From Fuzzy to Funny: Implementing Facial Deblurring Models

Using Autoencoders & Generative Adversarial Networks

APCOMP 221 Final Project

By Omar Abdel Haq

Table of Contents

I. Motivation & Research Question

In recent years, significant strides have been made in the field of computer vision, particularly in advancing techniques to improve image quality – tasks like deblurring and depixelating faces. This project embarks on a comprehensive review of existing literature on deblurring methods and associated ethical considerations, alongside an implementation segment demonstrating modern deblurring techniques. As machine learning algorithms improve and advancements in the field continue, ethical concerns surrounding data privacy and informed consent have gained prominence. Deblurring methods raise worries about potential for misuse, such as for unauthorized surveillance, revealing more information from a photo than may have been intended to. It's crucial to scrutinize the current applications of these technologies and their potential for misuse both now and in the future, and anticipate forthcoming issues as AI situates itself within society.

This project has two main objectives: first, to examine and compare various methods used for deblurring in computer vision, specifically in improving the clarity of facial images (headshots); and second, to delve into the ethical dilemmas associated with deblurring techniques, such as concerns about privacy, consent, and the current applications of this technology. The implementations compared will mainly be autoencoders and generative adversarial networks (GANs), with models trained on three different types of facial obstruction in images.

II. Existing Literature

Deblurring models are used to enhance or restore the sharpness of an image that has been degraded due to blur. Blur in images can be caused by various factors such as Gaussian blur, motion blur, or pixelation (and so details of images are not visible), to name a few. These obstructions happen naturally, such as when a photo is taken but the camera isn't held steadily, but at other times the obstruction is deliberate. Deblurring models aim to reverse the blurring process and recover the original sharp image.

One common approach in reasoning about blurring is to model the blur as a convolution operation between the sharp image and a blur kernel. The blur kernel describes how each pixel in the sharp image is spread or mixed with neighboring pixels to create the blurred effect; besides different types of blurring, this posits that there are different "strengths" to the process as well. Depending on the type of model used, traditionally deblurring algorithms were divided into two camps, depending on whether or not the blurring "strength" is known:

1. *Blind Deconvolution*: This approach aims to estimate both the sharp image and the blur kernel simultaneously from the observed blurred image. It typically involves iterative optimization methods that alternate between updating the estimated sharp image and the blur kernel.
2. Non-blind Deconvolution: In cases where the blur kernel is known or can be estimated separately, non-blind deconvolution methods are used – these methods directly perform deconvolution using the known blur kernel.

However, this past dichotomy became more trivial with current advances in neural network models and deep learning. Models based on convolutional neural networks are trained on pairs of blurred and sharp images and learn to estimate the sharp images directly from blurred inputs [1].

III. Methods

Data Generation

Figure A. *CelebA Images with Different Blurring Applied*

The CelebA dataset, short for Celebrities Attributes, is a widely utilized collection comprising more than 200,000 images of celebrities. Each image within this dataset is annotated with 40 binary attribute labels, such as "smiling" or "wearing glasses." Researchers and practitioners often leverage the CelebA dataset for tasks such as facial recognition, attribute prediction, and image generation. Its extensive size and detailed annotations make it a cornerstone resource in the field of facial analysis and recognition, and one used for this project [2].

We create a subset of the CelebA dataset by filtering to images where the person is not wearing eyeglasses (as that often means they are wearing sunglasses), and the image is not labeled as "blurry" according to CelebA. Then, we use the OpenCV library to select images where we can select forward-facing photos as opposed to side profiles. OpenCV, which stands for Open Source Computer Vision Library, is a powerful open-source library primarily aimed at real-time computer vision. Developed by Intel, it has since become a community-driven project with Python bindings [3].

