University of BRISTOL

Department of Computer Science

# Automatic Photograph Filtering

Akhil Jalla

A dissertation submitted to the University of Bristol in accordance with the requirements of the degree of Master of Science in the Faculty of Engineering

September 2019 | CSMSC-19

0000059561

# Declaration:

This dissertation is submitted to the University of Bristol in accordance with the requirements of the degree of Master of Science in the Faculty of Engineering. It has not been submitted for any other degree or diploma of any examining body. Except where specifically acknowledged, it is all the work of the Author.

Akhil Jalla, September 2019

# Executive Summary

With the advancement of technology, digital cameras have become so affordable that it can be incorporated and used in many technological devices such as mobile phones, CCTV, Spy camera and so on. As a result of this, an enormous amount of images are being captured everyday either by humans through their devices or by other devices that automatically captures images. The devices in which the photos are stored also tend to have photos of the same object from different angles out of which only a few photos are worth saving in the storage space. Thus image overload becomes an issue where storage space is wasted by storing images that are not aesthetically pleasing and are of low quality. This project aims to address this issue by using various image processing technologies and a training based model to implement a framework that predicts the quality of the image and further ranks the images which are of high quality.

To achieve the aim, the project implemented the following:

- A detailed research of features which are observed by common people (subjective features) while assessing the quality of the image.
- A detailed research of how these subjective features can be evaluated from the image as discussed in Chapter 2.
- Implementing a framework which calculates the feature score of high quality and low quality images and creates a CSV file of the result,
- A python implementation of a machine learning model which is trained and validated on an image dataset.

As an outcome of the work done, following results were achieved for the project:

- A framework is created which evaluates the scores of the subjective features of the image having humans as region of interest in it and predicts whether the image is of high quality or of low quality.
- The framework produced and trained, provided an accuracy of 86.56%
- The model was further compared with the results of subjective analysis done by humans and provided an accuracy of 76.66% in the real world.

# Acknowledgment

Throughout writing this dissertation, I have received a great deal of support and assistance. Firstly, I would like to thank my supervisor, Dr. Simon Lock, for his dedicated support and guidance. He continuously provided encouragement and was always willing and enthusiastic to assist in any way he could throughout the dissertation.

Furthermore, I would like to thank University of Bristol for providing a wonderful experience throughout my masters. The university has been providing ultimate facilities like library resources, technical labs, and such wonderful staff team who have been there to support and guide throughout. I would like to thank all the professors for providing such good lectures which overall helped me to write my dissertation successfully.

In addition, I would like to thank all the participants who actively participated and spent their valuable time for my survey. My sincere thanks to all my friends who helped me to gain the audience for my survey and actively participated in the survey as well. I would like to extend my sincere gratitude to my friends for their support and for encouraging me throughout this process.

I am extremely grateful to my parents for their love, prayers, care, and sacrifices for educating me and preparing me for the future. Sincere thanks to my other family members for always showing the confidence in me and lending a sympathetic ear whenever I needed it.

# Table of Content

# List of Figures

# List of Tables

# 1. Introduction

With the advent of ubiquitous cameras such as CCTV, cameras in mobile devices, life capturing technologies and so on, images are everywhere and capture everything. According to the InfoTrends statistics, a total of 1.2 trillion digital photographs were taken globally in the year of 2017, which is approximately 160 photographs for each of the approximately 7.5 billion individuals living on planet earth [46]. Although the statistics only shows the number of images taken by smartphones, the actual number of images captured will be comparatively higher if we consider other devices such as CCTV, Spy cameras, Smart-watch camera and so on. This leads to the problem of image overload issue wherein too many images are to be viewed and let alone admired. One such instance which leads to photograph overload is when people are taking more than one photograph of the same scenario but from different angles which creates a non-identical duplicates [26].

Currently humans have to explicitly organise the image gallery and delete the images that are of low quality or images that are not aesthetically appealing. Thus, selecting and retaining appealing and better quality images becomes a troubling and time-consuming job. With so many images present in a device, the images that are not pleasing are also saved in the storage space of a device resulting in the wastage of the storage disk.

This research project explores the current state-of-the-art approaches to image analysis and image ranking system, helping to develop a higher-level management framework that carries out a number of real-world studies to assess the analytical data from the image and further filtering them according to their quality. This research also helps to draw empirically underpinned conclusions regarding the most effective combination of analysis and filtering mechanisms.

There are two approaches used to assess the quality of the image: Subjective approach and Objective approach [6]. The subjective approach is based on perceptual assessment of a human being wherein humans assesses the quality of the images based on the features such as blurriness, contrast, brightness and so on, whereas in thes case of objective approach, mathematical models which calculates the values of different features such as dot and line quality, colour registration, tone reproduction and so on [53]. Based on the scores of the different features, a computational model tries to assess the quality of the image. This research takes inspiration from the fact that subjective analysis always yield better results than the objective analysis. In the project, a framework has been designed and implemented such that there is a computational model which categorises the image either as of high quality or low quality by evaluating different feature scores from the image which are considered by humans during subjective analysis such as blurriness, brightness and so on. This helps to eliminate the use of those tools which assesses the quality score of an image based on feature which are only understandable to professional photographers such as tone reproduction, colour harmony and so on.

## 1.1. Aims and Objective

The main aim of the project is to design a framework which is based on existing state of the art methodologies that can predict whether the image is of high quality or low quality and further rank the images which are regarded as high quality according to their quality assessment score. To narrow down the cost of computation resources required this project only considers those images for evaluation that have humans as region of interest or the main subject in the image. Following are the core objective to achieve the aim of the project:

- Review the existing methodologies available to score different features analytically from an image.
- Design and implement an algorithm that assesses the score of different features from an image which will be done by combining and analysing the strengths and weaknesses of different methodologies proposed by researchers in the past.
- Implement a Machine Learning model which learns the corresponding feature score of both high quality and low quality image and then predict the quality of the image on a validation image dataset. This will further help to evaluate the accuracy of the model proposed and the features score methodology used to create the framework.
- Implement an algorithm that ranks the images from a set of predicted high quality images according to their overall quality assessment score.
- Further to evaluate the model implemented in this project, results predicted by the model will be compared with the result of subject analysis of the image by human beings so as to check the accuracy of the model with respect to human opinion.

The project will help to address the issue of image overload such that devices can be integrated with the framework which will automatically classify the quality of images and prompt the user about the number of low quality images. Appropriate actions which can be taken by the user on the low quality images detected by the framework can then be designed.

# 2. Background and Context

This section reviews the different types of Image Quality Assessment (IQA) methods developed and will also concentrate on the technical background of the characteristic features that can be used for the assessment of the quality of an image. Reviewing the technical background of these features will show the weaknesses and strengths of the present prototypes which will further help to build the project.

## 2.1. Classification of Image Quality Assessment models

The goal of an Objective Image Quality Assessment (IQA) is to develop mathematical models that can correctly and accurately predict the quality of the image. Based on the classification, an Objective IQA can be classified into three types[3]. These are:

### 2.1.1. Full-Reference IQA (FR-IQA)

In this type, a reference image is provided to the framework such that the model will check the quality score of the image by comparing the reference image with the distorted version of the same image. Some of the performance metrics used in the algorithms that are based on this type are Structural Similarity Index Measure (SSIM), Most Apparent Distortion (MAD), Normalized Perceptual Information Distance (NPID), Multi Scale Structural Similarity Index Measure (MS-SSIM), Feature Based Structural Similarity Index (FSIM) and so on[22].

### 2.1.2. Reduced-Reference IQA (RR-IQA)

In this method, a complete reference image is not present at the receiver side of the IQA framework. Only partial information about the reference image is sent by the sender side such that quality assessment on the distorted image is done based on the information about the reference image that is provided by the sender side of the framework. As discussed in [57], some of the approaches proposed under this type are RR-IQA based on visual information fidelity, RR-IQA based on statically modelling the Discrete Cosine Transform (DCT) distribution, RR-IQA method with estimation of SSIM, which is mostly utilized by FR-IQA and so on.

### 2.1.3. No-Reference IQA (NR-IQA)

In this method, no reference image or no information about the reference image is provided to the system. The only thing provided in this framework is the distorted image such that the IQA model should evaluate the score of different model using the information extracted only from the distorted image. This classification is also called as Blind IQA. Although it has been seen that both FR-IQA and RR-IQA yields better results than NR-IQA, but in a real world scenario reference image is not present so as to compare it with the distorted image for its quality assessment. Thus, NR-IQA is a better real-world framework for assessing the quality score of the image. Many researchers have made some progress in this method such as NR-IQA based on Human Visual System (HVS) [21], NR-IQA based on sharpness metric sensitive to blur and noise [58], NR-IQA for JPEG compressed images [52].

## 2.2. Features used for Image Quality Assessment models

This project is based on a No-Reference Image Quality Assessment, wherein no prior information about the image to be assessed is provided to the framework. A framework can be created which calculates the numeric score of different features (discussed in the subsequent section). Some of the image quality attributes or features which are considered while assessing the image by human beings are blurriness in the image, sharpness of the image, proper contrast in the image, noise present in the image and so on. The following sub-section provides the details of the features that can be used for image quality assessment and previous research work done related to the subsequent features.

### 2.2.1. Region of Interest

Most of the research previously done on NR-IQA utilizes image as a whole. However, human beings generally do not evaluate the image as a whole. Individuals generally convey their views on only some areas of an image which are called as Region of Interest (ROI) [2]. Fig.1 shows an example of a Region of interest segmented separately in the image.

**Fig.1 – Region of Interest segmented in an Image**

The region of interest can change according to the images. For example – Humans as ROI in an image, Flowers as ROI in an image, Objects as ROI in an image, Cancer cells as ROI in medical images, foreground as ROI in an image and so on. Hence it becomes extremely important to assess the quality of image on the basis of region of interest to get a better result. Many researches have been done in the past to evaluate the quality of an image based on the region of interest present in the image. In [2], the authors proposed a methodology which segmented the foreground from the background and then calculates the quality score of both background and foreground. In the methodology, they used an algorithm called as Piece-wise Painting algorithm (PPA) which is based on a linear filtering technique and local Otsu's algorithm. In their approach, initially, the PPA is began by splitting the image into several vertical strips of a particular size. The intensity values are modified in each stripe row by the average intensity value of all the rows. Each of stripes are then binarized using Otsu's algorithm such that the image then consists of black and white areas only. The black areas obtained defines the foreground of the image whereas the white areas defines the background of the image. Once the image has been segmented, k-means clustering and Difference of Gaussian (DoG) features have been used such that the similarity between these indexes determines the quality of each of the patch. The average weightage of each patch gives the final quality of the image.

In [31], the authors proposed a visual region of interest and structural similarity (SSIM) image quality assessment technique which combines the areas of human visual system and structural resemblance. In their proposed approach, they considered an input image which has n regions $R_n$, where n = 1,2,3,….,n, and each covering an area $S_n$. Then by low pass filtering, a reference image of each region is obtained. Once the reference image is obtained, the structural similarity of the two structures and weight sum of structural similarity is used to calculate the quality of the image. Further, to improve the accuracy of the model, the authors further calculate the quality score of the image as a whole as well. Although both the approaches mentioned above provided efficient results but one of the drawbacks of these methodologies were lack of usage of the other features, that can be important while calculating the quality of the image, which are particularly observed by human beings while assessing the quality of the image.

### 2.2.2. Blurriness

Blur is defined as a visual phenomenon which causes the edges of the text or images to emerge hazy or out of focus. The blurred region in an image can fall into two categories : Motion blurred regions and the other one is Out-of-focus blurred region. Fig.2 shows two images where in (a) is an example of an image which has motion blurred region and (b) an image with out-of-focus blurred region.

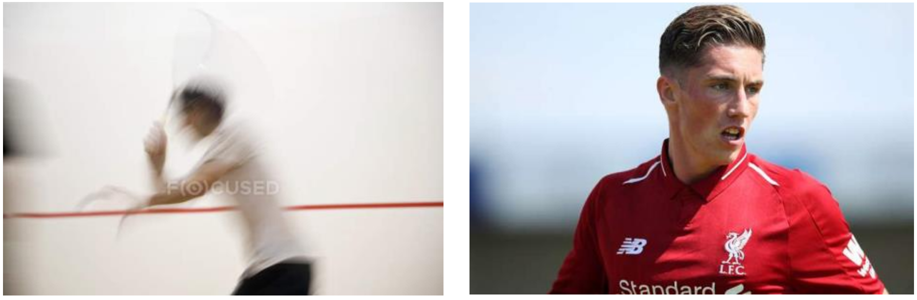



**Fig.2 - (a) motion blurred image (b) Out-of-focus blurred image**

The authors in [45], attempts to classify the blurred region into any one of the two mentioned categories. It becomes quite important for an IQA model to correctly identify the category of blurriness present in the image as most of the times professional photographers intentionally adds blurriness to the image as it helps to provide depth to an image. Thus, to have a better model that determines the category of blurriness, the segmentation of foreground from the background should be done in an efficient manner.

In [11], the authors proposed a methodology that assesses the amount of blurriness with no reference image provided. In this, a probabilistic Support Vector Machine (SVM) is applied that computes the confidence values which represent the distance between the training image set and the test image. The detail image is then used to refine the blur measurements. In the final step of their proposed approach, the blur information is pooled to predict the quality of blurriness present in the image. The authors in [14], proposed a methodology detecting blurriness in the image by using Fast Fourier Transform (FFT) such that it examines the number of high frequency and low frequency present in the image. It was observed that if there are low amount of high frequency present in the image, the image was classified as a blurry image. However, defining a threshold value of the frequency above which the image can be classified as blurry or non-blurry is difficult as this threshold value can vary based on the images.

