Profile Photo Rating Service — Using Deep Learning Models

Authors: Venkatakrishna Panga, Bhavna Saxena, Debiprasad Banerjee, and Manish Sinha

Abstract: Websites and apps like Match, Tinder, LinkedIn, Care.com, etc., use profile pictures to introduce a person to their users. They say a 'picture is worth a thousand words', and it couldn't be truer for these websites and apps. Profile pictures from the first and often, a lasting impression of the user. A professionally clicked profile picture conveys sincerity, tardiness, and creates an emotional connection with the users. And a collection of such profiles with good profile pictures make the website or the app look professional, which in turn, leads to higher website and app visits and greater conversions.

We looked at various aspects of these pictures like sharpness, brightness, and contrast using OpenCV based deep learning (DL) algorithms. We also looked at several subjects in a picture (a profile picture should have only one). We also analyzed the subject itself for the area of the face, the angle of the face, whether it is centered or not, whether it is covered or not, etc. Using these parameters, we built an AI algorithm to rate a picture on a scale of 1 to 10, 10 being the highest rating one can receive.

Disclaimer: *Gender, color, age, and aesthetic beauty are not taken into any consideration by this algorithm*

Keywords: Profile photo rating, Facial expressions, Face detection, Image quality, Background detection, Image score, Object detection.

I. **Introduction**: Computer vision [1]-[2] is one of the most popular research fields of Deep Learning. There are many models implemented for face recognition, object detections, segmentation, etc. Machines' computational power and processing ability have increased tremendously over the past few decades, which in turn, has helped us run high computational machine learning (ML) models with low processing time. We have developed a photo rating algorithm by using pre-trained models and statistical equations. Identified patterns, measured raw values, and the applied custom logic on top of it all convert to a 'score' or a number between the range of 1 to 10, where the maximum score is 10 and the minimum score is 1.

We have used a quality matrix to measure the quality of the entire picture, used deep learning [3]-[5] models, and statistical calculations to detect faces, different parts of the face, age, objects in the picture, and the background. With this, we have covered all parts of the profile picture and considered all parameters while rating.

The remaining paper is divided into the following sections

II. Image quality,

III. Facial attributes,

IV. Background detection,

V. Integration,

VI. Validation and Rectification block,

VII. Weights calculation,

VIII. Scores Distribution

II. **Image quality**: Image quality has a key role to play in the profile picture rating process. It increases user engagement by grabbing more attention

and establishing friendliness and approachability. Three parameters that we considered to assist quality were — brightness, sharpness, and contrast.

II.I. Brightness: Brightness [6] is the perceived intensity of light coming from a screen. It gives information about brighter pixels and darker pixels on the picture. On a color screen, it is the average of the red, green, and blue pixels that transforms into a “perceived brightness”. On a gray screen, it is the average of the gray pixels. The formula to calculate brightness is:

$$r,\ g,\ b\ =\ \Sigma\frac{r}{n},\ \Sigma\frac{g}{n},\ \Sigma\frac{b}{n}$$

$$\text{brightness}\ =\ \sqrt{(0.241\times \mathrm{r}^2)+\ (0.691\times \mathrm{g}^2)\ +\ \ (0.068\times b^2)}$$

(1). r, g, and b are pixels of the picture

II.II. Contrast: It is defined as the degree of difference between the lightest and the darkest parts of a picture. The human visual system is more sensitive to contrast hence it plays an important role in assisting picture quality. In mathematical terms, the entropy of the picture is used as a measure of contrast. Contrast is calculated by taking the sum of the logarithmic values of histograms.

$$\text{contrast}\ =\ -\ \Sigma(\mathrm{p}\times \log_2(\mathrm{p}))$$

(2). where p contains the histogram counts

II.III. Sharpness: It describes the clarity of detail in a picture and can be a valuable creative tool for emphasizing texture. When the subject in the picture is sharp, he/she appears clear and lifelike, with detail, contrast, and texture rendered in high detail. Pictures with low blurriness will have high sharpness. A variation of the Laplacian was used to calculate the sharpness

