AGE AND GENDER CLASSIFICATION FROM A LOW-QUALITY HUMAN FACE IMAGES

SHAABAN, MOHAMMED NAJIB AHMED

SESSION 2020/2021

FACULTY OF COMPUTING AND INFORMATICS
MULTIMEDIA UNIVERSITY
FEBRUARY 20

AGE AND GENDER CLASSIFICATION FROM A LOW-QUALITY HUMAN FACE IMAGES

BY

SHAABAN, MOHAMMED NAJIB AHMED

SESSION
2020/2021

THIS PROJECT IS
PREPARED FOR

FACULTY OF COMPUTING AND INFORMATICS
MULTIMEDIA UNIVERSITY
IN PARTIAL FULFILLMENT
FOR

BACHELOR OF COMPUTER SCIENCE
B.CS(HONS) DATA SCIENCE

FACULTY OF COMPUTING AND INFORMATICS

MULTIMEDIA UNIVERSITY
FEBRUARY 2021

DECLARATION

I hereby declare that the work has been done by myself and no portion of the work contained in this thesis has been submitted in support of any application for any other degree or qualification of this or any other university or institute of learning.

Shaaban, Mohammed Najib Ahmed
Faculty of Computing & Informatics
Multimedia University
Date: 10:02:2021

ACKNOWLEDGMENT

Firstly, I would like to introduce my thanks and gratitude to my project supervisor DR. Loh Yuen Peng, for her assistance to accomplish this project

I would also like to introduce my thanks and love to my dear father, for his support along my journey of study. I would also like to extend my thanks to the management of Multimedia University, As they were always keen to provide for us the latest learning technologies

Finally, I would like to present my thanks to all my lecturers, who taught me along my journey of study on MMU as well as my friends.

ABSTRACT

Age and gender are two of the most important facial features whereas they are involved in many intelligent applications such as visual surveillance and access control. However, such application might be applied in a messy environment that leads to capture of low-quality human face images, which results in wrong age and gender classification, however most of the existing methods didn't address the mentioned problem.

In this project we aim to improve the pre-existing age and gender classification model developed by Levi and Hassncer (2015) by developing an autoencoder based image enhancement technique that handles and improves the quality of different types of low-quality images (Low-light, Low-resolution, and Blurred images). We evaluated our proposed enhancement technique in two different approaches: Application evaluation process and image quality evaluation process. The results showed a significant improvement in pre-existing age and gender classification model for two types of low-quality images (Low-light and Blurred image), however, further analysis and investigations have been planted as a future work to further increase the accuracy of the proposed autoencoder based enhancement technique in the low-quality pixelated images.

Table of Contents

List of Tables

List of Figures

CHAPTER 1

Introduction

As Today's technology became the backbone of the new era, the need to constantly improve it became an essential matter whereas the human significantly relies on it to accomplish most of their daily activities and tasks. Age and gender classification systems have gained much attention in the computer vision community in recent years and played an important role in today's technology due to its related connection to several intelligent applications such as access control and visual surveillance. However, such intelligent applications mostly applied in a bit messy environment such as a security guardhouse or a train station. Such environments might capture low-quality human face images, the thing that might distort or hide the image facial features and eventually lead to wrong classification results. However, most of the existing methods such as Levi and Hassncer (2015) and Duan , Yang and Li (2018) didn’t address the mentioned problem (Age and gender classification from a low-quality human face image).

In this project we attempt to solve this problem by developing an autoencoder based image enhancement technique that is able to handle and enhance low-quality images of three different types: Blurred, Low-light ,and Low-resolution. To the end we evaluated our model in two different approaches: Application evaluation process and Image quality evaluation process using two of the objective image quality measurements: PSNR and SSIM. The result shows a significant improvement of the proposed model in the pre-existing age and gender classification model developed by Levi and Hassncer (2015) in two type of low-quality images :Blurred and Low-light, however an investigation and analysis planned for the future work to achieve better accuracy for the low-quality pixelated images.

1.1 Problem Statement

Age and gender are two of the key facial attributes, play a fundamental role in social interactions. Several intelligence applications such as control access and visual surveillance required such attributes data to achieve their missions. whereas having the wrong data in such applications increases the probability of system failure saturation. While such applications mostly applied in a bit messy environment such as a security guardhouse or a train station. Such environments might capture a low-quality human face image, the thing that might cause the classifier system to detect wrong facial features and therefore produce wrong data.

1.2 Research Objectives

1- To Improve the accuracy of the pre-existing age and gender classification models

2- To propose an autoencoder based image enhancement technique

3- To benchmark the proposed method with a number of the state of art methods (To be achieved in part 2)

1.3 Scope

The scope of this project is to develop an autoencoder based image enhancement technique that is able to handle and enhance low-quality images of different types and evaluate it on the pre-existing age and gender classification model for the aim of improving the model.

CHAPTER 2

Literature Review

This chapter aims to summarize an overall study and investigations on a number of state-of-the-art methods and algorithms for age and gender classification from a human face images and provides a cursory overview of deep convolutional networks in an attempt of us to highlights their advantages and drawbacks that can be either reused or improved on the future work, at the end we provided an overall study and investigations over a number existing datasets that can be used for age and gender classification future work.

2.1 Separate Age and gender Classification

2.1.1 Age Classification

Automatic Age classification from a human face image has received increasing attention in the last few years due to the rapid development in technology and the prosperity of social platforms and social media. Many methods have been proposed, an early method Kwon and Vitoria (1999) is based on craniofacial development theory and skin wrinkle analysis, the method divided into two-phase on the first phase the facial features (i.e., mouth, nose, eye, chin and the side of the face) are identified from the face image. While in the second phase, it calculates the ratio between these facial feature measurements to identify to which age category the face is.

More recently Reid, Samangooei, Chen, Nixon and Ross (2013) use a similar approach to perform age classification for individuals under 18 years old. However, The drawback of this method is that it required precise facial features identification from an image to be able to calculate the accurate ratio, which might be a tricky task to be performed in those images taken from unconstrained environments.

Different approach from that is Guo, Dyer and Huang (2008), they developed a method called orthogonal locality preserving projections (OLPP) that was used to reduce the dimensionality of the dataset. They used the Support Vector Machine (SVM) and Support Vector Regression (SVR) methods to predict the estimated age based on the age manifold respectively. They then applied a locally adjusted robust regression (LARR) to improve the age estimation performance and accuracy. The final results show that the LARR method gives better age estimation than the purely robust regression by SVR or purely classification by SVM.

2.1.2 Gender Classification

A detailed survey of the early methods used for the human gender classification from a human face image can be found in Makinen and Raisamo (2008). An early method developed by O'toole, Vetter, Troje and Bülthoff (1997) was mainly focused on the comparison between the three-dimensional structure and grey-level image data on the process of classifying the human gender from visual images. They applied the principal component analysis (PCA) separately to both analytics. The individual components captured the information related to the human face from both analytics. They were able to compare the quantity of the available data obtained from both analytics. At the end The results show that the three-dimensional structure more supports the process of classifying human gender than the grey-level image data.

Different approach from that is Ullah, Hussain, Aboalsamh, Muhammad, Mirza, and Bebis (2012), They proposed a new features descriptor based on Dyadic wavelet Transform (DyWT) and Local Binary Pattern (LBP) that can be used for the classification of human gender from the human face image. The LBP is one of the best-known texture descriptors that work to separate various objects on the image. while the DyWt is a multi-scale image converting technique that works to differentiate the features at different scales by decomposing the image into various groups. The dimension of the feature space generated by the new proposed descriptor was quite high and to solve this problem they applied the feature subset selection (FSS) technique as a means to reduce it. They also applied a minimum distance classifier for the aim of keeping the system complicity simple, and its performance is based on the matrices. They examined the minimum distance classifier based on three matrices: Euclidean distance, City block distance, and Chi-square. They validate the proposed method using two public domain databases: FERET and Multi-PIE. The results show that the City block distance minimum distance classifier gave the best result with the proposed DyWT-LBP descriptor.

2.2 Integrate Age and Gender Classification

Age and gender are two key facial feature attributes that are involved in many of the contemporary intelligent applications and that made them attention issues for many researchers in the computer vision fields in the last few years. One of the state-of-art methods Bekhouche, Ouafi, Benlamoudi, Taleb-Ahmed and Hadid (2015) is based on Multi Level Local Phase Quantization (ML-LPQ) features which are extracted from the normalized face image with two different Support Vector Machines (SVM) models used to predict the age group and the gender of a person.

Different approach from that is Azarmehr, Laganiere, Lee, Xu and Laroche (2015). They proposed a method for age and gender classification that works for unconstrained environments videos. They Introduced a segmental dimensionality reduction method to reduce the memory requirements up to 99.5%. They also introduced several improvements for face alignment, illumination normalization, and feature extraction using a multi-resolution binary pattern method. Finally, they used the SVM+RBF classifier along with a discriminative demographics classification strategy to improve the performance. The advantage of this method is that it required less computational power and less memory space.

In Fazl-Ersi, Mousa-Pasandi, Laganiere and Awad (2014) they proposed a novel method for age and gender estimation that makes use of multiple visual descriptors rather than a single visual descriptor as most of the state-of-art methods do. The proposed method relied on the selection of information features that just permits the regions that can best separate face images of various segment classes (as for age and gender) to be added to the face representations.

In Bukar, Ugail and Connah (2016), they proposed a supervised appearance model (SAM) that improves the traditional active appearance model (AAM) When used for facial feature extraction. by replacing PCA with partial least-squares regression. The model then used for the task of age and gender classification from a human face image and the results show that SAM outperform most of the state-of-art methods that used AAM as facial feature extractor.

2.3 Age and Gender Classification Using a Convolutional Neural Network

An early work on the age and gender classification using a deep convolutional neural network is Levi and Hassncer (2015). They proposed a simple convolutional neural network architecture. That can be utilized even with a small amount of learning data and designed to handle the challenges of unconstrained imaging conditions, apart from that they applied two extra methods to further reduce the risk of overfitting. They first applied a dropout learning method Hinton, Srivastava, Krizhevsky, Sutskever and Salakhutdinov (2012). Second, they used data augmentation by taking an irregular crop of 227×227 pixels from the 256×256 input picture and arbitrarily reflected it in each forward-backward training pass. Their network has been tested on the Adience benchmark (A challenging dataset of unconstrained imaging conditions). The results show that a significant improvement in the task of automatic estimation of age and gender from a human face image can be obtained with the use of the deep convolutional neural network.

In Agbo-Ajala and Viriri (2020) They proposed a novel method to predict age and gender classification from unfiltered real-world environments, using two-level CNN architecture which includes feature detection and prediction. they tackled the large variations in the unfiltered real-world images with a powerful image pre-processing algorithm that equips and processes those images before being fed into the CNN model. They validate their model with an OUI-Audience benchmark dataset.

In Duan , Yang and Li (2018) they present a hybrid structure that maxed both Convolutional Neural Network (CNN) and Extreme Learning Machine (ELM) to build an integrated age and gender classification system. Basically they tried to use the best advantage of each network of them as the CNN is used to extract the facial feature and the ELM is used for classification purposes. They applied different methods to reduce the risk of overfitting. Their result was validated in two datasets MORPHII and Adience benchmark.

In Fang, Yuan, Lu and Feng (2019) they proposed a multistage learning method. The method is divided into two stages: object detection learning stage and prediction learning stage. In the first stage, an encoder-decoder based on a segmentation network is proposed to mark-up the object region as human or not-human and only the human region involves the second stage. The second stage is to perform age and gender prediction tasks based on a deep convolutional neural network that contains two branches: gender classifier branch and age regressor branch. They validate the proposed method in three public challenging datasets. and the results proved the effectiveness of the proposed method. However, the disadvantage of this method is that it can be used to classify the age and gender from facial images that contain one and only one person. however, images with more than one person or with no person may exist on the real-world applications. The overall flowchart of Fang, Yuan, Lu and Feng (2019) multistage method is shown in figure 2-1.

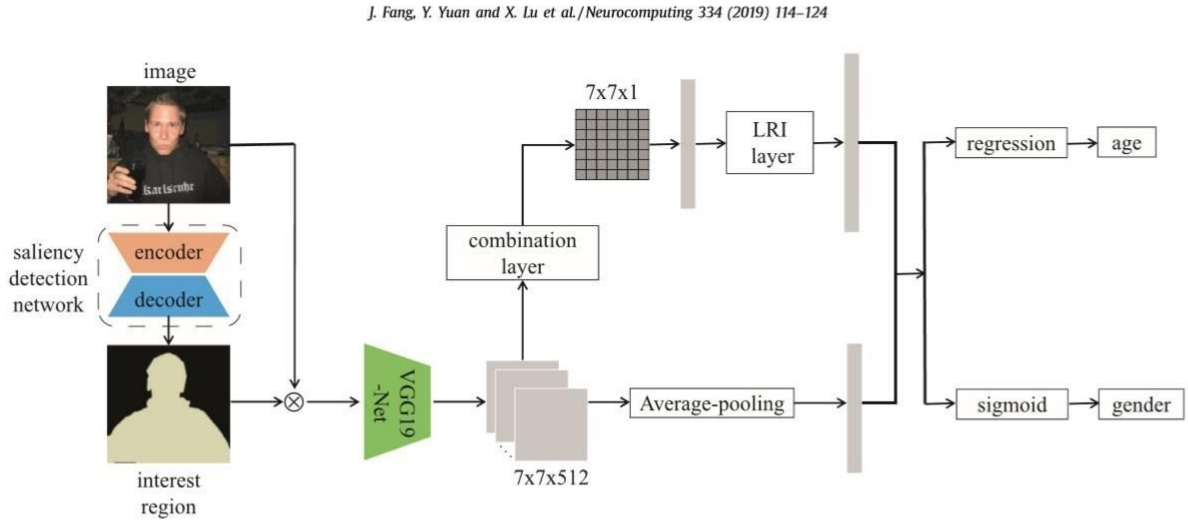


Figure 2-1. Multistage Learning method. Fang, Yuan, Lnd, Feng,, 2019

2.4 Datasets

Automatic age and gender classification from a human face images problem gained much attention on the last few years and several of the datasets with age and gender labels were released for public access as a part of the contributions done by the computer vision and machine learning experts in an attempt of them to facilitate the process of train accurate artificial intelligence models for age and gender recognition.

One of the most famous datasets for age and gender classification from a human face image is the Adience dataset Tal, Shai, Eran and Roee (2015) which includes a collection of 26,580 images for 2,248 individuals taken at different stages of life from their born and childhood until their adultness and aging. The images were taken in a way that represents various challenges of real-world imaging conditions, in an attempt to represent the various appearance, lighting, pose, noise, and more conditions the real-world images might have, and that made it a very powerful dataset to train and validate the age and gender classification models. The images on the dataset obtained from the Flickr albums, Gathered by automatic upload from iPhone5 and newer smartphone devices, and exposed to the general public by their author under the Creative Commons (CC) license. Samples images of Tal, Shai, Eran and Roee (2015) Adience dataset is shown in figure 2-2.

