iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** I **Month of publication:** January 2025

DOI: https://doi.org/10.22214/ijraset.2025.66189

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue I Jan 2025- Available at www.ijraset.com

Face Unblurring

Chaithanyasankar S[1], Abhishek R[2], Kiyasath Rahman[3], Shraddha SreeKumar[4], Mrs. Vidya Myageri[5]
[1,2,3,4]*Student,* [5]*Assistant Professor, Dept. Computer Science and Engineering, A J institute of engineering and technology Mangaluru, India*

Abstract: ***The human face is one of the most interesting subjects involved in numerous applications. Significant progress has been made towards the image deblurring problem, however, existing generic deblurring methods are not able to achieve satisfying results on blurry face images. The success of the state-of-the-art image deblurring methods stems mainly from implicit or explicit restoration of salient edges for kernel estimation. As a common method of deep learning, a convolutional neural network (CNN) shows excellent performance in face recognition. +e features extracted by traditional face recognition methods are greatly influenced by subjective factors and are time-consuming and laborious. In addition, these images are susceptible to illumination, expression, occlusion, posture, and other factors, which bring a lot of interference to the computer face recognition and increase recognition difficulty. Deep learning is the most important technical means in the field of computer vision.***

I. INTRODUCTION

In recent years, identification technology has grown greatly in importance and throughout today's society. The fields of finance and criminal investigation have a higher wall of security than before. The traditional methods of identifying a person whether through using for example certificates, keys or passwords, leaves much to be desired in the contemporary society. While the image of old ways of identifying an individual is very advanced, sadly it needs to be supplemented with further interventions such as why a mobile phone verification code is included. Yet, identification technology has to depend on other methods of identifying an individual, which may characteristics of the body.

The fear of misunderstanding and being misidentified poses its own challenges as well. Biometric identification system is that which deals with biological containment and identification methods of the human being, that is face recognition, fingerprint verification, voice identification, and secure iris recognition. Current Status of Research in the World and Foreign Countries. In recent years, many researchers have conducted thorough studies in deep learning, face assessment and other fields. According to Vanitha CN, the face recognition machine has outgrown its intended purpose and is now employed in any place where there are faces to detect, recognize, and analyze them. One of its applications is recognition of a picture image with the help of artificial intelligence. AI face recognition system takes out any image of the person in a video and saves it for future reference of that particular image for which analysis is to be done. But then it also means that this approach comes with herculean data as well as no appropriate face recognition system [1]. Litjens et al. firstly apprised the efficacy of radial basis function classified dimensionality reduction methods.. Various, let us say, state-of-the-art deep learning framework is optimized to combine all these feature representations and yield the best performance in terms of classification. The performance of the classifier based deep learning approach has managed to demonstrate itself on publicly available hyperspectral imaging datasets. Due to the intricacy of carrying out the study, the findings tend to have a low level of precision [2].



©IJRASET: All Rights are Reserved

Figure of face unblurring results. We utilize the semantic information of the face within a deep convolutional neural network based system for the task of face image deblurring. (a) Ground truth images (b) Blurred images (c) Ours w/o semantics (d) Ours w/ semantics.

Here, we are concerned with face image deblurring and we present a simple and practical deep Convolutional Neural Network (CNN) based solution to the anisotropic face deblurring problem. As face images have a high degree of structure and contain homogenous regions, the shape features form a strong statistical prior for restoration. Accordingly, we advocate the use of face semantic labels as global prior information and local constraints during the image deblurring of face images.

Specifically, the first step involves the creation of the semantic labels for the blurred input images through the use of face segmentation network. In addition to this, we apply a local structure loss on some important face parts (such as eye, nose, and mouth) to push the network to generate details reasonably well.

Figure presents the deblurred results with and without the proposed systemic priors and loss functions. The function presents a higher degree of realistic facial details than the one that was only trained with pixel-wise L1 loss without any semantics. Since our approach is fully end-to-end, with no estimation of the blur kernel, and no post-modifications done, the time taken for execution is vastly lesser in comparison to the recent best-performing MAP techniques.

