 doi:10.20944/preprints202108.0392.v1

# An Integration of Deep Network with Random Forests Framework for Image Quality Assessment in Real-Time

Zahi Al Chami*[1,2], Chady Abou Jaoude†[1], and Richard Chbeir‡[2]

[1]Antonine University, Faculty of Engineering - TICKET Lab, Beirut, Lebanon

[2]Université de Pau et des Pays de l'Adour, E2S UPPA, LIUPPA, Anglet, France

**Abstract**

In recent years, data providers are generating and streaming a large number of images. More particularly, processing images that contain faces have received great attention due to its numerous applications, such as entertainment and social media apps. The enormous amount of images shared on these applications presents serious challenges and requires massive computing resources to ensure efficient data processing. However, images are subject to a wide range of distortions in real application scenarios during the processing, transmission, sharing, or combination of many factors. So, there is a need to guarantee acceptable delivery content, even though some distorted images do not have access to their original version. In this paper, we present a framework developed to estimate the images' quality while processing a large number of images in real-time. Our quality evaluation is measured using an integration of a deep network with random forests. In addition, a face alignment metric is used to assess the facial features. Experimental results have been conducted on two artificially distorted benchmark datasets, LIVE and TID2013. We show that our proposed approach outperforms the state-of-art methods, having a Pearson Correlation Coefficient (PCC) and Spearman Rank Order Correlation Correlation Coefficient (SROCC) with subjective human scores of almost 0.942 and 0.931 while minimizing the processing time from 4.8ms to 1.8ms.

**Keywords:** Image Quality Assessment, Real-Time Image Processing, Image Functions Adaptation, Convolutional Neural Network, Face Alignment, Deep Neural Network, Random Forest

## 1 Introduction

With the ongoing advances in technology, data providers are producing and streaming a significant amount of data. In particular, the huge interest in the development and usage of multimedia-based applications

*Corresponding Author: zahi.chami@ua.edu.lb

†chady.aboujaoude@ua.edu.lb

‡richard.chbeir@univ-pau.fr

has led to an enormous multimedia traffic production on the global inter-network. Newsfeeds, podcasts, live interviews, and real-time content delivery are examples of real-time multimedia applications, systems, and solutions that have provoked the rapid growth in multimedia traffic. In addition, the huge amount of multimedia data received and produced at a rapid pace brings concerns towards data content processing since the traditional approaches do not scale well for data streaming scenarios. These approaches require data to be first stored before being processed, which takes a significant amount of time.

More particularly, the majority of the streamed and generated data are images due to the social media photo-sharing applications' growth. According to [1], over 300 million images are posted daily to Facebook, while over 95 million photos are uploaded daily to Instagram. These streams could be exposed to alteration or modification, such as applying adaptation/protection functions to satisfy users' needs. For example, blurring a person's face in an interview to conceal his/her identity, removing critical information from a Twitter stream, or shedding light on only the salient objects in a video because of certain hardware or network limitations. Hence, the content's outcome must be assessed to guarantee an acceptable trade-off between the quality of the delivered content and the expected result. More specifically, and in recent years, treating images that contain faces has gained a lot of attention due to its wide range of applications, including video surveillance, entertainment, etc. More specifically, and in recent years, treating images that contain faces has gained a lot of attention due to its wide range of applications, including video surveillance, entertainment, etc. Since low-quality images limit the utility of these applications, it is necessary to assess the images' visual features before publishing. This can be done through structure and semantic content evaluation. For example, ensuring that the faces are kept intact when applying content adaptation functions and guaranteeing that some useful information can still be extracted after using content protection functions.

Normally, estimating an image quality is achieved by comparing the distorted image with an ideal imaging model or perfect reference image (a.k.a Full Reference Image Quality Assessment, abbr. FR-IQA). But, in most real-time streaming scenarios, the original image is not available. In such cases, the FR-IQA approaches (for example, SSIM [2], PSNR [3], Content-Based Image Retrieval [4], etc.) can't be used to measure image quality degradation since they require the presence of the distortion-free images. Consequently, evaluating an image's quality blindly has been becoming increasingly important (a.k.a No Reference Image Quality Assessment, abbr. NR-IQA). Therefore, we need to directly quantify image degradations by exploiting features that are discriminant for image degradation. Several approaches have been suggested and received attention in this field. They extracted Natural Scene Statistics (NSS) using the wavelet transform [5–7] or the DCT transform [8]. These methods are very slow since costly image transformations were used. Motivated by the recent success of the Convolutional Neural Networks (CNNs) for image classification

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 18 August 2021 doi:10.20944/preprints202108.0392.v1

tasks due to its deeper structure [9], there are several existing approaches [10–12] that quantify the image quality degradation based on the CNN. Despite the fact that these approaches achieve high accuracy, they have high time complexity due to an excessive number of multiplications between the images and the layers, which will increase its response time when classifying these images. Therefore, this will limit their use in real-time applications. To clarify the previous points, we provide the following scenario.

### 1.1 Motivating scenario

Let us take, for example, the situation of a photo-sharing company (shown in Figure 1), which offers its customers the ability to share and publish images online. The company is demanded to instantly process these images as it receives an unbounded stream of distorted and undistorted images. Moreover, the company provides additional services such as:

- Protecting his/her identity to avoid disclosing sensitive information using several techniques, such as masking functions.
- Adapting their images to meet the limitations imposed by the available resources—for example, image compression.

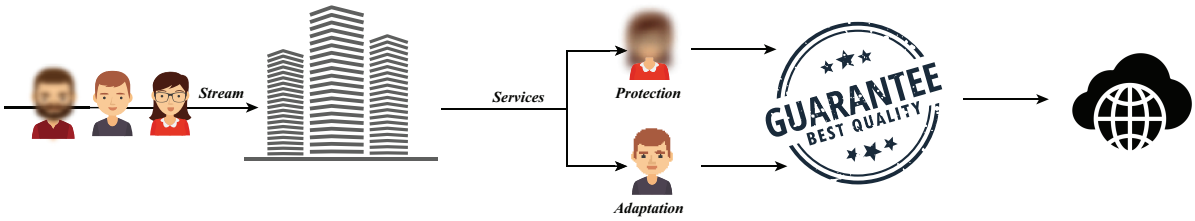



Figure 1: Scenario.

Although the users are getting benefits from these services, the images' content could be damaged or removed. Therefore, the customers start sending their complaints about image quality degradation. So, the company is looking for a new solution that satisfies the needs of the users. In summary, the framework that should be implemented in the company must be able to:

- Estimate the images' quality: assessing the quality of the distorted images that were already affected during the transmission or processing phase. These images are considered the most challenging cases since they do not have access to their reference images.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 18 August 2021 doi:10.20944/preprints202108.0392.v1

- Preserve the images' quality: ensuring that the modified images' features, such as color, shape, texture, etc., are remained intact and can still be extracted from the output after applying the previously mentioned services.

- Handle the unbounded stream of images: processing the multimedia data content requires higher bandwidth, bigger memory, and faster computational resources. Therefore, it forces strict quality of service requirements and demands efficient network architecture. For this reason, we need to find a way to treat the significant number of images by ensuring a successful image processing and reliable delivery of the data instantly and in real-time, even though the previously mentioned services, along with the quality assessment process, could take much time.

Several existing works tried to meet these objectives using many methods/techniques cited, along with their limitations, in the next section before presenting our proposed approach.

## 2 Related Work

In this section, we compare various existing works to our approach based on the following criteria: 1) The extent of the useful images' information that remained intact when addressing business or users' needs, such as adaptation or protection, 2) The metrics used to estimate the images' quality; this will indicate the number of features learned as well as the quality prediction accuracy, and 3) The amount of time it takes to process the images, especially the proposed approaches that work in real-time.

Image quality assessment measures are typically classified into three categories: Full-Reference Image Quality Assessment (FR-IQA), reduced-reference (RR) IQA, and No-Reference (NR) IQA. To assess the quality of the distorted images, FR-IQA requires reference images. To compute the quality measure in RR-IQA, partial information from a reference image is needed. As for the NR-IQA, it provides the quality without the need for any reference image. In the following sections, we will present the techniques used to assess the images' quality and their limitations. However, we mainly focus on the FR and NR categories since they are mostly encountered in real-time streaming scenarios.

### 2.1 Full-Reference Image Quality Assessment

Although traditional signal fidelity measures like mean square error (MSE) and peak signal-to-noise ratio (PSNR) have no consideration of characteristics of an image signal and the HVS, they are still widely used as FR measures [3]. However, it does not correlate with perceived visual consistency. This resulted in

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 18 August 2021 doi:10.20944/preprints202108.0392.v1

developing a whole zoo of image quality metrics to improve the agreement with human perceptions of image quality.

In [13], the authors proposed a method to lossy image compression that generates files 2.5 times smaller than JPEG and JPEG 2000 while maintaining images' quality. The proposed approaches in [14, 15] assessed the distorted faces in real-time using objective quality methods. While these techniques provide valuable results, they assessed the distorted images by analyzing only the color and structure features without considering the other features that may be damaged and led to content degradation. In [16], a hybrid feature descriptor-based method is proposed to recognize human emotions from their facial expressions. A combination of a spatial bag of features (SBoFs) with spatial scale-invariant feature transform (SBoF-SSIFT), and SBoFs with spatial speeded up robust transform are utilized to improve the ability to recognize facial expressions. A new feature descriptor called Histogram of Oriented Gradients from Three Orthogonal Planes (HOG-TOP) is proposed in [17] to extract dynamic textures from video sequences to characterize facial appearance changes. And a new effective geometric feature derived from the warp transformation of facial landmarks is proposed to capture facial configuration changes.