Figure 2-2. The Adience Dataset. Tal, Shai, Eran and Roee (2015)

Other than the Adience dataset is the Morph Longitudinal database developed by Ricanek and Tesafaye (2006) which is considered as the largest longitudinal facial recognition database in the world. The database is licensed for commercial and developmental uses. It is a combination of two large databases each containing nearly 200,000 face images, producing a database with more than 400,000 face images of approximately 70,000 individuals taken at different stages of their life (born, childhood, adultness, aging). The images are 8-bit color and may vary in the size. The database contains information for race, weight, height, age, gender, and eye coordinates more info of Morph database can be found in figure 2-3.

Gender/Ancestry Images Count

Gender	African	Asian	European	Ancestry Hispanic	Indian	Other	Unknown	Grand Total
Female	24,898	536	109,132	1,880	66	82	10	**136,604**
Male	155,783	1,150	99,093	8,908	322	93	102	**265,451**
Grand Total	**180,681**	**1,686**	**208,225**	**10,788**	**388**	**175**	**112**	**402,055**

Gender/Decade of life Image Count

Gender	< 20	20 - 29	30 - 39	Created Bins 40 - 49	50 - 59	60 - 69	70+	Grand Total
Female	9,773	43,756	37,841	34,644	9,010	1,305	275	**136,604**
Male	27,203	89,794	61,644	58,339	24,436	3,716	319	**265,451**
Grand Total	**36,976**	**133,550**	**99,485**	**92,983**	**33,446**	**5,021**	**594**	**402,055**

Gender/Ancestry Subjects

Gender	African	Asian	European	Ancestry Hispanic	Indian	Other	Unknown	Grand Total
Female	5,937	168	25,874	611	12	28	5	**32,635**
Male	21,125	284	11,019	2,398	63	32	44	**34,965**
Grand Total	**27,062**	**452**	**36,893**	**3,009**	**75**	**60**	**49**	**67,600**

Fig. 2-3. The Morph Dataset. Ricanek and Tesafaye (2006)

Another dataset that can be used for age and gender classification is the IMDB-WIKI dataset developed by Rasmus, Radu, and Luc (2015) and considered to be one of the largest datasets with age and gender labels and available for public access. Around 460,723 images on the dataset was collected from the IMDB website by scanning the profiles for more than 100,000 actors to gather all the images related to that actor along with information regarding his name, age, gender, race, and date of birth. The dataset also contains around 62,328 images collected from individuals' profiles in Wikipedia with the same meta information. The dataset adopts only the images with a timestamp and other images without it were removed from the dataset. The age was assigned to each actor on the images according to the timestamp or movie production time. However, this can be a disadvantage of the dataset as the assigned age might be incorrect due to wrong timestamps and movies extended production times. Figure 2-4 shows a set of images obtained from IMDB-WIKI dataset.

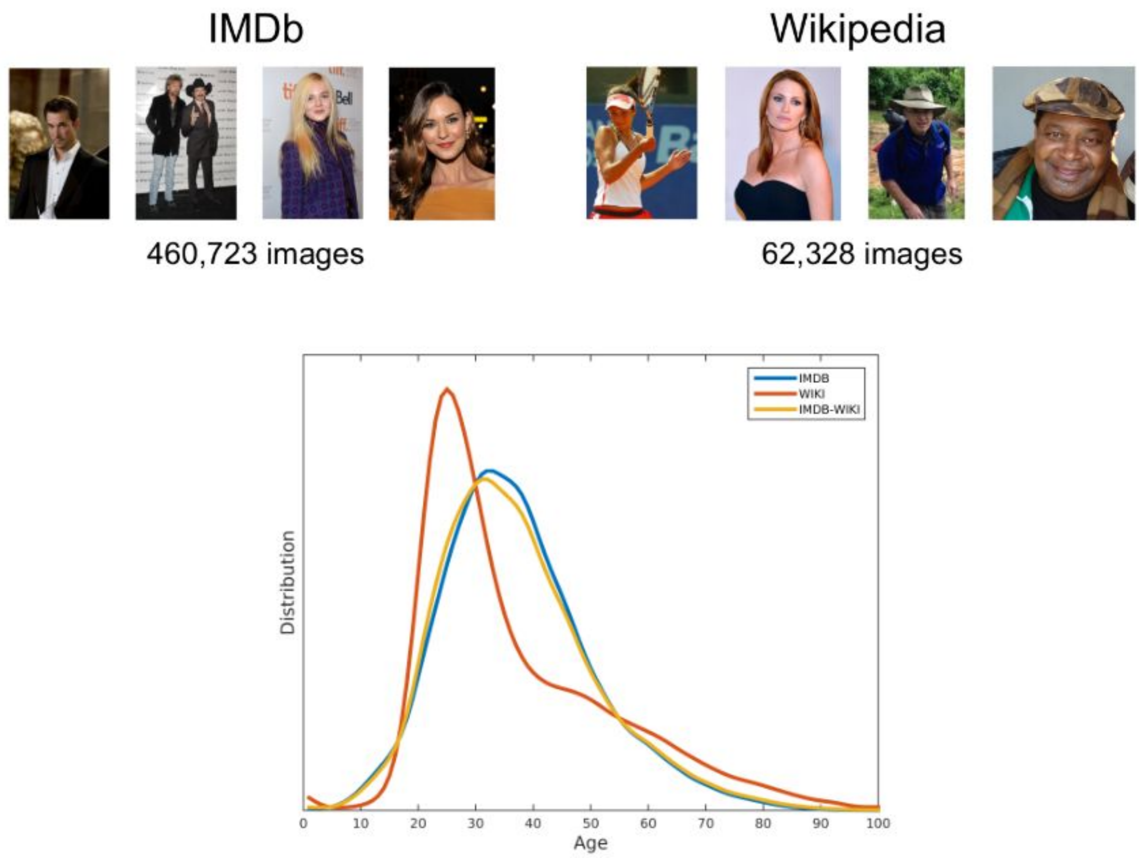


Figure 2-4. IMDB--WIKI Dataset. Rasmus, Radu, and Luc (2015)

Another dataset that can be used for the age and gender classification from a human face image is Tero, Samuli, and Timo (2014) Flickr-Faces-HQ Dataset (FFHQ). The dataset contains 52000 PNG high-quality human face images with resolution of 512x512. The dataset maintains considerable diversity in terms of ethnicity, age and image background. it also maintains considerable diversity in terms of accessories such as hats, eyeglasses, sunglasses , etc. The images were originally extracted from Flicker, and because of that they maintain all the biases of that website. and automatically aligned and cropped using dlib. Their dataset contains only images under legal licenses. different types of filters were used to prune the dataset. and finally Amazon Mechanical Turk was used to remove the unnecessary images. Figure 2-5 illustrates how the images look like on the Flicker-Faces-HQ dataset.

Figure 2-5. The Flickr-Faces-HQ Dataset (FFHQ). Tero, Samuli, and Timo (2014),

Different dataset from those described above is the All-Age-Faces dataset developed by (Cheng, Li , Wang, Yu., & Wang, 2019) the dataset contains 5,941 human male face images and 7,381 human female face images together to produce a dataset of 13'322 human face images for individuals of the range of 2 to 80. The dataset contains four different folders to store the original face images, aligned face images, facial landmarks, and examples of landmark distribution separately from each other. Each image on the dataset has its own unique name that represents the serial number and specific age. Images with the serial number of 0000 to 07389 are all human female images and images with the serial number of 07381 to 13321 are all human male images. The dataset has been randomly split into two sets, one for training and another one for validation. However, the All-Age-Faces dataset appears to be a very powerful dataset for the problem of age and gender classification from human face images it suffers from a diversity limitation as most of the images contained on the dataset are for Asian race people the thing that might reflect negatively on the newly facial feature based analysis model performance while dealing with different race of people than the Aisain. However such limitations will not affect our models as they are both aimed to learn how to increase the quality of the images and has nothing to do with the facial feature of the person on the image. Figure 2-6 shows a set of images obtained from the All-Age-Faces dataset.

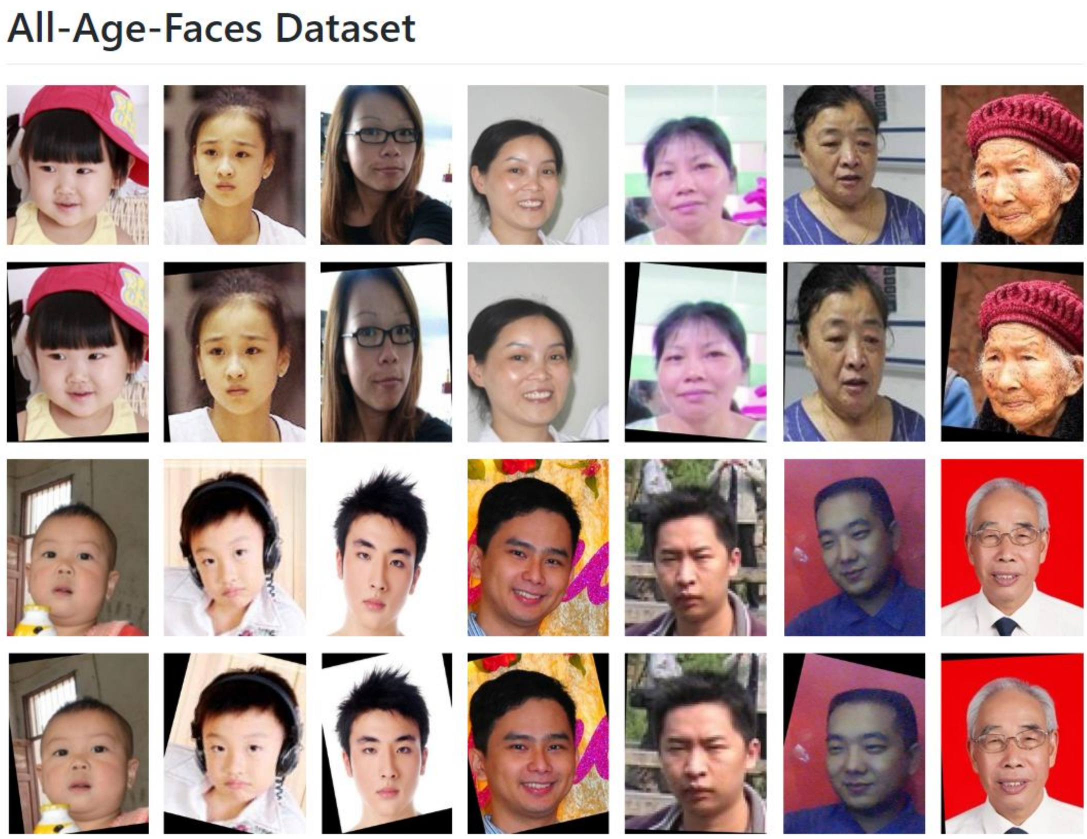


Figure. 2-6. All-Age-Faces Dataset. (Cheng, Li , Wang, Yu., & Wang, 2019)

2.4.1 Datasets Table of Comparison

Table 2-1. Datasets comparison

No	Name	Size (Number of images)	Diversity of Ethnicity	Advantage/Disadvantage
1	Adience Dataset	26,580	Hight	Advantage: The dataset is designed to be as similar as

				possible as real-world challenging face images
2	Morph Database	200,000 (Has been doubled recently)	High	Advantage: The largest longitudinal facial recognition database in the world
3	IMDB-WIKI Dataset	460,723	High	Disadvantage: inaccurate age labelled, the assigned age might be incorrect due to wrong timestamps and movies extended production times
4	Flickr-Faces-HQ Dataset (FFHQ)	52000	High	Advantage: The images are in vary high quality with resolution of 512 x 512 Advantage: It has very good diversity in terms of age, ethnicity and image background Advantage: it has very good diversity items of accessories such as sunglasses, eyewear ..etc
5	All_Age_Faces datasset	13322	Low	Disadvantage: The dataset suffer low ethnicity diversity as it contains mostly images of Asian race

CHAPTER 3

Theoretical Framework

This chapter aims to describe the overall framework of this project by introducing and discussing the main theories and components we used to build our autoencoder based image enhancement technique.

3.1 Artificial Intelligent

Artificial intelligence also known as AI is one of the most important components of the technology today as it became a more powerful technique with the presentes of a number of intelligent algorithms such as Decision Trees, Artificial Neural Networks, Support Vector Machine and others. Artificial intelligence endeavors to build machines that are capable to perform too complex tasks that require human interaction. According to O'Carroll, (2017, October 24) there are three types of artificial intelligence: narrow artificial intelligence, general artificial intelligent and superintelligent artificial intelligence. The narrow artificial intelligence also denoted as weak AI is the one that has a narrow range of abilities and attempts to handle specific or single tasks such as email spam detection, facial recognition, music recommendation service and many others. Broad artificial intelligence is another type of artificial intelligence that works to achieve a wide range of tasks instead of only a single task and it works to mimic the way our brain works to achieve those tasks. while the superintelligent AI is the one who doesn't only attempt to mimic the way our brain works to process and handle various tasks; but instead of that it enables the machine to become more self-aware and superpass the capability of the human brain. However according to Girablog (2020, June 26) majority of the experts agree that the points of general and superintelligent artificial intelligences haven’t been reached yet by the technology today. Figure 3-1 illustrates the three different types of artificial intelligence.

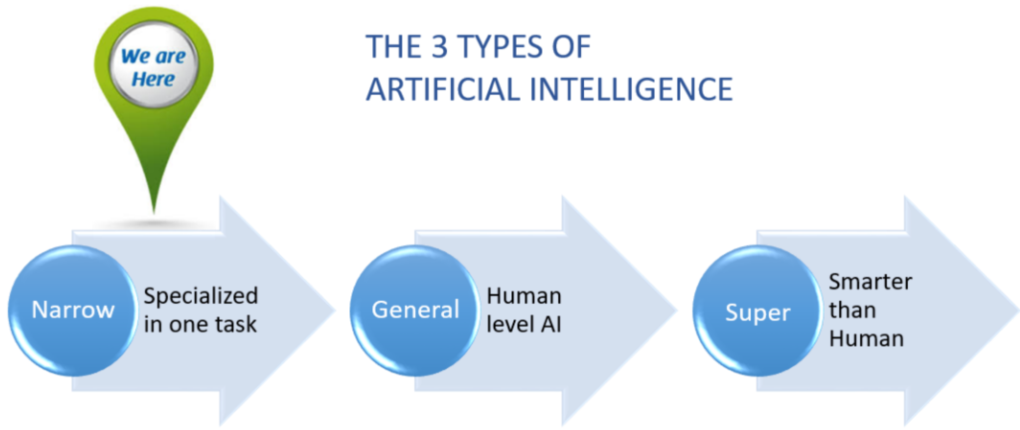


Figure 3-1. Types of artificial intelligence, Girablog (2020, June 26).

3.2 Machine Learning

Machine learning is a part of artificial intelligence (AI) that teaches the systems on how to perform specific tasks without being humanly programmed, by making the system learn from experiences using a bunch of data. Machine learning becomes today one of the most powerful techniques that enable us to make data driven decisions which result in the growth and prosperity of our business. It enables us to complete our task more efficiently and quickly.