[7], [8]. We took a single channel of a picture (presumably grayscale) and convolved it with the 3 x 3 kernel. We, then, took the variance (i.e., standard deviation squared) of the response. If the variance fell below a pre-defined threshold, then the picture was considered of low sharpness.

$$\nabla^2 image = 0$$

$$\partial^2 image = 0$$

3). the second-order partial derivative of the picture

III. **Facial attributes:** Facial attributes describe facial expressions, face alignment, face tilted measurement, and face visibility. These parameters play a major role while rating a picture — a good profile picture should have a single person in the photo with him/her face straight, and looking at the camera, center-aligned with no tilted head. We have used DL models for face detection, facial parts' location identification, and age calculation. Outputs of these models were passed to statistical equations to measure facial attributes and parameters. A total of seven parameters were implemented — face count, face area, face distance, angle, smile, Open/closed eye analysis, and age.

III.I. Face count: An ideal profile picture should only have a single person in the frame, so it is important to count the number of people in a picture. The profile photo rating will decrease as the number of people in it, increase. We have used DL models for face detection and facial parts' location identification.

See sections III.I.I Face detection Model and III.I.II Face parts identification model for model information.

We have developed detection scripts and tested with Haar cascade [9]-[10][18], Histograms methods, and some other naive DL models, but they were not up to the mark. So, we finalized the ResNet model [12] as it was accurate and faster than the others. This face detection model gives face co-ordinates with confidence — confidence tells us the probability of having a face in the bounding box. A high probability indicates higher chances of having a face inside the frame. We then calculated the number of faces using this model and passed the results to Face ROI for facial parts' detection model [11], [13]. It identified eyes, nose, mouth, jaw, left eyebrow, and right eyebrow with the help of 68 vectors. We then measured the remaining parameters using these 68 vectors.

III.I.I Face detection model: Fast, accurate face detection can be performed with this model [12]. This deep learning face detector is based on the Single Shot Detector (SSD) framework with a ResNet base network (unlike other SSDs that use MobileNet as the base network). This is also favorable as it is a lightweight model that can produce faster results.

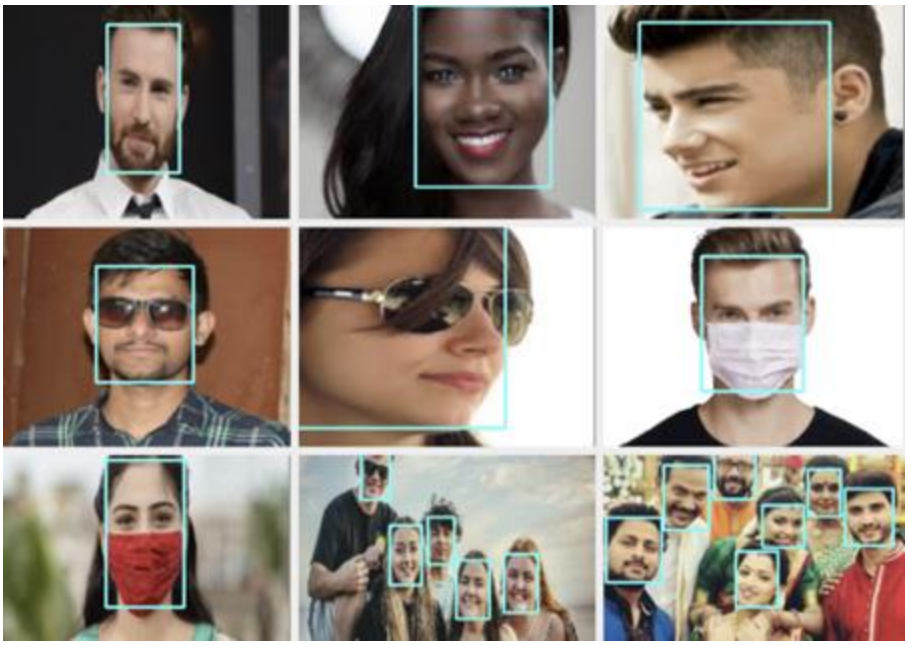

Fig. 1: Face detection on various pictures

III.I.II Facial parts identification model: Next, we needed to detect key facial parts in the face region. A training set was generated by manually labeling specific (x, y)-coordinates of the regions surrounding each facial part. We also took into consideration the distance between the pairs of input pixels, hence determining the distance from one facial part to another. An ensemble of regression trees was trained with this training data to estimate the positions of the facial parts in an average human face. This is a 68 point-based landmarks detector. This can identify eyes, nose, mouth, jaw, left eyebrow, and right eyebrow. Find the complete description of the model in [11],[13], [19].

