 1073
TN 1
FP 129
FN 1
Specificity 76.92307692307693
Precision 89.26788685524126
Recall 99.90689013035382
NPV 50.0
FPR 99.23076923076923
FNR 0.0931098696461825
FDR 10.732113144758735
Accuracy 93.20265780730897
```

REFERENCES

[1] Levi, G., &Hassncer, T. (2015). Age and gender classification using convolutional neural networks.Computer Vision and Pattern Recognition Workshops (pp.34-42).IEEE.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImagenetClassification with Deep Convolutional Neural Networks," Advances in Neural Information Processing Systems, pp. 1097-1105, 2012.

[3] Zhou, S. K., Georgescu, B., Zhou, X., &Comaniciu, D. (2010). Method for performing image based regression using boosting. US, US7804999.

[4] D. Strigl, K. Kofler, and S. Podlipnig, "Performance and Scalability of GPU-Based Convolutional Neural Networks," Euromicro Conference on Parallel, Distributed, and Network-Based Processing, pp. 317-324, 2010.

[5] D. Cireşan, U. Meier, and J. Schmidhuber, "Multi-column Deep Neural Networks for Image Classification," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2012.

[6] D. Cireşan, et al., "Flexible, High Performance Convolutional Neural Networks for Image Classification," in International Joint Conference on Artificial Intelligence, pp.1237-1242, 2011.

[7] Geng, X., Zhou, Z. H., Zhang, Y., Li, G., & Dai, H. (2006). Learning from facial aging patterns for automatic age estimation.