OpenCV's facial detection capability is implemented using the Haar Cascade Classifier, an approach that detects objects in images based on their features, developed by Paul Viola and Michael Jones. OpenCV provides a pre-trained Haar Cascade Classifier specifically designed for facial detection, making it relatively easy to integrate facial detection functionality into Python. The process begins with a training phase where positive images (containing faces) and negative images (without faces) are utilized to teach the classifier. Haar features, which capture localized intensity changes, are computed from these images (detecting things like edges and lines). Following this, a feature selection step occurs, employing Adaboost to determine the most discriminative features. Adaboost is a shallow tree-based algorithm which iteratively adjusts feature weights to minimize classification errors, resulting in the selection of the most relevant features. These selected features are then combined into a strong classifier using weighted sums. In the detection phase, features are applied to overlapping windows in the image. To optimize efficiency, a cascade of classifiers is employed, checking for an increasingly larger number of features per stage, quickly eliminating portions of an image from consideration as potentially part of a face. Only windows that successfully pass through all stages are considered as potential faces [4].

Only images where a face was detected by the classifier is used in this project. After the selection phase, three additional versions of the dataset are created, each with a different obscurement method: Gaussian blurring, pixelation, and motion blurring. Each involve a different mathematical operation applied to each image's face:

Gaussian Blurring: This method works by convolving the image with a Gaussian kernel, which is a 2D matrix where the values follow a Gaussian distribution. Each pixel in the image is replaced with a weighted average of its neighboring pixels, with the weights determined by the Gaussian function. Mathematically, this operation can be expressed as:

$$G(x, y) = \frac{1}{2\pi\sigma^2} \cdot e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Where $G(x, y)$ represents the Gaussian kernel, σ controls the standard deviation (spread) of the distribution, and (x, y) are the coordinates relative to the center of the kernel. By adjusting the standard deviation, different levels of blurring are achieved [1].

Pixelation: This method involves dividing the image into regions and then replacing the color values of each region with the average color value of that region. Mathematically, if we denote the size of the regions as $n \times n$, then the new color value for each region R can be calculated as:

$$R = \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} I(i, j)$$

Where $I(i, j)$ represents the color intensity of the pixel at position (i, j) within the region. This operation effectively reduces the resolution of the image and creates a blocky appearance. Adjusting the size of the regions determines the degree of pixelation applied to the image.

Motion Blurring: This method works by averaging the pixel values along the direction of motion. Mathematically, motion blurring can be represented as a linear convolution operation

with a motion blur kernel. To apply motion blur to an image I horizontally, we convolve the image with the motion blur kernel:

$$I_{\text{blurred}}(x, y) = \sum_{i=0}^{n-1} \frac{1}{n} I(x - i, y)$$

Where $I_{\text{blurred}}(x, y)$ represents the pixel value of the blurred image at position (x, y). This operation effectively smears the image along the direction of motion, simulating the effect of motion blur. Adjusting the length of the motion blur kernel determines the extent of blurring applied to the image [5].

The product of this step was four datasets: one with the original images, one with a Gaussian blur filter applied, one where images were pixelated, and another where horizontal motion blur was applied. Of the three, the Gaussian blurred images intentionally didn't include a strong blurring effect, while the other two datasets did. Samples from the datasets can be seen in Figure A.

Models

Models are trained such that the blurred image is provided, and the goal is to predict the original image before blurring. The two models used in this project are convolutional autoencoders and Generative Adversarial Networks (GANs).

An autoencoder consists of two parts, the encoder and decoder. The encoder part takes an input image and compresses it into a latent representation, typically of lower dimension than the original image. This compression is achieved through a series of convolutional and pooling layers, which extract important features from the input image while reducing its spatial dimensions. Then, the decoder part of the autoencoder takes this encoded representation and reconstructs the original image. The decoder consists of a series of convolutional and upsampling layers, which gradually increase the spatial dimensions of the encoded representation while also decoding the features learned by the encoder. The final output of the decoder is a reconstructed image that ideally closely resembles the original input image [6].

A GAN is a type of deep learning model composed of two neural networks, known as the generator and the discriminator, which are trained simultaneously through a competitive process. The generator's role in a GAN is to create synthetic data, such as images or sounds, that resemble real data from a given dataset. It takes data (like blurry images in our case) as input and generates samples that ideally mimic the distribution of the training data. The generator would, in this case, try to deblur the image to resemble a sharp version of itself.