Fig. 2: Face part detection output in each stage

III.II. Face Area: The face area describes the average area that the face occupies in the entire profile picture. Visibility of the face is more critical to rating profile pictures than the visibility of other parts, such as the head and the shoulders. Also, pictures taken from a distance have less face area than close shots. We used the face co-ordinates given by the face detection model to calculate the face area. The ratio of face area to the total area in the picture gives the face area percentage.

$$\text{area} = \frac{(h \times w)}{(H \times W)} \times 100$$

(4). h and w are the height and width of the face. H and W are the height and width of the picture

III.III. Face Distance: Pictures with centered face alignment get a higher rating than cornered or side-aligned faces. We have taken facial parts locations from the model, calculated the face-centered point,x and then, divided the entire picture into four coordinates, and identified the coordinate in which the face lies. We assigned positive values for the first and second coordinates and negative values for the third and fourth. Positive values get more weightage than negative values — a detailed explanation of weights calculation will be discussed in VI.

$$\text{face distance} = \sqrt{(\frac{W}{2} - x_m)^2 + (\frac{H}{2} - y_m)^2}$$

$$\text{diagonal distancee} = \sqrt{W^2 + H^2}$$

$$\text{face distance ratio} = \text{face distance} / \text{diagonal distance}$$

(5). xm, ym is the center point of the face

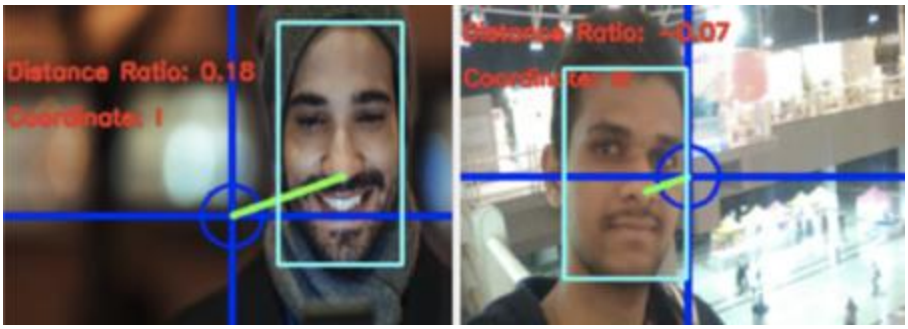

Fig. 3: Face distance results

III.IV. Angle: Angle parameter is used to measure the head pose — whether it is straight and looking at the camera or titled towards the right or left. Good profile pictures should not have a tilted head in them. The angle is zero degrees when the head is straight and increases as the head tilts. By looking at the deviation between the two eyes from a straight line we measured slop, and then converted slop to angle. Here is the detailed calculation:

$$dY = Y_{le} - Y_{re}$$

$$dX = X_{le} - X_{re}$$

$$\text{radians} = \tan^{-1}\left(\frac{dY}{dX}\right)$$

(6). le refers to the left eye and re refers to the right eye

Fig. 4: Angle measurement results

III.V. Open/Closed Eye analysis: The eye aspect ratio is an elegant solution that involves a very simple calculation based on the ratio of distances [14] between facial landmarks of the eyes. Each eye is represented by 6 (x, y)-coordinates, starting at the left corner of the eye (as if you were looking at the person), and then working clockwise around the remainder of the region. With this, we have been able to derive an equation that reflects this relation called the eye aspect ratio (EAR).