According to Madza (2018, May 27) machine learning approaches are divided into three broad categories: supervised, unsupervised, and reinforcement learning. in supervised learning, we provide the machine with input and output data and make the machine continuously learn by experience to find a rule that maps the input to the output, while in unsupervised learning the machine is provided with only input data without any output (labels) and works to enable the machine to learn from experience to discover hidden pattern on data, the most common machine learning unsupervised learning method is clustering which aims to separate the input data into separate groups or categories. whereas the reinforcement learning system reacts in an uncertain, potentially complex environment and learns to achieve a certain task such as playing against an opponent or driving a vehicle. Figure 3-2 shows the machine learning components.

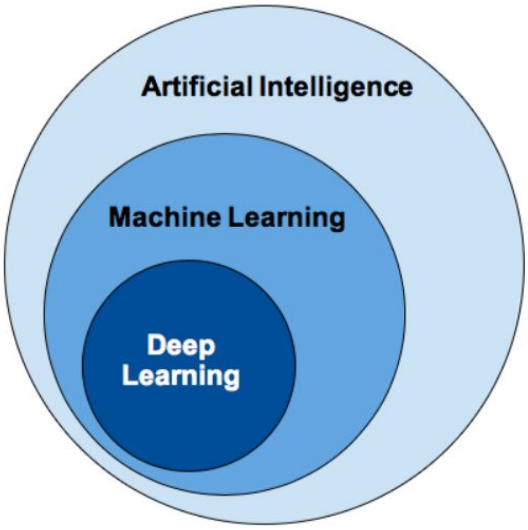


Figure 3-2 . Machine Learning Components, Wasicek, A. (2018, October 11),

3.3 Deep Learning

Deep learning is a broad branch of machine learning which is based on artificial neural networks that consist of multiple layers that aim to extract high-level features from ground truth input, for example, in image processing it detects and analyzes the presence of edges (low-level feature) to extract a digit (high-level feature). Deep learning tends to mimic the way our brain analyzes and processes the data to be used by the machine to achieve various tasks such as object detection, speech recognition, and decision making, deep learning divided into two broad categories which is: supervised and unsupervised. in supervised deep learning, the machine is provided with a bunch of input and output data and learns on how to map the input data to the output data. There are various type supervised deep learning models such as Classic Neural Networks (Multilayer Perceptrons), Convolutional Neural Networks (CNNs), and Recurrent Neural Networks (RNNs), whereas in unsupervised deep learning the machine is provided with only input data without output data and learns how to find hidden patterns and rules from the input data. There are different type unsupervised deep learning models such as Self-Organizing Maps (SOMs), Boltzmann Machines, and AutoEncoders. Figure 3-3 shows the differences between traditional machine learning and deep learning.

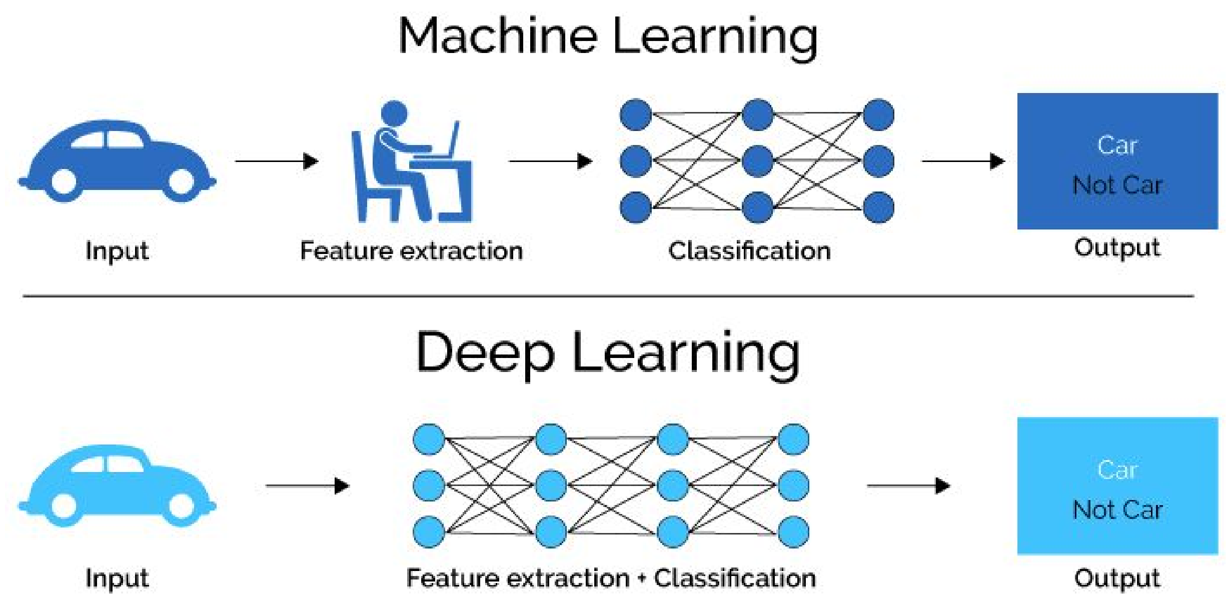


Fig. 3-3. Machine learning Vs Deep learning. Lawtomated. (2020, February 02).

3.4 Artificial Neural Networks

Artificial neural networks are one of the most important techniques used in machine learning. They endeavour to mimic the way our brain functions to process a large number of data. They were made of many layers including input layer, output layer and hidden layers. They mostly used to analyze or extract too complex patterns from a large number of data which may required thousands of programming lines. Neural networks techniques have been around since 1940 however they only on the few decades become a major part of machine learning, the reason of this is because their inclusion of a new property called "backpropagation" which is a property of a neural networks that allow them to adjust their hidden layer each iteration in an attempt of them to minimize the distance between the input data and the output it endeavors to achieve.There are many types of neural networks such as Classic Neural Networks (Multilayer Perceptrons), Convolutional Neural Networks (CNNs), and Recurrent Neural Networks (RNNs). Each different type of the neural networks work in a certain way that can be used for specific tasks. Neural networks can be applied to achieve complex tasks such as car automatic driving, handwriting recognition, human face recognition and moreover. Figure 3-4 illustrates different types of artificial neural networks.

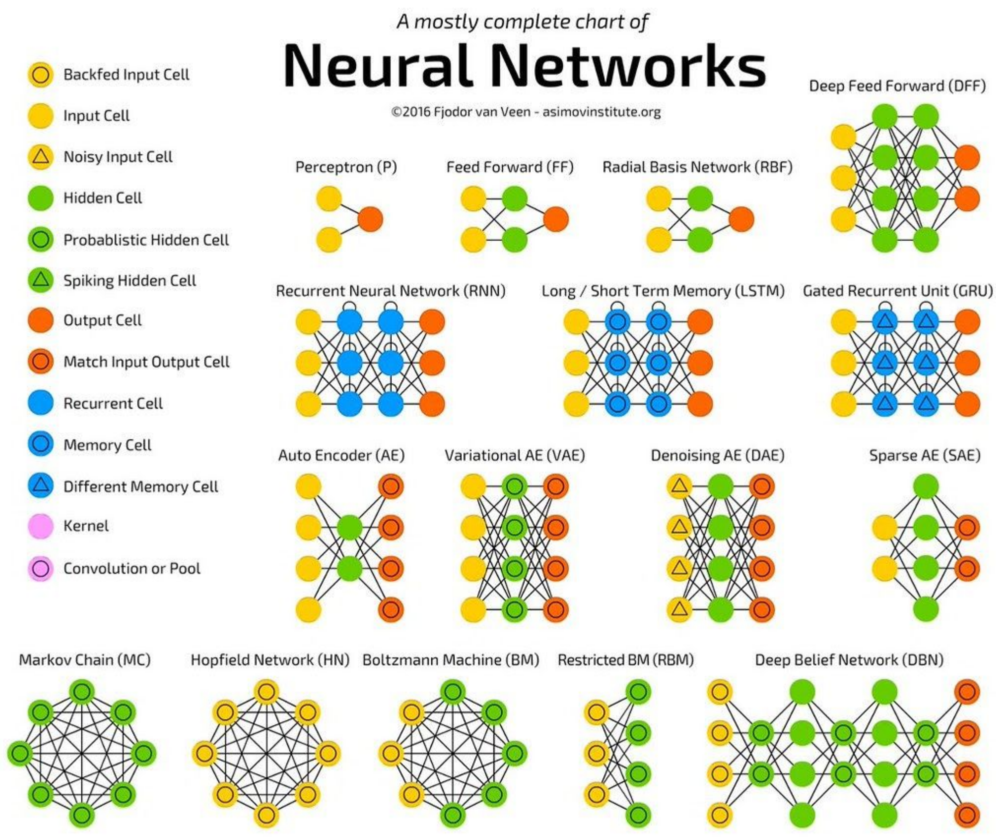


Figure. 3-4. Neural network different type architecture, Jen Stirrup. (2019, October 30)

3.5 Convolutional Neural Network

A convolutional neural network is a special type of neural network in which it contains convolutional layers. It is mainly designed to work for image processing, segmentation, classification, and other tasks. it consists of three major components: Convolutional layers, Pooling layers, and Fully connected layers. Below is a brief description of each layer.

3.5.1 Convolutional Layers

The first layer of the convolutional neural network is the convolutional layer. It consists of a set of the filters with weight and height smaller than the input image volume. it maintains the association between the input image pixels as each filter needs to be applied to every small portion of the input image. The dot products between the input and filter are computed at every spatial position to create what is called a feature map that summarizes the feature on the input image. In other words the convolutional layer is a mathematical function that takes two inputs such as an image matrix and a filter or kernel and performs the dot product between the two inputs in every small portion of the image to obtain a set image feature.

3.5.2 Pooling

The pooling layer is one of the main components of the convolutional neural network. It aims to reduce the complexity of the network by reducing the number of parameters the network has to learn. This reflects in increasing the overall accuracy and performance of the network as it reduces the chance of overfitting. According to Prabhu (2019, November 21) there are various type of polling such as max pooling, min pooling and average pooling. in max pooling each feature map produced from the previous convolution layer is replaced with the maximum within it, while the minimum pooling is to replace each feature map with the minimum value within it and the average pooling is to replace the feature map with average of all the values within it. Figure 3-5 illustrates the process of max pooling.

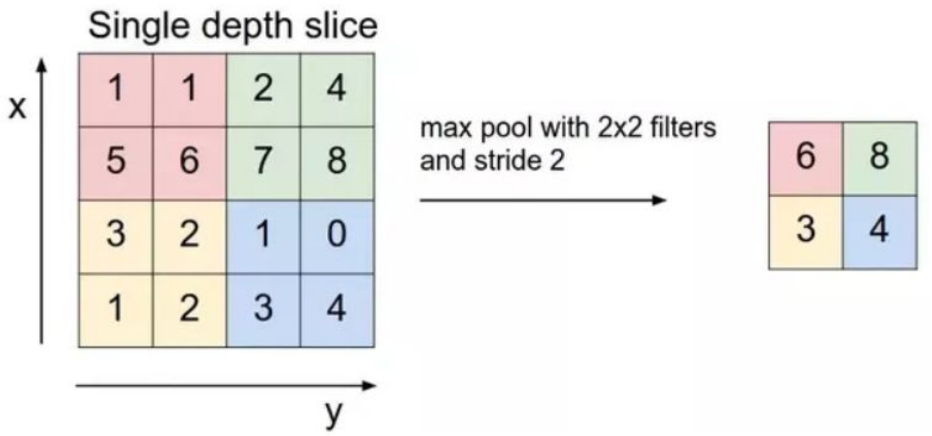


Figure. 3-5. The Pooling Layer, Prabhu. (2019, November 21)

3.5.3 Fully-Connected Layer

The fully-connected layer is one of the main building blocks of the convolutional neural network. it consists of a set of fully connected layers such that each node in one layer is connected to every activation unit of the following layer. The fully connected layer feeded by the output of the preceding convolutional and pooling layers to classify the image into a label. The fully-connected layer is considered the second time-consuming layer after the convolutional layer. The feature maps produced from the proceeding convolutional and pooling layers are flattened into a single vector of values, each value represents a probability. Each value is a probability that indicates that a certain feature belongs to a certain label or not. For example, if we have an image of a tree, the feature maps that represent things like leaves or trunks should have a high chance to be labeled as a tree. Figure 3-6
illustrates the structure of the fully-connected layer in convolutional neural networks.

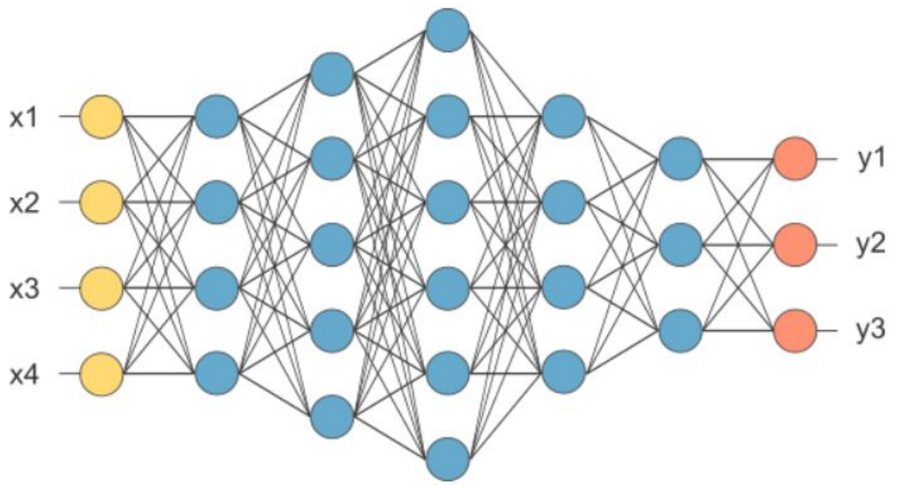


Figure 3-6. The fully connected layer Prabhu. (2019, November 21)

3.6 AutoEncoders

Autoencoder is a member of the family of artificial neural networks that works in an unsupervised approach. There are four main components of the autoencoder: Encoder, Bottleneck, Decoder and Reconstruction Loss. The function of the encoder is to receive the input data and the model starts to learn how to reduce the dimensions of input data and compress them into encoded representation. The Bottleneck contains the output of the encoder (the encoded representation) which is the lowest possible dimension of the input data. The data is then delivered to the decoder in which the model starts to learn how to reconstruct the encoded representation to produce output that is as close as possible to the original input. the output of the encoder can then be collected to evaluate the overall performance of the autoencoder using reconstruction loss method. The network architecture for autoencoders can vary between a simple FeedForward network, LSTM network or Convolutional Neural Network depending on the use case. The autoencoder can be used in many applications such as denoising (removing noise from an image) , Dimensionality reduction (reduce the dimensional space of the input data) and outlier detection (detect inconsistent data points on a dataset). Figure 3-7 illustrates the Autoencoder main components.