The key contributions of this research can be stated as follows: to begin with, a concise overview of deep learning concepts is provided, followed by an in-depth explanation of the principle underlying CNN. The next stage involves the implementation of Caffe's superior deep learning engine which is used to construct the architecture of a deep CNN. In this section, however, we put forth a better-trained CNN focused fused face recognition system and demonstrate through experiments how well this model performs in extracting face features.

CNNs show remarkable potential in super-resolution using single images (SISR). They learn low-resolution and high-resolution pairs of images over a huge dataset. However, despite being scored in high decibels using general metrics like PSNR and SSIM, hallucinated results are inadequate for actual photo visualization, especially in the case of face SR. Face SR targets fine and detailed reconstruction with photorealism in SR images as compared to general SR, which adds a lot of complexity.

Recent techniques have combined optical geometry priors, such as landmarks, parsing maps, and 3D models, to achieve improved reconstruction output from HR face images. In addition to these, supporting tasks such as facial region mask generation, heat maps, and parsing maps are also aimed at better-quality image production. Such techniques tend to be costly in computation and highly dependent on labelled datasets. This may restrict their scalability and make them impractical for real-world applications.

A face detection and recognition system is also developed and put into practice. This system has trained the two algorithm models that were addressed in this research and performed quite well. Hence, a convolution based multilayer neural network feature recognition face fusion neural network structure is designed. Feature fusion serves as a primary architecture for the extraction and combination of all convolution pooling features. The techniques for extracting features utilized in for PCA, LDA, and LPP approaches utilize the layer features from convolutional structures, which were implemented and evaluated on the face data set. Lastly, the test results and their interpretation are presented.

II. RELATED WORKS

Deblurring a single image usually involves two forms, which are non-blind and blind deblurring. This classification depends on whether one knows the blur kernel or not. This section is dedicated to blind image deblurring.

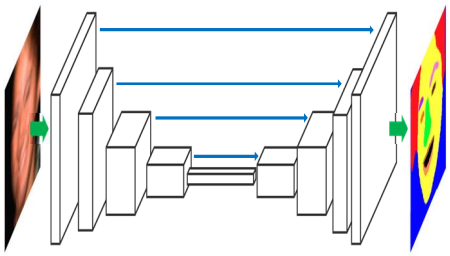
The rapid advancement made recently on single-image blind deblurring owes its success to effective priors that have been established for natural images, including: sparse image gradient prior [8, 23], normalized sparsity measure [17], patch prior [42], L0 gradient [48], color-line model [18], low-rank prior [34], self-similarity [27], and dark channel prior [31]. In all these works, strong edges have been implicitly restored for deriving blur kernels and latent sharp images by optimizing the various image priors within the MAP framework. Unfortunately, the solution to those highly nonlinear priors via several optimization stages usually comes with immense computational burden. This explains why edge-selection-based methods [6, 46] prefer the simplest priors for images (for example, L2 gradients) with filters applied (for example, shock filter) to restore or select strong edges explicitly. Although generic image deblurring methods hold the state of the art, photographs of faces differ in statistical properties from natural scenes and hence cannot be restored well using the above methods.

To handle images from specific categories, several domain-specific image deblurring approaches have been developed. Pan et al. [30] introduce the L0-regularized priors on both intensity and image gradients for text image deblurring, as text images typically contain nearly uniform intensity. To address extreme cases such as low-light images, Hu et al. [13] detect light streaks in images for

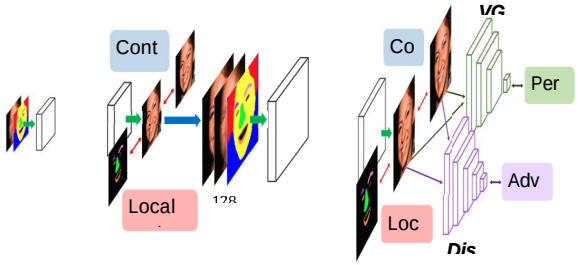
estimating blur kernels. Anwar et al. [2] propose a frequency-domain class-specific prior to restore the band-pass frequency components. Additionally, some methods utilize reference images as guidance for non-blind [43] and blind deblurring [11].