In [41], the authors provided a way of evaluating a score for the amount of blurriness present in the image. They used a Laplacian operator which is a second derivative operator for passing high spatial frequencies, which are associated with sharp edges. Like the Sobel and Scharr operators, the Laplacian highlights areas of an image with rapid intensity modifications. Once the image has been convolved with Laplacian operator, the variance of the resultant is obtained such that lower the variance score, the greater the probability of the region in the image to be blurry and vice versa.

### 2.2.3. Brightness

According to authors in [54], brightness is defined as a characteristic of a visual sensation where the visual stimulus appears to be more or less intense; or the region where the visual stimulus appears to emit more or less amount of light or luminance. Various methodologies have been proposed by researchers in the past to estimate the amount of brightness present in the image. The authors in [4] mention some of the models used to calculate the brightness of an image. The research mentions a popular brightness substitution Luma which according to ITU-R BT.601 standard is brightness equivalent in MPEG and JPEG algorithms and is calculated as

$$\text{Brightness} = 0.299r + 0.587g + 0.114b$$

where r, g and b are stimulus RGB coordinates of the image. Another way of brightness estimation discussed in the same research is the Arithmetic mean model which takes the mean of the sum of all r, g and b pixels present in the image such that

$$\text{Brightness} = \frac{(r+g+b)}{3}$$

The research then further mentions one more model which is an HSB (Hue, Saturation and Brightness) model according to which the brightness value of the image is

$$\text{Brightness} = \max(r, g, b)$$

The major disadvantage of calculating brightness with these formulas are that it is possible to get the same brightness value for different combinations of r, g and b pixels such that the model will predict the same brightness value even though the luminance differs by a large value.

Another model mentioned in the same research uses stimulus length as a measure of brightness which is introduced in BCH (brightness, Chroma, Hue) model. The length in this approach is calculated according to Cohen metrics [12] such that

$$\text{Brightness} = \sqrt{D^2 + E^2 + F^2}$$

$$\begin{bmatrix} D \\ E \\ F \end{bmatrix} = \begin{bmatrix} 0.2053 & 0.7125 & 0.4670 \\ 1.8537 & -1.2797 & -0.4429 \\ -0.3655 & 1.0120 & -0.6104 \end{bmatrix} . \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

where X, Y and Z are Tristimulus values. The main advantage of this model mentioned was that it simplifies the design of an algorithm that performs only the intended operations without unwilling concurrent modification to other image parameters.

### 2.2.4. Contrast

According to Wikipedia [16], contrast is defined as the difference in luminance or colour which makes an object in the image distinguishable and in the real world which can be determined by the difference in the colour and brightness of the object & other objects within the same field of view. Many researches have been done to find the local or global contrast measure of an image. In [37], the author proposed a formula which is called as the Michelson formula and is used to calculate the global contrast of an image which is confined to periodic patterns of symmetrical deviation of luminance such that there is a maximum and minimum luminance in the image. The formula proposed was:

$$\text{Contrast} = \frac{I_{max} - I_{min}}{I_{max} + I_{min}}$$

where $I_{max}$ and $I_{min}$ represents the maximum luminance and the minimum luminance of the image respectively. In [9], the authors linked contrast with lightness and amount of chroma present in the image such that they came up with an equation to measure the perceived contrast of the image :

$$\text{Perceived Contrast} = -1.505 + 0.131 \,.\, C + 0.151 \,.\, L + 666.216 \,.\, S$$

where C, L and S are the standard deviation of chroma, lightness and high-passed lightness of the image respectively. In [1], the authors proposed a methodology according to which the local contrast of the image I can be calculated by

$$C[x,y] = \frac{B[x,y]}{L[x,y]} - 1$$

where B[x,y] is the image obtained when the image I is convolved with a low pass Gaussian filter and L[x,y] is the image obtained when the image B[x,y] obtained from the previous step, is convolved with another Gaussian filter.

The author in [42] proposed a pyramidal image-contrast structure where a pyramid is a set of lowpass or bandpass copies of an image. The author defined contrast as the ratio of the bandpass filtered image to the low pass filtered image for each pyramid. Each pyramid here is obtained by multiplying the Fourier transform of the image I(x,y) and the cosine log filter centred at frequency of 2'cycles/image such that

$$\text{Pyramid } A(u,v) = F(r,\theta).G(r)$$

The contrast at each pyramid is then represented as a two dimensional array c(x,y) is given by

$$c_i(x,y) = \frac{a_i(x,y)}{I_i(x,y)}$$

where $a_i(x, y)$ is the corresponding local luminance mean image and $I_i(x, y)$ is a low pass filtered version of the image. One of the main limitations of this algorithm is that the quality contrast score of high contrast image and the low contrast image is evaluated as the same which is not the way ideally a feature quality assessment algorithm should perform.

## 2.2.5. Simplicity

Often it is seen that images which are simple and straightforward is admired by the people. Simplicity can be defined as keeping the image background simple and less colourful such that the foreground of the image can be clearly focused by the human. Fig.3 shows two images where (a) is aesthetically pleasing and (b) is aesthetically less pleasing. It can be seen from the Fig. 3 that although (a) is aesthetically more pleasing than (b), it is still less colourful therefore more simpler whereas (b) image is more colourful and therefore less simpler.

**Fig.3 - (a) Aesthetically pleasing (b) Aesthetically less pleasing**

In [55], the author defines hue count as a measure of simplicity. The research suggests that the number of unique hues present in a professional or a high quality image is less than that of a snapshot or a low quality image although each colour may be richer in tones. One can expect to have more number of objects in a low quality image or a snapshot thereby each object having its own colour. Therefore, the number of unique hue counts of an image will be more in case of a low quality image. The author assessed the simplicity score of the image by first converting an image to its HSV representation. A 20 bin histogram H is computed on the good hue values wherein the good hue values are from the image pixels having brightness between 0.15 to 0.95

and saturation greater than 0.2 so as to remove inaccuracy or bias while hue calculation. The simplicity quality of the image is then calculated by the equation

$$q_i = 20 - \| N \|$$

where N is the set of bins with a value greater than the threshold value which is equal to a scalar multiplied by maximum value of histogram m such that

$$N = \{ i \mid H(i) > \alpha.m \}$$

In [56], the author extracted two features to get a better prediction of Simplicity of the image:- area of the region of interest and simplicity assessment score inspired from the research work done in [35]. In [35], the authors describes the usage of simplicity is to reduce the attention distraction by the objects in the background. To calculate the simplicity of an image, the author initially quantized each of the R, G and B channel into 16 values, creating a histogram $H_{is}$ of 4096 bins which gave the counts of quantized colours present in the background. The simplicity $f_s$ was then calculated by using the equation

$$f_s = ( \|S\| / 4096 ) \times 100 \%$$

where $S = \{ i \mid H_{is}(i) >= \alpha H_{max} \}$. Here, $H_{max}$ is the maximum count in the histogram and $\alpha$ is a scalar unit. It was noticed in their research that the simplicity feature of high quality image falls in the range (0 , 1.5%] and that for low quality images falls in the range [0.5% , 5%].

The author in [13], compared simplicity with colourfulness by first dividing the RGB colour space into 64 cubic blocks with four equal partitions along each dimension. The colourfulness score of the image was then calculated by the following equation

$$\text{Colourfulness} = \text{emd}(D_1 , D_2 , \{d(a,b) \mid 0 <= a, b <= 63\})$$

where emd is the Earth Mover's distance which is a measure of simplicity between two weighted distributions, $D_1$ is generated as the colour distribution of a hypothetical image, $D_2$ is computed from the given image by finding the frequency of occurrence of colour, $d(a,b) = \|rgb2luv(c_a) - rgb2luv(c_b)\|$ such that $c_i$ is the Euclidean distance between the geometric center of each cube i after conversion to LUV space.

### 2.2.6. Rule of thirds

Rule of thirds is an important feature which when followed makes the subject in the image aesthetically more pleasant. If an image is divided into two equally spaced horizontal lines and two equally spaced vertical lines such that there are four points of intersection, then according to the Rule of third, an image will be aesthetically pleasant if the region of interest present in the image lies on any of the four points of intersections. The Fig.4 below shows an image which follows the Rule of third

**Fig.4 - An image following Rule of thirds**

In Fig.4, it can be seen that the girl in the image which is region of interest in the image as well lies on the intersection points of imaginary horizontal lines and the vertical lines.

The author in [56] proposed a methodology of calculating the quality score of the image based on the rule of thirds. In his research, the author first segmented the image into homogeneous patches using graph based segmentation such that a saliency value is assigned to each pixel which is the difference between the colour vector of the pixel and the average colour vector for the entire image. A final Saliency value is then obtained by averaging the saliency for the pixel that covered a segmented patch. The author then calculates the quality score of 'rule of thirds' feature of the image by the following formula

$$\text{Score} = \frac{1}{\sum_i A_i S_i} \sum_i A_i S_i e^{\frac{-D^2}{2\alpha}}$$

where $A_i$ is the size of the patch $i$, $S_i$ is the Saliency value of the segmented patch $i$, $\alpha = 0.17$ and D is the closest distance of the center of patch to one of the four intersection points.

In 2011, another method was proposed by Long Mai [36] which was based on a machine learning model in which initially for each saliency map, three types of features were extracted, namely centroid, third map and raw saliency map. The machine learning model was trained to learn the scores of these features on 2089 images following rule of thirds and also on 2051 images that did not follow the rule of thirds. The model was trained on 75% of the total images (training image set) and the rest 25% images were used for validating the prediction (test image set). The result in their proposed approach could have been even more accurate if a better approach for segmenting the foreground or the saliency map of the object in the image was applied.

### 2.2.7. Golden Ratio

Fibonacci spiral or golden spiral is a way of evaluating the subject of focus in the image. Fig.5 shows the imaginary spiral created by rectangular framework using golden ratio taken from flicker.com [20]. In accordance with this principle, on the image if an imaginary spiral is produced then the subject or the region of interest in the image is closer to the reduced part of the spiral radius.

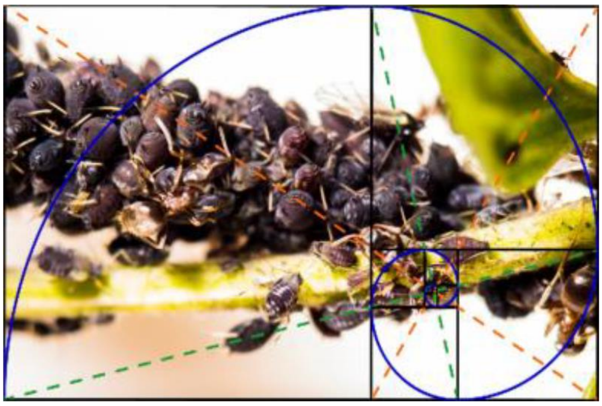


**Fig.5 - Image with imaginary spiral lines following the golden ratio. The subject of the image lies on the lower radius spiral region**

These spirals are created by splitting the image into many rectangular frames for which the subsequent rectangular frame regions are equal to the golden ratio (1.61803). In [5], the author uses the golden ratio as a feature to check the quality of the image. Fig. 6 has been referenced from research [5] where Fig.6(a) shows an image which follows the golden ratio and Fig.6(b) shows an image which doesn’t follow the rule. In both the images horizon separates the image into two rectangular frames and the golden ratio is evaluated according to the line of horizon only.

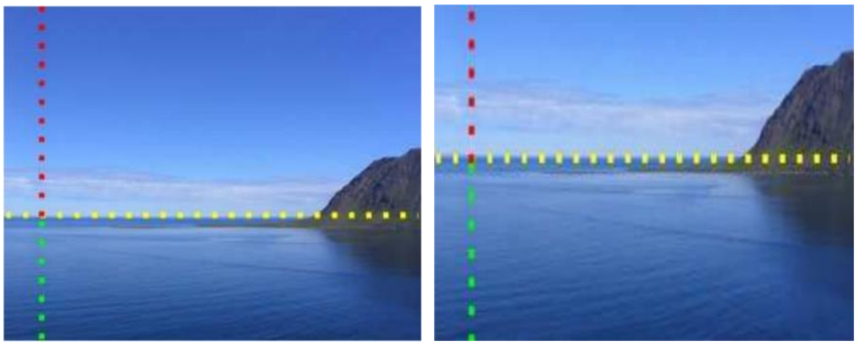


**Fig.6 - (a) Image following Golden ratio (b) Image not following Golden ratio**

The author then verifies if the image follows the golden ratio rule by using the following equation

$$\frac{Y_G}{Y_R} = \frac{Y_R}{Y_R + Y_G} = \alpha\ ,\ Y_R > Y_G$$

where $Y_G$ and $Y_R$ is the length of green dotted line and red dotted line in Fig.6 respectively, and α is the golden ratio. If the calculated value of the ratio is nearer to the golden ratio of 1.61803 then the authors concluded that those images follow the golden ratio and are aesthetically more pleasant. For Fig.6 (a), the ratios observed by the author were 1.6011 and 1.5934 which is nearer to the golden ratio and for Fig.6 (b), the ratios observed by the author were 0.4533 and 0.6743 which is far away from the golden ratio, thus the image shown in Fig.6(b) does not follow the rule of golden ratio. Since then many researches on image quality assessment have been done which uses the golden ratio as one of the features for assessing the quality of the image. In [34] and [44], the authors uses machine learning models to evaluate the quality of an image wherein one of the features used in both of the researches is golden ratio thereby providing evidence of relation of aesthetic quality of an image with the golden ratio.