The suggested solution in [18] provides a novel DQAMLearn framework that aims to support mobile learner's seamless access to educational multimedia content from a variety of mobile devices with different characteristics. Moreover, as mobile users are increasingly becoming quality-aware, the framework integrates novel mechanisms for decreasing the video quality in a controlled way, with the aim to support a good learner quality of experience (QoE) even in resource-constrained situations. In [19], the authors propose an application-layer and middleware-based solutions that increase network reliability and flexibility and provide Quality of Service (QoS) control based on Scalable Video Coding (SVC). The open-source Scalable Video-streaming Evaluation Framework (SVEF) tool has been used to assess the video transmission performance with performance metrics, Viz. Peak Signal to Noise Ratio (PSNR) and Mean Opinion Score (MOS). In [20], The authors present the architecture of an adaptive multimedia learning service, where their engine enables users to identify the best combination of adaptive features of visual and audio content.

Machine learning has been partially adopted in Full-Reference image quality assessment (FR-IQA). Instead of directly combining quality-related image features, some IQA metrics are based on learning techniques for feature discovery and integration. An obvious advantage of using machine learning techniques in feature integration is that the model can be mathematically optimal and therefore has superior performance. A singular value decomposition (SVD) based measure was proposed in [21], and the authors first calculated the distance between the singular values of the reference image blocks and distorted image blocks. They then computed a global value from each block to represent the final quality. Liu et al. [22] introduced a

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 18 August 2021 doi:10.20944/preprints202108.0392.v1

novel parallel boosting measure that inherited the advantages of some state-of-the-art FR measures. Specifically, the authors utilized the SVR to integrate the quality features extracted by state-of-the-art FR measures. In [23], multiple features were extracted from the difference of Gaussian frequency bands and regressed onto the quality score.

### 2.2 No-Reference Image Quality Assessment

Most of the state-of-the-art IQA methods [24–26] train their network to predict the distorted images' quality using human subjective quality scores that are available in several datasets, e.g., the images in the TID2013 [27] and LIVE [28] databases. All the previously mentioned methods follow a two-step framework: feature extraction and model regression by human scores. The IQA metric in [24] creates and combines image pairs within individual databases as the training set, which effectively bypasses the issue of perceptual scale realignment. The authors compute a continuous quality annotation for each pair from the corresponding human opinions, indicating the probability of one image having better perceptual quality. The authors in [26] proposed an IQA metric based on deep meta-learning. They first trained the model based on a number of distortion-specific NR-IQA tasks to learn a meta-model. The latter can capture the humans' shared meta-knowledge when evaluating images with various distortions, enabling fast adaptation to the NR-IQA task of unknown distortions.

There are several existing studies using no-reference IQMs to assess faces' quality. The "MagFace" approach from Meng et al. [29] expanded on the idea of FR with integrated Face Image Quality Assessment (FIQA). In contrast to previous approaches such as ProbFace [30], the data uncertainty learning approach from [31], or PFE [32], MagFace does not have separate quality or uncertainty output at all. Instead, the quality is directly indicated by the magnitude of the FR feature vector. The approach works by extending the ArcFace [33] training loss, changing the angular margin to a magnitude-aware variant, and adding magnitude regularization.

Motivated by the recent success of CNNs for image classification tasks, there are several Deep Neural Network-based approaches [12,34,35] to image quality assessment. It can be used in a no-reference as well as in a full-reference IQA. The authors in [36] rely on extracting feature vectors from the distorted and reference image to be then concatenated together while assigning weights for each region. They showed superior performance compared to the state-of-the-art. However, the authors did not show the execution latency and time complexity. This method may take more time to assess an image since the authors consider each region and the distorted parts, limiting their use in real-time applications. In [37], they train a CNN to predict the objective scores of all metrics by their proposed Multi-Task Learning (MTL) framework. Afterward, they use

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 18 August 2021 doi:10.20944/preprints202108.0392.v1

this framework to extract features to train another small regression network for subjective score prediction. We present, in the following Table, the previously mentioned approaches along with their limitations.

Table 1: Showing the content and the limitations of each approach

| Cited approaches | Content | Limitations |
|---|---|---|
| Full-Reference IQA | | |
| [13–15] | The proposed methods maintain the image quality, while processing the images in real-time | Their use in many real-time streaming scenarios could be unpractical, especially when the original image is not present |
| [18–20] | They provide an end-to-end quality assessment framework that could guarantee a high level of QoS | They assessed a limited number of features in order to identify the remaining useful images' features when adapting the content |
| [21–23] | They train a regression model to predict the image quality score using multiple features | |
| No-Reference IQA | | |
| [29–31] | A universal 512-D face feature representation is provided to measure the quality of a given face | Even though these techniques achieve state-of-arts results, their use could be limited in real-time applications due to the structural complexity of the trained models |
| [26, 34–36] | The authors used the FR-IQA methods to annotate and train their CNN models. | |

Our work aims to find a fair trade-off between the quality of the altered content and the users' expected outcome in real-time, as detailed in the next section.

## 3 Contributions

According to [38], the authors showed that the random forests have the best time complexity among all machine learning models. This will allow us to process and predict the images' quality score within a short time. However, the authors revealed that their use for large-scale multi-class image classification is unpractical since they cannot classify data of high dimensions. For this reason, we propose a faces quality estimation contained in the images by integrating a Deep Neural Network with the random forests while processing these images in real-time. Hence, we first apply a feature extraction process through the use of a Deep Neural Network to reduce the images' dimension by keeping the useful information before classifying

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 18 August 2021 doi:10.20944/preprints202108.0392.v1

the images according to their quality scores using the random forests. Thus, this combination will minimize the inference time comparing to any single-handedly Deep Network. We assume that the faces are affected by adaptation or protection functions to be evaluated on the fly.

Our contributions can be summarized as follow:

- We propose a faces quality estimation in the images by integrating a Deep Neural Network with the random forests. We opt to choose the Convolutional Neural Network (CNN) as our Deep Network since it is the most commonly used when it comes to analyzing images. After extracting the features, we train the random forests using three Full-Reference metrics:
  - Structural Similarity Index (SSIM) [2].
  - Content-Based Image Retrieval (CBIR) [4].
  - Perceptual Coherence Measure (PCM) [14].
- We come up with a second metric that will allow us to assess the facial features in an image using a face alignment metric while reducing the dimension of the face feature vector.

  It is worth mentioning that the two main differences between our method and the state-of-arts methods are: 1) it is the first time an integration of CNN with the random forest has been used to predict the images' quality while considering such a number of features to train our model, and 2) an unprecedented combination composed of a face alignment metric with the previously trained model.
- We develop a framework with the capability of evaluating a stream of images efficiently while estimating its quality.

The remainder of this article is organized as follows. Section §4 presents some definitions and terminologies used in our work. The Machine Learning Model for image quality assessment and the face alignment metric are described in section §5, while the proposed framework is then detailed in section §6. We evaluate our proposed approach in section §7 through a set of experiments. Conclusions and future work are summarized in section §8.

## 4 Definitions

In this section, we present the data model and data manipulation functions needed to fully understand the proposed framework.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 18 August 2021 doi:10.20944/preprints202108.0392.v1

### 4.1 Data model

**Definition 1 (image)** *An image, denoted by **im**, is a basic data structure consisting of attributes that provide clues about its content. It is written as follows:*

$$im \prec \textbf{DESC, F, SO} \succ$$

*where,*

- ***DESC** is a user-provided set of textual descriptions, keywords, or annotations.*
- ***F** is the set of features that depicts an image. It can describe an entire image or a feature at a specific location.*
- ***SO** is a set of salient objects representing objects of interest in an image detailed in the following definition.*

**Definition 2 (salient object)** *It is an object of interest, denoted by **so**, in an image. For example: a person's face. It is defined as:*

$$so \prec \textbf{w, h, coord, DESC, F} \succ$$

*where,*

- ***w** and **h** are the width and height of the salient object.*
- ***coord** indicates the coordinates of $so$.*
- ***DESC** is a set of textual descriptions related to $so$.*
- ***F** is the set of features revealing a salient object's visual content such as color, texture, and shape.*

**Definition 3 (entity)** *It is a semantic object that exists by itself (e.g., person, vehicle) and is expressed as **e**. Each entity is represented by a set of salient objects, which can be either distorted or not. A relationship, $e \rightarrow \{so_1, so'_2, ..., so_n\}$, done via manual or automatic annotation, shows the salient objects $\{so_1, so'_2, ..., so_n\}$ that are associated with entity e, where $so'$ represents a modified salient object.*

**Definition 4 (multimedia data stream)** *It represents an infinite sequence of images, designated as **mds**, that may contain a mix of distorted and distortion-free images. It is formally defined as follows:*

$$\textit{mds}= im_1, im'_2, ..., im_k \textit{ where } k \in \mathbb{N}^*$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 18 August 2021 doi:10.20944/preprints202108.0392.v1

### 4.2 Data manipulation functions

This section defines the functions used to modify the images in the multimedia data stream; either protect or adapt the salient objects in these images. Our presumptions focus mainly on the identification of the salient objects that might be protected or adapted based on predefined rules in authorization or adaptation schemes, which are out of the scope of this paper. We assume that the protection and adaptation functions are known and that they can be called implicitly on a subset of specified entities.