However, the performance of such methods relies on the similarity of the reference images and the quality of dense correspondence. Since face images have fewer textures and edges, existing algorithms based on implicit or explicit edge restoration are less effective. Pan et al. [29] search for similar faces from a face dataset and extract reference exemplar contours for estimating blur kernels. However, this approach requires manual annotations of the face contours and involves computationally expensive optimization processes of blur kernels and latent images in the MAP framework. In contrast, we train an end-to-end deep CNN to bypass the blur kernel estimation step and do not use any reference images or manual annotations when deblurring an image.

Deep Convolutional Neural Networks have been used for different purposes in the field of image restoration like denoising,[26], JPEG deblocking [7], dehazing [35], and super resolution [16, 19] recently. Approach so far has developed image deblurring with deep CNNs, non-blind deconvolution [37, 47, 50] blur kernel estimation [38] dynamic scene deblurring [28]. Chakrabarti et al. [4] teach a deep network to predict the Fourier coefficients of a deconvolution filter. Although computationally efficient, CNNs do not match the best state-of-the-art MAP approaches for such an image restoration task, mainly when large motion kernels are involved. This is because text images often consist of much more uniform intensities and have sparser texture regions, making end-to-end deep networks [12] good at performing denoising with large noise levels. Xu et al. [49] attempt to provide combined deblurring and super-resolution to very low-resolution blurred face and text images, usually blurred by Gaussian-like kernel degradation. Here, we concentrate more on deblurring facial images with severe motion blur. We exploit global and local semantic cues, as well as perceptual [14] and adversarial [10] losses, to achieve photorealistic face images with high detail.



(a) Semantic face parsing network



III. SCOPE

In dedicated image deblurring problems, facial image deblurring is one of them, and therefore the relevant solution has been upgraded and developed by the general deblurring means. Few reviews summarized the existing general deblurring models [6, 7]. Li [6] briefly summarized the traditional and depth-represented image deblurring methods, while Zhang et al. [7] elaborated on the deep learning-based image deblurring approaches.

Furthermore, unlike these studies, this survey delves deeper into the summarization of the deblurring research purposely made on facial images. Facial images, as an application context, possess less texture and edge information than that of general scene images; therefore, the usually proposed general deblurring methods never yield satisfactory performances on facial photographs. Further, even if an identity gets changed, several faces consist of fixed components that can then work as prior information for better performance of general methods. Accordingly, investigations have been carried out focusing solely on facial image deblurring, and we have reviewed such works.

Prior to 2015, the techniques for face deblurring were primarily model-based. Some methods boost the performance of recognition, and others enhance the general image deblurring methods in the spatial domain. To be specific, this paper discusses the after-2010

research done under the domain of face deblurring. Since 2016, methods based on deep learning have slowly yet steadily emerged and excelled in their performance as compared to other approaches, owing to the strong fitting ability of a convolutional neural network. Hence, we mostly present deep-learning-based approaches in this work.

IV. FACE SYNTHESIS USING GAN

Recently, a different approach has been witnessed in the field of Generative Models-from synthesizing faces to frontalizing them [5], [30] to completing face images [17], [31] and even using SR techniques [13], [32] for them. GANs create images through min-max-optimization where the generator and discriminator tommy-huru on the image through a very noisy sampling instance. GANs have contributed immensely in generating high-resolution images from low-resolution references.

Karras et al. [32], [33] created high-definition images using a GAN trained without any condition. Facial features are very highly acquired by this network. However, a shortcoming is that this requires an important amount of power during computation. So far no study has used a GAN to deal with the problems of blur and LR images simultaneously.