## 2.2.8. Face detection and clarity

Human faces can be seen as a region of interest which is considered by humans according to their personal preference while doing subjective analysis of the image. Viola-Jones in [51], described one of the most efficient ways of detecting the face in an image. This algorithm uses Haar cascade and Adaboost training to detect the faces present in the image. Several researches have been done to assess the quality of faces present in the image. In [33], the authors proposed a methodology to assess the facial quality in the image so as to improve the facial recognition systems. Initially they segmented the face present in the image and then 15 low level features were extracted from the facial region such as sharpness, contrast, colour and so on. These features were then used to train various machine learning models on image dataset of high quality and low quality facial image such that the model will be able to predict the quality of the test image dataset. In [50], author uses Convolutional Neural Network (CNN) to predict the quality of facial images. In their proposed methodology, the faces present in the images are extracted by using the algorithm proposed in [51]. Then on the segmented faces, CNN is used to model the performance which accepts the entire 2D image as an input. The CNN internally here extracts the features present in the segmented region and these features are extracted using the techniques: Local binary pattern (LBP) and Histogram of oriented gradients (HOG) wherein LBP helps to extract the shape and texture information of the region and HOG helps to get the information regarding the distribution of intensity or edge detection to describe the quality of the image.

## 2.3. Use of Machine learning for Image Quality Assessment

Machine learning algorithms are the computational algorithms which gives a system the capacity to learn and enhance automatically from experience without being explicitly programmed. Various researchers have mostly implemented machine learning algorithms to assess the quality of an image where no reference of the image is provided. Since for a NR-IQA, there is an absence of a reference image, researchers have been trying to design an elaborate feature characteristics that can discriminate distorted image from their original image. In [56], the author uses machine learning techniques to rank the images according to their quality score. The author initially assigned a quality score of a set of images and then extracted relevant features of the image such that a dataset was created which consisted of the scores of features. The dataset created is passed as an input to the training phase of machine learning algorithm and quality score of the image as the target/output value of the model. The model was allowed to learn the score of the features of a set of images and was then evaluated on a set of images which the machine learning model never saw during its training phase. The quality scores predicted by the model from a set of feature score input is then used to rank the image as an output. The model used by the author had a dataset of 12000 images where 6000 images were used in the training phase of the machine learning model and rest 6000 is used while testing the accuracy of the model. In [35], various machine learning algorithms such as Bayesian Classification, Support Vector Machine and Gentle Adaboost were used wherein features such as Clarity Contrast Feature, Lighting feature, Simplicity feature and so on were extracted from the images and the target variable is set as a binary where 1 indicates high quality image and 0 as a low quality image. The model were then trained on 6000 images having equal number of high quality images and low quality images from DPChallenge.com [15] such that the model learns the corresponding feature score of the high quality and low quality image. The models were then tested on 6000 random high quality and low quality images from [15]. The accuracy of these models have inspired researchers to use more deep learning technique such as Convolutional Neural Network (CNN), Deep Convolutional Neural Network (DCNN) and so on to assess the quality of the image more accurately and relatable to subjective analysis done by humans. In [27], the authors uses CNN to evaluate the quality of the image. They used a CNN in spatial domain wherein the input for the network are the patches of images which thereby helped to eliminate the use of hand crafted features extracted by authors in other researches. The authors used five convolutional networks in their research wherein within the network structure, feature learning and regression were integrated into one optimization process. The accuracy attained by the model on a LIVE [29] dataset is above 90% (varies for different types of images present in the LIVE dataset). In [32], the authors uses DCNN to evaluate the quality of image. Their methodology included using neural network having ResNet [24] of three convolutional layer which was followed by a fully connected network such that the neural network takes the input images which are labelled from 0 to 3 where 0 classifies the image of highest quality and 3 being of the lowest quality. The neural network was trained on a set of images such that the output of the network was the probability of the image belonging to a particular class label. The network attained an accuracy of 89.6%.

## 2.4. Various Ranking algorithms proposed for NR-IQA

Several researchers have worked previously to find a methodology to evaluate the quality score of an image as a whole. Some of the work include ranking the images according to their quality score such that the images having high quality scores are classified as high quality and the images with low quality scores are classified as a low quality image. In [39], the authors proposed a methodology which could assess the quality score of an image having natural scenes. The approach was based on the hypothesis that natural scenes in an image have statistical properties which when subjected to a distortion makes the image un-natural. The methodology proposed by them was called as 'Distortion Identification-based Image Verity and Integrity Evaluation' (DIIVINE) in which initially the distorted image is decomposed using a wavelet transform and forms an oriented band pass responses that helps to calculate the sub-band coefficients. This coefficient is then used to retrieve the statistical properties of the image such that a feature vector is created. Once the feature vector were extracted, a model was created which first finds the likelihood of the image being affected by a particular distortion class and secondly, the model mapped the feature vector to a set of quality score for each distortion category. The final quality score of the image was then calculated by combining distortion specific quality score with the probabilistic distortion identification value. The authors in [48] proposed a similar approach which is called as Blind Image Integrity Notator using DCT statistic (BLIIND-II) which was a follow up of BLIIND-I [47]. BLIIND-II also uses the hypothesis considered by [39]. In [48], a discrete cosine transform (DCT) coefficient of natural scene statistics model is used to calculate the statistical properties of the image. A Bayesian approach was then further used to train a simple probabilistic model which predicts the quality score of a natural scene image. If the extracted features from test images are known, then the quality score of the image is the value that maximizes the probability of determined inference model. For this project BRISQUE [38] is chosen and its working is explained in section 3.3. The advantage of choosing BRISQUE is that it is computationally faster than other methods discussed. The computation time required in DIIVINE is high because of the large number of features and BLIIND-II requires non-linear sorting of natural scene statistics which slows down the computational process. Table 1 compares the result of The Spearman's rank ordered correlation coefficient (SROCC), used to assess the quality of an assessment model, for the discussed quality assessment model discussed in this section performed on 1000 test-train combinations on LIVE dataset.

| **Models** | **Median SROCC score** |
|---|---|
| DIIVINE | 0.9250 |
| BLIND-II | 0.9124 |
| BRISQUE | 0.9395 |

**Table 1 – Median SROCC score of different quality assessment model**

Using the strengths and weaknesses of the researches done by the previous researchers, this project aims to evaluate different features scores which can be extracted from a given image such that a machine learning model can be trained to learn the corresponding feature scores of high quality images and low quality image which will further help the model to accurately predict the quality of the unknown images that the model has never seen before. The high quality images predicted by the model will then be ranked according to the quality assessment score of the image. The project will help to eliminate the need of running quality assessment score on low quality images thereby reducing the computational cost of the overall model.

# 3. Implementation

The methodology proposed in this project is a No-Reference image quality assessment which is used to find the quality of the image but only for those images wherein the subject or the foreground of the image is a human. Fig.7 shows a high-level architecture of the proposed implementation.

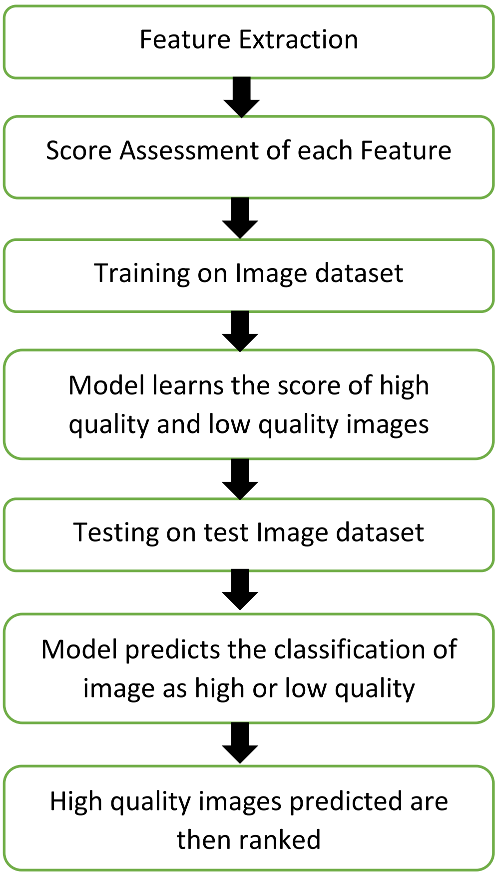



**Fig.7 – High level Architecture of implemented work done in this project**

The basic idea of the implementation is to first extract quality score of different features which can be obtained from the image. Then these scores are evaluated on images which are highly rated by the people and also on the image which is poorly related by the people such that we will have a dataset for a machine learning model which contains the corresponding feature scores of high quality as well as low quality images. The model will be trained such that it learns the corresponding feature scores of high quality image and low quality image. The model will further be checked for its accuracy and the prediction result will be checked on a test image dataset. The result will be further evaluated by checking the similarity of the model with the real subjective analysis by the humans on a set of images. Following steps were implemented to extract the quality of images having humans in it.

## 3.1. Obtaining the feature score of the image

Based on the research done (discussed in Chapter 2), following sections describes the methodology used to obtain the quality scores of the characteristic features of an image dataset.

### 3.1.1. Segmenting Region of Interest from the image

In this project only those images are assessed which has region of interest as humans in it. Hence it gets very important to segment out the human present in the image efficiently so as to prepare a model which can assess some of the features analysis on the subject of the image as well. In 2017, Google published and introduced a new framework named as Deeplab v3 [10] which could efficiently segment out different types of objects present in the image such as person, horses, dogs and so on. [10] uses atrous convolution that allowed the authors to enlarge the field of view of filters to incorporate larger context. Since the repeated combination of max-pooling at each consecutive layers on the Deep Convolutional Neural Network (DCNN) reduced the spatial resolution of the resulting map, the authors in [10] introduced atrous convolution which offered them to find the trade-off between accurate small field of view and a large field of view. With atrous convolution the author kept the size of the stride constant after a few layers of standard convolution such that it will maintain a larger field of view without increasing the number of parameters or the amount of computation power required. Once the image reaches the layer of atrous convolution, it is passed through Atrous Spatial Pyramid Pooling (ASPP) which provides a weight to different valid features present in the different feature map of the segmented layers. The performance score achieved by this framework on PASCAL VOC 2012 test set is 86.9% which was recorded as the highest till 2017.

In this project, region of interest is extracted by using the framework provided by Google i.e. Deeplab v3. The image once loaded is pre-processed such that the image is resized and colour values are changed from [0,255] to [0,1]. Then the image is passed through the convolutional layer of the Deeplab model such that each pixel is voted to be in one of the 21 classes of objects such as person, horse and so on. In this implementation all the classes of the objects were colour coded as black pixel except the person class in the image which is represented by white pixels. Fig.8(a) shows a pair of input images having a person in the image and Fig.8(b) shows its plot of segmented region of interest in the image using Deeplab v3.

**Fig.8(a) - a pair of presumably high quality image and a low quality image, each having person as the region of interest**

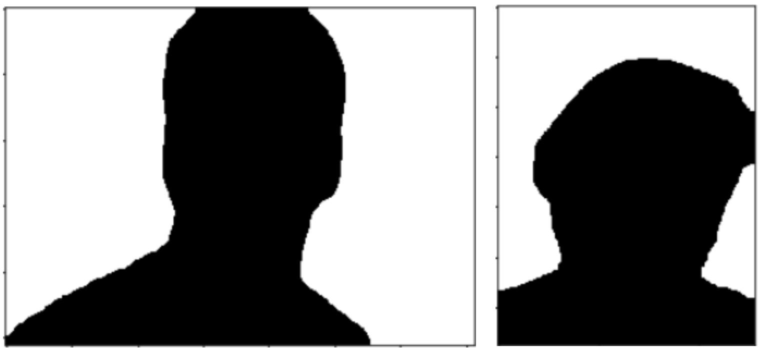


**Fig.8(b) - corresponding image showing segmented region of interest represented by black pixels**

**Fig.8(c) - corresponding image after original colour pixels are masked on the black pixels of the segmented image**

It can be seen that the framework performs efficiently in case of an image where the person is clearly visible in the image and also in the case where the region of interest present in the image is not clearly visible. The black and white pixels are then masked with corresponding colour values from the original image separately so as to segment the region of interest or foreground from the background. Fig.8(c) shows the corresponding image retrieved after coloured pixels from the original image, shown in Fig.8(a), are masked on the black pixels of the segmented image shown in Fig.8(b).

### 3.1.2. Assessing the Blurriness score of the Image

One of the first thing people check while assessing the quality of the image is if there is blurriness present in the image or not. As discussed in section 2.2.2. the blurriness of the object in the image can be of two types: motion blurred region or out-of-focus blurred region. Out of focus images are generally considered of good quality as in those images, the region of interest is clearly distinguished from the background of the image. Keeping this in mind, this implementation utilizes the result of Deeplab model applied on the image such that the algorithm will first assess the blurriness score of the foreground or the region of interest identified and background of the image separately. As a feature for the implementation of the proposed model, blurriness is calculated by using the methodology proposed in [41]. This approach is applied on two images: (a) image having only the foreground and all other pixels as white and (b) image having only the background and all other pixels as white. Each of the images are initially converted to a grayscale image such that all the values of the pixels are now in the range of [0,1]. The images are then convolved using a 3 x 3 Laplacian kernel

$$\text{Laplacian kernel} = \frac{1}{6}\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

The variance of the resultant is then calculated which acts as a blurriness feature score for the model. The Laplacian here acts as a second derivative of an image such that it highlights the region of the image having rapid intensity changes or highlights the edges detected in the region. It can be said that there are fewer edges visible in a blur image rather than an image that is not blurred. If the variance of the image is higher, then it can be assumed that there is a high possibility of a clearer image with sharp edges and vice-versa. To check whether the image is having out-of-focus blurred region, the blurriness score of the image having just the foreground of the image should be higher than that of the blurriness score of the image having just the background of the image. As a result of this part of the implementation, two feature scores are extracted i.e. ‘blurriness score of the foreground’ and ‘blurriness score of the background’.