**Definition 5 (image manipulation function)** *It is a low-level function, designated by **imf**, that modifies, suppresses, or removes a set of features attributed to so in an image* $im$. $imf(so, im)$ *takes a salient object so, the image* $im$ *that contains so, and returns a modified salient object denoted by **so'**.*

As previously mentioned, we focus on two types of functions: protection and adaptation. The first type is used to hide the images' content by deleting some of its features to conceal some sensitive information related to an entity (for example, his/her identity). The second type is used to meet resource constraints such as hardware limitations.

A group of manipulation functions could be applied on the entity that exists in the images. In our work, this group is known as entity manipulation function, and it is formally defined as follows;

**Definition 6 (entity manipulation function)** *It is denoted by **emf** and defined as:*

$$emf(e, mds)=(imf_1(so_1, im_1)\circ \ldots \circ imf_i(so_n, im_n))$$

*where,* $i$ *and* $n \in \mathbb{N}^*$. $emf$ *combines several image manipulation functions* ($imf_1(so_1, im_1)$, ... ,$imf_i(so_n, im_n)$), *which modifies the salient objects representing entity* $e$ *in the multimedia data stream mds, by altering their features. As a result, this function returns a set of modified salient objects **SO'** that represents entity **e**.*

## 5 Data Quality

### 5.1 Machine Learning Model For Image Quality Assessment

An overview of our Machine Learning Image Quality Assessment model is shown in Figure 2. It is composed of two main modules:

- The Convolutional Neural Network.
- The set of random forests.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 18 August 2021 doi:10.20944/preprints202108.0392.v1

But before going into the details of each module, we first divided the dataset into three subsets where each subset will be used to train a random forest and labeled using the following FR-IQA metrics:

- Structure Similarity Index.
- Content-Based Image Retrieval.
- Perceptual Coherence Measure.

We select these metrics since each one of them can assess the images' quality by considering different features. In our work, the labels represent the quality scores ranging from 0 to 1, while leaving a margin of 0.1 between the scores and expressed as $S = [s_1, ..., s_N]$ with $0 \leq s \leq 1$.

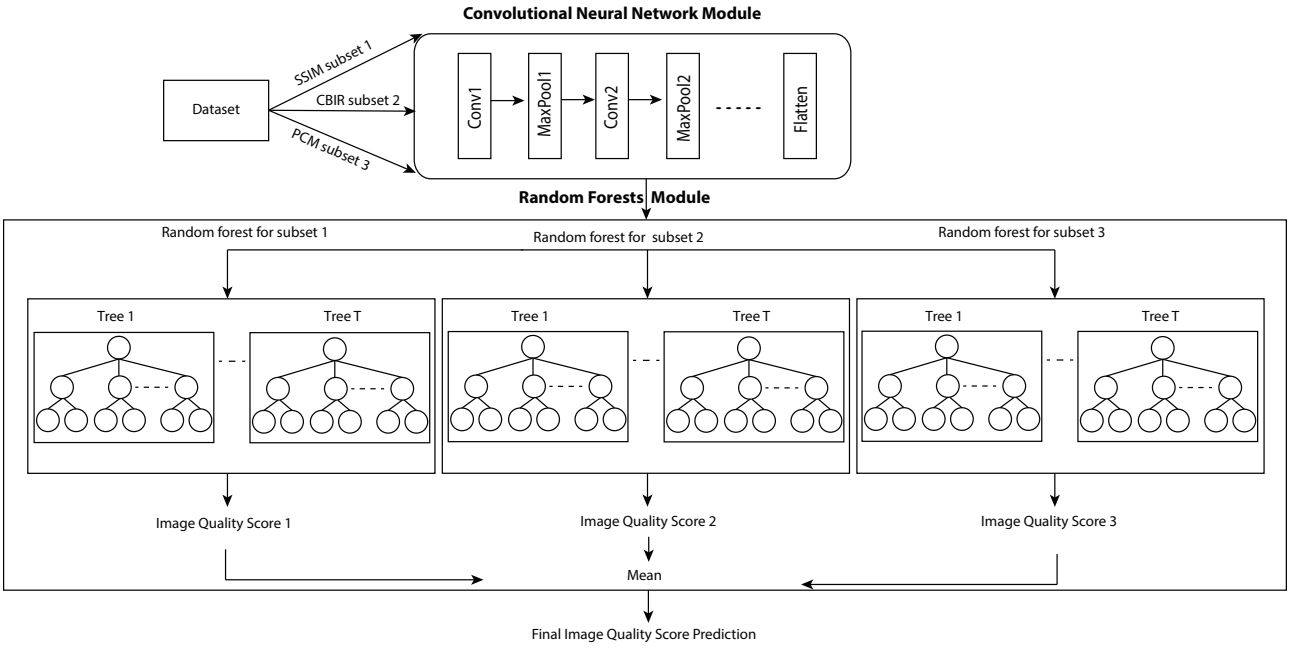



Figure 2: The Machine Learning Model For Image Quality Assessment.

### 5.1.1 Convolutional Neural Network Module

Usually, the number of parameters in a neural network grows rapidly as the number of layers increases; hence, tuning so many parameters can be a huge task and force the model to be computationally heavy. However, the Convolutional Neural Network (CNN) minimizes the time taken for tuning these parameters since it effectively reduces the number of parameters without losing the quality of models. Therefore, this sort of network is the most widely used when processing and analyzing images due to their high dimensionality. For this reason, we employ the Convolutional Neural Network.

Normally, a CNN is composed of two basic parts:

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 18 August 2021 doi:10.20944/preprints202108.0392.v1

- Feature extraction.
- Classification.

In our work, we adopt the CNN to apply a feature extraction using several convolutional, max-pooling and, flatten layers, as shown in Figure 2. The feature extraction process will allow us to reduce the image dimensionality while keeping the most useful information. Hence, it will lead to:

- Minimizing the processing time, especially that our solution is targeting real-time applications.
- Facilitating the learning and classification process of the random forests since they are not able to classify data of high dimensions.

So, this module will return a feature vector, denoted by $\mathcal{F}_{Conv}$, for any given image $im$.

#### 5.1.2 Random Forests Module

The random forest consists of a large number of individual decision trees that operate as an ensemble. Hence, in a random forest with $T$ trees we have $t \in \{ 1, ..., T \}$. The random forest gets a prediction from each tree and selects the best solution by means of voting. We use the random forest as a classifier to know what class (the quality score in our case) an image belongs to. The random forest model works so well because a large number of relatively uncorrelated trees operating as a committee will outperform any of the individual constituent models.

The low correlation between trees is the key. Uncorrelated trees can produce ensemble predictions that are more accurate than any of the individual predictions. The reason for this wonderful effect is that the trees protect each other from their respective errors. In our work, we are ensuring that the trees are uncorrelated since they are trained separately using different features.

Usually, the random forest predicts the probabilities, from each decision tree, of the input image belonging to each given class. In our case, we define eleven classes for each decision tree that represent the ground-truth image quality scores provided by the FR-IQA metrics to train these trees. Therefore, we define the ground-truth vector of probabilities, denoted as $P = [p_{s_1}, ..., p_{s_N}]$, with $\sum_{i=1}^{N} p_{s_i} = 1$ and where $p_{s_i}$ represents the probability of a quality score falling in the $i$th bucket. In our work, we make use of three random forests to predict the image quality score by training each one using a specific subset, as shown in Figure 2.

After finishing the training phase, and during the testing process, each feature vector $\mathcal{F}_{Conv}$ of an image $im$ is simultaneously pushed through all random forests, where each one will predict an image quality score,

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 18 August 2021 doi:10.20944/preprints202108.0392.v1

denoted by $IQS$, as follows:

$$IQS(\mathcal{F}_{Conv}) = \frac{1}{T}\sum_{i=1}^{T} Q_i \tag{1}$$

where:

$$Q = \hat{s}_i \mid \hat{p}_{s_i} = \max_{j \in 1,\ldots,N}(\hat{p}_{s_j}), with \begin{cases} 1 \leq i \leq N \\ \hat{s}_i \; the \; predicted \; quality \; score \\ \hat{p}_{s_i} \; the \; probability \; of \; the \; predicted \; quality \; score \; \hat{s}_i \end{cases} \tag{2}$$

Finally, we calculate the final score by averaging the values from the random forests as shown below:

$$\mu_{im} = \frac{1}{3}\sum_{k=1}^{3} IQS_k(\mathcal{F}_{Conv}) \tag{3}$$

As a result, $\mu$ will return a value between 0 and 1. Higher values indicate image quality preservation.