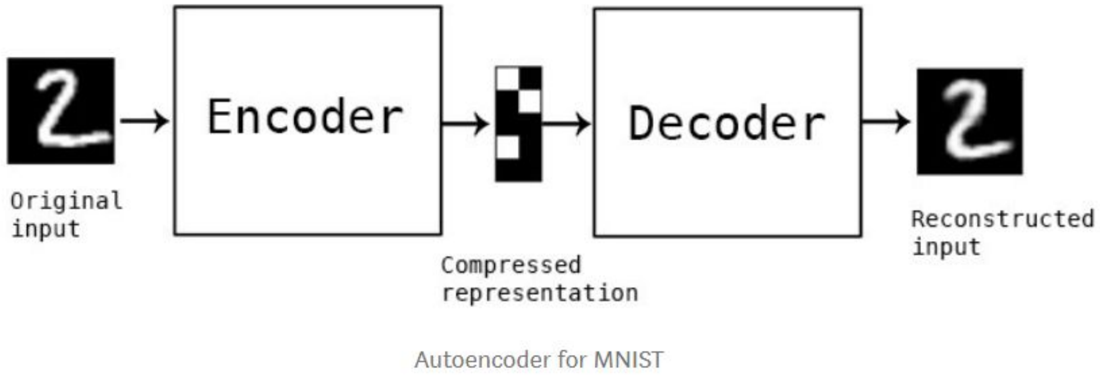


Fig. 3-7. Autoencoder Components, Badr, W. (2019, July 01)

3.7 Artificial Neural Network Training Process

Once the neural network structure has been built successfully the network then is ready to be trained. The neural network training process is an iterative learning process that can be divided into two broad approaches: supervised and unsupervised. In supervised learning the input data are associated with random weights and the weights will continually adjust each round based on the error calculated to measure distance between the input and output data. Finally the network will stop training when at catch the weights that best minimize the distance between the input data and output data. while in unsupervised learning the network provided with only input data without any outputs. the network itself needs to find a way that enables it to discover the hidden pattern from it or group similar groups together. This type of training is also called self-organization or adaptation. At the current time, unsupervised learning is a bit ambiguous. A neural network learning process is also called "connectionist learning," as the units of the network are connected to each other. one of the most important features of neural networks is their ability to classify patterns on which they have not been trained as well as to tolerate the noisy data. Neural network learning processes can be divided into two broad approaches supervised and unsupervised.

3.8 Optimizer

An optimizer is a method or algorithm that is used to repetitively change the neural network attributes such as the learning rage and weights in order to minimize the error loss. Optimizers enable the network to learn faster. The following are some of the most popular optimizers.

3.8.1 Adam

Adam stands for "Adaptive Moment Estimation", it was created by Jimmy Ba and Diederik Kingma in Their 2015 paper "Adam: A Method for Stochastic Optimization". Adam is an optimization method that can be used to change the networks weights iteratively based on training data, it can be considered a combination of stochastic Gradient Descent with momentum and RMSprop. It scales the learning rate like RMSprop by using the squared gradient and it uses the moving average of the gradient rather than using the gradient itself. There are many advantages of using Adam for non-convex problems such as it requires less computational power, easy to implement, appropriate for non-stationary objectives, and Invariant to diagonal rescale of the gradients.

3.8.2 SGD

SGD stands for Stochastic Gradient Descent, is an optimizer that works to optimize an objective function by using one stationary learning rate for all parameters during the training process. SGD can be considered as a stochastic approximation of gradient descent optimization, since it uses an estimation (computed from a random subset of the data) to substitute the actual gradient (computed from the whole dataset). The advantage of using SGD optimizer is that it can reduce the computational power for high-dimensional optimization problems.

3.9 Loss Function

The loss function is a function that endeavors to measure how well an algorithm models a dataset. if the predicted output is very far from the actual results, that means the loss function will hold a large number. loss function works together along with the optimization function to minimize the distance between the input and the output. According to Parmar, R. (2018, September 02) the loss functions can be divided into two broad categories: Classification losses and Regression losses. classification loss function is most suitable for classification problems such as color classification or handwritten digit classification. Some examples of classification loss include Cross Entropy Loss, Hinge Loss and Multi class SVM Loss while regression loss functions are most suitable regression problems such as predicting house price given its data. Some examples of regression loss functions are Mean Absolute Error, Mean Bias Error, and Mean Square Error The choice of which loss function to use is based on many factors such as the type of problem we are trying to solve as well as the type of machine learning algorithm chosen. Figure 3-8 illustrates on how the loss function works

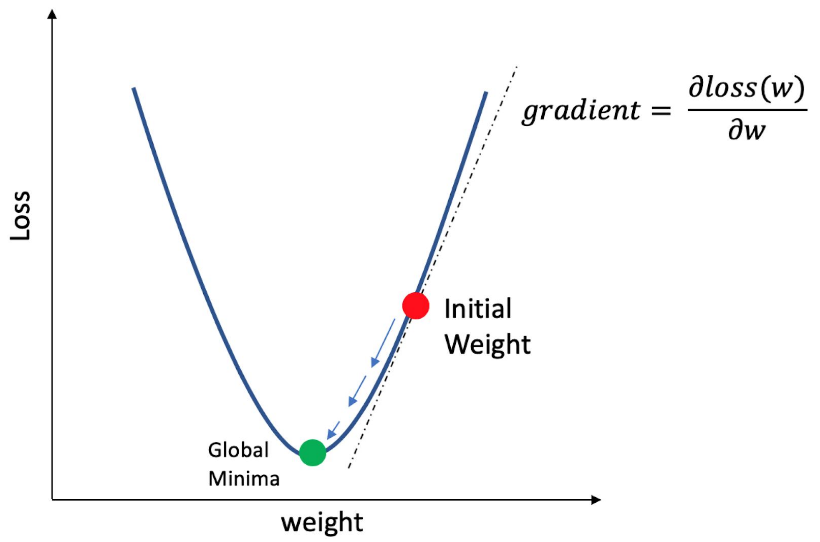


Figure 3-8. Loss Function, Rengasamy, Jafari, Rothwell, Chen & Figueredo (2020, January 28)

CHAPTER 4

Research Methodology

In this project we aim to propose an autoencoder based image enhancement technique that works to enhance and handle the low-quality images of three different types (Blurred, Low-light ,and Low-resolution). The methodology we are going to follow in this paper is divided into two phases: in the first phase we build our autoencoder based image enhancement model and apply a number of low-quality images of different type to it to generate a number of enhanced images, while in the second phase we evaluate our model using two different approaches: in the first approach we apply each pair of original and enhanced images to the pre-existing age and gender classification model developed by (Levi & Hassncer, 2015) and record the result for each pair respectively, while the second approach we used an objectives image quality measurements such as PSNR and SSIM to evaluate the quality of enhanced images, further explanation on that would be found on the evaluation section

4.1 Types of Low-Quality Images

4.1.1 Low Lighting Images Problem

One of the most common problems that cause degradation in image quality is to capture the image in low-light condition. Such images might have high noise levels with low dynamic ranges, the thing that can reduce the overall performance of the computer vision algorithms. In order for us to solve this problem and maintain the computer vision algorithm's accuracy and performance we have to make use of an image enhancement technique that can improve the image visibility.

4.1.2 Blurred Images Problem

Blurred images are considered a type of low-quality images. . Blurred refers to a visual effect that makes the edges of text or images appear fuzzy or out of focus. Blurred images are the results of several causes such as hand shaking, object movement or camera's autofocus problem, such images may affect the overall performance of the computer vision algorithms due to their high level of noise or low dynamic range.

4.1.3 Pixelated Images Problem

Pixlelated is another type of low-quality images. pixelated simply means that the image is represented with a small number of pixels. it causes the image to lose details as one pixel would represent a large portion of the image and that case the appearance of squares on the image. pixelated images can be a result of several factors such as trying to use a small file size to store the images, downloading the images from websites or taking a photo from a camera with poor sensor light, like blurred and low-light images pixelated images may affect the overall performance of the computer vision algorithms due to their high levels of nosy or low dynamic ranges.

4.2 Dataset

In this project we made use of the All_Age_Faces dataset as the main dataset to train our proposed autoencoder based image enhancement models. The reason for choosing this dataset is because its ideal number of images as will as the images resolutions and alignments. However, the limitation of ethnicity the dataset suffering from won't affect our model training process as the dataset being used to train the model to enhance the quality of different images without any interaction for the human ethnicity.

4.3 Low-quality Images Enhancement Model

Several methods have been proposed to address the low-quality images problems such as blurred, pixelated and low-lighting. One of the most powerful techniques that can be used to address the low-quality images is by using deep learning Autoencoder neural networks. Autoencoder is a type of unsupervised neural network that consists of two parts: Encoder and Decoder. The encoder is used to compress and reduce dimensionality level of the images so that only the important feature of the image are kept, the decoder then to be used to reconstruct the original images from the low dimensional space produced by the encoder. in this project we trained our autoencoder deep learning based model into three different type of images (Blurred, Pixelated ,and Low-light) separately.

4.4 Model Architecture

4.4.1 Encoder Architecture

An encoder is mainly used to extract the most strong features from the input data and compress and encode them into much smaller space. It has been built using different layers. The first layer of the encoder is the input layer which indicates the size of the input image as well as the number of color channels. in this project we used input images of size 256 * 256 with 3 color channels. The second layer of the encoder is the first convolutional layer. The function of the convolutional layers is to apply a set of different filters to the input image, to extract the strong feature from it, in order to reduce the dimensional space. They have been created using Conv2D function from keras with relu activation function, and regularizer.l1(10e-10) as activation_regularizer (used to regularize the output after the activation function). The first convolutional layer uses a set of 128 3 x 3 different filters.The second convolutional layer is applied to the output of the first convolutional layer with the same set of filters and settings as the preceding one. The purpose of the second convolutional layer is to strengthen the features we got from the first convolutional layer. After applying the second convolutional layer the input image will begin to descend into smaller space. so that we have to start slowly down scale the image in a much smaller space. and to do so we apply the fourth layer which uses the MaxPool2D function from keras (The default size of the MaxPool2D is (2,2). so it reduces the size of the output by a factor of 2) . The remaining layers are repetitive layers of the above processes with a different number of filters to extract the interesting features from different dimensional space. Finally we applied the last convolutional layer with the same settings as the previous convolutional layers but it passed 256 3 x 3 filters instead. Figure 4-1 shows the architecture of the Encoder.

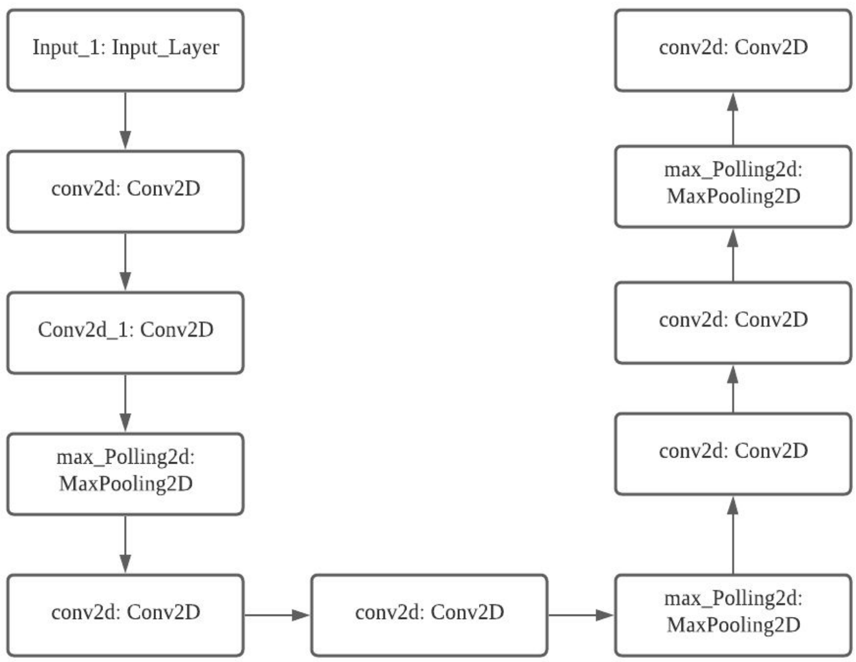


Figure 4-1. Encoder Architecture

4.4.2 Decoder Architecture

The doctor performs the deconvolution function which is the reverse function to the encoder convolution function. It used to reconstruct an output image from a low dimensional encoded space. The first layer of the decoder is to receive the output from the encoder and upscales it using the upsampling2D keras function. The second layer is a convolutional layer that applies a set of 342 3 x 3 filters to the output of the first layer to extract the interesting features from it. The third layer is also a convolutional layer with the same number of filters of the second layer. and the purpose of it is to strengthen the features we got from the second layer. The fourth layer is the merge layer which performs an add operation with the corresponding layer in our encoder, it enables the network to not lose information by going deeper (having a high number of layers). The remaining layers are repetitive layers of the above processes with a different number of filters to extract the interesting features from different dimensional space. Finally, to make the final image to be with the correct number of color channels which is 3 RGP. Figure 4-2 shows the architecture of the Decoder.