### 3.1.3. Assessing the Brightness score of the image

Brightness is one of the features which humans consider while doing subjective analysis of an image. In this project, the brightness score of an image is calculated by the formula proposed in [19]. According to [19], brightness of a coloured image can be calculated by using the following formula

$$\text{Brightness Score} = \sqrt{0.299R^2 + 0.587G^2 + 0.114B^2}$$

where R, G and B represents the mean of the intensity of red pixels , green pixels and blue pixels respectively. The score obtained from this formula acts as another feature score for the proposed NR-IQA model in this research.

### 3.1.4. Assessing the Contrast score of the image

For the implementation, contrast per pixel is selected as another feature which could be used for assessing the quality of the image. One of the most commonly used contrast metrics for assessing the contrast score of an image is RMS contrast [28,16], which is based on the standard deviation of luminance level of the stimulus. RMS contrast metric is considered better for those models which are considered very descriptive for stimulus with complex or asymmetric brightness distributions. According to [16], RMS contrast of the image can be defined as the standard deviation of the intensity of the pixels and is calculated by the following formula

$$\text{RMS Contrast} = \sqrt{\frac{1}{H.W}\sum_{i=0}^{W-1}\sum_{j=0}^{H-1}(I_{ij} - brightness)^2}$$

wherein H and W denotes the height and width of the image I, $I_{ij}$ denotes the intensity of the pixel at a particular position and brightness is the average brightness of the image which can be calculated as proposed in the section 3.3.

### 3.1.5. Assessing the Simplicity score of the image

Simplicity can be correlated to the term colourfulness as discussed in section 2.2.5. A feature which calculates a score which represents how colourful or simple is the image will be beneficial for the model while predicting the quality of the image. In this project, colourfulness or simplicity is calculated by the methodology proposed in [39]. In [39], the author derived a simple metric that correlated with the result of subjective evaluation done in the research. According to the methodology, initially the image colour metric is derived by

$$rg = R - G$$

$$yb = 0.5(R + G) - B$$

Here, the above two equation represents the opponent colour representation where R, G and B are the red, green and blue channel present in the image respectively. It was found by the author in [23] that the mean and standard deviation of the opponent colour representation correlates to almost 95% of the actual experimental result conducted during the research. Once the opposite colour representation has been defined, the mean and standard deviation are calculated by the following formula

$$\sigma_{rgyb} = \sqrt{\sigma_{rg}^2 + \sigma_{yb}^2}$$

$$\mu_{rgyb} = \sqrt{\mu_{rg}^2 + \mu_{yb}^2}$$

wherein σ represents the standard deviation and μ represents the mean. The colourfulness metric of simplicity metric of the image is then calculated using the following formula

$$\text{Simplicity or Colourfulness score} = \sigma_{rgyb} + 0.3 \text{ x } \mu_{rgyb}$$

such that the higher the score obtained by the above equation, higher is the colourfulness of the image and lower is its simplicity and vice-versa.

### 3.1.6. Assessing the Rule of thirds score of the image

As discussed in section 2.2.6., Rule of thirds is one of the features which makes an image look aesthetically more pleasant. Hence the quality assessment of the image on the basis of rule of thirds will act as another feature for a model that can predict the quality of the images. To calculate the rule of third score of the image, the image is initially divided into two equally spaced horizontal lines and two equally spaced vertical lines such that there are four points of intersection. Fig.9 shows an image wherein green lines indicates the imaginary lines and blue dot indicates the point of intersection of these lines.

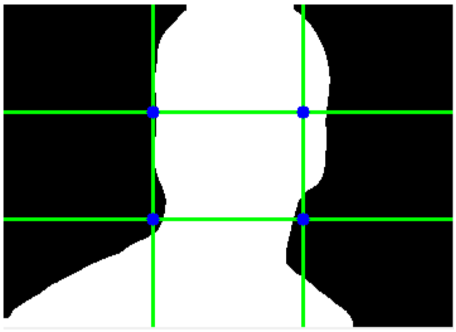


**Fig.9 – image shows the imaginary lines and intersection points retrieved from the input image. This is applied on the image where the foreground (white pixels) is segmented from the background (black pixels)**

The coordinates of the points of intersection are identified such that they can be further used to check the rule of third score of the object in the image. To check if the subject of the image lies nearer to one of the points of intersection of the line, the centroid of the foreground region is calculated.

Fig.10 shows an image wherein the white pixels and black pixels are the foreground and background of the image respectively and red dot indicates the coordinates of the centroid of the foreground.

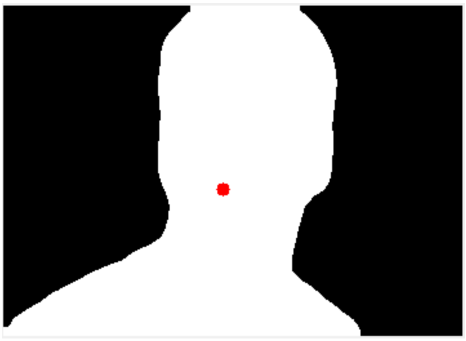


**Fig.10 - Image wherein red dot shows the centroid of the foreground of the image segmented by white pixels**

The Rule of third score of an image can then be calculated using the methodology proposed by the authors in [40]. According to the proposed methodology in [40], initially a value is calculated which represents the minimum distance of the centroid of foreground from any of the four points of intersection such that

$$Distance_{ROT} = min\ (\sqrt{(cX - x1)^2 + (cY - y1)^2}, \sqrt{(cX - x2)^2 + (cY - y2)^2}, \sqrt{(cX - x3)^2 + (cY - y3)^2}, \sqrt{(cX - x4)^2 + (cY - y4)^2}\ )$$

where (*cX, cY*) represents the coordinates of the centroid of the foreground of the image and (*x1, y1*), (*x2, y2*), (*x3, y3*) and (*x4, y4*) are the coordinates of the points of intersection of the imaginary horizontal lines and vertical lines.

The quality score of an image based on Rule of Third (ROT) is then calculated by using the following formula

$$\text{ROT score} = \frac{Area\ of\ segmented\ foreground\ \times 0.05}{(image.height \times image.width)} \times \left(1 - \frac{Distance_{\ ROT}}{\sqrt{(image.height)^2 + (image.width)^2}}\right)$$

The value obtained from the above equation will be in the range [0, 1] wherein 0 indicates the worst score and 1 indicates the best score obtained from the image.

### 3.1.7. Assessing the symmetry score of the region of interest in the image

A short survey was conducted for the evaluation of the implemented model, wherein people were also asked about the features they check while assessing the quality of the image. One of the features which got significant number of votes was the symmetry of the region of interest to the image. The region of interest will be symmetrical if the region of interest lies exactly on the centre of the image. To compute the score of this feature, initially the coordinates of the centroid of the foreground or region of interest is found out. Then an imaginary vertical line is made on the image such that the line passes through the centre coordinates of the image such that the line also passes from coordinates (x1, y1) and (x2, y2) where x1=x2= 0.5*(width of the image), y1 = 0 and y2 = (height of the image). The perpendicular distance of the centroid to the imaginary line is calculated by using the following formula [17]

$$\text{Distance} = \frac{(\mathrm{y2} - \mathrm{y1})*\mathrm{cX} - (\mathrm{x2} - \mathrm{x1})*\mathrm{cY} + \mathrm{x2}*\mathrm{y1} - \mathrm{y2}*\mathrm{x1}}{\sqrt{(y2-y1)^2 + (x2-x1)^2}}$$

where (cX, cY) is the coordinates of the centroid of the foreground or the region of interest segmented in the image. Thus, to calculate how close the centroid is to the centre alignment of the image, the ratio of the distance calculated to (0.5* width of the image is calculated) which is considered here as the score of this feature.

$$\text{Symmetry score} = \frac{Distance}{0.5*image.width}$$

According to the result obtained, lower the symmetry score of foreground of the image, better is the center alignment of the foreground to the image.

### 3.1.8. Assessing the area of foreground present in the image

It was also observed from the online survey conducted, participants suggested that one of the things which they look while assessing the quality of the image is the area occupied by the region of interest in the image. The votes suggested that the while assessing the image quality, people also check whether the region of interest in the image is not too small enough to be not visible and also at the same time it should not be big enough to cover the whole area of the image such that there is no background in the image. To check this, we compute the ratio of the area of region of interest present in the image to the area of the image such that

$$\text{Area ratio} = \frac{\text{Area of region of interest}}{Image.width * Image.height}$$

This feature also helps the model to have a correlation of blurriness of foreground and background of the image when the region of interest cover different values of area in the image.

### 3.1.9. Assessing the amount of noise present in the image

According to Wikipedia [18], Image noise can be defined as a random variation of colour information or brightness of the image and is generally an aspect of electronic noise. Often, during pre-processing noise gets added into the image, thereby degrading the quality of the image. Fig.11 shows an image which is disturbed with a presence of noise.

**Fig.11 - image which is disturbed with a presence of noise**

To calculate the amount of noise present in the image, the methodology proposed in [25] has been applied for this project. In [25], the authors first convolve the input image using a 3x3 noise estimation operator.

$$\text{Noise Estimator operator} = \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

which has a mean of 0 and a variance of $36\sigma_n^2$, where $\sigma_n$ represents the standard deviation of the noise present at each pixel.

The standard deviation $\sigma_n$ which represents the amount of noise present in the image is then calculated by using the absolute deviation caused by assuming it as a Gaussian distribution of mean zero and variance σ such that

$$\int_{-\infty}^{\infty} |t| \frac{1}{\sqrt{2\pi}\sigma} exp\left(\frac{-t^2}{2\sigma^2}\right) dt = \sqrt{\frac{2}{\pi}}\sigma$$

which then helps to further compute $\sigma_n$ that represents the standard deviation of the noise present in the image. The standard deviation can be then computed by using the equation below

$$\sigma_n = \sqrt{\frac{\pi}{2}}\left(\frac{1}{6(W-2)(H-2}\right).S$$

where S is the result of summation of pixels of the resultant of the image *I(x,y)* convolved with the noise estimator operator such that

$$\mathrm{S} = \textstyle\sum_{image\ I} |I(x,y) * N|$$

## 3.1.10. Implementing the feature score on image set

The assessment scores of all the features discussed in the above section are then evaluated on a set of images. A set of 1600 images wherein the region of interest in the image is a human is downloaded from [43] and [20] such that there are an equal number of high quality images and low quality images. To make the training of the model proposed unbiased, all the images downloaded are of width 640px and height is a variable which is adjusted according to the aspect ratio. The images downloaded are initially categorised as high quality or low quality image based on the rating given by the people to the image. A python implementation of feature quality assessment of an image based on section [3.1.1 – 3.1.9.] is made such that the image dataset is loaded into the model so as to retrieve the feature scores of corresponding low quality images and high quality image. A csv file is created which holds the feature quality score of all the 1600 images downloaded as the image dataset. Here another column is added which represents the quality of image such that whenever the feature assessment is being done on high quality image, the 'Output' variable will be set as 1 and whenever the feature assessment is being done on low quality images, the 'Output' variable will be set as 0. This csv file acts as a dataset for the image quality assessment model for this project having 1600 records of the corresponding features of a set of the image containing both low quality and high quality images.

### 3.1.11. Reasons for not implementing Face quality assessment in image

According to the survey conducted where people were asked what characteristics they look while assessing the quality of an image in which the region of interest is a human, one of the most common answers noted was if the faces in the image are clearly visible and is of high quality. Although, the region of interest for the image dataset is human, but one cannot be sure if that the image to be assessed will always have a person whose orientation is in the same way as trained by the model in case of classifier used in [51]. For example, the face detection algorithm proposed in [51] will not yield a desirable result if the person in the image is yawning and this is due to the change in dynamics and variations of the mouth. One more scenario where face detection will be difficult to use is the case when there is a person in the image wearing sunglasses. Fig.12 shows a set of three images which has person as the region of interest in the image but the face detection algorithm proposed in [51] was not able to detect any faces in the image.

**Fig.12 – A set of 3 images in which no face was detected while using frontal face cascade classifier for face detection algorithm proposed in [51]**

Since one cannot anticipate the orientation and appearance of the person in the image, multiple cascade classifier needs to be used such as classifier for frontal face, classifier for side face, classifier for faces with glasses and so on to detect the facial region of the person in the image. But this comes with a disadvantage in the form of computational speed of the process. Adding more cascade classifier slows down the processing time due to the fact that the algorithm will have to run the number times equal to the number of classifiers.