V. MULTILAYER FEATURE FUSIONFACE RECOGNITION EXPERIMENT OF CNN

In this work, the ORL and Yale datasets are chosen as the training sets for the following deep neural network. The basic introduction of the two data sets is provided below:

The ORL Face Data Set. The ORL Face Data Set comes from AT&T Laboratories in Cambridge University, U.K. This face data base includes a total of 40 persons out of which each has 10 gray-scale pictures of face. Data set contains all face pictures of the same norm and with individual black back ground. In order to acquire their face feature more elegantly, each photograph has certain angling changes and change in expression.

Yale Face Data Set. Also from the AT&T Labs, Yale consists of 165 grayscale images of faces, which has 11 images for every of the 15 people in different poses, angles and light conditions.

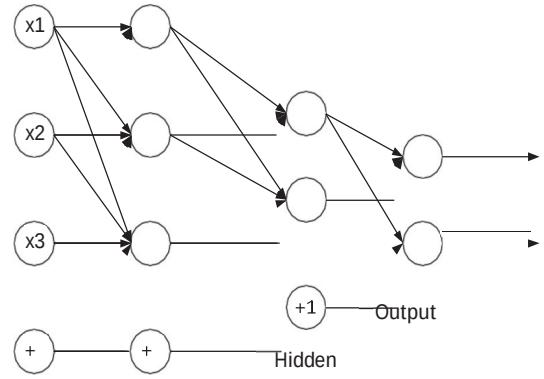


Figure (a) Multi-hidden neural network structure

VI. OVERVIEW OF PROPOSED METHOD

This web application was created with the aid of modules essential for the face deblurring application. It is hosted using Flask, which serves as the web application framework for uploading images for deblurred results and use Werkzeug to secure file handling. OpenCV is responsible for image processing that implements the Richardson-Lucy deblurring algorithm for improving image quality. NumPy performs numerical operations with heaped image arrays to contribute towards the deblurring process. Finally, Joblib is used for loading a pre-trained machine learning model for face recognition, which, using images from the deblurring process, identifies the subjects within these images. In this way, all these modules work well together, providing smooth and complete solutions to image enhancement and recognition applications.

A. Flask Module

The application begins with, `app = Flask(__name__)`, to initialize Flask. This creates the core flask application object which handles all incoming web requests and directs them to appropriate functions. The argument `__name__` is to inform Flask to consider the current module as the root of the example application.The next step would be to define the Flask configuration settings.

The application defines an upload folder and the allowed file extensions for the uploaded images (for example, ".jpg", ".jpeg", ".png", "and .gif").

These configurations are necessary to check the validity of an uploaded image file against the allowed formats.The main function of the application is to upload images for users. The above routes Flask listens to a user's action of uploading an image through the web interface; Flask captures that with a POST request on the route just defined as the handler of the upload form.

The uploaded image file is saved to the server, and upon successful uploading, an image processing algorithm - deblurring (Richardson-Lucy) - is applied to restore the quality of the image. In this, OpenCV's `filter2D` function is used to transform the image through iterative processing and revert blurring.

B. Werkzeug Module

Werkzeug, a library very complete and useful, which comes with everything that can be needed by a Flask application in dealing with various factors of HTTP, takes care of very important subjects concerning the requests and routing and also the session handling.

In the use of the deblur application, the function of Werkzeug very prominently stands out and is `secure_filename`---it secures that what has been uploaded via the web interface can be stored safely on the server. When one uploads image files to a flask application, it must be ensured that the file name does not include any pending dangerous characters or patterns that could be exploited as an entry for weaknesses, including special characters or sequences that would allow the attacker to simulate this particular advertisement of file system exploitation, such as relative paths (../..). The route that this safe method takes is thus sanitizing and washing away potentially insecure characters in the filename or replacing some with safe ones to ensure the safe storage of the otherwise risky file.