$$A = \sqrt{(Y_{p2} - Y_{p6})^2 + (X_{p2} - X_{p6})^2}$$

$$B = \sqrt{(Y_{p3} - Y_{p5})^2 + (X_{p3} - X_{p5})^2}$$

$$C = \sqrt{(Y_{p1} - Y_{p4})^2 + (X_{p1} - X_{p4})^2}$$

$$EAR = \frac{A + B}{2 \times C}$$

(7). p1 refers to the first point in the eyes, a total of 6 points denote eye

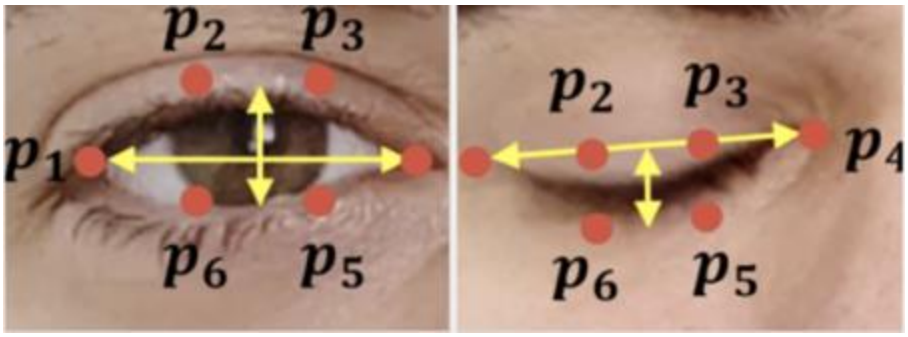

Fig. 5: Eye detection using 6 points.

III.VI. Smile: As it is believed that people smiling in their profile pictures makes them look more friendly and approachable, we considered smile as an important parameter to rate profile pictures. We detected smile status by taking the ratio of the distance [14] between the two endpoints of the lips. Mouth and lips locations were received from the DL model. MAR (mouth aspect ratio) is the ratio of vertical distance to horizontal distance. We calculated MAR and experimented with several values of several pictures before deciding the final threshold point. If the MAR ratio is greater than the threshold, then the person in the picture is said to be smiling.

$$\mathrm{A} = \sqrt{(\mathrm{Y_{m51}} - \mathrm{Y_{m59}})^2 + (\mathrm{X_{m51}} - \mathrm{X_{m59}})^2}$$

$$\mathrm{B} = \sqrt{(\mathrm{Y_{m52}} - \mathrm{Y_{m58}})^2 + (\mathrm{X_{m52}} - \mathrm{X_{m58}})^2}$$

$$\mathrm{C} = \sqrt{(\mathrm{Y_{m53}} - \mathrm{Y_{m57}})^2 + (\mathrm{X_{m53}} - \mathrm{X_{m57}})^2}$$

$$\mathrm{D} = \sqrt{(\mathrm{Y_{m49}} - \mathrm{Y_{m55}})^2 + (\mathrm{X_{m49}} - \mathrm{X_{m55}})^2}$$

$$\mathrm{MAR} = (\mathrm{A} + \mathrm{B} + \mathrm{C})/(3 \times \mathrm{D})$$

(8). MAR formula

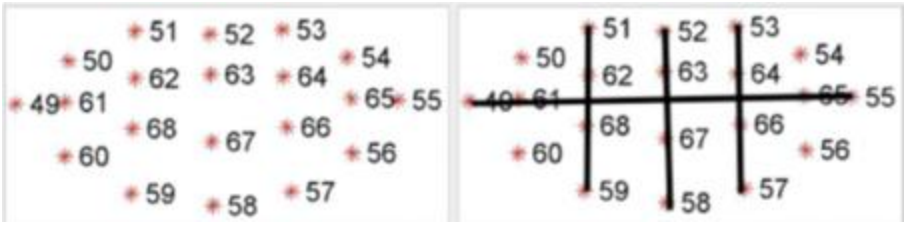

Fig. 6: Mouth detection using 20 points

III.VII. Age detection: Age detection is the process of automatically discerning the age of a person solely from a picture of their face. Extracted face Region of Interest (ROI) is applied to the age detector model to predict the age of the person. We divided age into eight segments and converted the problem into a classification problem. This is not to be considered a regression problem since age prediction is inherently subjective and based solely on appearance. No weightage was allocated for rating the picture, but this was solely used to filter out kids when there was any in the profile picture.