## 3.2. Implementing a learning based model to predict the quality of Image

In this project, machine learning based models are implemented to obtain a framework which predicts the quality of the image. Initially the dataset created as discussed in section [3.1.10] is pre-processed to check if there are many missing information present at any index. If there is any missing index present, then that index value is replaced by the mean of the values of the quality score of that particular feature from all images. Any machine learning model is implemented in two phases: training phase and testing phase. Training phase is the time where the model is learning the relation between the different input features with the target variable and a test phase is the time when the model checks its accuracy on a set of records it has never seen before. Keeping this in mind, the dataset is divided into two sets, namely training image dataset, which contains 80% records of the dataset, and a test image dataset which includes the rest 20% records of the dataset. The training dataset is used by machine learning models so as to learn the corresponding scores for an "Output' variable which indicates the quality of an image whereas the test dataset is a dataset which the model has never seen during the training phase and will thus check the prediction of the 'Output' variable so as to check the accuracy of the model. There are many machine learning algorithms for a classification model. For this project, two machine learning classifiers were considered and compared to predict the quality of the image. The two classifiers used are:

### 3.2.1. Decision Tree Classifier

According to [49], decision tree is a tree like structure wherein each leaf node represent the outcome of target class, nodes other than leaf nodes are the features and branches in the decision tree are the set of decision rules. In a decision tree classifier, the prediction by the model is done by making a decision tree such that each node has a value which when analysed layer by layer will result to an outcome of a prediction. The decision tree algorithm operates by choosing recursively the best feature attribute by dividing the dataset and expanding the decision tree's leaf node till a stop criteria is met. In doing so, it uses a mutually exclusive and exhaustive if-then rule such that these rules are learned sequentially and one at a time during the training phase of the model. Once training phase on a particular feature is done, the tuples covered by that decision rule is removed and this process continues until met by a termination condition. Although decision tree algorithm is easy to construct and inexpensive, these are easily prone to over-fitting and also in the case of large decision tree, it gets difficult to interpret and the result provided by the model may seem counter-intuitive, thus having low bias but more variance. To counter the disadvantages of this methodology, one more classifier is considered and used, which is discussed in section [3.2.2].

### 3.2.2. Random Forest Classifier

Random Forest [7] is a supervised machine learning algorithm and in its simplest form is a collection of decision trees randomly selected from a subset of training set and are relatively uncorrelated to each other such that the classifier aggregates the votes from all the decision trees to decide the outcome of a test set. To ensure that all the decision trees are not related to each other, the training algorithm applies a technique called as bagging or feature bagging wherein random forest allows each decision tree to randomly sample from the training dataset with replacement that results into a new decision tree. If after some training phase, two or more features are found to be strongly related to the output variable, then those features will be selected in many of the trees, thereby causing their output to be correlated. The unseen input is then fed to the model where the output is predicted using the majority votes from each of the decision trees. Random forest overcomes the disadvantages of decision tree. In random forest there is a reduced chance of over-fitting as in this case the model is averaging several decision trees and also bagging operation reduces the variance of the model without raising the bias leading to a stronger model than decision tree.

## 3.3. Ranking the predicted High Quality Images

This project further intends to display all the high quality image according to their quality. Once the machine learning model implemented as described in section 3.2, the algorithm will make a list of images which the model has predicted as high quality from a set of images passed by the user to the model. The list of high quality images predicted by the model is then passed through an implementation which ranks the images according to the quality assessment score. [38] describes a general image quality assessment score implementation for an image and is called as Blind/Referenceless Image Spatial Quality Evaluator (BRISQUE). The work done in [38] is based on the fact that the pixel intensity distribution of a natural image varies from that of a distorted image. It was observed by the authors that pixel intensities of natural images follows a Gaussian distribution after normalization whereas the same was not observed for distorted images. To calculate the quality score of the image, the author proposed a methodology to find the amount of deviation of pixel intensities of images from an ideal Gaussian distribution curve. To normalize the intensity of the pixels in the image, Mean Subtracted Contrast Normalization (MSCN) is used. MSCN coefficient is calculated by the following equation

$$MSCN(i,j) = \frac{I(i,j) - \mu(i,j)}{\sigma(i,j) + C}$$

where *I(i, j)* is the intensity of the pixel at position *(i,j)*, $\mu$ is the local mean field or the gaussian blur of the original image and $\sigma$ is the local variance field or the gaussian blur of the square of the difference of the image and local mean field. This method doesn't only restricts till finding the MSCN of pixels but also tries to establish the relationship between the neighbouring pixels using pair-wise products of MSCN of the image with a shifted version giving MSCN of neighbouring pixels. Pair-wise MSCN coefficient of four orientation are found i.e. for horizontal neighbour, vertical neighbour, main diagonal neighbour and secondary diagonal neighbour such that there are 5 images derived from the original image: MSCN image and four pair-wise product MSCN image referring to neighbouring pixels. Two features are extracted by fitting the MSCN image to a Generalized Gaussian Distribution (GGD) and 16 features are extracted by fitting the other pair-wise product images to an Asymmetric Generalized Gaussian Distribution (AGGD) such that there are 18 features in total. By using the same methodology another 18 features are extracted by reducing the size of the image such that in total there are 36x1 feature vector. This process is then applied on a training image dataset and using Support vector machine (SVM), the model is trained to predict the quality score of an image. Here the quality score of an image will lie in the range [0,100] where 0 indicates the best quality and 100 indicates the worst quality.

Once the quality score of the images are obtained, the high quality images retrieved from the approach discussed in section 3.2 are then ranked according to the quality score of the images. The methodology reduces the time BRISQUE has to be performed, as all the low quality images are separated from the high quality images and thus BRISQUE needs to be implemented only on the high quality images.

# 4. Analysis of results obtained

This section discusses some of the analysis of the results observed from the methodology used in this project. As discussed in section 3.2, the model was trained on a set of 1600 images where in 1280 images (80% of total images) were used to during the training phases of the model and rest 320 images were used during the testing phase of the model. For this project, Random forest classifier was chosen over Decision tree classifier because of the advantages of random forest over the later as discussed in section 3.2.1 and 3.2.2. Also it was seen that Random Forest Classifier was providing better results for the dataset created than the Decision tree classifier. The model once trained was able to classify the high quality images and low quality images with an accuracy of 86.56%. With the accuracy score attained, satisfactory answer is retrieved for the hypothesis in the project that an image can be evaluated by a machine according to their quality by extracting different features from the image. The set of features extracted from the image are those features which humans generally consider while doing the subjective analysis of the image. Further the evaluation of the model will be discussed in Chapter 5. The feature score dataset created from a set of high quality and low quality images also helped to analyse the relationship of different features with each other (Feature Correlation) and the relationship of each feature with the target variable of the dataset (Feature Importance). Some of the key observations noted are listed below.

## 4.1. Feature Importance

This section discusses the features that are highly correlated to the output variable of the model. A feature can be considered important if the shuffling of values increases the error rate produced by the model. If even by shuffling the values of the features, the error rate obtained from the model is constant then one can consider that feature as not important. Once the Random forest classifier model used in this project has been trained, the most important features along with the score of their importance can be extracted. Table 2 shows the list of features and their importance score relative to the output variable arranged in descending order.

| Feature | Importance Score |
|---|---|
| Brightness Score | 0.2442 |
| Noise Score | 0.1562 |
| Background Blur Score | 0.1328 |

| | |
|---|---|
| Area Ratio | 0.1244 |
| Foreground Blur Score | 0.0865 |
| Rule of third score | 0.0851 |
| Simplicity score | 0.0713 |
| Contrast score | 0.0559 |
| Portrait mode | 0.0291 |
| Landscape Mode | 0.0140 |

**Table 2 - List of features along with their importance score**

According to Table 2. the important features which affects the quality of the model is the brightness of the image followed by noise and blurriness of background such that together these features hold an importance of over 50% while determining the quality of an image

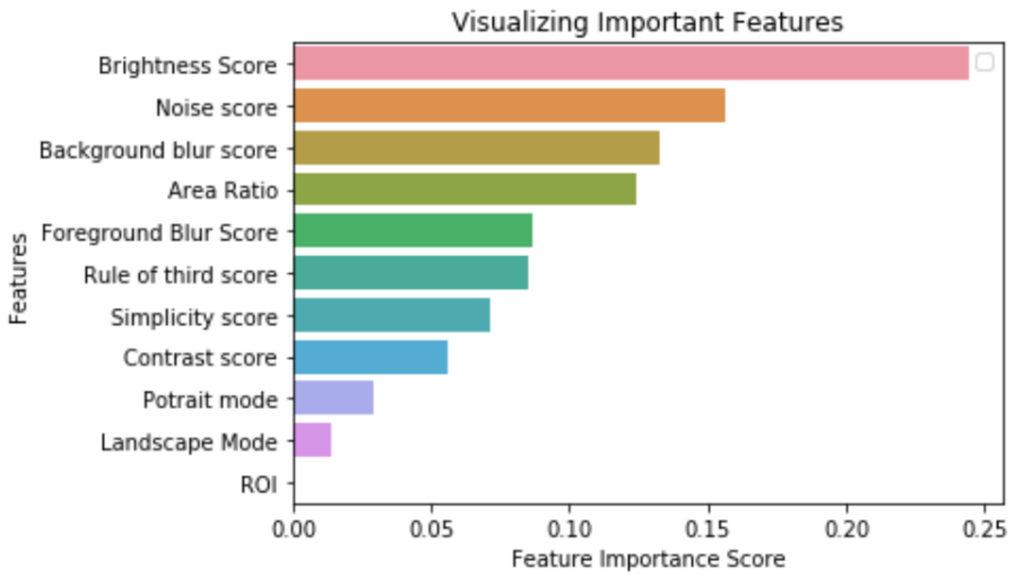



**Fig.13 - Visualizing Important Feature of the model**

Fig. 13 shows a visualization of the feature importance score relative to the output or target variable of the model. Using the data of feature importance, a more accurate model can be created in which only the most important features are selected for training the model and the features which are not that important with respect to the output variable are dropped from the model. Although the top 3 features shown in Table 2 holds the overall importance of over 50%, only one feature was dropped from the dataset which is Symmetry score that had a feature importance of less than 1%.

## 4.2. Feature Correlation

The statistical relationship between two features in a dataset is called as feature correlation. For example, one feature of the dataset can be highly or loosely associated to another feature in the same dataset. A feature correlation can be classified into three categories: Positive correlation, where both the features are changing in the same direction; Neutral correlation, where both the features are independent of each other and the value of one feature doesn't affect the value of the second feature; and Negative Correlation, where both the feature input variables change in the opposite direction i.e. with the increase in the value of one feature, the value of the other feature decreases. Heatmap is an excellent visualization used to compare any two feature variables in a dataset with their corresponding values. Fig.14 shows a heatmap showing the correlation of the feature scores extracted from a set of high quality and low quality images.

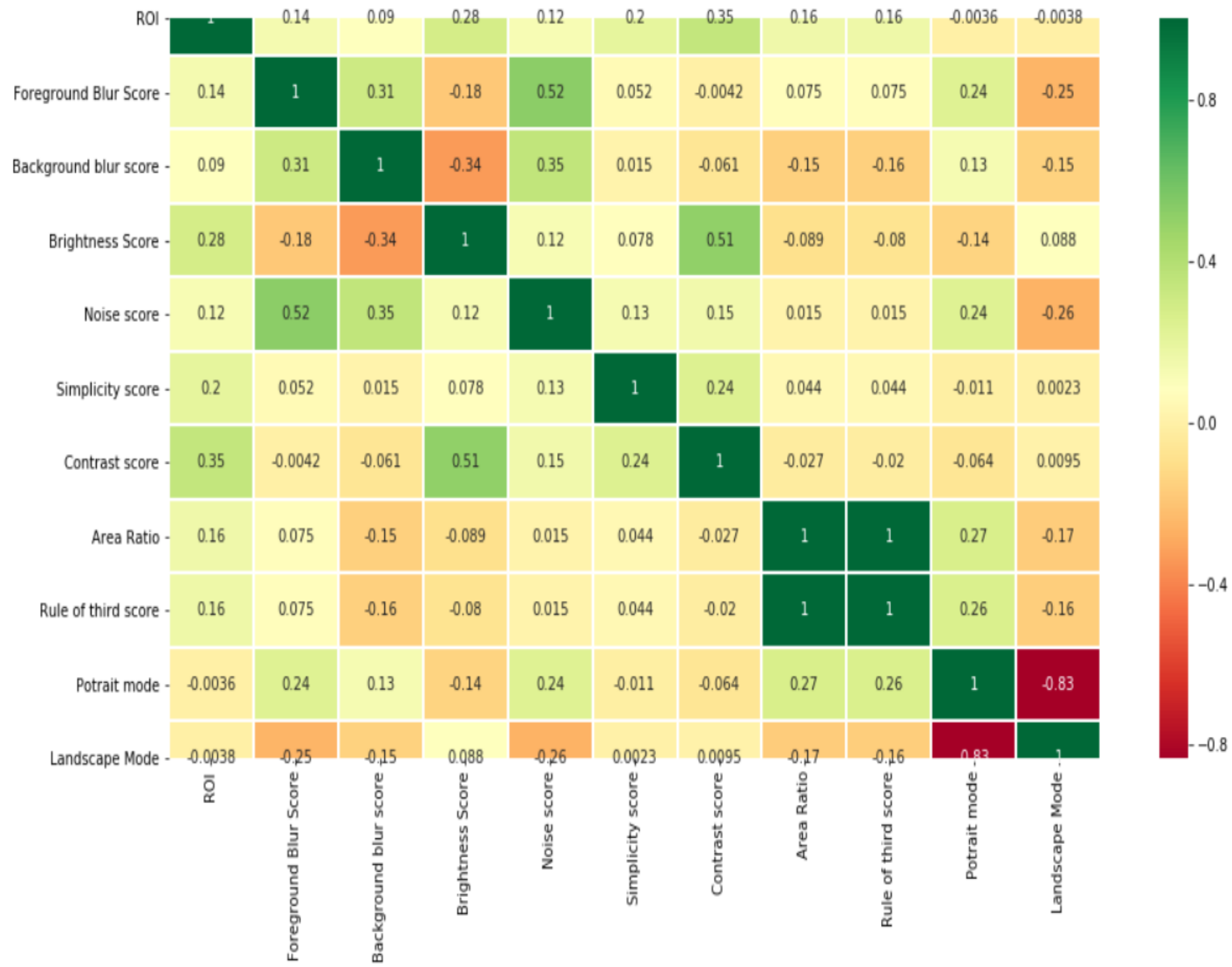



**Fig.14 – Correlation Heatmap**

In Fig.14, shade of green represents positive correlation, shade of white represents neutral Correlation and shade of red represents negative correlation. The diagonal of a heatmap represents the relation of each feature with itself and hence it is always strictly positively correlated. According to the heatmap generated by the dataset, it can be seen that the Area ratio score is highly and positively correlated to Rule of third score of an image. Some other positively correlated features are Noise score – Foreground blur score, Noise score – background blur score, Contrast score – Brightness score. It will be interesting to note that the Simplicity Score is neutrally correlated with almost all other features extracted from the image in this project except contrast score whereas brightness score – background blur score, Area ratio score – background blur score are negatively correlated to each other.