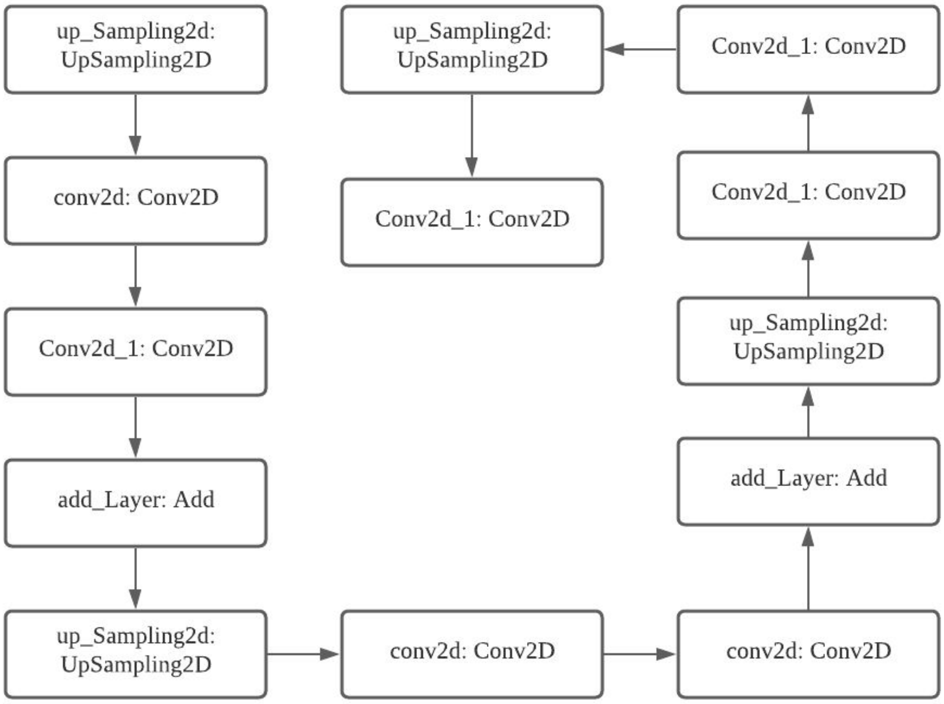


Figure 4-2. Decoder Architecture

4.4.3 Autoencoder Summary

```
Model: "model"
__________________________________________________________________________________________________
Layer (type)                   Output Shape         Param #     Connected to
==================================================================================================
input_1 (InputLayer)           [(None, 256, 256, 3) 0
__________________________________________________________________________________________________
conv2d (Conv2D)                (None, 256, 256, 128 3584        input_1[0][0]
__________________________________________________________________________________________________
conv2d_1 (Conv2D)              (None, 256, 256, 128 147584      conv2d[0][0]
__________________________________________________________________________________________________
max_pooling2d (MaxPooling2D)   (None, 128, 128, 128 0           conv2d_1[0][0]
__________________________________________________________________________________________________
conv2d_2 (Conv2D)              (None, 128, 128, 256 295168      max_pooling2d[0][0]
__________________________________________________________________________________________________
conv2d_3 (Conv2D)              (None, 128, 128, 256 590080      conv2d_2[0][0]
__________________________________________________________________________________________________
max_pooling2d_1 (MaxPooling2D) (None, 64, 64, 256)  0           conv2d_3[0][0]
__________________________________________________________________________________________________
conv2d_4 (Conv2D)              (None, 64, 64, 342)  788310      max_pooling2d_1[0][0]
__________________________________________________________________________________________________
conv2d_5 (Conv2D)              (None, 64, 64, 342)  1053018     conv2d_4[0][0]
__________________________________________________________________________________________________
max_pooling2d_2 (MaxPooling2D) (None, 32, 32, 342)  0           conv2d_5[0][0]
__________________________________________________________________________________________________
conv2d_6 (Conv2D)              (None, 32, 32, 512)  1576448     max_pooling2d_2[0][0]
__________________________________________________________________________________________________
up_sampling2d (UpSampling2D)   (None, 64, 64, 512)  0           conv2d_6[0][0]
__________________________________________________________________________________________________
conv2d_7 (Conv2D)              (None, 64, 64, 342)  1576278     up_sampling2d[0][0]
__________________________________________________________________________________________________
conv2d_8 (Conv2D)              (None, 64, 64, 342)  1053018     conv2d_7[0][0]
__________________________________________________________________________________________________
add (Add)                      (None, 64, 64, 342)  0           conv2d_5[0][0]
                                                                 conv2d_8[0][0]
__________________________________________________________________________________________________
up_sampling2d_1 (UpSampling2D) (None, 128, 128, 342 0           add[0][0]
__________________________________________________________________________________________________
conv2d_9 (Conv2D)              (None, 128, 128, 256 788224      up_sampling2d_1[0][0]
__________________________________________________________________________________________________
conv2d_10 (Conv2D)             (None, 128, 128, 256 590080      conv2d_9[0][0]
__________________________________________________________________________________________________
add_1 (Add)                    (None, 128, 128, 256 0           conv2d_3[0][0]
                                                                 conv2d_10[0][0]
__________________________________________________________________________________________________
up_sampling2d_2 (UpSampling2D) (None, 256, 256, 256 0           add_1[0][0]
__________________________________________________________________________________________________
conv2d_11 (Conv2D)             (None, 256, 256, 128 295040      up_sampling2d_2[0][0]
__________________________________________________________________________________________________
conv2d_12 (Conv2D)             (None, 256, 256, 128 147584      conv2d_11[0][0]
__________________________________________________________________________________________________
add_2 (Add)                    (None, 256, 256, 128 0           conv2d_1[0][0]
                                                                 conv2d_12[0][0]
__________________________________________________________________________________________________
conv2d_13 (Conv2D)             (None, 256, 256, 3)  3459        add_2[0][0]
==================================================================================================
Total params: 8,907,875
Trainable params: 8,907,875
Non-trainable params: 0
```

Figure 4-3. Autoencoder Summary

4.5 Model Training

4.5.1 The Loss Function

The mean square error is a loss function that measures the square average distance between the input data and predicted output. The mean square error tells us how far a set of points is far from a regression line. It achieves this by measuring the distances between the regression line to the set point and square them. The purpose of the squaring is to convert the negative sign to positive sign. The mean square error loss function aims to penalize those ponts with large differences from the regression line by giving them more weights. The smaller the mean square error value the more close input data to the prediction output. In this project we made use of the regression loss Mean Squared Error provided by keras as a mean loss function to train our proposed autoencoder based enhancement model. Figure 4-4 shows the formula of the Mean Squared Error loss function

The mean Square Error Formula

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i^2)$$

MSE = mean squared error

n = number of data points

Y_i = observed values

$\hat{Y}_i$ = predicted values

Figure 4-4. The Mean Square Error Loss Function Formula, Stephanie (2020, August 03).

4.5.2 Optimizer

Adadelta optimizer is a stochastic gradient descent method that aims to minimize its aggressive Adadelta optimizer aims to handle two problems which is "the constantly degradation of the learning rate during the training process " and "The necessity of global learning rate selected manually" based on adaptive learning rate per dimension. Adedelta is an improvement of the Adagrad optimizer that adjusts the learning rate based on a movement of the gradient updates window. rather than accumulating all the previous gradients, this means that even if many updates have been done Adadelta is able for continuous learning. Using Adadelta we don't have to initial the learning rate.

In this project we decided to make use of Adadelta optimizer for the purpose of training our model due to its ability for continuous learning even if many updates have been made.

4.5.3 Image Processing

Training an artificial neural neural network requires the availability of a ground truth dataset as well as training dataset. In our project the ground truth dataset images stand for the high-quality images while the training dataset images stand for the low-quality images. In order for us to provide three different types of low-quality images (Blurred, Pixelated, and Low-Lighting) the dataset " All-Age-Faces" have been undergone into three different image processing functions that convert the high-quality image from the dataset into low-quality image. Each function produced a different type of low-quality image (Low light, Pixelated and Blurred), each type of them is then used to train our autoencoder deep learning based model. The below paragraphs is to describe how the image processing functions works:

4.5.3.1 Blurred Image Processing

The main objective of the blurred image processing function is to convert a dataset of high-quality images to a dataset of low-quality of different blur levels. The blurred function accepts a single image and it calculates a random numbers "n" and "n1" and based on these numbers it chooses the type of python blurred function it should use as well as the amount of blur it should apply to the image. The reason for this step is to avoid overfitting situations during the training process. The blurred image processing uses three different types of blur predefined python blurred functions which is: GaussianBlur(), medianBlur(), and the uniform weight blur() function. The Gaussian Blur uses the gaussian function it gives the center pixel the higher weight and the weight will gradually reduce for surrounding pixels, it blur an image by computing the average value of all the pixels under the kernel window and replace the center pixel with the average computed value. in order to use the GaussianBlur function we should pass to it the input image as well as the weight and height of the filter window and the standard deviation in the X and Y directions, in the other hand, the medianBlur sort all the pixels under the kernel window in descending order to identify the median value of them and then replace the center pixel value with the median value it computed the medianBlur function takes the input image as will as the size of the kernel which must be a positive and odd number. while the uniform weight blur uses a uniform weight which is 1 for all the pixels under the kernel window, calculates the average value and replaces the center pixel value with the average computed value. The blur function takes the input image as well as the kernel size.

Gaussian Blur 3x3 Kernel

$$\frac{1}{16}\begin{array}{|c|c|c|}\hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline\end{array}$$

Figure 4-5. Gaussian Blur 3x3 Kernel

Median Blur Process

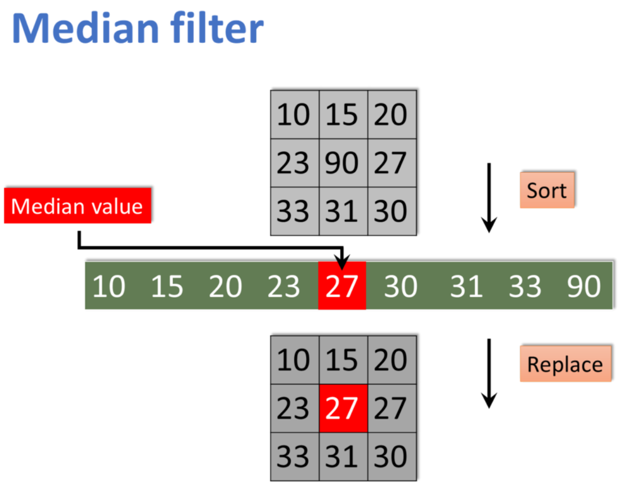


Figure 4-6. Median Blur. Datahacker (2020, February 29).

Uniform Weight Blur 3x3 Kernel

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Figure 4-7. Uniform Weight Blur, Boricha, V. (2018, April 18)

Blurred Image Processing

Figure 4-8. Blurred Image Processing Function Output

4.5.3.2 Pixelated Image Processing

The pixelating image processing function endeavors to generate a new dataset consisting of images with different levels of resolution. The pixelating function accepts a single image and it calculates a random number "n" based on the random number it calculates it chose the resolution level it should pixelate the image to. The reason for this step is to avoid overfitting problems during the training process. It is done by using the predefined python function cv2.resize(). The method accepts a single image as well as two numerical parameters indicating the weight and height of the new image it should generate. The pixelated function uses five different levels to pixelate the images finally it resizes the images to 256 x 256 the uniform image size of all the images. Figure 4-9 shows the five different levels we used to generate the low-resolution dataset .

Pixelating Image Processing Output

Figure 4-9. Pixelating Processing Function Output

4.5.3.3 Low-light Image Processing

The image processing low_lighting function aims to generate a new dataset consisting of low-lighting images. The function accepts a single image and it calculates a random number "n" based on the random number it calculates it chose the level of the darkness it should apply to the image. The reason for this step is to avoid overfitting problems during the training process. The low_lighting function applies the predefined python gammaCorrection() function to the images. The gamma correction function aims to adjust the luminance level for those images with low brightness caused mostly by the capturing device. it maps the image luminance values which are mostly between 0 and 1 to a new range of values that could enhance the images brightness. In our project we used the gamma correction function to map the image luminance values to new ranges that brought more darken images. The gamma correction taks the input image as parameter as well as a numerical value indicating gamma level. Figures 4-10 & 4-11 illustrate how the gamma correction function and the outputs of the low_lighting function respectively.

Gamma Correction Process

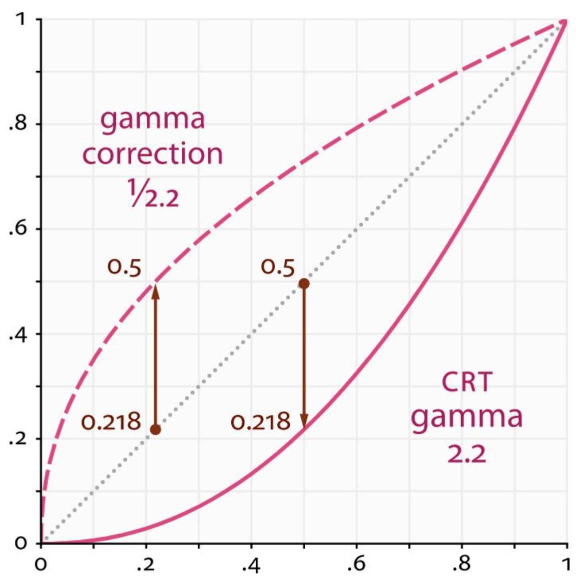


Figure 4-10. Xah Lee. (2010, February 24) Gamma Correction

Low-Light Image Processing Output

Figure 4-11. Low-light Function Output

CHAPTER 5

Evaluation

The evaluation process aims to evaluate the performance and accuracy of our image enhancement autoencoder based model, it helps us to identify how well we achieved our goal and what are the points of weakness that need further improvement. in this project we evaluate our model using two different approaches: Application evaluation process and image quality evaluation process : The application evaluation process aims to evaluate the performance of the pre-existing Age and gender classification model developed by (Levi & Hassncer, 2015) in both type of images : original and enhanced, while the second evaluation process aims to evaluate image quality using objective image quality measurements such as PSNR and SSIM.

5.1 Application Based Evaluation

The application evaluation process is the process of evaluating the performance of the pre-existing Age and gender classification model developed by (Levi & Hassncer, 2015) before and after applying our proposed Image enhancement techniques . It enables us to measure the improvement of the pre-existing Age and gender classification model before and after applying our proposed image enhancement technique.

5.1.1 Steps to Conduct the Evaluation

1- Randomly select up to 100 different types low-quality images (Pixelated, Low-light ,and Blurred images) to form the original images dataset

2- The 100 low-quality image have been subject to the enhancement process using our proposed image enhancement auto-encoder based technique to generate another 100 enhanced images dataset called enhanced image

3- Applying each pair of original and enhanced images into the pre-existing Age and gender classification and recording the result of each image respectively.

4- Gathering the results and computing the accuracy of each dataset individually

Figures: 5-1, 5-2, 5-3 show the process of conducting the application evaluation process using different types of low-quality images.

Application Evaluation process using Low-light images

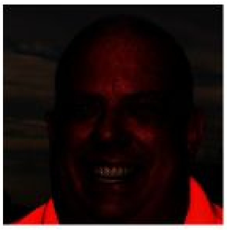
No	Image Before Applying the mode	Image After Apply Model	Actual Age and Gender	Result before apply the model	Result after Applying the model	Correct/ Wrong
1			Male (25 - 32)	Male (8 - 12)	Male (25 - 32)	correct
2			Male (25 - 32)	Male (8 - 12)	Male (25 - 32)	Correct
3			Male (60 - 100)	Female (25 - 32)	Female (38 - 43)	Wrong
4			Female (4 - 6)	Male (4 - 6)	Female (4 - 6)	Correct

Figure 5-1: Application Evaluation Sample Results Using Low-light Images

Application Evaluation Process using Blurred Images

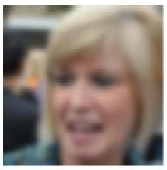
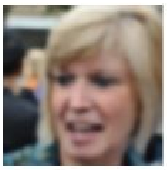
No	Image Before Applying the model	Image After Apply Model	Actual Age and Gender	Result before apply the model	Result after Applying the model	Correct/ Wrong
1			Male (25 - 32)	Female (8 - 12)	Male (25 - 32)	Correct
2			Male (25 - 32)	Male (38 - 43)	Male (25 - 32)	Correct
3			Female (48 - 53)	Male (4 - 6)	Female (4-6)	Wrong
4			Female (25 - 32)	Male (25 - 32)	Male (25 - 32)	Wrong

Figure 5-2: Application Evaluation Sample Results Using Blurred Images

Application Evaluation Process using Pixelated Images

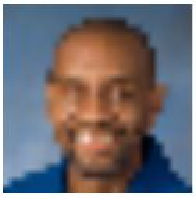
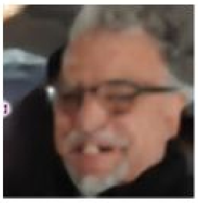
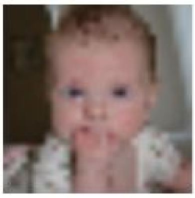
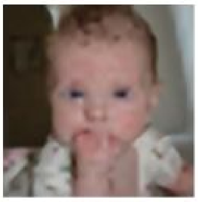
No	Image Before Applying the mode	Image After Apply Model	Actual Age and Gender	Result before apply the model	Result after Applying the model	Correct/ Wrong
1			Female (25-32)	Female (8-12)	Female (8-12)	Wrong
2			Male (25-32)	Male (25-32)	Male (25-32)	Correct
3			Male (6-100)	Not-able to detect the face	Not-able to detect the face	Wrong
4			Male (4 - 6)	Male (4 - 6)	Male (4-6)	correct

Figure 5-3: Application Evaluation Sample Results Using Low-resolution Images

5.1.3 Application Evaluation Process Result

The application evaluation process has been done by collecting up to 100 low-quality images equally distributed of different type such that 33 blurred images , 34 low-light images and 33 pixelated images, the images then were exposed to a quality enhancement process by our proposed aut-encoder based enhancement technique to generate 100 enhanced images. The original images along with the enhanced images were then applied to the pre-existing Age and gender classification model. The table below shows the test results of each type of the low-quality images, each cell value represents the number of correctly labeled images for the corresponding low-quality type and dataset.