III.VII.I. Age detector model: The deep learning age detector model we used here was implemented and trained by Levi and Hassner in their 2015 publication [16]. The authors propose a simplistic AlexNet-like architecture that learns a total of eight age buckets: 0–2 years, 4–6 years, 8–12 years, 15–20 years, 25–32 years, 38–43 years, 48–53 years, and 60–100 years.

IV. Background detection: It is the process of identifying objects present in the background. Good profile pictures should have a plain background with no objects present in the frame. The person should be standing closer to the camera, giving him/her higher visibility. The two parameters developed in these criteria are (i) Object detection, and (ii) Person occupied area.

IV.I. Object detection: YOLOV3[17] model was used to detect objects in the frame. It is a custom object detector network that looks at the whole picture

at once and can make accurate predictions. YOLOV3 is one of the faster object detection algorithms out there, capable of performing at 30 fps. A total of 80 objects can be detected in a single frame. The model has also been trained on the COCO data set, based on IOU and Non-max suppression techniques. With this, the model can predict bounding boxes of objects. We have used this model to detect the number of objects in one frame and allot weights to every number. The more the number of objects in the picture, the lesser the rating.

Fig. 7: Object detection results

IV.II. Person Occupied Area: Person area percentage is the measure of the body-covering portion in the entire picture. Pictures having their face and body occupying most of the frame get a higher rating than a picture where the face is occupying only a small portion or a corner of the frame. Person Body ROI received from YOLOV3 is used to calculate person occupied area. The ratio between person occupied area to the entire frame area gives Person occupied area percentage. Here's the formula:

$$\text{area} = \frac{(\text{h} \times \text{w})}{(\text{H} \times \text{W})} \times 100$$

(9). h and w are the height and width of the person. H and W are the height and width of the picture

V. Integration: It is the process of designing an integration flow and combining each parameter according to the flow. Image quality parameters and background detection parameters were calculated irrespective of the number of people in the picture, but facial attributes were calculated for pictures with one or two people only. For pictures with two people, we used the age detection model to eliminate the child in the picture to consider only the one adult for rating.

For pictures with three or more people in them, we did not follow through with the calculation of the facial attributes as it becomes difficult to measure the exact facial attributes of each face, and there is a high chance of getting an error. We combined the raw values of each parameter as per the integration flow and then sent them to the Validation and Rectification block.

VI. Validation and Rectification block: We developed this block to verify the face count of a face detection model with the person count of the YOLOV3 model. Sometimes models give wrong results — one model detection may differ from another, and to eliminate such errors and build an error-free algorithm, we implemented this block. When the face count and the person count didn’t match, we have introduced a penalty.

Confidence is the other parameter we implemented at this stage to measure face detection accuracy. Pictures, where the faces are covered with either sunglasses or hair were assigned less confidence. We experimented with many pictures, categorized them into different batches, analyzed results, and identified the confidence threshold for each category. All parameters were then sent to the next block for the final score calculation.

VII. Weights calculation: So far, we had calculated the raw value of each parameter. We converted raw values into weights and produced a final score out of 10. Good pictures got the maximum score of 10, while really bad pictures got a score of 1. We distributed weights for each parameter based on their importance. We developed normalized functions and quartiles for parameters to convert raw values into weights as allocated for each parameter. The Validation and Rectification block added penalties whenever the detections didn't match. We did recursive testing and experimented, validated scores with pictures, and then finalized weights.

Below is the process flow diagram with calculated weights of each parameter for a sample picture (Fig. 8).

Fig. 8: Image with the final score. This image with it's each parameter weights shown in the Flow diagram (Fig. 9)