### 5.2 Face Alignment

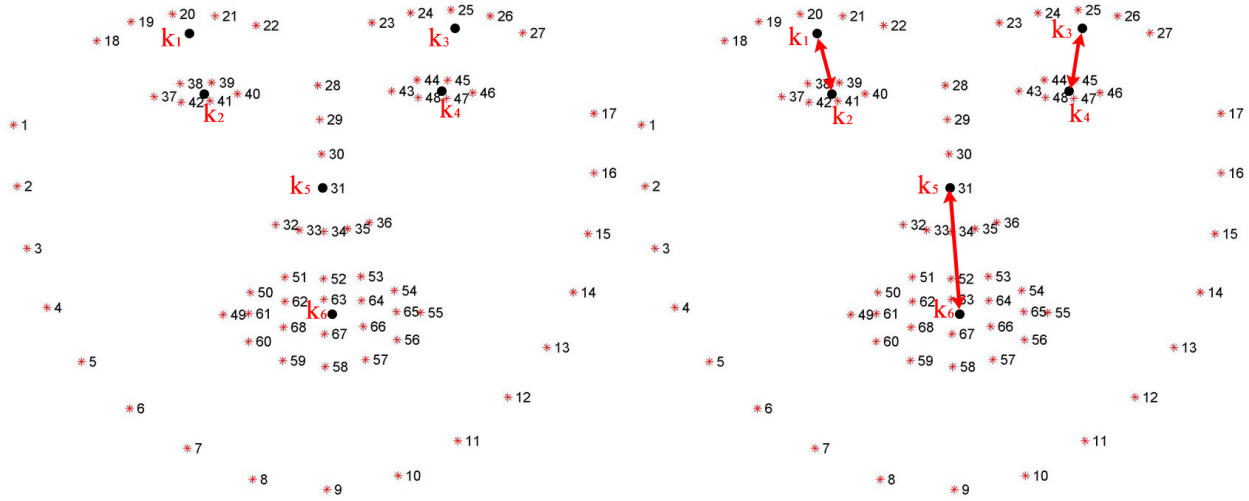
Face alignment aims to locate a group of predefined facial landmarks points (eye corners, mouth corners, etc.) in an image containing faces. Many computer vision applications, including face verification, facial expression recognition, human-computer interaction, rely on face alignment. In our work, we used this metric to ensure that the facial expressions can still be recognized from a distorted face by checking that the facial landmarks points are correctly positioned.

Some facial landmarks points are considered more representative than others. As stated in [39], the eyes and the mouth are very representative parts of a person's face. The authors showed that these two parts achieve an emotion classification accuracy equal to 81.5%. For this reason, we refer to the points that are relative to these parts as $key \; points$, denoted by $k(x, y)$, with $x$ and $y$ are the coordinates of point $k$.

More Specifically, we select six main key points, as shown in Figure 3a:

- The midpoint of the eyebrows ($k_1$ and $k_3$).
- The centroids of the two eyes ($k_2$ and $k_4$).
- The center of the face ($k_5$).
- The midpoint of the mouth ($k_6$)

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 18 August 2021 doi:10.20944/preprints202108.0392.v1




(a) The key points.

(b) The Distances between key points.

Figure 3: The key points and the distances between them

The centroids of the two eyes are calculated from the six facial landmarks of each eye. These centroids will represent the eyes' key points. We then find the eyebrows' midpoint by computing the distances between #18 and #22 for the left eyebrow, #23 and #27 for the right eyebrow. For the center of the face, we directly use the landmark #31 while finding the midpoint of the mouth using the landmarks #49 and #55. As shown in Figure 3b, we make use of these key points to find the following three distances:

- $\mathcal{D}_{so}(k_1, k_2)$: left eyebrow midpoint - left eye centroid
- $\mathcal{D}_{so}(k_3, k_4)$: right eyebrow midpoint - right eye centroid
- $\mathcal{D}_{so}(k_5, k_6)$: #31 - mouth midpoint

These distances are calculated using the Euclidean Distance as follows:

$$\mathcal{D}_{so}(k_i, k_{i+1}) = \frac{\sqrt{(k_i.x - k_{i+1}.x)^2 + (k_i.y - k_{i+1}.y)^2}}{h} \tag{4}$$

With $\{i \in 1, ..., 5 \mid i \neq 2, 4\}$ and $h$ is the diagonal bounding box used to normalize the value and computed as: $h = \sqrt{so.w^2 + so.h^2}$. These distances will create a 3D feature vector, denoted by $FA$, representing the face $so$. We then train an unsupervised neural network by applying kmeans on the feature vectors. This mechanism will produce K clusters, denoted by $C = \{c_1, c_2, ..., c_K\}$, with K $\in \mathbb{N}$. Thus, each cluster $c_i$ will have a centroid represented by the mean feature vector of this cluster, designated as $\mathcal{FA}_i$. After completing the training phase, we assessed the facial landmarks of a distorted face by finding the deviation

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 18 August 2021 doi:10.20944/preprints202108.0392.v1

between its landmarks points and the trained points using the relative error formula as follows:

$$\delta(so') = \min_{i \in 1...K, j \in 0...2} \frac{|FA'(j) - \mathcal{FA}_i(j)|}{\mathcal{FA}_i(j)} \tag{5}$$

As a result, $\delta$ will contain a value between 0 and 1. The closer the value is to 0, the more likely the facial landmarks will be aligned. Therefore, and after finding the deviation of the landmarks points for a face, we compute the face alignment for all of the faces contained in a distorted image $im'$ as follows:

$$\mathcal{A}_{im'}(SO') = \frac{1}{count(SO')} \sum_{so' \in SO'} \delta(so'_m) \tag{6}$$

Finally, we subtracted the result from 1 (as shown in the equation below) to adjust the measure with the outputted score of our machine learning model, described in the previous subsection. Therefore, higher scores will lead to face alignment preservation.

$$\mathcal{F}_{alignment} = 1 - \mathcal{A}_{im'}(SO') \tag{7}$$

### 5.3 Image Score

In the end, we combine the previous methods to calculate the final image quality score. We compute the Image Score, denoted by IS, as follows:

$$IS = \frac{w_1 \times \mathcal{F}_{alignment} + w_2 \times \mu_{im}}{\sum_{i=1}^{2} w_i} \tag{8}$$

where $w_1$ and $w_2$ are weights between 0 and 1 with their sum equal to 1. The administrator chooses these weights to indicate the importance of each method based on his preferred features. The value of $IS$ has a range from 0 to 1, and higher scores indicate quality preservation.

## 6 Framework

An overview of our framework is shown in Figure 4. It consists of two main modules:

- Stream Processing.
- Back-end.

In the following, we present in detail the framework's modules.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 18 August 2021 doi:10.20944/preprints202108.0392.v1

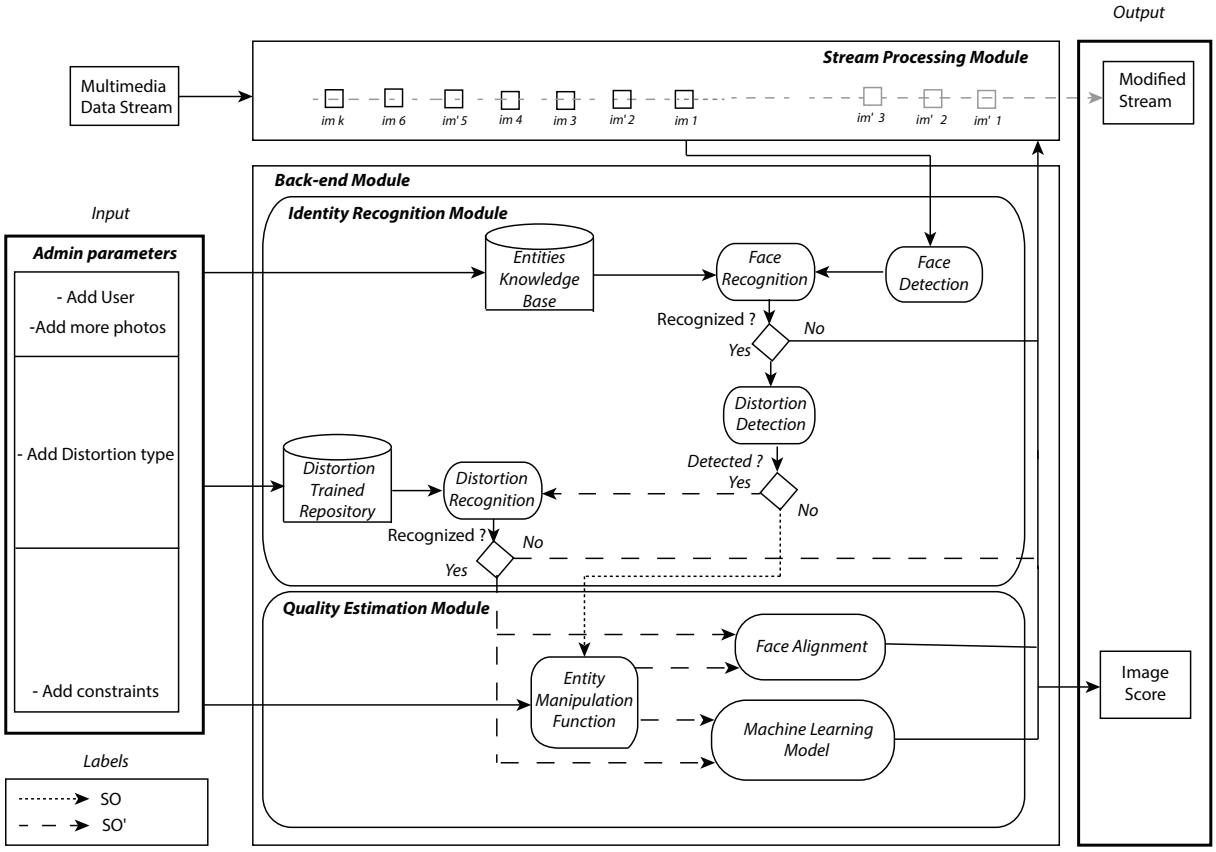



Figure 4: Framework.

## 6.1 Stream Processing Module

This module is responsible for processing continuous, never-ending data streams with no beginning or end that provide a constant feed of data that can be utilized or acted upon without being downloaded first. By using Stream Processing Module, data streams can be instantly processed and analyzed as it is generated in real-time. Therefore, administrators will have the ability to query continuous data streams and detect conditions within a short amount of time from the date of receipt of these data. In our work, we took Twitter as a source for multimedia data streaming while processing only images.