As an illustrative example, perhaps if a file had the following filename: `image<script>.jpg`, then what `secure_filename` would clean it into would be: `image_script_.jpg`, and therefore save it on a server, conflict free and without introducing issues of security. Once after the file is sanitized, then the name is used to form the correct path for saving such images uploaded in an appropriate folder on the server.For Flask, Werkzeug is tightly integrated because Flask is built on Werkzeug.

It has all necessary tools for any forms of requests, file uploads, and many other things. Although many functions are given in Werkzeug for performing many actions like sessions and cookies, its focus is only on securing the overall uploading of files in this project so that the uploaded image will certainly not drop under any risk of threats.

C. Joblib Module

The module joblib is the engine of your application face deblurring, since it will be used to load and manage the pre-trained machine learning model in charge for face recognition. In fact, in face deblurring, when the image is deblurred, identifying the person from the image is a must. The role of the joblib module comes into play here. The machine learning model for face recognition is named in a file (usually a `.pkl` file) and is loaded with **joblib** into memory efficiently at the time of running the web application. The `joblib.load()` function is called to read the model from disk into the variable (`knn_clf`), which can then be used to make predictions for the deblurred face images. This process allows avoiding retraining of the model each time the application use is done, thus saving time and resources.

The usage of joblib plays a significant role in making your application possible, in that it will allow for the storage and retrieval of large machine learning models, like KNN classifier in this particular case, efficiently.

Joblib is optimized for the serialization process in such a way that the model would be saved in as little a format as possible while considering the accuracy of the model predictions. It also increases the pace of loading, which is the main thing in real-time applications like face deblurring, where speed is key.

This makes it possible for joblib to be the main body of the system that deblurs faces, providing easy predictions about the identity of the person in the image made deblurred with the help of a trained model. Simplifying the work of storing and retrieving machine-learning models makes the application efficient without delays for whatever reason.

D. Numpy Module

The importance of the NumPy module for the application of face deblurring is to ensure optimum speed and agility when processing image data in both deblurring and preparation of image data for face recognition.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

Typically, an image entered into the program is represented in a multi-dimensional array called such that every pixel in the image has its corresponding value in the array. NumPy was designed to manipulate these arrays for maximum performance, thereby important for image processing operations such as deblurring.

In deblurring, it requires the **Richardson-Lucy deconvolution algorithm**, an iterative mathematical operation on the image array that involves multiplication, division, and add operation of the arrays to make an accurate image over many iterations. Operations are made efficient and hassle-free by NumPy.

NumPy is also concerned with the operations through which the image can be resized or flattened before face recognition. For example, images are usually resized first into some standard, say into a 100x100 pixel array, before being admission into the machine learning model for face identification. Resizing and transformation of the image array into a desired state are made possible using array manipulation functions in NumPy.

Further, normalization of the PSF used in deblurring is carried out where the PSF is redressed to allow the algorithm function.

This way, NumPy makes the whole processing of images smooth while performing the expected mathematical transformations and, thus ensuring smooth and fast operations while deblurring and recognizing the images.

E. Opencv Module

In this project of face deblurring, the Open Source Computer Vision Library (OpenCV) is really helpful for processing and manipulating the images, from the use of loading of images to the conversion and finally enhancing them through the entire deblurring pipeline.

The input image typically comes as a blurred image and is loaded using OpenCV's `cv2.imread()` function.

To improve visualization experiences in the web app, with submission of image into the application, this image is converted from BGR (which is the default color format in OpenCV) to RGB using the `cv2.cvtColor()` function.

The foundation of the face deblurring process incorporates the convolution operation, which is facilitated by OpenCV's `cv2.filter2D()`. Included in your code, this function applies a blur kernel to the image as part of the Richardson-Lucy deblurring algorithm, which iteratively changes the image to reduce blur based on pixel values determined from the point spread function (PSF) along with the blurred image.