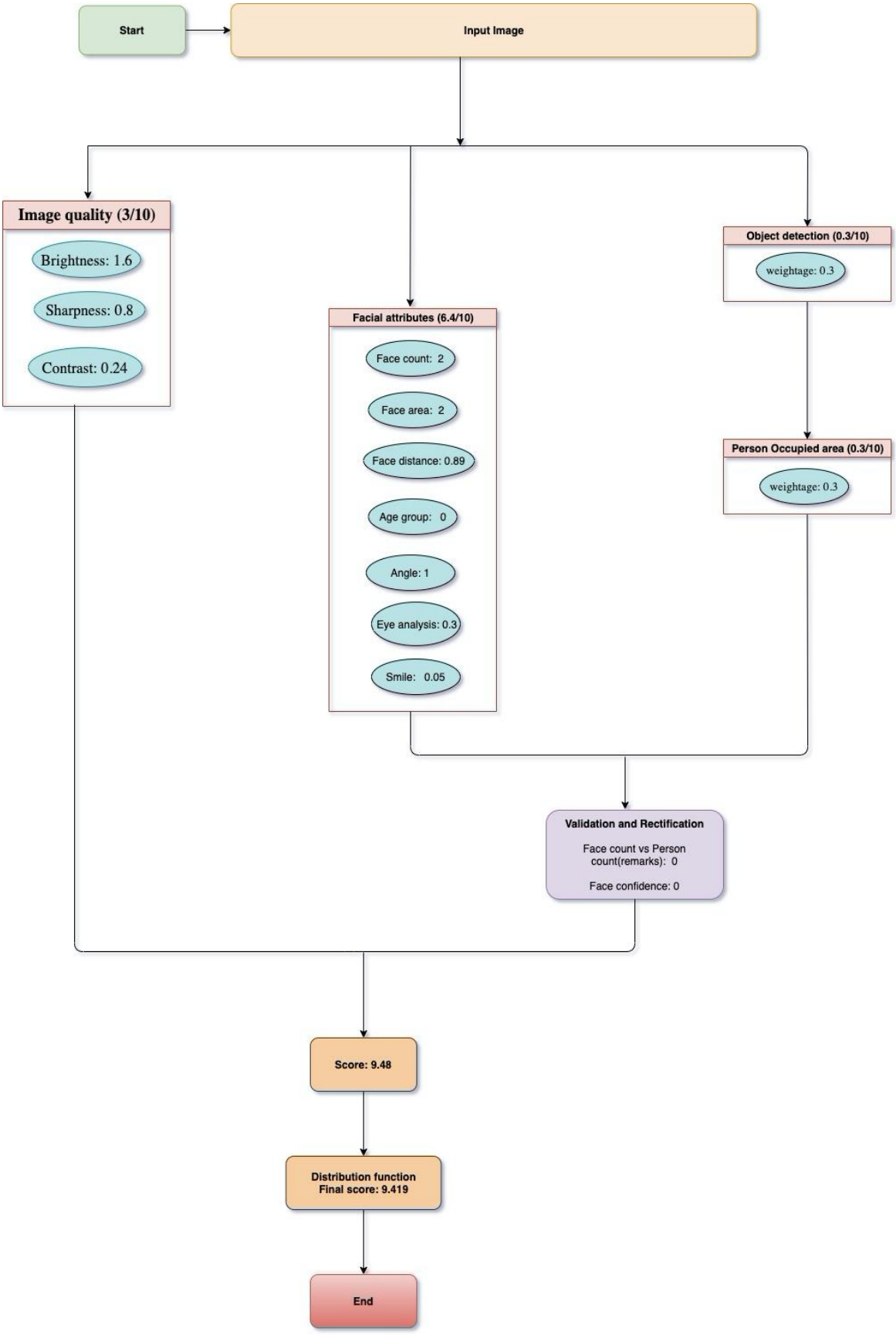
Start
Input Image
Image quality (3/10)
Brightness: 1.6
Sharpness: 0.8
Contrast: 0.24
Facial attributes (6.4/10)
Face count: 2
Face area: 2
Face distance: 0.89
Age group: 0
Angle: 1
Eye analysis: 0.3
Smile: 0.05
Object detection (0.3/10)
weightage: 0.3
Person Occupied area (0.3/10)
weightage: 0.3
Validation and Rectification
Face count vs Person count(remarks): 0
Face confidence: 0
Score: 9.48
Distribution function
Final score: 9.419
End

Fig. 9: Flow diagram and weights of each parameter

In the above flow diagram, we can see the total score is being divided into three parts:

1. Image quality section allocated with a maximum weightage of 3
2. Facial attributes section allocated with a maximum weightage of 6.4, and,
3. Background section allocated with a maximum weightage of 0.6.