# 5. Critical Evaluation

This section discusses the critical evaluation tests done on the model implemented in this project. Some of the evaluation metrics which can be used to check the performance of a classification model are

- Accuracy
- Precision, Recall and F1 score
- Cohen's Kappa Statistic
- Brier Score Loss
- Area under ROC curve

Further to evaluate the model, the results of the model are compared with the results of subjective analysis done by 50 participants on a set of images and also the evaluation of the model was done on a new image dataset downloaded using pexels API which is an API to fetch high quality images from pexels.com.

## 5.1. Accuracy of the model

Accuracy of the model can be found by testing the trained model onto a test image dataset which the model has never seen before. The model then predicts the target variable of test image dataset and using that, one can find the accuracy of the model by comparing the predicted outcome from the model with the ground truth. Table 3 shows a confusion matrix of general classification model.

| | | **Actual** | |
|---|---|---|---|
| | | **Positive** | **Negative** |
| **Predicted** | **Positive** | True Positives | False Positives |
| | **Negative** | False Negatives | True Negatives |

**Table 3 – Confusion matrix of general classification model**

For this project, there are 320 images (20% of total images) present in test dataset. Fig.15 below shows the confusion matrix which represents the number of times the model predicted the quality of the image which matches with the ground truth of those 320 images.

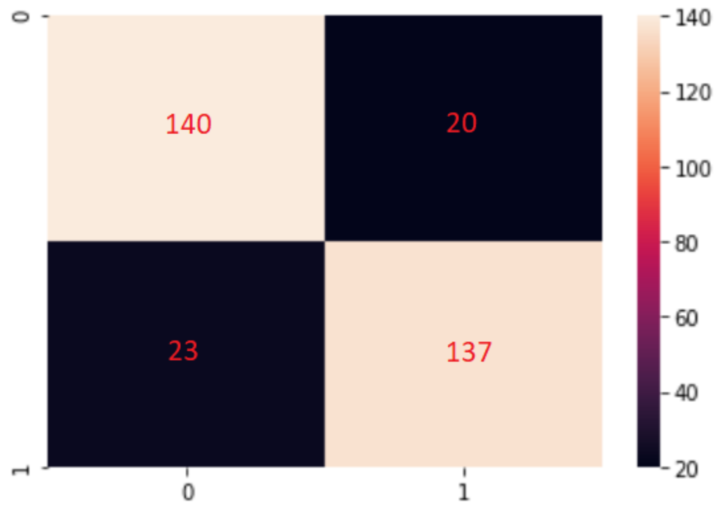



**Fig.15 – Confusion matrix**

According to the confusion matrix, the model correctly predicted 137 low quality images and 140 high quality images. The model wrongly predicted the quality of 43 images (23+20) only. The accuracy attained by the model by testing it on a test dataset is 86.56% which is significantly good. The evaluation of the model also relates two to other metrics which are precision and recall such that

$$Precision = \frac{true\ positives}{true\ positives + false\ negatives}$$

and

$$Recall = \frac{true\ positives}{true\ positives + false\ positives}$$

An F1 score is then obtained to balance precision and recall. F1 score can be calculated by the following equation

$$F1\ score = 2\ \times \frac{precision\ \ \times recall}{precision + recall}$$

In an ideal model F1 score is 1. The precision and recall of the model implemented in this project is 0.8726 and 0.8562 such that the F1 score obtained is 0.8643.

## 5.2. Cohen's Kappa Statistic

Cohen's Kappa statistic is an under-utilized evaluation metrics for a classification machine learning model. The value of Kappa tells us how much the model is better than a model in which the classifier randomly predicts the output variable based on the frequency of a class. Hence it can be said that Kappa statistic is a metric value which compares the observed accuracy with an expected accuracy (in this case from a random classifier). The Kappa value is an excellent measure that can manage multi-class and imbalanced class issue. The Cohen's Kappa Statistic can be defined by the following equation

$$Kappa = \frac{p_o - p_e}{1 - p_e} = 1 - \frac{1 - p_o}{1 - p_e}$$

where $p_o$ represents the observed agreement and $p_e$ represents the expected agreement from a random classifier. The value of Kappa statistic is always less than or equal to 1 where according to [30], a value of less than 0 indicates no agreement between the two models and thus interprets that the classifier is not at all a good classifier, a value between 0 to 0.20 indicates slight agreement, 0.21 to 0.40 indicates a fair agreement, 0.41 to 0.60 indicates a moderate agreement, 0.61 to 0.80 indicates substantial agreement and 0.81 to 1.0 indicates an almost perfect agreement. For the model proposed in this project the value of Cohen's Kappa Statistic attained is 0.7312.

## 5.4. Brier Score Loss

Brier score [8] is a scoring function which calculates accuracy of predicted probabilities such that Brier score can be calculated by taking the mean squared error between the expected value of the target variable and the predicted probability of the model for a particular target class. The equation for Brier score can be written as

$$Brier\ Score = \frac{1}{N}\sum_{t=1}^{N}(f_t - o_t)^2$$

where $f_t$ and $o_t$ are the predicted probability and actual outcome of the event at an instance t and N is the number of prediction instances. This equation is an appropriate scoring rule for a binary event in which the target variable is either 1 (in this project – 1 represents High quality image) or 0 (in our project – 0 represents Low quality image). The Brier score of a model always lies in the range [0,1] where 0 indicates a model with the perfect skills. To have a good machine learning classification model, the Brier score of the model should be as close as possible to value 0. If it is nearer to 1, the Brier score represents that the model obtained high error values between the prediction of probabilities for a particular class and the actual class. The Brier score obtained by the machine learning model in this project is 0.103 which is closer to 0.0 and thus indicates that the model proposed in this project is efficiently predicting the probability of true target classes.

## 5.3. Area under ROC Curve

Area under ROC curve is an evaluation metrics which can be used when the output of the target class can be binarized. The area under the ROC curve represents the ability of the model to discriminate between the negative and the positive labels. For an ideal model which predicts all the target values accurately, the area under ROC curve will be 1.0. Area of 0.5 will represent a model which is completely random. To have a good model the area under ROC curve should be closer to 1.0 and greater than 0.5. Fig. 16 represents the Area under ROC curve for the model implemented in this project.

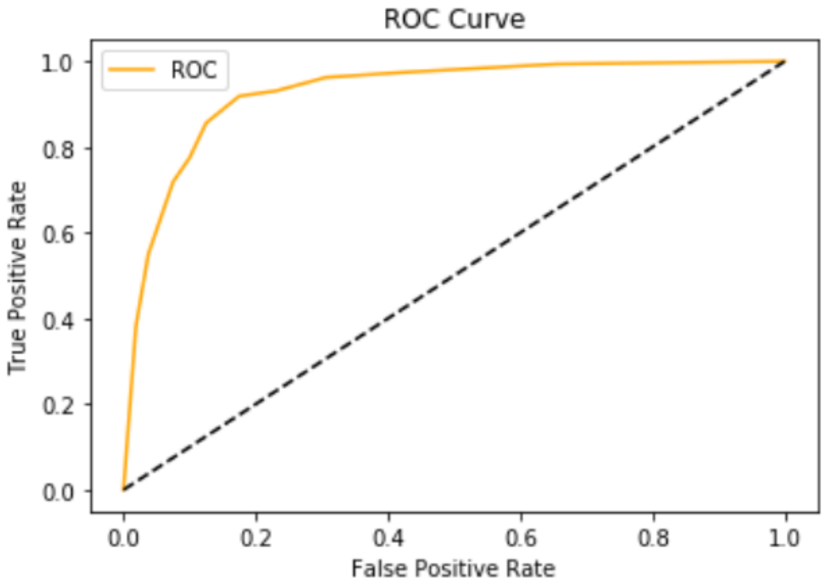



**Fig.16 – ROC curve for the implemented model**

the Area under ROC curve for the implemented model is 0.9278 which is closer to 1.0 indicating the model efficiently discriminates between the positive and the negative classes of the output variable given the values of the input feature scores.

## 5.4. Comparison with subjective analysis

To check how efficient and accurate is the machine learning model proposed in this research for predicting high quality images and low quality images related to the subjective analysis done by the humans, a survey was conducted. In this survey, 50 human participants were shown a set of 30 images such that the participants have to classify each of the image either as high quality image or as a low quality image. The final class of each image after the survey was decided by the most number of votes given by the participants to a particular class which determines the quality of the image such that if more than 50% people voted an image as high quality, the image will be considered as high quality so as to compare it with the predicted output class for the same image by the machine learning model implemented in this project. Table 4 shows the comparison between the performance of the model in predicting the quality of the image where the ground truth is taken from the subjective analysis done by the humans.

| | | **By the proposed model** | |
|---|---|---|---|
| | | **Positive** | **Negative** |
| **By Humans** | **Positive** | 9 | 3 |
| | **Negative** | 4 | 14 |

**Table 4 – Comparison of proposed model with subjective analysis**

According to Table 4, it can be said that the number of times the image predicted the correct high quality images or true positive predicted by the model is 9, number of times the correct low quality images or true negatives predicted by the model is 14, number of times the model predicted high quality image as a low quality image or the false positive predicted by the model is 3 and number of times the model predicted low quality images as high quality image or false negatives attained by the model is 4. Thus it can be said that the model implemented in this project attained an accuracy of 76.67% with respect to subjective analysis done by humans.

## 5.5. Testing the model on an Image dataset

As such the performance of the model cannot be compared with the results of the other models because the other models are trained and tested on specific dataset such as LIVE dataset, DPChallenge.com. The model proposed in this project cannot be tested on these image dataset because these datasets doesn't exclusively contains images having a person as the region of interest and thus requires a dataset in which images having a person as the region of interest can only be considered. To evaluate the performance of this model, 200 random images, having region of interest as a person, are downloaded from pexels.com using pexels API which is a website having a large dataset of high quality images only. Table 5 shows the performance of the proposed model on 200 high quality images randomly downloaded from pexels.com.

| | | By the proposed model | |
|---|---|---|---|
| | | **Positive** | **Negative** |
| **Ground Truth** | **Positive** | 171 | 29 |
| | **Negative** | 0 | 0 |

**Table 5 – Performance of proposed model 200 random images from pexels.com**

The model was able to classify 171 images out of 200 images as high quality images such that the number of true positives predicted by the is 171 and the number of false positives predicted by the model is 29. Since there were no low quality images downloaded from pexels.com, the number of false negative and the number of true negatives predicted by the model is 0.

# 6. Limitations of the proposed approach

Although the model proposed in this project gave satisfactory results, this project was limited in its scope. Following is a list of limitations in this project.

- Quality assessment of image in this project is restricted to only those images which have human as the region of interest in the image. To make a generic image quality assessment model, the model should evaluate the quality scores of the features present in the image wherein the region of interest is not restricted to 'person' class and can be any object.
- The model was trained on a private image dataset of 1600 images and hence could not be evaluated and compared with other models proposed by other researchers.
- Due to time complexity, only selected features were extracted from the image. To further improve the prediction by the model, more features can be extracted from the image such as face clarity, aesthetic score and so on.
- The implementation uses single machine learning classifier to train the model which can predict the quality of the image whereas researchers in the past have used more than one classifier to increase the accuracy of the model.
- The model does not work well for the images in which both foreground and background of the image is human i.e. there should not be any human present in the background of the image.
- The model has been specifically trained on images with width 640px and height adjusted to its aspect ratio so that there is no bias created while learning the feature scores of different features due to the size of the image. The model should be designed in such a way that the size of the image doesn't affect the quality scores of any feature in the image and also it should be comparable to any image of different sizes.