5.1.3.1 Application Evaluation Table of Results

Table 5-1: Application Evaluation Table of Result

No	Dataset	Low-Light	Blurred	Pixelated
1	Original images	(3/34)%	(5/33)%	(7/33)%
3	Enhanced images	(22/34)%	(19/33)%	(9/33)%

5.2 Image Quality Evaluation Process

The second evaluation process is to measure the improvement of the images quality before and after applying our proposed image enhancement aut-enchoder based technique. Several objective image enhancement techniques are available for use. In this project we make use of two different objective image quality measurements: PSNR and SSIM. The PSNR stands for peak signal-to-noise ratio.The PSNR aims to measure the quality between two images. In our case it used to measure the quality between the original image and the enhanced image. The higher the PSNR value the better the quality of the enhanced images, while the SSIM stands for structural similarity index measure. It is a technique used to measure the quality of the images and videos. The result of the SSIM indicates how similar two images are. Figures 5-4 & 5-5 shows the formula used by PSNR and SSIM respectively.

$$MSE(m) = \frac{1}{N} \sum_{i,j} (Y_{out}(i,j,m) - Y_{in}(i,j,m))^2$$

$$PSNR(m) = 10\log_{10}\left(\frac{(2^{B-1})^2}{MSE(m)}\right)$$

MSE: mean square error
N: number of data points
Y_{out}: predicted values
Y_{in}: observed values
PSNR: peak signal-to-noise ratio

Figure 5-4. PSNR Formula, Pantech. (2017, September 30)

$$\mathrm{SSIM}(x,y) = \frac{(2\mu_x\mu_y + C_1) + (2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}$$

u

SSIM: structural similarity index measure
μ_x: local mean of first image
μ_y:local mean of the second image
σ_x: local standard deviation of the first image
σ_y: local standard deviation of the second image
σ_{xy}: cross-covariance for images
C_1, C2: constants

Figure 5-5. SSIM Formula, Mamun, I. (2019, February 12).

5.2.1 Steps to Conduct the Evaluation :

1- Randomly select 150 high quality human face images that form the original dataset

2- Expose the original dataset to the image enhancement process to generate another 150 images of low-quality images of different type (Blurred, Low-light, and Low-resolution) (50 for each type)

3- Expose the 150 low quality images to our proposed image enhancement auto-encoder based technique to generate another 150 enhanced images that formed the enhanced dataset

4- Apply each pair of original and enhanced images to the PSNR and SSIM python functions

5- Record the result for each pair respectively

6- Find the average PSNR and SSIM values

5.2.2 Image Quality Evaluation Results

Table 5-2: Image Quality Evaluation Result

No	Test Name	Low-light Images	Blurred Images	Pixelated Images
1	PSNR	27.88	31.17	30.92
2	SSIM	0.56	0.68	0.67

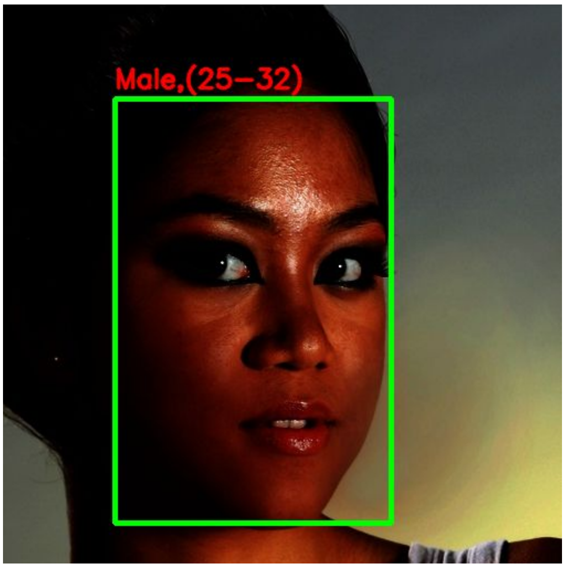
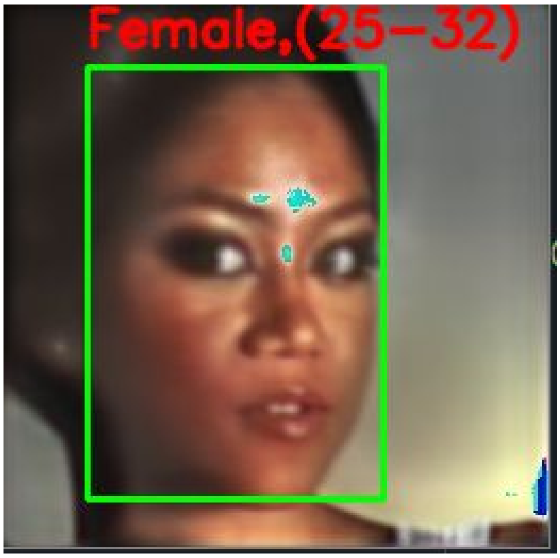
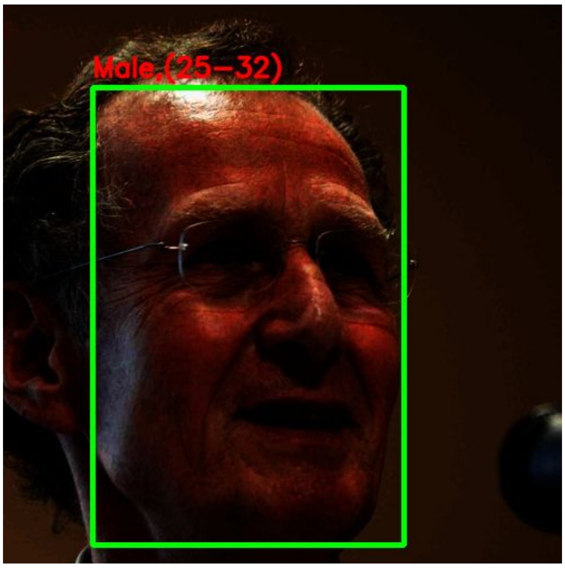
Result

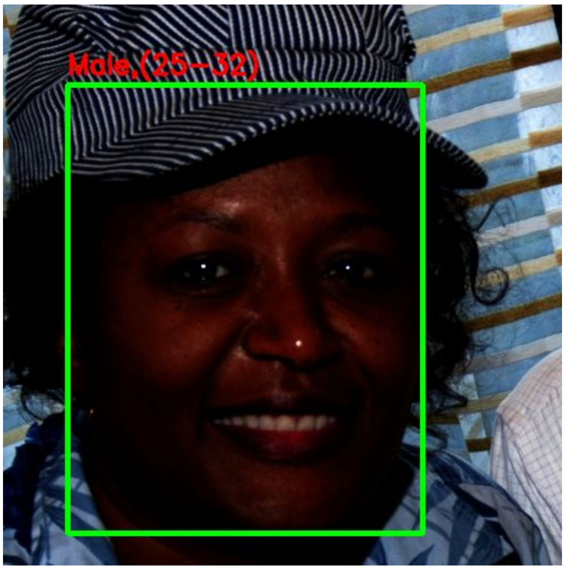
A sample of results of this project is shown on tables: 5-2, 5-3 and 5-4.

From table tables: 5-2, 5-3 and 5-4 we can see that applying our proposed image enhancement autoencoder based technique brought a significant improvement for the pre-existing age end gender classification model in two types of low-quality images: Low-light and Blurred. however the result shows a low improvement on the low-quality pixelated images, due to that future plan have been set for further investigation and analysis.

Low-quality Blurred Images Result

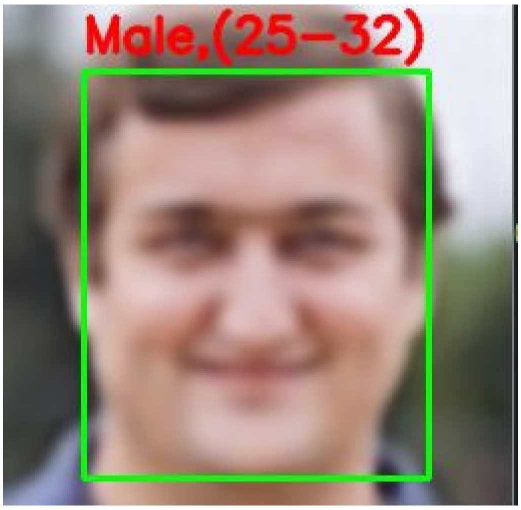
Table 5-3: Results in Low-quality low-light Image

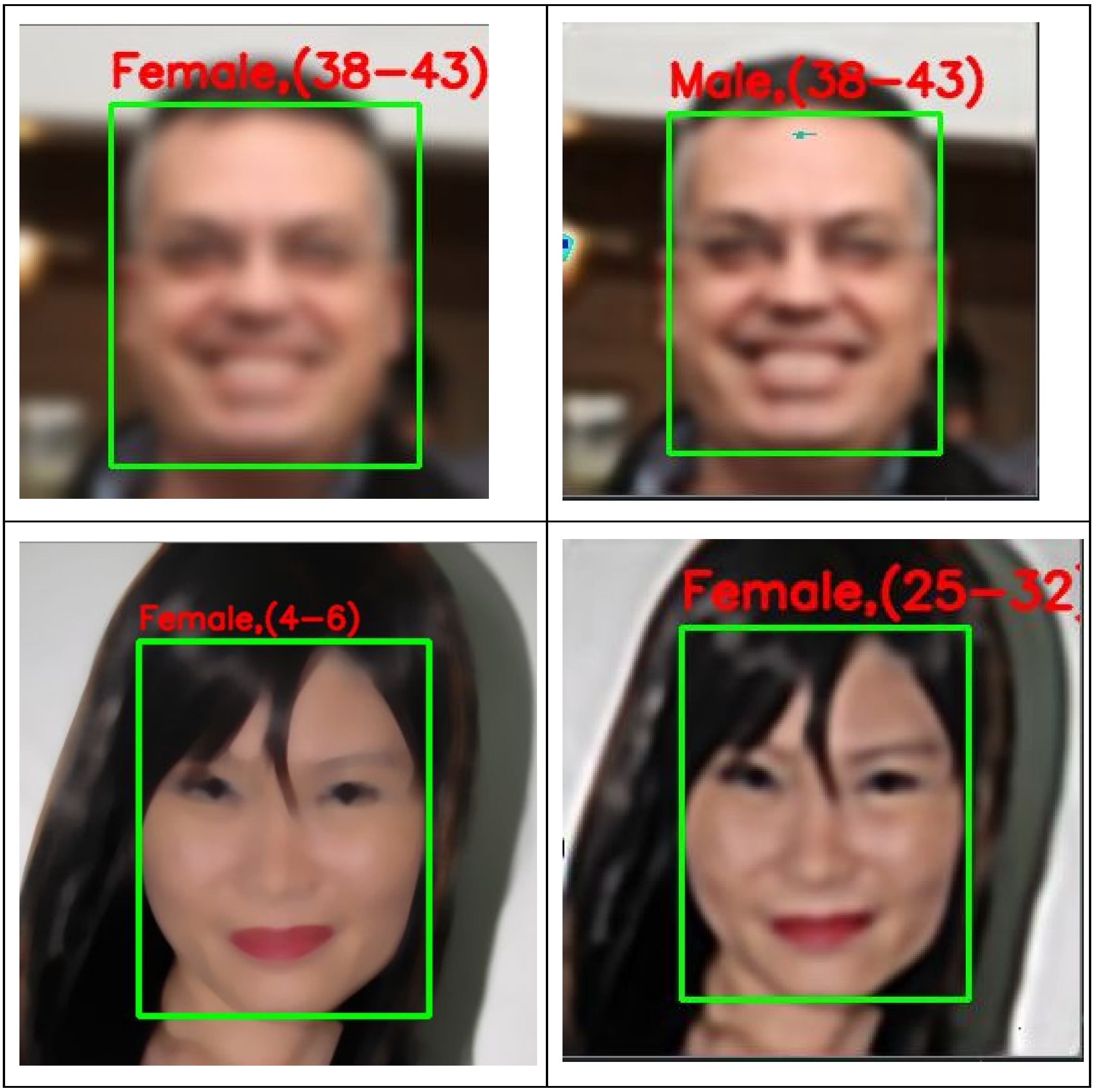
	
	

Low-quality Blurred Images Result

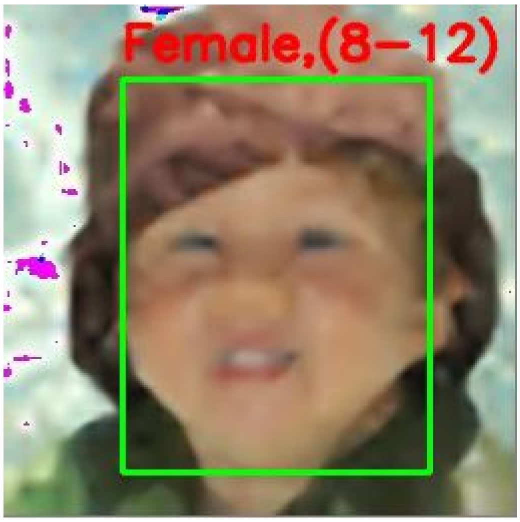
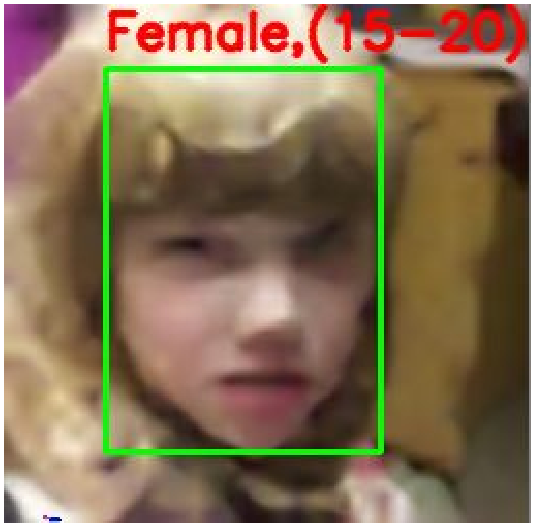
Table 5-4: Result in Low-quality Blurred Images