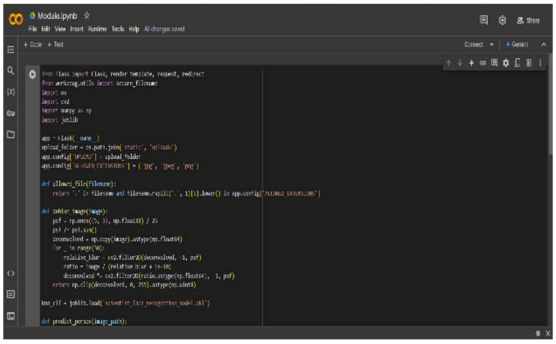
Finally, the last function used in scenarios that were described by OpenCV as: `cv2.imwrite()`, saves the deblurred images onto the disk where the images can be replayed or called on in the web application.

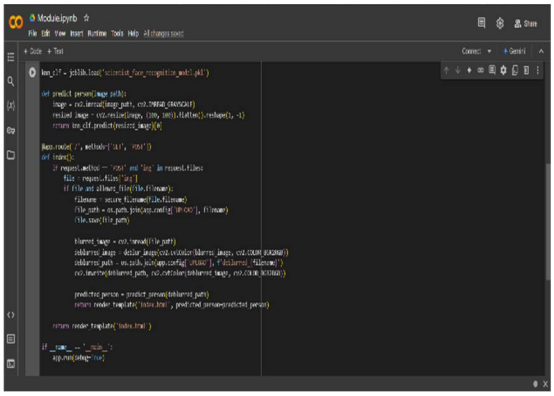
It becomes significant in that both the original blurred image and later processed deblurred one will go through this functionality against the two images.

All together, OpenCV acts as the backbone for image manipulations in this face deblurring code; such as reading images, converting them, filtering, and saving them necessary to enhance processed images quality and usability.

VII. IMPLEMENTATION DETAILS

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |





The project is implemented in a Flask web application for image deblurring and face recognition. An image is received through the user interface, and the first check is for the supported formats, which include JPG, PNG, and GIF. Upon validation, the file is saved on the server, and deblurring starts using the Richardson-Lucy algorithm, which allows for the iterative refinement of the image with a `5x5` average kernel. This process helps to reverse the blurriness and to restore the image quality, which is then shown side-by-side with the original image. The second section focuses on face-recognition. Here the application employs a K-Nearest Neighbors (KNN) classifier that has been trained using the dataset of scientist face images. The uploaded face gets preprocessed into a feature vector and compared against those that are stored to make a prediction. When the face is identified, the system retrieves and displays the description associated with that identity.

We use a CNN architecture for face deblurring. The structure has two scales each scale having 6 convolution blocks entirely sum up to 18 convolutional layers. All convolution layers, but the first one, use 5×5 kernel with 64 channels. By using a 4×4 transposed convolution layer, a scaling is made by a factor of up to 2 in upsampling. The CNN is trained at 16 batch-size while the learning rate is at 5e−6 at the parsing network and 4e−5 at the deblurring network. Semantic face parsing is trained first and converges after 60,000 iterations, taking less than a day. The deblurring network, however, is trained for 17 million iterations and takes about 5 days to do so on an NVIDIA Titan X GPU. During training, the parsing network is fixed after convergence, and the deblurring network is developed as an independent end-to-end trained system. Sits now, this modeling system is implemented with the MatConvNet toolbox [45], and when made to work, it has a direct feed into the browser. Any user can open it and instruct the processing of the image with face description and open a window to view the result. Thus, this architecture incorporates deep learning techniques, considering image restoration and face recognition for a more user-friendly and efficient tool for image enhancement and identity prediction.