As shown in Figure 4, our framework has the ability to process two kinds of images:

- A distorted image.
- A distortion-free image.

Moreover, the images in the stream are marked from one to k, where k is equal to an infinite number, indicating that we are processing images without any bounds. In the end, $im'$ is the resulted images that are returned from the back-end module.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 18 August 2021 doi:10.20944/preprints202108.0392.v1

### 6.2 Back-End Module

It consists of two main submodules: a) Identity Recognition and b) Quality Estimation.

The first one is responsible for detecting and recognizing:

- The entities.
- The distortion if available .

These tasks are done through the use of the following components: 1) *Face Detection*, 2) *Face Recognition*, 3) *Entities Knowledge Base*, 4) *Distortion Detection*, 5) *Distortion Recognition*, and 6) *Distortion Trained Repository*.

The second submodule's duty is to assess the image quality with the help of the remaining components: 7) *Entity Manipulation Function*, 8) *Face Alignment*, and 9) *Machine Learning Model*. We will detail each component in the upcoming sections.

#### 6.2.1 Identity Recognition Module

##### 6.2.1.1 Face Detection

*Face Detection* plays an important role as the first step in many key applications, including face tracking, face analysis, and facial recognition. In our work, *Face Detection* is considered an essential component for face recognition. While receiving the multimedia data stream, this component is responsible for detecting the entities' faces who appeared in the images. The detected faces are then sent to the *Face Recognition* component.

##### 6.2.1.2 Face Recognition

This component attempts to recognize the individuals' faces by comparing the detected faces with those stored in the *Entities Knowledge Base*, which will be detailed in the next section. If a match is found, the image is forwarded to the *Distortion Detection*. Otherwise, the image will return to the *Stream Processing Module*.

##### 6.2.1.3 Entities Knowledge Base

This component represents the database in which the trained entities reside. An administrator can add new entities to the database to build his schema and train more images to current entities. This will provide us with the opportunity to increase recognition accuracy.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 18 August 2021 doi:10.20944/preprints202108.0392.v1

#### 6.2.1.4 Distortion Detection

This component is in charge of determining whether the image is distorted or not. If a distortion is detected, the image is sent to the *Distortion Recognition* component. Otherwise, it will be headed to the *Entity Manipulation Function*.

#### 6.2.1.5 Distortion Recognition

After detecting the distortion, and in order to identify its type, this component compares the latter with those stored in the *Distortion Trained Repository*. If the distortion is recognized, the image is sent to the *Quality Estimation Module* to assess the quality of the distorted image. Otherwise, it will be redirected to the *Stream Processing Module*.

#### 6.2.1.6 Distortion Trained Repository

This component represents the repository where different types of distortion are trained and stored. It can recognize five kinds of distortions: Median blurring, Gaussian blurring, Pixelate, additive Gaussian noise, and compression. In addition, an administrator can add more distortion types.

### 6.2.2 Quality Estimation Module

#### 6.2.2.1 Entity Manipulation Function

This component modifies the salient objects of the entities, which are considered the persons' faces in our work. As shown in Figure 4, the administrator could add and impose constraints by applying a series of image manipulation functions on the salient objects of entity $e$. Hence, this component will return a set of altered salient objects. We recall that the image manipulation functions are divided into two types:

- Protection to hide the users' identity as we used four main protection functions: Pixelate, Gaussian blurring, Median blurring, and additive Gaussian noise.
- Adaptation to meet some limitations imposed by the available resources. We used two compression techniques: lossy and lossless.

In fact, the functions differ in terms of the number of features that they preserve. For example, a median blur returns a modified image so that certain visual and multimedia features are damaged while keeping the semantic features intact. Considering the fact that each function maintains specific features, we apply the

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 18 August 2021 doi:10.20944/preprints202108.0392.v1

previously mentioned list of functions to find the most suitable one that could guarantee an acceptable image quality with the help of the following image quality assessment metrics.

#### 6.2.2.2 Face Alignment

After modifying the faces, the first metric used to assess the image's quality is the *Face Alignment* component by extracting the facial features. In this component, we measure the divergence between the landmarks points of a distorted face and the trained landmarks points to ensure that the facial expressions can still be extracted and recognized from the distorted face. As a result, the *Face Alignment* will return a value between 0 and 1.

#### 6.2.2.3 Machine Learning Model

This model will help us estimate the images' quality by considering a various number of features. We recall that we trained this model using three FR-IQA metrics: SSIM, CBIR, and PCM. When the training phase is completed, the model will determine the image quality by assessing several important features that are related to the FR-IQA metrics. Therefore, the model will output a value between 0 and 1 for any distorted image.

#### Image Score

The final component, which resides in the output, is the *Image Score*. It is responsible for:

- Aggregating the scores, which are returned from the previous *Machine Learning Model* and the *Face Alignment* component while assigning them weights based on the administrator preferred features selection.
- Selecting the image manipulation function that has preserved the images' features when imposing the constraints.
- Displaying the image score, which is calculated using equation 8.

Simultaneously, the distorted images (im') coming from the stream and the recently modified images will return to the *Stream Processing Module* to be then published.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 18 August 2021 doi:10.20944/preprints202108.0392.v1

# 7 Experiments

In this section, we first present the experimental setup and protocol before testing our framework's efficiency.

## 7.1 Experimental Setup

First of all, we start by training the Machine Learning Model using two different scenarios:

- In the first one, we used the CSIQ [39] dataset. It consists of 30 reference images and 866 distorted images with five different distortions: JPEG compression, JPEG-2000 compression, global contrast decrements, additive pink Gaussian noise, and Gaussian blurring. In this scenario, we aim to prove our approach's validity by comparing the model's prediction accuracy with the state-of-art methods. For this purpose, we use the ResNeSt269 as our backbone network due to its highest accuracy, as highlighted in the dark gray color in Table 2.

- In the second one, we used CelebA [40] dataset. It contains 202,599 face images with 10,177 identities, from which we select 52,800 images to prepare our training set. We then applied three manipulation functions on these images while considering many distortion levels: Pixelation (a.k.a mosaicking), Gaussian blurring, and Median blurring. Those three functions are the most commonly encountered distortions in practical applications. Our goal here is to use this model to predict the faces' quality in real-time. For this reason, we change the backbone network to SE-ResNeXt101_64x4d, which has an excellent trade-off between inference time and memory consumption, as shown in Table 2 in the light gray color. We note that the inference time represents the time needed for feature extraction and classification. In our work, the CNN processing time is lower than the presented values in Table 2 since we only use this network for feature extraction.

In both scenarios, we divided the dataset into three subsets using the FR-IQA metrics, mentioned in subsection §5.1, and we distributed the images to the eleven classes according to their quality score. Each manipulation function will have 230 images per class; hence, each class will contain 690 distorted images for training and 173 for validating. We recall that we choose eleven classes ranging from 0 to 1 while keeping a margin of 0.1 between each class. We tried several scenarios by minimizing and maximizing the margin between the classes, and we noticed that the margin of 0.1 gives us the best result in terms of accuracy. Moreover, we run into several situations to select the ideal number of trees. We notice that 505 trees with a maximum depth of 70 deliver the most acceptable result.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 18 August 2021 doi:10.20944/preprints202108.0392.v1

Table 2: Benchmark Analysis of Convolutional Neural Network Architectures [41–43]

| Backbone Network | Accuracy in % | Inference time in ms | Memory consumption in GB |
|---|---|---|---|
| MobileNetV3 | 75.3 | 42.94 | 0.51 |
| ResNet50_V2 | 77.1 | 11.9 | 0.72 |
| InceptionV3 | 78.8 | 9.14 | 1.05 |
| ResNet152_V2 | 79.2 | 4.90 | 0.72 |
| Inception-ResNet_V2 | 80.3 | 4.88 | 0.95 |
| ResNeXt101_32x4d | 80.4 | 5.12 | 0.60 |
| ResNeXt101_64x4d | 80.7 | 2.96 | 0.98 |
| SE-ResNeXt101_32x4d | 80.9 | 4.16 | 0.60 |
| SE-ResNeXt101_64x4d | 81 | 2.62 | 0.98 |
| ResNeSt200 | 83.7 | 1.35 | 2.78 |
| ResNeSt269 | 84.3 | 0.63 | 4.68 |

In order to train the unsupervised neural network of the face alignment, we select 100 celebrities having each 50 images from the CelebA dataset. We then extract their facial landmarks to build a 3D face feature vector. We create in this training 2 clusters by referring to two main metrics: Inertia (Intra Distance) and Dunn-index (Inter Distance). Inertia tells how far away the points within a cluster are. Therefore, lower Inertia values mean that clusters are internally coherent as the range of inertia's value starts from zero and goes up. The Dunn-index aims to identify sets of clusters that are compact, with a small variance between members of the cluster, and well separated. Thus, for a given assignment of clusters, a higher Dunn-index indicates better clustering. We notice that the Dunn-index value drops to 0.03 between clusters 2 and 3, while the Inertia value decreases slightly from 2.61 to 2.09. For this purpose, the optimal number of clusters is 2.

Finally, we gathered images for each celebrity to train an identity recognition model. However, the CelebA dataset is not designed for face recognition tasks. Therefore, we first grouped images of the same individuals based on the identity annotations provided by the CelebA dataset. We then considered the celebrities that have 30 images and more, which resulted in almost 2,600 identities.