On the other hand, the discriminator is trained to distinguish between real data from the training set and synthetic data produced by the generator. It receives both deblurred and original samples as input and assigns a probability of each sample being real. The discriminator is essentially a binary classifier, aiming to maximize the probability of correctly identifying real samples and fake samples. The two elements are trained through a competitive process where each tries to become better at its respective task [7].

Variations of these two model types were trained to build deblurrers, using slightly different layer architectures. The different versions were trained and evaluated on train and validation folds of the selected CelebA images.

IV. Results

Firstly, we attempt to build a simple autoencoder to remove Gaussian blur from images. We do so using downsampling layers and two convolutions with 32 kernels each in the encoder, and the same but with upsampling layers for the decoder. Sample images and their deblurred counterparts can be seen in Figure B:

Figure B. *Images Deblurred Using a Simple Convolutional Autoencoder*

We see that even with a relatively simple architecture (29,507) trainable parameters, we can get pretty good results for this deblurring task. Although the quality of the produced images is not as high as the targets, the blur is removed while preserving features and making them look somewhat sharper. However, the behavior of such a model on more corrupted images is of more interest. Next, we investigate the use of autoencoders on slightly more augmented images – the pixelated data – and evaluate the performance of different architectures on those.

In building autoencoders for pixelated faces, six potential architectures were examined; all architectures included two convolutional layers in both the encoder and decoder segments, as deeper networks would have overwhelmingly increased training time. A plot of the mean squared error between the target and depixelated images on a withheld validation warrants some interesting observations, and can be seen in Figure C:

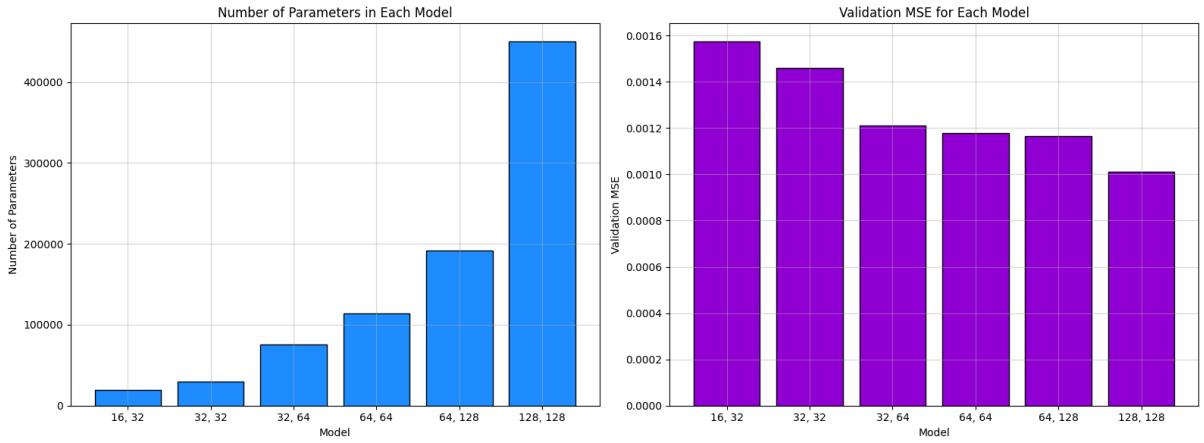

***Figure C**. Autoencoder Complexity and Corresponding Validation Set Performance; Bar Labels are Kernel Sizes of the Two Layers in the Encoder & Decoder (Reverse)*

Although each architecture has nearly double the number of trainable parameters as the one before it, the quality of the resulting model only improves by a tiny margin. The training time also nearly doubles for increasingly complex models, raising into question how good a shallow autoencoder could be at a deblurring task. Sample images and their depixelated counterparts can be seen below in Figure D:

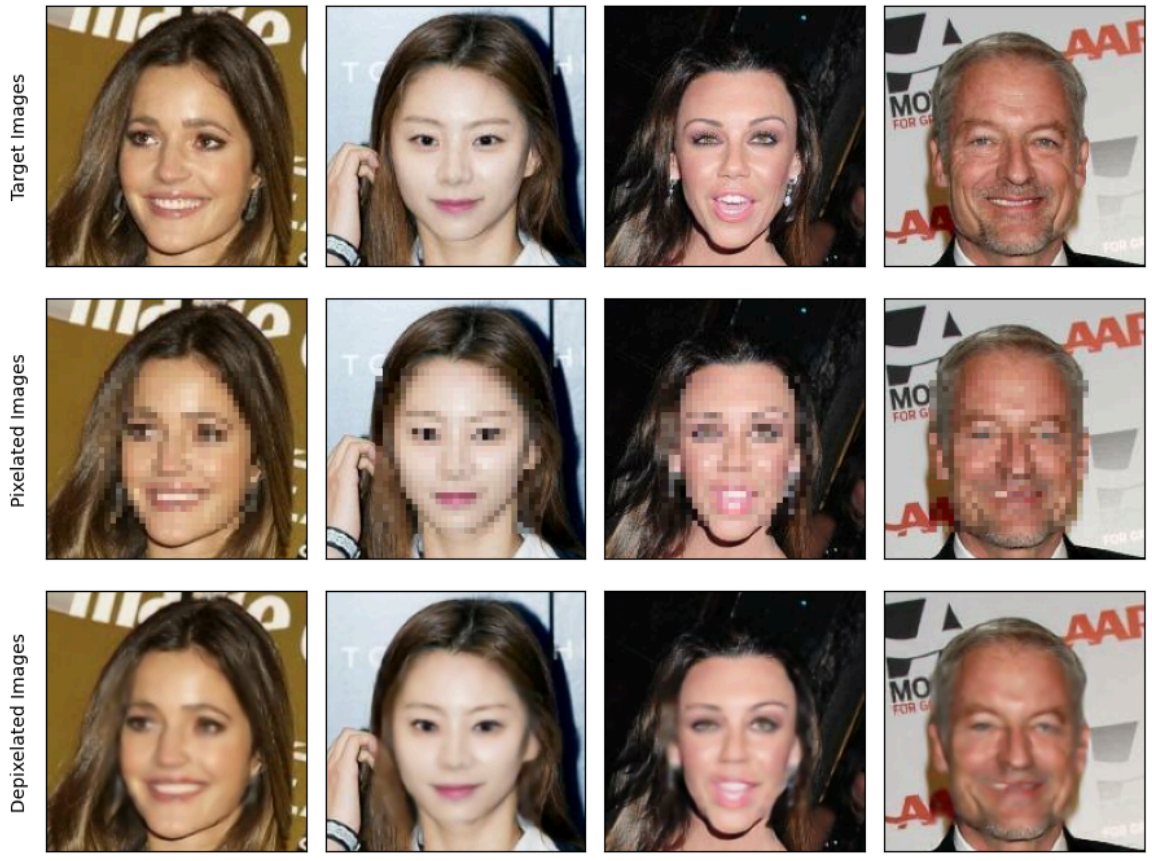

***Figure D**. Depixelated Images using an Autoencoder*

From the output, we see that the autoencoder performs well for some images (like the first two columns), but then performs rather poorly for others (the last two columns). Particularly, we see that images with more complex mouths (ajar or only slightly smirking) are a lot harder for the model to deblur.