Each parameter was calculated individually and then passed to the distribution function to get the final score.

VIII. Scores Distribution: We developed a statistical distribution function to get to the final scores for the profile pictures. This aimed to evenly distribute scores across all ranges. The calculated score in the previous stage is passed to this function for the final score conversion. The output of this function gives results like the gaussian curve (bell curve), as most of the ratings lie in the upper scale range.

Fig. 10: Final scores

IX. Advantages: This service is capable of measuring image quality, facial attributes, and background detection parameters in the picture, which

means we have looked at every possible corner of the picture and considered all criteria for implementation. We have used advanced deep learning models for predictions and statistical equations for calculations. The entire project was developed using a single library — OpenCV.

We also implemented backward compatibility, so that the feature releases can be easily integrable, and existing functionalities can be reused. This service is capable of scoring colored as well as pictures in grayscale. Extra validation and rectification blocks were implemented to minimize errors. Distribution functions were developed to evenly distribute scores across all ranges. The measured parameters were prioritized, and weights were allocated accordingly. The final score is the consolidated score of every parameter, hence if there is a wrong calculation in any one of them, it would not impact the final result.

X. Future work: Age is not detected properly in a few of the pictures. Although age does not have any weightage, it still is considered to eliminate the child in the pictures. Faces covered with either sunglasses or hair were not identified properly in a few pictures. Multiple pre-trained models were used to measure parameters, and the same can be solved using a single custom-built model. These are a few of the scenarios we are aiming to develop as part of the next release.

XI. **Conclusion**: Gender, Color, Age, and Beauty are not taken into consideration for rating the profile pictures. We have considered all criteria for rating photos. It was tested and verified with several pictures, it measured most of the parameters accurately. These ratings can be used as a sorting parameter along with other attributes to showcase profiles.

References

[1] D. H. Ballard and C. M. Brown: “Computer Vision”, NJ, Englewood Cliffs: Prentice-Hall, 1982

[2] Shirai: “Three-Dimensional Computer Vision”, West Germany, Berlin: Springer, 1987.

[3] Sotiris B Kotsiantis, I Zaharakis, and P Pintelas: “Supervised machine learning: A review of classification techniques. Emerging artificial intelligence applications in computer engineering”, 160:3–24, 2007.

[4]. Bhiksha Wang, and HaohanandRaj: “On the origin of deep learning”. arXiv preprint arXiv:1702.07800, 2017.

[5] Amitha Mathew, P.Amudha, and S.Sivakumari3: “Deep Learning Techniques: An Overview”, India

[6] Sergey Bezryadin, Pavel Bourov, and Dmitry Ilinih: “Brightness Calculation in Digital Image Processing”, India, Research gate

[7] Said Pertuz a,n, Domenec Puig a, and Miguel Angel Garcia b: “Analysis of focus measure operators for shape-from-focus”, 2012.

[8] Online: https://www.pyimagesearch.com/2015/09/07/blur-detection-with-opencv/ *by* Adrian Rosebrock *on* September 7, 2015.

[9] Online: https://github.com/abidrahmank/OpenCV2-Python-Tutorials/blob/master/source/py_tutorials/py_objdetect/py_face_detection/py_face_detection.rst

[10] Online: https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_objdetect/py_face_detection/py_face_detection.html

[11] V. Kazemi and, J. Sullivan: “One millisecond face alignment with an ensemble of regression trees”, 2014.

[12] Online: https://github.com/opencv/opencv/tree/master/samples/dnn
[13] Online: https://ibug.doc.ic.ac.uk/resources/facial-point-annotations/
[14] Barrett O'Neill: "in Elementary Differential Geometry (Second Edition)", 2006, 1.2 Definition
[15] Tereza Soukupova, and Jan Cech: "Real-Time Eye Blink Detection using Facial Landmarks" in 2016
[16] Gil Levi and, Tal Hassner: "Age and Gender Classification using Convolutional Neural Networks".
[17] Redmon: "You Only Look Once: Unified, Real-Time Object Detection" in 2015
[18] Online: https://github.com/skvark/opencv-python
[19] Online: https://github.com/davisking/dlib