After completing the training above, a software was built in Java using eclipse on a desktop computer with a 2.66 GHz core 2 duo and 4 GB RAM running Linux Ubuntu 14.04 64 bit. After running the program on one computer, the framework described in section §6 is deployed in a distributed system called Apache Storm [44]. In order for the storm cluster to run successfully, we must implement all of its components. To do so, we show in Tables 3 and 4 the Apache Storm configuration as well as the needed libraries.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 18 August 2021 doi:10.20944/preprints202108.0392.v1

Table 3: Showing the Apache Storm Configuration

| Apache Storm Configuration | |
|---|---|
| Machine | Service |
| Client node [45] | It tests the framework locally before deploying it to the cluster |
| Nimbus node [46] | It is the master in a Storm cluster that is responsible for distributing the application code across various worker nodes. |
| Three Zookeeper nodes [46] | They handle the communication between the Nimbus and the supervisors. |
| Eleven Supervisor nodes [46] | Each worker node runs a daemon called the Supervisor, which listens for work assigned to its machine and starts and stops worker processes based on what Nimbus has assigned to it. |

Table 4: Showing the needed libraries for Apache Storm

| Implemented Libraries | |
|---|---|
| Libraries' name | Service |
| Apache Storm 0.9.3, and zookeeper 3.4.6 | They should be distributed and installed on all nodes to successfully run the storm cluster. |
| OpenCV 3.4.3, Python dlib, and MTCNN | We use the manipulation functions from OpenCV, facial landmarks extraction using dlib and performing face detection via MTCNN. |
| Trained ResNet Model | It is used to recognize the distortion and the faces. |

## 7.2 Experimental Protocol

We conducted three sets of experiments as shown below:

- Firstly, we evaluated our model's efficiency by measuring the quality score prediction accuracy of the No-Reference images from two datasets: LIVE [47], and TID2013 [27]. A subjective quality score, i.e., the mean opinion score (MOS), is assigned for each image in the dataset. The LIVE database consists of 29 original images along with their 779 distorted versions having five types of distortions on various levels: JPEG2000 compression (JP2K), JPEG compression, additive white Gaussian noise (WN), Gaussian blurring (GB), and simulated fast fading Rayleigh channel (FF). The TID2013 database is composed of 25 original images and 3,000 distorted images with 24 types of distortions. But, and as in many previous works [36, 48, 49], we only consider three types of distortions that are

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 18 August 2021 doi:10.20944/preprints202108.0392.v1

common to the two databases: JPEG compression, additive Gaussian noise, and Gaussian blurring. We then compare our results with the state-of-art methods.

- Secondly, we assessed the images' quality from Twitter stream that may be affected after applying a manipulation function. We limit the processed images' size to 9,000 as our goal is to determine the image quality using the Machine Learning Model and the Face Alignment defined in section §5. We started by varying the number of faces in the images from 1 to 3 and applying the existing list of manipulation functions to find the appropriate one that returns the highest score and preserves most of the images' features. In these two scenarios, the framework is tested only on a local cluster without being uploaded to the distributed system.

- Thirdly, we implemented our framework on Apache Storm as we evaluated in real-time its performance in terms of:

  1. Execution latency: The average amount of time it takes for an image to be executed.
  2. Number of nodes: Number of supervisors engaged in processing the images.

  To do so, the following scenario was executed:

  1. We distributed the libraries on all nodes.
  2. We uploaded the framework to the cluster while processing 50,000 images from Twitter stream.

We repeat step 2 several times while incrementing in each run the number of nodes by 2 to evaluate the Apache Storm's performance in terms of execution latency.

## 7.3 Results

### 7.3.1 Performances Comparison

This test aims to evaluate the performance of our Machine Learning Model by comparing the prediction scores of the No-Reference images to the subjective ratings. As mentioned before, the two largest publicly available subject-related databases used are: LIVE [47], and TID2013 [27]. Two correlation coefficients between the prediction results and the subjective scores have been adopted to evaluate the performance of our method:

- Spearman Rank Order Correlation Coefficient (SROCC) assesses how well the relationship between two variables can be described using a monotonic function.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 18 August 2021 doi:10.20944/preprints202108.0392.v1

- Pearson Correlation Coefficient (PCC) measures the degree of relationship between two random variables.

A high correlation coefficient (close to 1) with the subjective score MOS indicates a good method. After processing the images from LIVE and TID2013 datasets, we obtain the results along with the state-of-arts methods shown in Table 5.

Table 5: Performance comparison on the two artificially distorted benchmark databases. Italicized are Full-Reference methods and the results of the best methods are listed in bold.

| Dataset | LIVE | | TID2013 | | Average values | |
|---|---|---|---|---|---|---|
| Correlation Coefficients | PCC | SROCC | PCC | SROCC | PCC | SROCC |
| *SSIM* [2] | 0.945 | 0.948 | 0.789 | 0.741 | 0.867 | 0.844 |
| *PSNR* [3] | 0.870 | 0.875 | 0.672 | 0.639 | 0.771 | 0.757 |
| Zhang et al. [24] | - | 0.961 | - | 0.855 | - | - |
| NIMA [25] | 0.698 | 0.637 | **0.941** | **0.944** | 0.819 | 0.790 |
| MetaIQA [26] | 0.835 | 0.802 | 0.784 | 0.766 | 0.809 | 0.784 |
| UNIQUE [34] | 0.968 | 0.969 | 0.873 | 0.858 | 0.920 | 0.913 |
| DeepFL-IQA [37] | **0.978** | **0.972** | 0.876 | 0.858 | 0.927 | 0.915 |
| CORNIA [50] | 0.961 | 0.951 | 0.705 | 0.612 | 0.833 | 0.781 |
| Our approach | 0.949 | 0.945 | 0.935 | 0.918 | **0.942** | **0.931** |

We first start by comparing our method to the FR-IQA metrics: PSNR and SSIM. Table 5 shows that our approach gives a good SROCC and PCC in both datasets and outperforms the PSNR metric. Moreover, our approach has slightly better results than SSIM, especially in the LIVE dataset. The reason behind this correlation is due to the fact that this metric was taken into consideration when training our model. In general, our approach achieves good results than the FR metrics in the overall datasets.

As for the NR-IQA metrics, we can notice that our approach achieves state-of-the-art performance on only TID2013. More specifically, our method was not able to outperform DeepFL-IQA (the best method in Table 5) on the LIVE dataset since the authors trained and tested their network on this dataset. However, our approach reaches better average values on both datasets. We believe the performance improvement arises in both datasets for two main reasons:

- Our model is able to assess an important feature, i.e., the color component. The human vision system and the subjective quality scores are very sensitive to color information, which is the most critical and straightforward feature that humans perceive when viewing an image.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 18 August 2021 doi:10.20944/preprints202108.0392.v1

- The random forest can generate new datasets from existing data by creating samples with replacement. Therefore, and in contrast to the deep networks that need large datasets, the random forest achieves high accuracy on small datasets since multiple versions of the dataset are generated.

### 7.3.2 Evaluating The Images Quality After Applying a Manipulation Function

Our objective is to estimate the images' quality that may be affected after applying a manipulation function. Hence, our goal is to find the most suitable function that could guarantee an acceptable trade-off between the images' quality and the constraints imposed by an administrator/user. We use in this study three manipulation functions, which are mainly considered as protection functions: Pixelation (a.k.a mosaicking), Gaussian blurring, and Median blurring. For each manipulation function, we choose fixed parameters while allowing users to select weights for the quality methods based on their preferred images' features. Table 6 shows the manipulation functions' parameters and the quality methods' weights.

Table 6: Parameters List

| Manipulation Functions Parameters List | | | |
|---|---|---|---|
| Manipulation Functions | Gaussian Blur | Median Blur | Pixelate |
| Kernel Size | 31x31 | 31x31 | - |
| standard deviation | 5 | 5 | - |
| Pixel Size | - | - | 10 |
| Quality Methods Weights | | | |
| Quality Methods | $\mu_{im}$ | $\mathcal{F}_{alignment}$ | - |
| Weights | 0.4 | 0.6 | - |

These parameters are considered average intensity values according to the literature. As for the weights, we prioritize the Face Alignment over the remaining features due to its high importance in face detection and facial expression analysis. In order to find the most appropriate manipulation function, we process 9,000 images from Twitter stream as we partition them into three equal parts based on the number of persons (from one to three persons) contained in each image. After applying the manipulation functions on each image, we obtain Table 7 and the graph shown in Figure 5. These results represent the average values of the predicted quality scores from the Machine Learning Model ($\mu_{im}$) and the Face Alignment ($\mathcal{F}_{alignment}$) for each manipulation function over 9,000 images.

According to the below graph (Figure 5), the manipulation function with the highest image score is the Gaussian blur. Moreover, and as stated in Table 6, this function will meet the users' needs since it could

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 18 August 2021 doi:10.20944/preprints202108.0392.v1

preserve several features, including structure, contrast, color, and most importantly, the location of the facial landmarks points. Hence, this function will guarantee that the facial expressions are still recognized while maintaining the remaining features. We recall that the image quality assessment is achieved through the use of the Machine Learning Model ($\mu_{im}$) and face alignment ($\mathcal{F}_{alignment}$).