Finally, we try out a General Adversarial Network for removing motion blur from images. GANs have risen in popularity due to their fascinating competing generator-discriminator paradigm, their ability to generate novel data, and in our case for its ability to deblur images. The input to the GAN is a blurred image, which the generator converts into a deblurred image; the discriminator attempts to tell apart deblurred images from original counterparts. Model results can be seen below in Figure E:

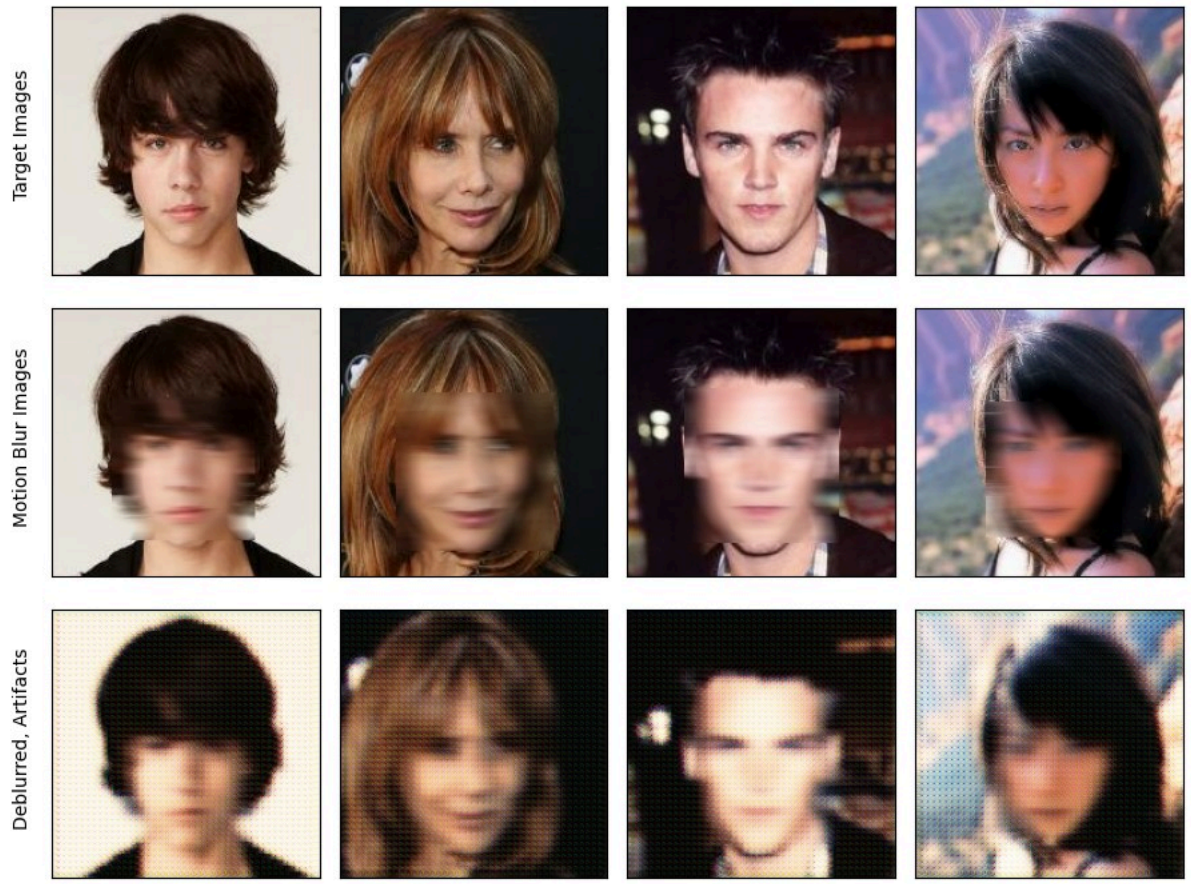

***Figure E**. GAN Deblurring on Motion Blurred Input with Heavy Artifacts*

One notable result of this model is the presence of artifacts in the images, with a checkerboard pattern. This is a well-documented issue in GANs, which happens as a result of deconvolutional layers. These layers essentially allow the network to use information from small representations and upsample larger ones. However, deconvolution can create uneven distribution of details, known as "uneven overlap," especially when the output window size isn't divisible by the spacing between points. While networks can theoretically learn to avoid this, in practice, it remains a challenge [8].