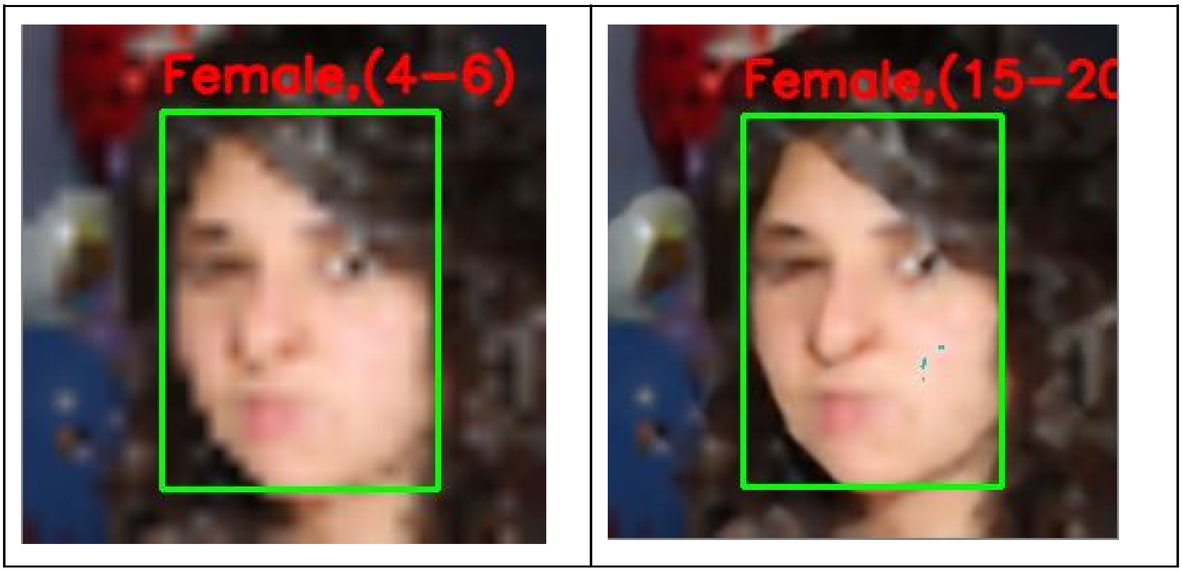
Original Images	Enhanced images
 	


Female,(38-43)
Male,(38-43)
Female,(4-6)
Female,(25-32

Low-quality Pixelated Images Result

Table 5-5: Result in Low-quality Pixleted Images

Original Image	Enhanced Image
	
	


Female,(4-6)
Female,(15-2

CHAPTER 6

Future Plan

Project Part 2 - Future Plan

Table 6-1: Future Plan

Week	Date	Tasks	Details
1	23 Nov - 29 Nov	Project API Design	Python based API to be designed for this project
2	30 Nov - 6 Dec	Project API Implementation	Python based API to be implemented during this week
3	7 Dec - 13 Dec	Project API Testing	Python based API to be tested b the end of this week
4	14 Dec - 20 Dec	Investigation and analysis for model improvement	Investigating different approach for the purpose of model enhancement
5	21 Dec - 27 Dec	Investigation and analysis for model improvement	Investigating different approach for the purpose of model enhancement
6	28 Dec - 3 Jan	Investigation and analysis for model improvement	Investigating different approach for the purpose of model enhancement
7	4 Jan - 10 Jan	Project general comparison with the state of art methods	Conducting a general comparison between our proposed method and a number of state of art methods
8	11 Jan - 17 Jan	Project general comparison with the state of art methods	Conducting a general comparison between our proposed method and a number of state of art methods
9	18 Jan - 24 Jan	Project general	Conducting a general

		comparison with the state of art methods	comparison between our proposed method and a number of state of art methods
10	25 Jan - 31 Jan	Final revision of the project	conducting an overall review of the project during this week
11	1 Feb - 7 Feb	Project Submission	Project to be submitted by the end of this week

References

1. G. Levi and T. Hassncer, "Age and gender classification using convolutional neural networks," 2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Boston, MA, 2015, pp. 34-42, doi: 10.1109/CVPRW.2015.7301352.

2. Eidinger, E., Enbar, R. and Hassner, T., 2014. Age and gender estimation of unfiltered faces. *IEEE Transactions on Information Forensics and Security*, *9*(12), pp.2170-2179.

3. Duan, M., Li, K., Yang, C. and Li, K., 2018. A hybrid deep learning CNN–ELM for age and gender classification. *Neurocomputing*, *275*, pp.448-461.

4. Kwon, Y.H. and da Vitoria Lobo, N., 1999. Age classification from facial images. *Computer vision and image understanding*, *74*(1), pp.1-21.

5. Reid, D.A., Samangooei, S., Chen, C., Nixon, M.S. and Ross, A., 2013. Soft biometrics for surveillance: an overview. In *Handbook of statistics* (Vol. 31, pp. 327-352). Elsevier.

6. Guo, G., Fu, Y., Dyer, C.R. and Huang, T.S., 2008. Image-based human age estimation by manifold learning and locally adjusted robust regression. *IEEE Transactions on Image Processing*, *17*(7), pp.1178-1188.

7. O'toole, A.J., Vetter, T., Troje, N.F. and Bülthoff, H.H., 1997. Sex classification is better with three-dimensional head structure than with image intensity information. *Perception*, *26*(1), pp.75-84.

8. Makinen, E. and Raisamo, R., 2008. Evaluation of gender classification methods with automatically detected and aligned faces. *IEEE transactions on pattern analysis and machine intelligence*, *30*(3), pp.541-547.

9. Agbo-Ajala, O. and Viriri, S., 2020. Deeply Learned Classifiers for Age and Gender Predictions of Unfiltered Faces. *The Scientific World Journal*, *2020*.

10. Bekhouche, S.E., Ouafi, A., Benlamoudi, A., Taleb-Ahmed, A. and Hadid, A., 2015, May. Facial age estimation and gender classification using multi level local phase quantization. In *2015 3rd International Conference on Control, Engineering & Information Technology (CEIT)* (pp. 1-4). IEEE.

11. Fang, J., Yuan, Y., Lu, X. and Feng, Y., 2019. Muti-stage learning for gender and age prediction. *Neurocomputing*, *334*, pp.114-124.

12. Bukar, A.M., Ugail, H. and Connah, D., 2016. Automatic age and gender classification using supervised appearance model. *Journal of Electronic Imaging*, *25*(6), p.061605.

13. Fazl-Ersi, E., Mousa-Pasandi, M.E., Laganiere, R. and Awad, M., 2014, October. Age and gender recognition using informative features of various types. In *2014 IEEE International Conference on Image Processing (ICIP)* (pp. 5891-5895). IEEE.

14. Ullah, I., Hussain, M., Aboalsamh, H., Muhammad, G., Mirza, A.M. and Bebis, G., 2012, July. Gender recognition from face images with dyadic wavelet transform and

local binary pattern. In *International Symposium on Visual Computing* (pp. 409-419). Springer, Berlin, Heidelberg.

15. Azarmehr, R., Laganiere, R., Lee, W.S., Xu, C. and Laroche, D., 2015. Real-time embedded age and gender classification in unconstrained video. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops* (pp. 57-65).

16. Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R.R., 2012. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.

17. Tal Hassner, Shai Harel*, Eran Paz* and Roee Enbar. *Effective Face Frontalization in Unconstrained Images.* IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Boston, 2015

18. Ricanek, K., & Tesafaye, T. (2006, April). Morph: A longitudinal image database of normal adult age-progression. In *7th International Conference on Automatic Face and Gesture Recognition (FGR06)* (pp. 341-345). IEEE.

19. Wikipedia contributors. (2020, September 26). Machine learning. In *Wikipedia, The Free Encyclopedia*. Retrieved 03:09, September 27, 2020, from https://en.wikipedia.org/w/index.php?title=Machine_learning&oldid=980484021

20. Rasmus, R, Radu ,T, Luc ,G, (2015). IMDB-WIKI – 500k+ face images with age and gender labels. Retrieved October 18, from https://data.vision.ee.ethz.ch/cvl/rrothe/imdb-wiki/

21. Tero, K, Samuli, L, Timo, A, (2014). Flickr-Faces-HQ Dataset (FFHQ). Retrieved October 18, 2020, from https://github.com/NVlabs/ffhq-dataset

22. Madza, A. (2018, May 27). Machine Learning: An Explanation for the Novice. Retrieved October 18, 2020, from https://medium.com/@The_Mad_Zaafa/machine-learning-an-explanation-for-the-novice-eb22d6515a32

23. Wasicek, A. (2018, October 11). Artificial Intelligence vs. Machine Learning vs. Deep Learning: What's the Difference? Retrieved October 18, 2020, from https://www.sumologic.com/blog/machine-learning-deep-learning/

24. Lawtomated. (2020, February 02). A.I. technical - Machine Learning vs. Deep Learning. Retrieved October 18, 2020, from https://lawtomated.com/a-i-technical-machine-vs-deep-learning/

25. Jen Stirrup. (2019, October 30). How do you evaluate the performance of a Neural Network? Focus on AzureML. Retrieved October 18, 2020, from https://jenstirrup.com/2017/08/01/how-do-you-evaluate-the-performance-of-a-neural-network-focus-on-azureml/

26. O'Carroll, B. (2017, October 24). What are the 3 types of AI? A guide to narrow, general, and super artificial intelligence. Retrieved October 18, 2020, from https://codebots.com/artificial-intelligence/the-3-types-of-ai-is-the-third-even-possible

27. Girablog. (2020, June 26). What Are The 3 Types Of AI? Retrieved October 18, 2020, from https://www.deccanherald.com/brandspot/pr-spot/what-are-the-3-types-of-ai-853275.html

28. Prabhu. (2019, November 21). Understanding of Convolutional Neural Network (CNN) - Deep Learning. Retrieved October 18, 2020, from https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148

29. Badr, W. (2019, July 01). Auto-Encoder: What Is It? And What Is It Used For? (Part 1). Retrieved October 18, 2020, from https://towardsdatascience.com/auto-encoder-what-is-it-and-what-is-it-used-for-part-1-3e5c6f017726

30. Rengasamy, D., Jafari, M., Rothwell, B., Chen, X., & Figueredo, G. (2020, January 28). Deep Learning with Dynamically Weighted Loss Function for Sensor-Based Prognostics and Health Management. Retrieved October 18, 2020, from https://www.mdpi.com/1424-8220/20/3/723
31. Parmar, R. (2018, September 02). Common Loss functions in machine learning. Retrieved October 18, 2020, from https://towardsdatascience.com/common-loss-functions-in-machine-learning-46af0ffc4d23
32. Stephanie. (2020, August 03). Mean Squared Error: Definition and Example. Retrieved October 18, 2020, from https://www.statisticshowto.com/mean-squared-error/
33. Datahacker.rs. (2020, February 29). OpenCV #004 Common Types of Noise. Retrieved October 18, 2020, from http://datahacker.rs/opencv-common-types-of-noise/
34. Boricha, V. (2018, April 18). Image filtering techniques in OpenCV. Retrieved October 18, 2020, from https://hub.packtpub.com/image-filtering-techniques-opencv/
35. Xah Lee. (2010, February 24). Retrieved October 19, 2020, from http://xahlee.info/img/what_is_gamma_correction.html
36. Mamun, I. (2019, February 12). Image Classification using SSIM. Retrieved October 19, 2020, from https://towardsdatascience.com/image-classification-using-ssim-34e549ec6e12
37. Pantech. (2017, September 30). Matlab Code for PSNR and MSE. Retrieved October 19, 2020, from https://www.pantechsolutions.net/blog/matlab-code-for-psnr-and-mse/

APPENDIX

Autoencoder Structure Python Code

```
Input_img = Input(shape=(256, 256, 3))

#encoding architecture
x1 = Conv2D(128, (3, 3), activation='relu', padding='same', kernel_regularizer=regularizers.l1(10e-10))(Input_img)
x2 = Conv2D(128, (3, 3), activation='relu', padding='same', kernel_regularizer=regularizers.l1(10e-10))(x1)
x3 = MaxPool2D(padding='same')(x2)
x4 = Conv2D(256, (3, 3), activation='relu', padding='same', kernel_regularizer=regularizers.l1(10e-10))(x3)
x5 = Conv2D(256, (3, 3), activation='relu', padding='same', kernel_regularizer=regularizers.l1(10e-10))(x4)
x6 = MaxPool2D(padding='same')(x5)
x7 = Conv2D(342, (3, 3), activation='relu', padding='same', kernel_regularizer=regularizers.l1(10e-10))(x6)
x8 =  Conv2D(342, (3, 3), activation='relu', padding='same', kernel_regularizer=regularizers.l1(10e-10))(x7)
x9 = MaxPool2D(padding='same')(x8)
x10 = Conv2D(512, (3, 3), activation='relu', padding='same', kernel_regularizer=regularizers.l1(10e-10))(x9)



#encoded = Conv2D(64, (3, 3), activation='relu', padding='same')(x2)
# decoding architecture
x11 = UpSampling2D()(x10)
x12 = Conv2D(342, (3, 3), activation='relu', padding='same', kernel_regularizer=regularizers.l1(10e-10))(x11)
x13 = Conv2D(342, (3, 3), activation='relu', padding='same', kernel_regularizer=regularizers.l1(10e-10))(x12)
x14 = Add()([x8, x13])
x15 = UpSampling2D()(x14)
x16 = Conv2D(256, (3, 3), activation='relu', padding='same', kernel_regularizer=regularizers.l1(10e-10))(x15)
x17 = Conv2D(256, (3, 3), activation='relu', padding='same', kernel_regularizer=regularizers.l1(10e-10))(x16)
x18 = Add()([x5, x17])
x19 = UpSampling2D()(x18)
x20 = Conv2D(128, (3, 3), activation='relu', padding='same', kernel_regularizer=regularizers.l1(10e-10))(x19)
x21 = Conv2D(128, (3, 3), activation='relu', padding='same', kernel_regularizer=regularizers.l1(10e-10))(x20)
x22 = Add()([x2, x21])
# x3 = UpSampling2D((2, 2))(x3)
# x2 = Conv2D(128, (3, 3), activation='relu', padding='same')(x3)
# x1 = Conv2D(256, (3, 3), activation='relu', padding='same')(x2)
decoded = Conv2D(3, (3, 3), padding='same',activation='relu', kernel_regularizer=regularizers.l1(10e-10))(x22)
autoencoder = Model(Input_img, decoded)
autoencoder.compile(optimizer='adam', loss='mse', metrics=['accuracy'])
```

Image Processing Python Code (Blurred)

```
import random
def smoothining(image):
    blur1 = []
    for i in image:
        n = random.randint(1,3)
        if(n == 1):
            n1 = random.randint(11,17)

            blur1.append(cv2.blur(i,(n1,n1)))
        elif(n == 2):
            n1 = random.randrange(21,31+1, 2)
            blur1.append(cv2.GaussianBlur(i,(n1,n1), 0))
        else :
            n1 = random.randrange(13, 21+1, 2)
            blur1.append(cv2.medianBlur(i, n1))
            blur1  = cv2.resize(img, (256, 256))

    return blur1
```

Image Processing Python Code (Low-light)

```
from _future_ import print_function
import numpy as np
import argparse
import cv2
import random

def adjust_gamma(image):
    gamma = 0
    n = random.randint(1,6)
    if (n == 1):
        gamma = 0.1
    elif (n == 2):
        gamma = 0.2
    elif(n == 3):
        gamma = 0.3
    elif(n == 4):
        gamma = 0.4
    elif (n == 5):
        gamma = 0.5
    else:
        gamma = 0.6


    invGamma = 1.0 / gamma
    table = np.array([((i / 255.0) ** invGamma) * 255
    for i in np.arange(0, 256)]).astype("uint8")
  # apply gamma correction using the lookup table
    return cv2.LUT(image, table)
```