Table 7: Quality scores when applying a manipulation function, where * represents the number of persons

| Quality scores before and after adding weights for each manipulation function | | | | | | |
|---|---|---|---|---|---|---|
| Manipulation Functions | Gaussian Blur | | Median Blur | | Pixelate | |
| Applying weights | Before | After | Before | After | Before | After |
| $\mu_{im}$ 1* | 0.9841 | 0.3936 | 0.9827 | 0.3930 | 0.9832 | 0.3932 |
| $\mu_{im}$ 2* | 0.9779 | 0.3911 | 0.9764 | 0.3905 | 0.9768 | 0.3907 |
| $\mu_{im}$ 3* | 0.9760 | 0.3904 | 0.9751 | 0.3900 | 0.9757 | 0.3902 |
| $\mathcal{F}_{alignment}$ 1* | 0.9735 | 0.5841 | 0.9255 | 0.5553 | 0.9502 | 0.5701 |
| $\mathcal{F}_{alignment}$ 2* | 0.9639 | 0.5783 | 0.9117 | 0.5470 | 0.9365 | 0.5619 |
| $\mathcal{F}_{alignment}$ 3* | 0.9344 | 0.5606 | 0.8892 | 0.5335 | 0.9277 | 0.5566 |

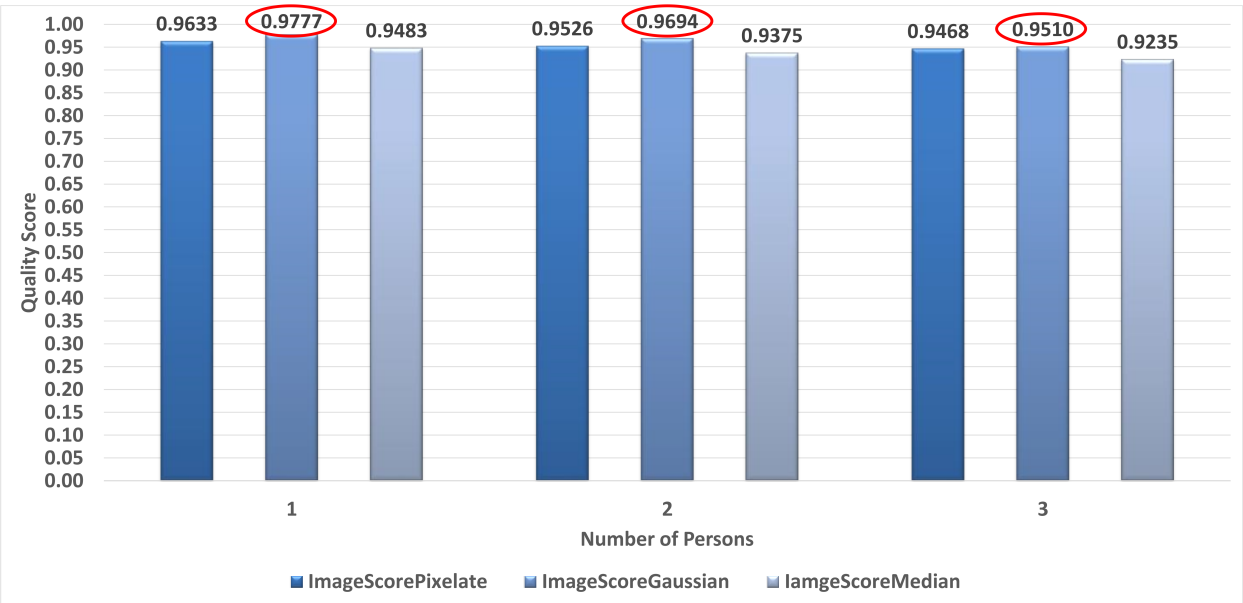



Figure 5: The dependence between a manipulation function and the image quality score.

### 7.3.3 Evaluating The Framework in Real-Time

Since IQA measures are often used in real-time applications, speed is an important issue in determining whether an IQA measure can be used in these applications. For this purpose, we treat 50,000 images from

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 18 August 2021 doi:10.20944/preprints202108.0392.v1

the Twitter stream. We deployed our framework in Apache Storm distributed system to measure the execution latency at each component by conducting two sets of experiments. In the first one, we vary the number of images from 5,000 to 50,000 while fixing the number of nodes to 4. As a result, we obtain the graph shown in Figure 6.

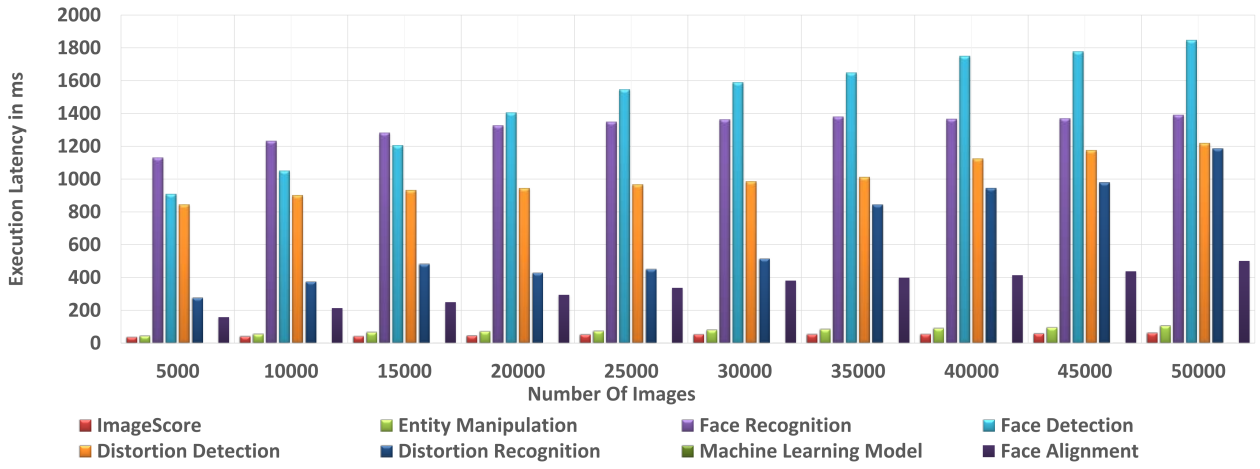



Figure 6: Execution latency per image at each component related to the number of images.

In this Figure, we can see that when the number of images is incremented by 5,000, the execution latency gets higher values due to the fact that a node needs to execute more images in each run.

In the second set, we vary the number of nodes from 2 to 7 while fixing the number of images to 50,000. Therefore, the results are shown in Figure 7.

We notice that while incrementing the number of nodes, the time required to process the images has decreased due to an increase in the number of workers, which may lead to an improvement in Apache Storm performance. More specifically, and according to Table 2, the backbone network used in this test needs 2.62 ms to process an image. However, as shown in Figure 7, the maximum time needed for our Machine Learning Model to predict the quality score is 1.811 ms (marked with red circle). We notice that the time is diminished since we are only using the CNN for feature extraction, and the random forest has better time complexity than the latter in the classification process. Consequently, our model is faster by 3 ms than the best approach in the state-of-art (DeepFL-IQA).

Furthermore, and according to the literature [31, 32], the face alignment takes an average of 1 sec to verify and recognize the facial expressions. In our work, the time is dropped to 0.643 sec (marked in red rectangle) since we are reducing the face feature vector dimension by taking only the most representative parts in the face.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 18 August 2021 doi:10.20944/preprints202108.0392.v1

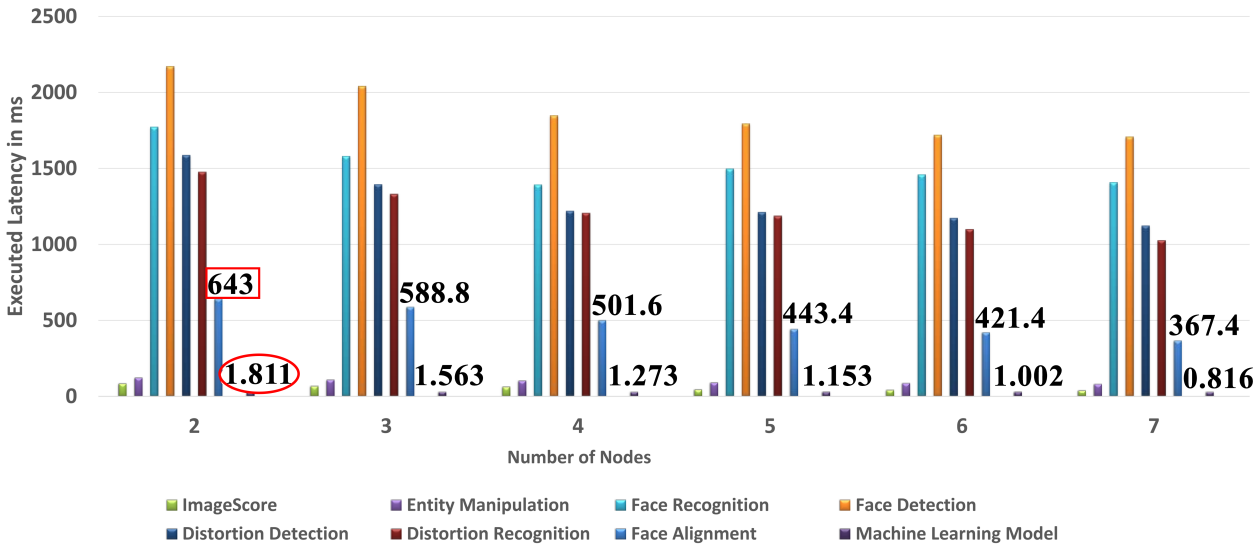



Figure 7: Execution latency per image at each component related to the number of nodes.

## 8 Conclusion

In this paper, we presented a framework that intends to assess the faces' quality in images that may be distorted during the processing and transmission phase while treating these images in real-time. The images' quality is estimated using a Machine Learning Model that represents an integration of a deep network with a random forests; we trained this model using three FR quality metrics. We also used face alignment as a second metric to estimate the faces' quality. Three sets of experiments have been tested in order to evaluate our approach.