# 7. Future Work

To overcome the limitations discussed in the previous section, further work can be done in the future which can lead to an even better model than the one implemented in this project. Some of the things which can be done in the future to make this project more accurate and efficient than the already implemented methodology in this project are:

- Having an image quality assessment in which region of interest in the image can be any object present in the image and should not be restricted to human only.
- With the help of the survey conducted, a question was asked to user about the features one looks for while assessing the quality of an image. Some features suggested by the participants were Depth of the image, aesthetic beauty of an image, camera angle related to the subject of image, face clarity and so on. With higher computational power of the production environment, these features can be extracted from the image which can certainly add to prediction accuracy of the model.
- Training and testing the model on a larger image dataset such that the model can more accurately learn scores of high quality and low quality images.
- Use of more than one machine learning classifiers such as use of Voting Classifier, which considers the output from more than one machine learning classifier and predicts the output based on the votes given by individual classifiers in the voting classifier.
- Comparing the model performance by evaluating the result of the model on an image dataset which is publicly available such as LIVE dataset, DPChallenge.com and so on. The performance of the model should be compared using these datasets because most of the image quality assessment models proposed by other researchers are evaluated on these datasets only.
- Currently this model works best for those images which have a width of 640px and height adjusted according to its aspect ratio. Further work can be done to remove this dependency on the size of the image.

# 8. Conclusion

This project aimed to research about the current state of the art methodologies present till date to evaluate different features present in the images and also how different features can be used to classify the image as high quality image or a low quality image. Using the results and findings of the research done by the previous researchers, an algorithm is designed which could assess if the image quality is high or low. In this approach all the features which are extracted from the image are some of the features which humans mostly check while subjective analysis of images done by them (from the survey conducted). The features extracted from the images are given a particular value which will be used by a machine learning model such that model learns the values of features corresponding to low quality images and high quality image. The model is trained on a set of 1600 images having equal number of different low quality images and high quality images which are actually rated by people online. Out of this 1600 images, 320 images were kept hidden from the model while training so as to check and evaluate the performance of the model. Although the scope of this project was limited to only those images in which the region of interest is human, the approach implemented in this project provided satisfactory results having overall accuracy of 86.56%. The model was evaluated using various metrics such as F1 score, precision, recall, Kappa's statistic and so on. In all these evaluation metrics the value retrieved for these metrics indicated that the proposed model efficiently discriminates between the positive classes (high quality image) and the negative classes (low quality images). The model was further evaluated by checking how well the model performs with respect to subjective analysis done by humans on a set of images. The comparison results show that the model performs well in comparison to subjective analysis where out of 30 images given for subjective analysis to humans, the model correctly predicted the output of 23 images, giving an accuracy score of 76.66% with respect to subjective analysis. Once the model has predicted the set of high quality images, it ranks the predicted high quality images according to their quality assessment score. This quality assessment score is calculated using a methodology called BRISQUE (discussed in section 3.3). The model also reduces the computational costs required by BRISQUE on a huge set of images by first reducing the huge set of images to a set of images having only high quality images and then arranging only the high quality images according to their assessment score.

Although the evaluation metrics suggests that the model implemented achieves a satisfactory results, the approach implemented in this project faces some limitations which sometimes affects the results of the model. To improve the model and make it more efficient more relevant features can be extracted from the image such as face clarity, aesthetic score of the image and so on and can be added as an important feature which determines the quality of the image. Also further work can be done to find the classification of the quality of the image wherein the region of interest in the image can be any object and is not only restricted to human.

# 9. Bibliography


1. Ahumada, A. and Beard, B. (1998). 43.1: A Simple Vision Model for Inhomogeneous Image-Quality Assessment. *SID Symposium Digest of Technical Papers*, 29(1), p.1109.
2. Alaei, A., Raveaux, R. and Conte, D. (2016). Image quality assessment based on regions of interest. *Signal, Image and Video Processing*, 11(4), pp.673-680.
3. Al Madeed, N., Awan, Z. and Al Madeed, S. (2018). Image Quality Assessment - A Survey of Recent Approaches. *Computer Science & Information Technology*.
4. Bezryadin, S., Bourov, P. and Ilinih, D. (2007). Brightness Calculation in Digital Image Processing. *International Symposium on Technologies for Digital Photo Fulfillment*, 2007(1), pp.10-15.
5. Bhattacharya, S., Sukthankar, R. and Shah, M. (2010). A framework for photo-quality assessment and enhancement based on visual aesthetics. *Proceedings of the international conference on Multimedia - MM '10*.
6. Boberg, A. (1992). *Subjective vs objective image quality*. [online] Pdfs.semanticscholar.org. Available at: https://pdfs.semanticscholar.org/df9f/1cc8560d655a26121df2bde8ed777a2acbeb.pdf [Accessed 25 Jun. 2019].
7. Breiman, L. (2001). *Machine Learning*, 45(1), pp.5-32.
8. BRIER, G. (1950). VERIFICATION OF FORECASTS EXPRESSED IN TERMS OF PROBABILITY. *Monthly Weather Review*, 78(1), pp.1-3.
9. Calabria, Anthony & Fairchild, M.D.. (2003). Perceived Image Contrast and Observer Preference II. Empirical Modeling of Perceived Image Contrast and Observer Preference Data. Journal of Imaging Science and Technology. 47. 494-508.
10. Chen, Liang-Chieh & Papandreou, George & Schroff, Florian & Adam, Hartwig. (2017). Rethinking Atrous Convolution for Semantic Image Segmentation.
11. Chen, M. and Bovik, A. (2011). No-reference image blur assessment using multiscale gradient. *EURASIP Journal on Image and Video Processing*, 2011(1).
12. Cohen, J. (2001). *Visual color and color mixture*. Urbana, Ill.: University of Illinois Press.

13. Datta, R., Joshi, D., Li, J. and Wang, J. (2006). Studying Aesthetics in Photographic Images Using a Computational Approach. *Computer Vision – ECCV 2006*, pp.288-301.
14. Dosselmann, R. and Yang, X. (2012). *No-Reference Noise and Blur Detection via the Fourier Transform*. University of Regina, Canada.
15. Dpchallenge.com. (n.d.). *DPChallenge - A Digital Photography Contest*. [online] Available at: https://www.dpchallenge.com/ [Accessed 6 Aug. 2019].
16. En.wikipedia.org. (2019). *Contrast (vision)*. [online] Available at: https://en.wikipedia.org/wiki/Contrast_(vision) [Accessed 27 Aug. 2019].
17. En.wikipedia.org. (2019). *Distance from a point to a line*. [online] Available at: https://en.wikipedia.org/wiki/Distance_from_a_point_to_a_line [Accessed 18 Jul. 2019].
18. En.wikipedia.org. (2019). *Image noise*. [online] Available at: https://en.wikipedia.org/wiki/Image_noise [Accessed 17 Aug. 2019].
19. Finley, D. (2006). *HSP Color Model - Alternative to HSV (HSB) and HSL*. [online] Alienryderflex.com. Available at: http://alienryderflex.com/hsp.html [Accessed 15 Jul. 2019].
20. Flickr. (2019). *Flickr*. [online] Available at: https://www.flickr.com [Accessed 8 Aug. 2019].
21. Gao, X., Lu, W., Tao, D. and Li, X. (2010). Image quality assessment and human visual system. *Visual Communications and Image Processing 2010*.
22. George, A. and Livingston, S. (2013). A SURVEY ON FULL REFERENCE IMAGE QUALITY ASSESSMENT ALGORITHMS. *International Journal of Research in Engineering and Technology*, 02(12), pp.303-307.
23. Hasler, D. and Suesstrunk, S. (2003). Measuring colorfulness in natural images. *Human Vision and Electronic Imaging VIII*.
24. He, K., Zhang, X., Ren, S. and Sun, J. (2016). Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
25. Immerkær, J. (1996). Fast Noise Variance Estimation. *Computer Vision and Image Understanding*, 64(2), pp.300-302.
26. Jaimes, A., Shih-Fu Chang and Lou, A. (2003). Detection of non-identical duplicate consumer photographs. *Fourth International Conference on Information,*

*Communications and Signal Processing, 2003 and the Fourth Pacific Rim Conference on Multimedia. Proceedings of the 2003 Joint*.

27. Kang, L., Ye, P., Li, Y. and Doermann, D. (2014). Convolutional Neural Networks for No-Reference Image Quality Assessment. *2014 IEEE Conference on Computer Vision and Pattern Recognition*.
28. Kukkonen, H., Rovamo, J., Tiippana, K. and Näsänen, R. (1993). Michelson contrast, RMS contrast and energy of various spatial stimuli at threshold. *Vision Research*, 33(10), pp.1431-1436.
29. Laboratory for Image and Video Engineering - The University of Texas at Austin. Live.ece.utexas.edu, 2003. http://live.ece.utexas.edu/research/Quality/.
30. Landis, J.R.; Koch, G.G. (1977). "The measurement of observer agreement for categorical data". *Biometrics* **33** (1): 159–174.
31. Li, D. and Cheng, T. (2015). Method of image quality assessment based on region of interest and Structural Similarity. *Proceedings of the 2015 6th International Conference on Manufacturing Science and Engineering*, pp.786-791.
32. Li, Y., Ye, X. and Li, Y. (2017). Image quality assessment using deep convolutional networks. *AIP Advances*, 7(12), p.125324.
33. Lienhard, A., Ladret, P. and Caplier, A. (2015). Low Level Features for Quality Assessment of Facial Images. Proceedings of the 10th International Conference on Computer Vision Theory and Applications, pp.545-552.
34. Lu, X., Lin, Z., Jin, H., Yang, J. and Wang, J. (2014). RAPID: Rating Pictorial Aesthetics using Deep Learning. *Proceedings of the ACM International Conference on Multimedia - MM '14*.
35. Luo, Y. and Tang, X. (2008). Photo and Video Quality Evaluation: Focusing on the Subject. *Lecture Notes in Computer Science*, pp.386-399.
36. Mai, L., Le, H., Niu, Y. and Liu, F. (2011). Rule of Thirds Detection from Photograph. *2011 IEEE International Symposium on Multimedia*.
37. Michelson, A. (1995). Studies in optics. New York: Dover Publications.
38. Mittal, A., Moorthy, A. and Bovik, A. (2012). No-Reference Image Quality Assessment in the Spatial Domain. IEEE Transactions on Image Processing, 21(12), pp.4695-4708.

39. Moorthy, A. and Bovik, A. (2011). Blind Image Quality Assessment: From Natural Scene Statistics to Perceptual Quality. IEEE Transactions on Image Processing, 20(12), pp.3350-3364.
40. Nguyen, N., Hoang, D., Hong, T., Pham, H. and Trawiński, B. (2018). Intelligent Information and Database Systems. Pp.466-475.
41. Pech-Pacheco, J., Cristobal, G., Chamorro-Martinez, J. and Fernandez-Valdivia, J. (2000). Diatom autofocusing in brightfield microscopy: a comparative study. *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*.
42. Peli, E. (1990). Contrast in complex images. *Journal of the Optical Society of America A*, 7(10), pp.2032-2040.
43. Pexels.com. (n.d.). Free stock photos · Pexels. [online] Available at: https://www.pexels.com [Accessed 8 Aug. 2019].
44. Pogačnik, D. (2012). Evaluating photo aesthetics using machine learning.
45. Renting, L., Zhaorong, L. and Jiaya, J. (2008). Image partial blur detection and classification. *2008 IEEE Conference on Computer Vision and Pattern Recognition*.
46. Richter, F. (2017). *Infographic: Smartphones Cause Photography Boom*. [online] Statista Infographics. Available at: https://www.statista.com/chart/10913/number-of-photos-taken-worldwide/ [Accessed 28 Jun. 2019].
47. Saad, M., Bovik, A. and Charrier, C. (2010). A DCT Statistics-Based Blind Image Quality Index. IEEE Signal Processing Letters, 17(6), pp.583-586.
48. Saad, M., Bovik, A. and Charrier, C. (2012). Blind Image Quality Assessment: A Natural Scene Statistics Approach in the DCT Domain. *IEEE Transactions on Image Processing*, 21(8), pp.3339-3352.
49. Sharma, Himani & Kumar, Sunil. (2016). A Survey on Decision Tree Algorithms of Classification in Data Mining. International Journal of Science and Research (IJSR).
50. Vignesh, S., Priya, K. and Channappayya, S. (2015). Face image quality assessment for face selection in surveillance video using convolutional neural networks. *2015 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*.
51. Viola, P. and Jones, M. (2004). Robust Real-Time Face Detection. *International Journal of Computer Vision*, 57(2), pp.137-154.
52. Wang, Z., Sheikh, H. and Bovik, A. (2002). No-reference perceptual quality assessment of JPEG compressed images. *Proceedings. International Conference on Image Processing*.

53. Wolin, D., Johnson, K. and Kipman, Y. (1998). The Importance of Objective Analysis in Image Quality Evaluation. In: *IS&Ts NIP 14: 1998 International Conference on Digital Printing Technologies*. [online] Society for Imaging Science and Technology, pp.603-606. Available at: https://pdfs.semanticscholar.org/8b2b/386cfac261fb07c2a168201ffeac62231421.pdf [Accessed 25 Jun. 2019].
54. Wyszecki, G. and Stiles, W. (2001). *Color science:Concepts and Methods, Quantitative Data and Formulae, 2nd Edition*. New York: Wiley.
55. Yan, K., Xiaoou, T. and Feng, J. (2006). The Design of High-Level Features for Photo Quality Assessment. *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1 (CVPR'06)*.
56. Yeh, C., Barsky, B. and Ouhyoung, M. (2014). Personalized Photograph Ranking and Selection System Considering Positive and Negative User Feedback. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 10(4), pp.1-20.
57. Yogita V, H. and Patil, H. (2015). A Survey on Image Quality Assessment Techniques, Challenges and Databases. *IJCA Proceedings on National Conference on Advances in Computing*, pp.34-38.
58. Zhu, X. and Milanfar, P. (2009). A no-reference sharpness metric sensitive to blur and noise. *2009 International Workshop on Quality of Multimedia Experience*.