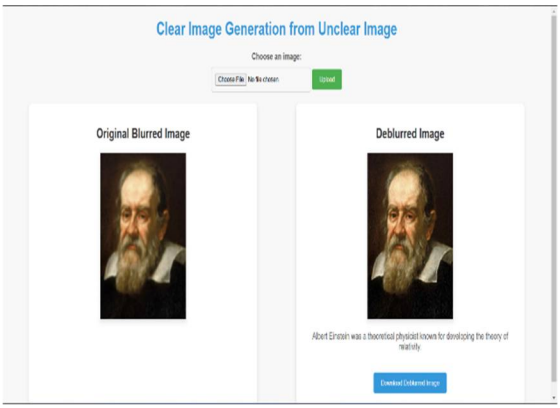
VIII. DATASETS

The dataset of blurred images includes photographs of faces particularly intended as input data for the face deblurring project. The images are purposely degraded to simulate realistic conditions, ranging from motion blur encountered during an action, low resolution due to poor camera lenses, or bad lighting that can usually be found in crime scenarios or recorded video footage. The dataset is meant to train deep learning models and treat them to pattern recognition and feature extraction techniques applying training on blurry images against clear ones. Learning from such images in conjunction with clear images will allow the model to reconstruct facial details and produce clean, recognizable outputs. This dataset is relevant because the model can be tested as to its

ability to process degraded visuals. It will be of value and application in forensic analysis, where investigative procedures can be greatly assisted through the recovery of facial information from blurry footage.

There are some well-defined images in the dataset, and those images are found in high quality and stored in the .jpg. These images depict the ground truth and are considered as the reference output for the face deblurring project. Each image is expected to correlate with a blurred version among the blurred images dataset, forming a paired dataset for supervised learning. The clear images are important input data for training the deep learning algorithm, as they establish the ground truth upon which accuracy and quality are measured during reconstruction. An example of one picture from this dataset, named adfweradfasdf.jpg, can further be analyzed by resolution, size, and other metadata to ensure that it is fit for purpose. These images are critical for assessing model performance using metrics such as PSNR and SSIM for strengthening the efficacy of deblurring techniques.

IX. RESULTS



The website has made it possible for the users to upload blurred images. This image uploaded by the user is deblurred by the principle of Richardson-Lucy. The image gets checked first for acceptable formats (JPG, PNG and GIF) and only then gets stored on the server to be retrieved for deblurring. The deblurred image is generated and displayed next to the original for comparison. The system also uses KNN classifier which is pre-trained to recognize faces. It predicts the identity of persons upon capturing them through images. It finally results in the display of the predicted name of the scientist with a brief description

Comparison	Metric	Value
Blurred vs Original	PSNR	24.5 dB
Blurred vs Original	SSIM	0.55
Deblurred vs Original	PSNR	28.7 dB
Deblurred vs Original	SSIM	0.85