In future work, we intend to provide a framework that can process a wide range of multimedia data, such as videos and audio. Moreover, we aim to consider more features in assessing the images to improve quality prediction accuracy while minimizing the execution latency, especially at the face detection and recognition components.

## References

[1] "Social media statistics in 2020," https://dustinstout.com/social-media-statistics/#instagram-stats, accessed: 2020-01-24.

[2] J. Nilsson and T. Akenine-Möller, "Understanding ssim," 2020.

[3] M. A. Baig, A. A. Moinuddin, and E. Khan, "Psnr of highest distortion region: An effective image quality assessment method," in *2019 International Conference on Electrical, Electronics and Computer Engineering (UPCON)*, 2019, pp. 1–4.

[4] M. K. Alsmadi, "Content-based image retrieval using color, shape and texture descriptors and features," *Arabian Journal for Science and Engineering*, vol. 45, no. 4, pp. 3317–3330, 2020.

[5] Q. Wang, J. Chu, L. Xu, and Q. Chen, "A new blind image quality framework based on natural color statistic," *Neurocomputing*, vol. 173, pp. 1798–1810, 2016.

[6] Y. Zhang, A. K. Moorthy, D. M. Chandler, and A. C. Bovik, "C-diivine: No-reference image quality assessment based on local magnitude and phase statistics of natural scenes," *Signal Processing: Image Communication*, vol. 29, no. 7, pp. 725–747, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0923596514000836

[7] L. Zhang, L. Zhang, and A. C. Bovik, "A feature-enriched completely blind image quality evaluator," *IEEE Transactions on Image Processing*, vol. 24, no. 8, pp. 2579–2591, 2015.

[8] S.-H. Bae and M. Kim, "A novel image quality assessment with globally and locally consilient visual quality perception," *IEEE Transactions on Image Processing*, vol. 25, no. 5, pp. 2392–2406, 2016.

[9] H. Lee and H. Kwon, "Going deeper with contextual cnn for hyperspectral image classification," *IEEE Transactions on Image Processing*, vol. 26, no. 10, pp. 4843–4855, 2017.

[10] W. Zhang, K. Ma, J. Yan, D. Deng, and Z. Wang, "Blind image quality assessment using a deep bilinear convolutional neural network," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 1, pp. 36–47, 2020.

[11] D. Yang, V.-T. Peltoketo, and J.-K. Kamarainen, "Cnn-based cross-dataset no-reference image quality assessment," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, Oct 2019.

[12] D. Varga, "Multi-pooled inception features for no-reference image quality assessment," *Applied Sciences*, vol. 10, no. 6, 2020. [Online]. Available: https://www.mdpi.com/2076-3417/10/6/2186

[13] O. Rippel and L. Bourdev, "Real-time adaptive image compression," 2017.

[14] Z. Chami, B. AL Bouna, C. Abou Jaoude, and R. Chbeir, "A real-time multimedia data quality assessment framework," 11 2019, pp. 270–276.

[15] ——, "A weighted feature-based image quality assessment framework in real-time," *Transactions on Large-Scale Data-and Knowledge-Centered Systems XLV: Special Issue on Data Management and Knowledge Extraction in Digital Ecosystems*, vol. 12390, p. 85, 2020.

[16] T. Kalsum, "Emotion recognition from facial expressions using hybrid feature descriptors," *IET Image Processing*, vol. 12, pp. 1004–1012(8), June 2018. [Online]. Available: https://digital-library.theiet.org/content/journals/10.1049/iet-ipr.2017.0499

[17] J. Chen, Z. Chen, Z. Chi, and H. Fu, "Facial expression recognition in video with multiple feature fusion," *IEEE Transactions on Affective Computing*, vol. 9, no. 1, pp. 38–50, 2016.

[18] A.-N. Moldovan and C. H. Muntean, "Dqamlearn: Device and qoe-aware adaptive multimedia mobile learning framework," *IEEE Transactions on Broadcasting*, vol. 67, no. 1, pp. 185–200, 2021.

[19] M. Al-hammouri, B. Madani, M. Aloqaily, I. A. Ridhawi, and Y. Jararweh, "Scalable video streaming for real-time multimedia applications over dds middleware for future internet architecture," in *2018 IEEE/ACS 15th International Conference on Computer Systems and Applications (AICCSA)*, 2018, pp. 1–6.

[20] A. Rueangprathum, S. Limsiroratana, and S. Witosurapot, "User-driven multimedia adaptation framework for context-aware learning content service," *Journal of Advances in Information Technology*, vol. 7, pp. 182–185, 2016.

[21] T.-J. Liu, K.-H. Liu, J. Y. Lin, W. Lin, and C.-C. J. Kuo, "A paraboost method to image quality assessment," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 1, pp. 107–121, 2017.

[22] L. He, D. Wang, Q. Liu, and W. Lu, "Fast image quality assessment via supervised iterative quantization method," *Neurocomputing*, vol. 212, pp. 121–127, 2016, chinese Conference on Computer Vision 2015 (CCCV 2015). [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231216306968

[23] S. Pei and L. Chen, "Image quality assessment using human visual dog model fused with random forest," *IEEE Transactions on Image Processing*, vol. 24, no. 11, pp. 3282–3292, 2015.

[24] W. Zhang, K. Zhai, G. Zhai, and X. Yang, "Learning to blindly assess image quality in the laboratory and wild," in *2020 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2020, pp. 111–115.

[25] H. Talebi and P. Milanfar, "Nima: Neural image assessment," *IEEE Transactions on Image Processing*, vol. 27, no. 8, pp. 3998–4011, 2018.

[26] H. Zhu, L. Li, J. Wu, W. Dong, and G. Shi, "Metaiqa: Deep meta-learning for no-reference image quality assessment," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 14 143–14 152.

[27] N. Ponomarenko, L. Jin, O. Ieremeiev, V. Lukin, K. Egiazarian, J. Astola, B. Vozel, K. Chehdi, M. Carli, F. Battisti, and C.-C. Jay Kuo, "Image database tid2013: Peculiarities, results and perspectives," *Signal Processing: Image Communication*, vol. 30, pp. 57 – 77, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0923596514001490

[28] "Live image quality assessment database." [Online]. Available: http://live.ece.utexas.edu/research/quality/subjective.htm

[29] Q. Meng, S. Zhao, Z. Huang, and F. Zhou, "Magface: A universal representation for face recognition and quality assessment," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 14 225–14 234.

[30] K. Chen, Q. Lv, and T. Yi, "Fast and reliable probabilistic face embeddings in the wild," 2021.

[31] J. Chang, Z. Lan, C. Cheng, and Y. Wei, "Data uncertainty learning in face recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[32] Y. Shi and A. K. Jain, "Probabilistic face embeddings," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[33] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[34] W. Zhang, K. Ma, G. Zhai, and X. Yang, "Uncertainty-aware blind image quality assessment in the laboratory and wild," *IEEE Transactions on Image Processing*, vol. 30, pp. 3474–3486, 2021.

[35] S. Ahn, Y. Choi, and K. Yoon, "Deep learning-based distortion sensitivity prediction for full-reference image quality assessment," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 344–353.

[36] S. Bosse, D. Maniry, K. Müller, T. Wiegand, and W. Samek, "Deep neural networks for no-reference and full-reference image quality assessment," *IEEE Transactions on Image Processing*, vol. 27, no. 1, pp. 206–219, 2018.

[37] H. Lin, V. Hosu, and D. Saupe, "Deepfl-iqa: Weak supervision for deep iqa feature learning," 2020.

[38] X. Solé, A. Ramisa, and C. Torras, "Evaluation of random forests on large-scale classification problems using a bag-of-visual-words representation," in *Artificial Intelligence Research and Development*. IOS Press, 2014, pp. 273–276.

[39] Z. Lian, Y. Li, J. Tao, J. Huang, and M.-Y. Niu, "Expression analysis based on face regions in read-world conditions," *International Journal of Automation and Computing*, vol. 17, 04 2019.

[40] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[41] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[42] S. Xie, R. Girshick, P. Dollar, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[43] H. Zhang, C. Wu, Z. Zhang, Y. Zhu, H. Lin, Z. Zhang, Y. Sun, T. He, J. Mueller, R. Manmatha, M. Li, and A. Smola, "Resnest: Split-attention networks," 2020.

[44] (2015) Apache storm - concepts. [Online]. Available: http://storm.apache.org/releases/current/Concepts.html

[45] (2015) Setting up a development environment. [Online]. Available: http://storm.apache.org/releases/1.0.6/Setting-up-development-environment.html

[46] (2018) Apache storm cluster architecture. [Online]. Available: http://storm.apache.org/releases/1.0.6/Setting-up-development-environment.html

[47] H. R. Sheikh, M. F. Sabir, and A. C. Bovik, "A statistical evaluation of recent full reference image quality assessment algorithms," *IEEE Transactions on Image Processing*, vol. 15, no. 11, pp. 3440–3451, 2006.

 doi:10.20944/preprints202108.0392.v1

[48] X. Liu, J. van de Weijer, and A. D. Bagdanov, "Rankiqa: Learning from rankings for no-reference image quality assessment," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[49] K.-Y. Lin and G. Wang, "Hallucinated-iqa: No-reference image quality assessment via adversarial learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[50] P. Ye, J. Kumar, L. Kang, and D. Doermann, "Unsupervised feature learning framework for no-reference image quality assessment," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 1098–1105.