V. Ethical Considerations

Privacy Concerns & Informed Consent

Privacy concerns surrounding deblurring and depixelating techniques stem from the possibility of potentially revealing intentionally obscured or blurred details in images or videos. These technologies, while aimed at enhancing visual quality, can inadvertently expose sensitive information. For example, enhancing low-resolution images or removing blurs might inadvertently unveil details like license plates or text that individuals or entities

deliberately obscured for privacy reasons. Anonymized facial data, once enhanced through these techniques, may become susceptible to pseudo-deanonymization, allowing individuals to be identified against their wishes. This unintentional disclosure could lead to various privacy breaches and pose risks to individuals' safety, security, and autonomy.

The application of these algorithms raises profound ethical questions regarding informed consent and data privacy. Even posting anonymized pictures online can be used to identify a person, and in so enable potential misuse like surveillance or stalking. Individuals must have agency over how their data is used and must be fully informed about the potential consequences of its enhancement, but this is difficult to do with this emerging technology. Without robust safeguards and transparency measures, these technologies risk exacerbating existing privacy concerns and undermining trust in data-driven systems [9].

Deceptive Naming Convention

Neural networks specialized in tasks like image enhancement often operate on a principle of generalization rather than pinpointing the exact causes of blurriness. Instead of directly addressing specific image defects, they leverage extensive training data to learn overarching patterns and features. Consequently, when confronted with blurry or low-resolution images, these networks excel at generating content that aligns with the common characteristics observed in their training set. This might involve inferring and adding plausible details like facial features such as eyes, nose, and mouth, even if they were not distinctly visible in the original input. By extrapolating from learned patterns, these networks produce outputs that are not merely sharpened versions of the input but rather plausible reconstructions imbued with contextually appropriate details.

Given that deblurring algorithms don't really "deblur" but fill in general facial features, calling such algorithms deblurring algorithms is deceptive. In reality, it is much more appropriate to call these algorithms detail extrapolators, as they make educated guesses about image details given a blurry version of an image. Calling such algorithms deblurring algorithms can also be problematic in the case that their popularity increases, especially if police departments start relying on them. If the output of these models is popularized, it is crucial to educate people, especially people using such technology, that its output is not a function of the input image alone, but a function of tens of thousands of images that taught the model how a face "looks."

Demographic Bias

Besides privacy concerns and naming conventions, the nature of the data that models are trained on raises serious implications. Although CelebA is only one of many face image datasets, it is a popular choice for many applications. However, it is crucial to note that the dataset is composed of approximately 90% white faces [10]. This means that any features learned by models are at risk of being significantly biased towards white features. An exacerbated case of such an issue is PULSE, an algorithm designed by students at Duke University, which deblurs but in doing so adds white masculine features that make the image somewhat unrecognizable [11]. For example, the algorithm can be applied to an image of Barack Obama [12]:

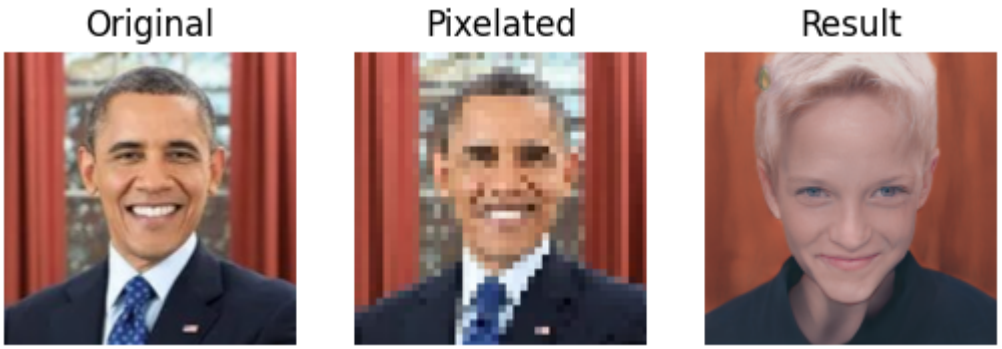

Figure F. Depixelated Image of Barack Obama Using PULSE [13]