Image Processing Python Code (Pixelated)

```
import random
def pixelating(i):
    pixelated = 0

    n = random.randint(1,5)
    if(n == 1):
        small = cv2.resize(i, (0,0), fx=0.5, fy=0.5);
        small = cv2.resize(small, (0,0), fx=0.1, fy=0.1)
        pixelated = small
    elif(n == 2):
        small2 = cv2.resize(i, (0,0), fx=0.4, fy=0.4);
        small2 = cv2.resize(small2, (0,0), fx=0.2, fy=0.2)
        pixelated = small2
    elif(n == 3):
        small3 = cv2.resize(i, (0,0), fx=0.3, fy=0.3);
        small3 = cv2.resize(small3, (0,0), fx=0.2, fy=0.2)
        pixelated = small3
    elif(n == 4):
        small4 = cv2.resize(i, (0,0), fx=0.5, fy=0.5);
        small4 = cv2.resize(small4, (0,0), fx=0.2, fy=0.2)
        pixelated = small4
    else:
        small5 = cv2.resize(i, (0,0), fx=0.3, fy=0.3);
        small5 = cv2.resize(small5, (0,0), fx=0.3, fy=0.3)
        pixelated =small5
    pixelated  = cv2.resize(pixelated, (256, 256))
    return pixelated
```

Weekly Logs

Week 1 Meeting Log

MMU MULTIMEDIA UNIVERSITY | FACULTY OF COMPUTING & INFORMATICS

TPT3101/TPR3321 Final Year Project (FYP1) Meeting Log
Trimester 1, 2020/21 (Trimester ID:2010)

Meeting Date: 07/07/2020	**Meeting No.:1**
Meeting Mode: Online Meeting	
Project ID:	**Project Type: Research-based**
Project Title : Age and gender classification from a low-quality human face images	
Student ID : 1151304220	**Student Name: Shaaban, Mohammed Najib Ahmed**
Student Programme and Specialisation: BACHELOR OF COMPUTER SCIENCE (HONORS) (DATA SCIENCE)	
Supervisor Name: Loh Yuen Peng	**Co-Supervisor Name:** **(if applicable)**
Collaborating Company: **(if applicable)**	**Company Supervisor Name:** **(if applicable)**

1. WORK DONE

Tasks: Project overall structure. Problem Formulation, Project Planning

Details (in point form):

- During the first meeting, we have discussed the overall structure of the project. We also planned out in what we going to work, for the coming weeks and finalize the problem statement.
- We identified some of the main research papers which we are going to rely on for our research
- We laid down the project objectives
- We discussed and organized the first part of the report which include project preamble and introduction.

2. WORK TO BE DONE

Tasks: Initializing the report: Introduction, Problem statement, Objectives, Scope.

Details (in point form):

- Work to be done on the first part of the report which include: introduction, problem statement, objectives and project scope

3. PROBLEMS ENCOUNTERED AND SOLUTIONS

No problem encountered during the first week

4. COMMENTS (Supervisor / Co-Supervisor / Company Supervisor)

....
Supervisor's Signature

....
Student's Signature

....
Co-Supervisor's Signature
(if applicable)

....
Company Supervisor's Signature
(if applicable)

Week 2 Meeting Log

FACULTY OF COMPUTING & INFORMATICS

MMU MULTIMEDIA UNIVERSITY

TPT3101/TPR3321 Final Year Project (FYP1) Meeting Log
Trimester 1, 2020/21 (Trimester ID:2010)

Meeting Date: 14/07/2020	**Meeting No.:2**
Meeting Mode: Online Meeting	
Project ID:	**Project Type: Research-based**
Project Title : Age and gender classification from a low-quality human face images	
Student ID : 1151304220	**Student Name: Shaaban, Mohammed Najib Ahmed**
Student Programme and Specialisation: BACHELOR OF COMPUTER SCIENCE (HONORS) (DATA SCIENCE)	
Supervisor Name: Loh Yuen Peng	**Co-Supervisor Name:** **(if applicable)**
Collaborating Company: **(if applicable)**	**Company Supervisor Name:** **(if applicable)**

1. WORK DONE

Tasks: Initializing the report: Introduction, Problem statement, Objectives, Scope.

Details (in point form):

- By the next meeting, I had submitted the first part of the report (Introduction, Problem, statement, Scope, Objectives)
- During the meeting we have discussed about the points that should be improved for the written part
- We have identified the first part of the methodology we are going to use for the proposed system.
- We discussed and organized the second part of the report which include: Literature review and background study

2. WORK TO BE DONE

Tasks: Literature review and background study

Details (in point form):

- Work to be done in literature review and background study part

3. PROBLEMS ENCOUNTERED AND SOLUTIONS

No problem encountered for the second week

4. COMMENTS (Supervisor / Co-Supervisor / Company Supervisor)

....
Supervisor's Signature

....
Student's Signature

....
Co-Supervisor's Signature
(if applicable)

....
Company Supervisor's Signature
(if applicable)

Week 3 Meeting Log

FACULTY OF COMPUTING & INFORMATICS

MULTIMEDIA UNIVERSITY

TPT3101/TPR3321 Final Year Project (FYP1) Meeting Log
Trimester 1, 2020/21 (Trimester ID:2010)

Meeting Date: 24/07/2020	**Meeting No.:3**
Meeting Mode: Online Meeting	
Project ID:	**Project Type: Research-based**
Project Title : Age and gender classification from a low-quality human face images	
Student ID : 1151304220	**Student Name: Shaaban, Mohammed Najib Ahmed**
Student Programme and Specialisation: BACHELOR OF COMPUTER SCIENCE (HONORS) (DATA SCIENCE)	
Supervisor Name: Loh Yuen Peng	**Co-Supervisor Name:** **(if applicable)**
Collaborating Company: **(if applicable)**	**Company Supervisor Name:** **(if applicable)**

1. WORK DONE

Tasks: Literature review and background study

Details (in point form):

- By the third meeting, I had submitted the second part of the report (Literature Review and Background study)
- During the meeting we have discussed about the points that should be improved for the written part
- We have identified the datasets we are going to use to validate the proposed model
- We have planned out the overall methodology we are going to use for the proposed system

2. WORK TO BE DONE

Tasks: Dataset study and investigations, Methodology

Details (in point form):

- Work to be done in investigating and reviewing the set available datasets to be used for validation task of the proposed model.
- Work to be done in methodology of the proposed model.

3. PROBLEMS ENCOUNTERED AND SOLUTIONS

No problem encountered for the third week.

....
Supervisor's Signature

....
Student's Signature

....
Co-Supervisor's Signature
(if applicable)

....
Company Supervisor's Signature
(if applicable)

Week4 Meeting Log

MULTIMEDIA UNIVERSITY | FACULTY OF COMPUTING & INFORMATICS

TPT3101/TPR3321 Final Year Project (FYP1) Meeting Log
Trimester 1, 2020/21 (Trimester ID:2010)

Meeting Date:07/08/2020	**Meeting No.:4**
Meeting Mode: Online Meeting	
Project ID:	**Project Type: Research Based**
Project Title : Age and gender classification from low-quality human face images	
Student ID : 1151304220	**Student Name: Shaaban, MohamMed Najib Ahmed**
Student Programme and Specialisation: : BACHELOR OF COMPUTER SCIENCE (HONORS) (DATA SCIENCE)	
Supervisor Name: Loh Yuen Peng	**Co-Supervisor Name:** **(if applicable)**
Collaborating Company: **(if applicable)**	**Company Supervisor Name:** **(if applicable)**

1. WORK DONE
Tasks: Research Methodology

Details (in point form):

- By the Fourth meeting, I had started the research methodology part
- During the meeting we have discussed about the overall methodology the project should follow
- We have Identified the type of neural work we are going to use to build our model
- We planned out the theoretical framework we are going to use on the project

2. WORK TO BE DONE
Tasks: Work to be done on the research Methodology part

Details (in point form):

- **By the end of this week the research methodology should be completed**
- **Work plan for the next week on implementing and testing the research methodology**

3. PROBLEMS ENCOUNTERED AND SOLUTIONS
No problem encounter

……………………………
Supervisor's Signature

……………………………..
Student's Signature

………………………………..
Co-Supervisor's Signature
(if applicable)

…………………………………….
Company Supervisor's Signature
(if applicable)

Week-5 Meeting Log

FACULTY OF COMPUTING & INFORMATICS

MMU MULTIMEDIA UNIVERSITY

TPT3101/TPR3321 Final Year Project (FYP1) Meeting Log
Trimester 1, 2020/21 (Trimester ID:2010)

Meeting Date: 14/08/2020	**Meeting No.:5**
Meeting Mode: Online Meeting	
Project ID:	**Project Type: Research Based**
Project Title : Age and gender classification from a low-quality human face images	
Student ID : 1151304220	**Student Name: Shaaban, Mohammed Najib Ahmed**
Student Programme and Specialisation: BACHELOR OF COMPUTER SCIENCE (HONORS) (DATA SCIENCE)	
Supervisor Name: Loh Yuen Peng	**Co-Supervisor Name:** **(if applicable)**
Collaborating Company: **(if applicable)**	**Company Supervisor Name:** **(if applicable)**

1. WORK DONE

Tasks:

1- Identification of the research methodology the project going to follow

2- Starting the implementation and testing of the research methodology

Details (in point form):

- **By the end of this week I have submitted the completed part of research methodology**
- **During the meeting we have discussed the implementation of the research methodology**
- **Solving some issues encountered during the methodology implementation process**

2. WORK TO BE DONE

Tasks:

- Complete the Research methodology implementation process
- Work plan to be done for the project theoretical framework

Details (in point form):

- Implement the research methodology to build the image enhancement model
- Test and evaluate the model
- Write down the project theoretical framework

3. PROBLEMS ENCOUNTERED AND SOLUTIONS

Errors and poor result during the implementation process of the proposed methodology

Solution: changing the overall structure of the technique solved the problem

4. COMMENTS (Supervisor / Co-Supervisor / Company Supervisor)

……………………………… ………………………………
Supervisor's Signature Student's Signature

……………………………… ………………………………
Co-Supervisor's Signature Company Supervisor's Signature
(if applicable) **(if applicable)**

Week 6 Meeting Log

TPT3101/TPR3321 Final Year Project (FYP1) Meeting Log
Trimester 1, 2020/21 (Trimester ID:2010)

Meeting Date: 04/09/2020	Meeting No.: 6
Meeting Mode: Online Meeting	
Project ID:	Project Type: Research Based
Project Title : Age and gender classification from a low-quality human face images	
Student ID : 1151304220	Student Name: Shaaban, Mohammed Najib Ahmed
Student Programme and Specialisation: BACHELOR OF COMPUTER SCIENCE (HONORS) (DATA SCIENCE)	
Supervisor Name: Loh Yuen Peng	Co-Supervisor Name: (if applicable)
Collaborating Company: (if applicable)	Company Supervisor Name: (if applicable)

2. WORK TO BE DONE

Tasks:
- Complete the project theoretical framework
- Starting the methodology evaluation process

Details (in point form):
- Work to be done on the project theoretical framework
- Work to be done the methodology evaluation process

1. WORK DONE
[Please write the details of the work done, after the last meeting]

Tasks:
- Methodology implementation
- Methodology testing

Details (in point form):
- By the end of this week I have completed the implementation of the methodology part
- Testing the performance of the methodology part
- During this week I was working on the project theoretical framework
- During the meeting we have discussed the two approaches we are going to follow in order to evaluate the model

3. PROBLEMS ENCOUNTERED AND SOLUTIONS
[Please write the details of the problems encountered, after the last meeting and provide the solutions / plan for the solutions]

4. COMMENTS (Supervisor / Co-Supervisor / Company Supervisor)

....
Supervisor's Signature

....
Student's Signature

....
Co-Supervisor's Signature
(if applicable)

....
Company Supervisor's Signature
(if applicable)

Week 7 meeting Log

FACULTY OF COMPUTING & INFORMATICS

MMU MULTIMEDIA UNIVERSITY

TPT3101/TPR3321 Final Year Project (FYP1) Meeting Log
Trimester 1, 2020/21 (Trimester ID:2010)

Meeting Date: 02/09/2020	**Meeting No.: 7**
Meeting Mode: Online Meeting	
Project ID:	**Project Type: Research-based**
Project Title : Age and gender classification from low-quality human face images	
Student ID : 1151304220	**Student Name: Shaaban, Mohammed Najib Ahmed**
Student Programme and Specialisation: BACHELOR OF COMPUTER SCIENCE (HONORS) (DATA SCIENCE)	
Supervisor Name: Loh Yuen Peng	**Co-Supervisor Name:** **(if applicable)**
Collaborating Company: **(if applicable)**	**Company Supervisor Name:** **(if applicable)**

1. WORK DONE
[Please write the details of the work done, after the last meeting]

Tasks:
- Completed the project theoretical framework
- Starting the project evaluation process

Details (in point form):
- **By the end of this week I have completed the project theoretical framework**
- **I have started the project evaluation process**
- **During the meeting we discussed the future plan for the coming part**

2. WORK TO BE DONE
[Please write the details of the work to be done, before the next meeting]

Tasks:
- Completing the project evaluation process
- Write down the project future plan
- Write down the result

Details (in point form):
- Work to be done in evaluating the project
- Work to be done on the future plan
- Planning out the future work

3. PROBLEMS ENCOUNTERED AND SOLUTIONS
[Please write the details of the problems encountered, after the last meeting and provide the solutions / plan for the solutions]

4. COMMENTS (Supervisor / Co-Supervisor / Company Supervisor)

.... Supervisor's Signature	.... Student's Signature
.... Co-Supervisor's Signature **(if applicable)**	.... Company Supervisor's Signature **(if applicable)**

Turnitin Check

turnitin

Digital Receipt

This receipt acknowledges that Turnitin received your paper. Below you will find the receipt information regarding your submission.

The first page of your submissions is displayed below.

Submission author:	Mohammed Najib Ahmed Shaaban
Assignment title:	Interim Report
Submission title:	FYP1
File name:	FYP1_Shaaban_Mohammed_Najib...
File size:	3.96M
Page count:	95
Word count:	11,254
Character count:	60,187
Submission date:	19-Oct-2020 08:47PM (UTC+0800)
Submission ID:	1419640122

AGE AND GENDER CLASSIFICATION FROM A LOW-QUALITY HUMAN FACE IMAGES

SHAABAN, MOHAMMED NAJIB AHMED

SESSION 2020/2021

FACULTY OF COMPUTING AND INFORMATICS
MULTIMEDIA UNIVERSITY
FEBRUARY 20

FYP1

ORIGINALITY REPORT

18%	11%	13%	7%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Multimedia University Student Paper	2%
2	www.hindawi.com Internet Source	1%
3	mafiadoc.com Internet Source	1%
4	www.cse.unr.edu Internet Source	1%
5	salah.bekhouche.com Internet Source	<1%
6	Zakariya Qawaqneh, Arafat Abu Mallouh, Buket D. Barkana. "Age and gender classification from speech and face images by jointly fine-tuned deep neural networks", Expert Systems with Applications, 2017 Publication	<1%
7	towardsdatascience.com Internet Source	<1%
	Jie Fang, Yuan Yuan, Xiaoqiang Lu, Yachuang	