# Appendix A

## Important code snippets

### Using deeplabv3 for image segmentation and extracting region of interest

```
deeplabv3 = models.segmentation.deeplabv3_resnet101(pretrained=1).eval()
preprocess = Tensor.Compose([Tensor.Resize(256),
#                 Tensor.CenterCrop(512),
                  Tensor.ToTensor(), Tensor.Normalize(mean = [0.485, 0.456, 0.406], std = [0.229, 0.224, 0.225])])
def decode_segmap(image, nc=21):
    label_colors = np.array([(255, 255, 255), (255, 255, 255), (255, 255, 255), (255, 255, 255), (255, 255, 255),
(255, 255, 255), (255, 255, 255), (255, 255, 255), (255, 255, 255), (255, 255, 255), (255, 255, 255), (255, 255,
255), (255, 255, 255), (255, 255, 255), (255, 255, 255), (0, 0, 0),(255, 255, 255), (255, 255, 255), (255, 255,
255),(255, 255, 255), (255, 255, 255)])
    r = np.zeros_like(image).astype(np.uint8)
    g = np.zeros_like(image).astype(np.uint8)
    b = np.zeros_like(image).astype(np.uint8)
    for l in range(0, nc):
        idx = image == l
        r[idx] = label_colors[l, 0]
        g[idx] = label_colors[l, 1]
        b[idx] = label_colors[l, 2]
    rgb = np.stack([r, g, b], axis=2)
    return rgb
def region_of_interest(img):
    number_of_black_pixels = np.count_nonzero(np.all(img==[0,0,0],axis=2))
    if number_of_black_pixels > 500:
        ROI = 1
        print('Region of interest is present i.e ROI = 1')
    else:
        ROI = 0
        print('Region of interest is not present')
    return ROI
```

### Blurriness score

```
def blurriness(img,img1):
    if ROI==1:
        segmented_person_gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
        background_gray = cv2.cvtColor(img1, cv2.COLOR_BGR2GRAY)
        blur_score_foreground = cv2.Laplacian(segmented_person_gray, cv2.CV_64F).var()
        blur_score_background = cv2.Laplacian(background_gray, cv2.CV_64F).var()
        print('Blur score of the foreground is ',blur_score_foreground)
        print('Blur score of the background is ',blur_score_background)
    else:
        blur_score_foreground = 0.0
        blur_score_background = 0.0
        print('Doesnt have an ROI')
    return blur_score_foreground, blur_score_background
```

## Brightness Score

```
def brightness(img):
    if ROI==1:
        img_pil = Image.fromarray(img)
        stat = ImageStat.Stat(img_pil)
        r,g,b = stat.mean
        brightness_score = math.sqrt(0.241*(r**2) + 0.691*(g**2) + 0.068*(b**2))
        print('Brightness score of the image is ', brightness_score)
    else:
        brightness_score = 0.0
        print('Doesnt have an ROI')
    return brightness_score
```

## Noise Score

```
def noise(img):
    if ROI==1:
        img_gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
        h,w = img_gray.shape
        M = [[1, -2, 1],[-2, 4, -2],[1, -2, 1]]
        sigma_noise = np.sum(np.sum(np.absolute(convolve2d(img_gray, M))))
        sigma_noise = sigma_noise * math.sqrt(0.5 * math.pi) / (6 * (w-2) * (h-2))
        print('the noise score of the image is ', sigma_noise)
    else:
        sigma_noise = 0.0
        print('Doesnt have an ROI')
    return sigma_noise
```

## Simplicity Score

```
def colorfulness(img):
    if ROI==1:
        (b,g,r) = cv2.split(img.astype('float'))
        rg = np.absolute(r-g)
        yb = np.absolute((0.5)*(r+g)-b)
        rbMean = np.mean(rg)
        ybMean = np.mean(yb)
        rbStd = np.std(rg)
        ybStd = np.std(yb)

        stdRoot = np.sqrt((rbStd ** 2) + (ybStd ** 2))
        meanRoot = np.sqrt((rbMean ** 2) + (ybMean ** 2))
        colorfulness_score = stdRoot + (0.3 * meanRoot)
        print('Colourfullness score of the image is ', colorfulness_score)
    else:
        colorfulness_score = 0.0
        print('Doesnt have an ROI')
    return colorfulness_score
```

## Contrast Score

```
def contrast(img):
    if ROI==1:
        img_pil = Image.fromarray(img)
```

```
        h,w = img_pil.size
        img_pil_array = np.array(img_pil)
        intensity = 0
        for i in range(img_pil_array.shape[0]):
            for j in range (img_pil_array.shape[1]):
                intensity += (img_pil_array[i,j,0] - brightness_score)**2
    #     print(intensity)
        total_pixel = h * w
        contrast_score = math.sqrt(intensity/total_pixel)
        print('the contrast score of the image is ',contrast_score)
    else:
        contrast_score = 0.0
        print('Doesnt have an ROI')
    return contrast_score
```

## Rule of Third Score

```
def ROT(img):
    if ROI ==1:
        new_image = np.copy(img)
        new_image[np.where((new_image == [255,255,255]).all(axis = 2))] = [125,125,125]
        new_image[np.where((new_image == [0,0,0]).all(axis = 2))] = [255,255,255]
        new_image[np.where((new_image == [125,125,125]).all(axis = 2))] = [0,0,0]
        gray_image = cv2.cvtColor(new_image, cv2.COLOR_BGR2GRAY)
        ret,thresh = cv2.threshold(gray_image,127,255,cv2.THRESH_BINARY)
        contours = cv2.findContours(thresh,cv2.RETR_TREE,cv2.CHAIN_APPROX_SIMPLE)[-2]
        countour_count = 0
        for c in contours:
            countour_count += 1

        M = cv2.moments(contours[0])
        # calculate x,y coordinate of centeroid
        if M["m00"]==0:
            cX = 0
            cY = 0
        else:
            cX = int(M["m10"] / M["m00"])
            cY = int(M["m01"] / M["m00"])

        abc = cv2.circle(new_image, (cX, cY), 5, (125, 125, 125), -1)
        width, height, channel = abc.shape
        horizontal_x1 = int((1/3)*height)
        horizontal_x2 = int((2/3)*height)
        vertical_x1 = int((1/3)*width)
        vertical_x2 = int((2/3)*width)

        x1,y1 = intersection_points(horizontal_x1,0,horizontal_x1,width-1,0,vertical_x1,height-1,vertical_x1)
        x2,y2 = intersection_points(horizontal_x2,0,horizontal_x2,width-1,0,vertical_x1,height-1,vertical_x1)
        x3,y3 = intersection_points(horizontal_x1,0,horizontal_x1,width-1,0,vertical_x2,height-1,vertical_x2)
        x4,y4 = intersection_points(horizontal_x2,0,horizontal_x2,width-1,0,vertical_x2,height-1,vertical_x2)

        dist_ROT = min((np.sqrt((cX-x1)**2 + (cY-y1)**2)),(np.sqrt((cX-x2)**2 + (cY-y2)**2)),(np.sqrt((cX-
x3)**2 + (cY-y3)**2)),(np.sqrt((cX-x4)**2 + (cY-y4)**2)))
        contour_area = cv2.contourArea(contours[0])
        Area_ratio = contour_area/(width*height)
```

```
        ROT_score = ((contour_area*0.05)/(height*width))*(1-((dist_ROT)/(np.sqrt(height**2 + width**2))))
        print('Ratio of area of foreground to the area of image is ', Area_ratio)
        print('The rule of third score of the image is ', ROT_score)
    else:
        Area_ratio = 0.0
        ROT_score = 0.0
        print('Doesnt have an ROI')
    return ROT_score, Area_ratio

def intersection_points(x1,y1,x2,y2,x3,y3,x4,y4):
    x_intercept = ( (x1*y2-y1*x2)*(x3-x4)-(x1-x2)*(x3*y4-y3*x4) ) / ( (x1-x2)*(y3-y4)-(y1-y2)*(x3-x4) )
    y_intercept = ( (x1*y2-y1*x2)*(y3-y4)-(y1-y2)*(x3*y4-y3*x4) ) / ( (x1-x2)*(y3-y4)-(y1-y2)*(x3-x4) )
    return x_intercept, y_intercept
```

## Symmetry Score

```
def distance_from_center(img):

    if ROI==1:
        new_image = np.copy(img)
        new_image[np.where((new_image == [255,255,255]).all(axis = 2))] = [125,125,125]
        new_image[np.where((new_image == [0,0,0]).all(axis = 2))] = [255,255,255]
        new_image[np.where((new_image == [125,125,125]).all(axis = 2))] = [0,0,0]
        gray_image = cv2.cvtColor(new_image, cv2.COLOR_BGR2GRAY)
        ret,thresh = cv2.threshold(gray_image,127,255,cv2.THRESH_BINARY)
        contours = cv2.findContours(thresh,cv2.RETR_TREE,cv2.CHAIN_APPROX_SIMPLE)[-2]
        countour_count = 0
        for c in contours:
            countour_count += 1
        M = cv2.moments(contours[0])
        # calculate x,y coordinate of centeroid
        if M["m00"]==0:
            cX = 0
            cY = 0
            ratio = 100.0
        else:
            cX = int(M["m10"] / M["m00"])
            cY = int(M["m01"] / M["m00"])
            width, height, channel = new_image.shape
            X1 = int((1/2)*height)
            Y1 = 0
            X2 = int((1/2)*height)
            Y2 = int(width - 1)
            dist = distance(cX,cY,X1,Y1,X2,Y2)
            ratio = dist/(0.5*height)
            print('Ratio of distance of the centroid from the center to the half of width of the image is ',ratio)
    else:
        ratio = 0.0
        print('Doesnt have an ROI')
    return ratio

def distance(x0,y0,x1,y1,x2,y2):
    numerator = abs((y2 - y1) * x0 - (x2 - x1) * y0 + x2 * y1 - y2 * x1)
    denominator = ((y2 - y1)**2 + (x2 - x1) ** 2) ** 0.5
    result = numerator/denominator
    return result
```

## Training the model

```
names = ['ROI', 'Foreground Blur Score', 'Background blur score', 'Brightness Score', 'Noise score', 'Simplicity score', 'Contrast score','Area Ratio','Rule of third score','Potrait mode','Landscape Mode','Output']
features = pd.read_csv(url, skiprows=1, names=names)
labels = np.array(features['Output'])
features= features.drop('Output', axis = 1)
feature_list = list(features.columns)

from sklearn.model_selection import train_test_split

train_features, test_features, train_labels, test_labels = train_test_split(features, labels, test_size = 0.20, random_state = 42)

from sklearn.ensemble import RandomForestClassifier

clf=RandomForestClassifier(n_estimators=10)

clf.fit(train_features,train_labels)

y_pred=clf.predict(test_features)

print(clf.score(test_features,test_labels))
```

## Image quality prediction using the machine learning model

```
for i in range(len(test_images)):
    rgb = deeplab(test_images[i])
    cv_image = cv2.cvtColor(np.array(rgb), cv2.COLOR_RGB2BGR)
    h,w,n = cv_image.shape
    resized_image = cv2.resize(test_images[i],(w,h))

    mask = (cv_image == 0)
    segmented_person = np.copy(cv_image)
    segmented_person[mask] = resized_image[mask]

    mask_background = (cv_image == 255)
    background = np.copy(cv_image)
    background[mask_background] = resized_image[mask_background]
    background[np.where((background == [0,0,0]).all(axis = 2))] = [255,255,255]

    print(i)
    ROI = region_of_interest(rgb)
    foreground_blurscore, background_blurscore = blurriness(segmented_person,background)
    brightness_score = brightness(test_images[i])
    sigma_noise = noise(resized_image)
    colorfulness_score = colorfulness(resized_image)
    contrast_score = contrast(resized_image)
    ROT_score , Area_ratio = ROT(cv_image)
    pot, land = potrait_or_landscape(cv_image)

    prediction = clf.predict([[ROI, foreground_blurscore, background_blurscore, brightness_score, sigma_noise, colorfulness_score, contrast_score, Area_ratio, ROT_score, pot, land]])
    print(prediction)
    if prediction == 1:
        high_quality_images.append(test_images[i])
```

# Appendix B

## Screenshot of survey form which is used for evaluation

- Mention at least five or more characteristic features you check while evaluating the quality of an image

  51 responses

  Color, Blurredness, Subject, Background, Lighting, Contrast

  Clarity, light, sharpness, orientation, depth

  No idea

  Blurness, red eye, focus, coverage, light

  Resolution sharpness brightness

  blurness, low light(dark), pixelated, pixelated on zooming, colour quality

  Lighting
  Sharpness
  Camera angle
  Blurness
  Background

  Depth, Noise, Edges, Colour, Skin tones

  Colour
  Clarity
  Noise level
  Brightness

- Do you find this image of high or low quality ? *

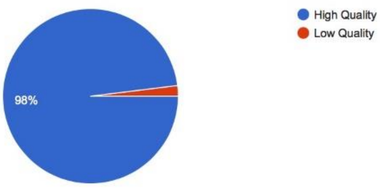
  ◯ High Quality

  ◯ Low Quality

  ◯ Other...

Do you find this image of high or low quality ?

51 responses



- Do you find this image of high or low quality ? *

○ High Quality

○ Low Quality

○ Other...

Do you find this image of high or low quality ?

51 responses