X. CONCLUSION

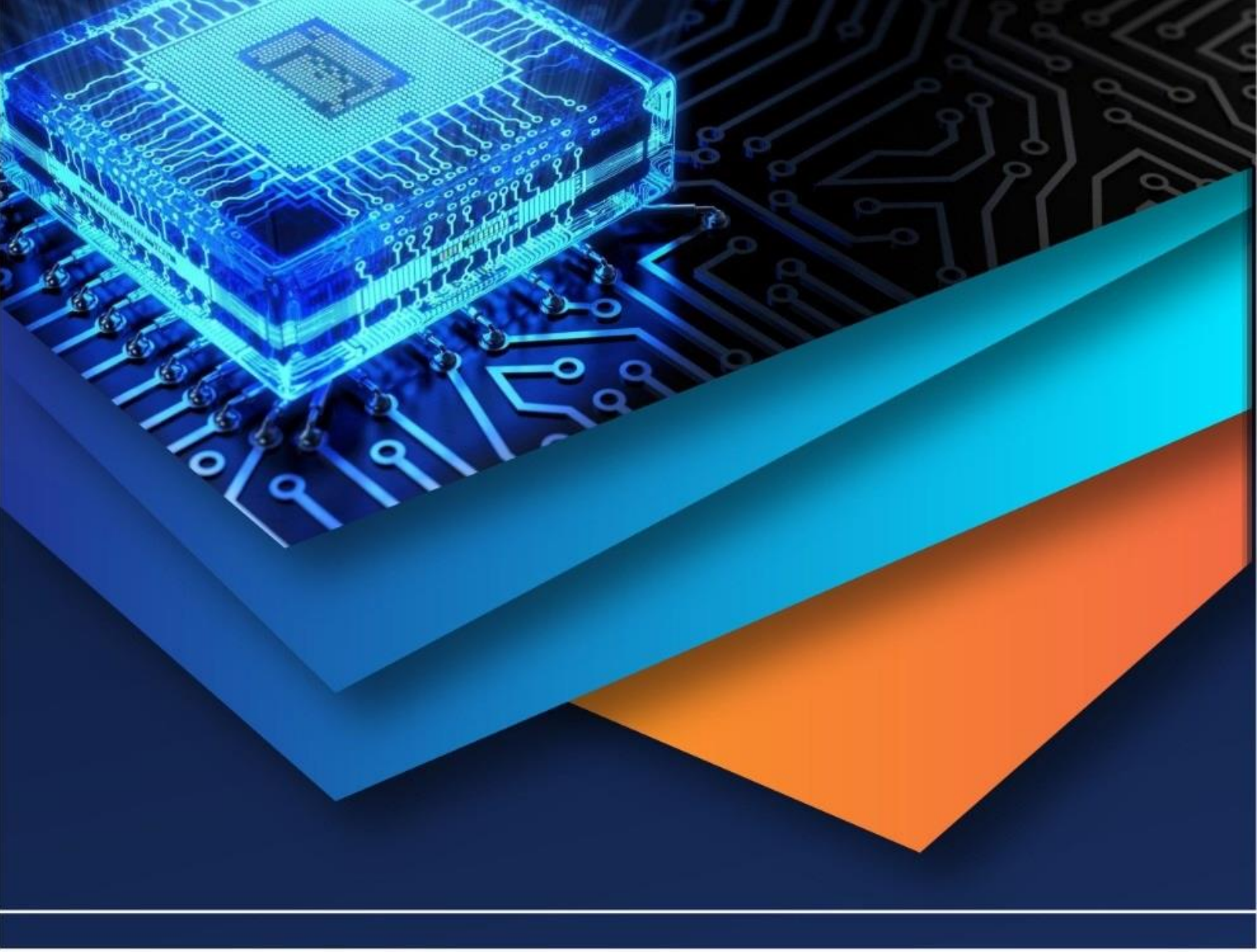
In reality, the face de-blur project is essentially a combination of both image processing and machine learning techniques to restore and analyze blurred facial image photos. It uses OpenCV for the Richardson-Lucy deblurring algorithm and an already trained face recognition model made used by joblib, and it delivers clarity in the image to retain essential facial characteristics necessary for proper identification. This is already incorporated into a web-based interface with the aid of Flask. This clearly demonstrates that such projects are feasible in refining facial image applications, particularly in crime scenes, for real-world law enforcement and security use.

REFERENCES

[1] J. U. Yun, B. Jo and I. K. Park, "Joint Face Super-Resolution and Deblurring Using Generative Adversarial Network," in IEEE Access, vol. 8, pp. 159661-159671, 2020,

[2] Ziyi Shen, Wei-Sheng Lai, Tingfa Xu, Jan Kautz, Ming-Hsuan Yang; Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 8260-8269
[3] Chrysos, Grigorios G. and Zafeiriou, Stefanos, "Deep Face Deblurring" Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, July, 2017, pp 69-78
[4] Pan, J., Hu, Z., Su, Z., Yang, MH. (2014). Deblurring Face Images with Exemplars. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds) Computer Vision – ECCV 2014. ECCV 2014. Lecture Notes in Computer Science, vol 8695. Springer, Cham.
[5] Bingnan Wang, Fanjiang Xu, Quan Zheng, "A survey on facial image deblurring" Computational Visual Media, 2024, Volume 10,
[6] Weilong Li, Jie Li, Junhui Zhou, "Advances in multimedia" Volume 2022, Article ID 9146711.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)