The resulting image does not resemble neither its pixelated version nor its original one, with vaguely masculine white features superimposed on the face. The same can be done to actress Michelle Yeoh, with similar results:

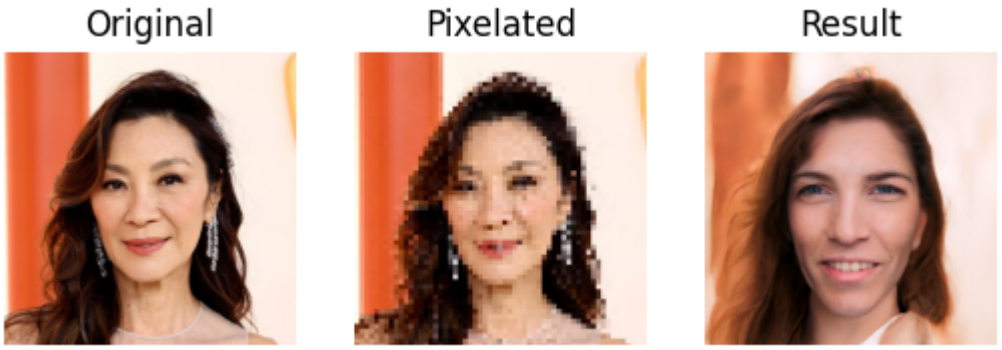

Figure G. Depixelated Image of Michelle Yeoh Using PULSE [14]

Similarly, vaguely masculine white features were superimposed on the actress's face. An issue like this one underscores how biases in training data and algorithms can perpetuate misrepresentation. The reliance on datasets that lack diversity, contributes to the biased outputs produced by AI systems.

VI. Conclusion

In this paper, we have explored the advancements, methodologies, and ethical considerations surrounding deblurring techniques in the field of computer vision. The rapid progress in machine learning algorithms, particularly in convolutional autoencoders and GANs has enabled significant improvements in image quality enhancement tasks, such as deblurring and depixelating faces. This project's review of existing literature provided insights into the evolution of deblurring methods, from traditional approaches like blind and non-blind deconvolution to modern deep learning-based techniques. Afterwards, different models were built to deblur images with different obstructions superimposed onto them.

Ethical considerations played a central role in our exploration, highlighting concerns regarding privacy, informed consent, and demographic bias. We discussed the potential risks associated with deblurring techniques, including unintentional disclosure of sensitive information and perpetuation of demographic biases in training data and algorithms. Furthermore, we addressed the deceptive naming conventions surrounding deblurring algorithms and the implications of relying on biased datasets, emphasizing the importance of transparency, education, and diversity in data collection and model development.

References

[1]. Zhang, K., Ren, W., Luo, W., Lai, W.-S., Stenger, B., Yang, M.-H., & Li, H. (2022). Deep Image Deblurring: A Survey. *International Journal of Computer Vision*, *130*(9), 2103–2130. https://doi.org/10.1007/s11263-022-01633-5

[2]. Liu, Z., Luo, P., Wang , X., & Tang, X. (n.d.). *Large-Scale CelebFaces Attributes (CelebA) Dataset*. CelebA Dataset. https://mmlab.ie.cuhk.edu.hk/projects/CelebA.html

[3]. *OpenCV Library*. OpenCV. (2024, March 27). https://opencv.org/

[4]. *Haar Cascade Classifier*. OpenCV. https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html

[5]. GeeksforGeeks. (2019, August 26). *OpenCV: Motion Blur in Python*. https://www.geeksforgeeks.org/opencv-motion-blur-in-python/

[6]. *What is an Autoencoder?*. IBM. https://www.ibm.com/topics/autoencoder

[7]. Li, C. A Survey on Image Deblurring. https://arxiv.org/pdf/2202.07456.pdf

[8]. Odena, A., Dumoulin, V., & Olah, C. (2016, October 17). *Deconvolution and Checkerboard Artifacts*. Distill. https://distill.pub/2016/deconv-checkerboard/

[9]. Al Ameen, Z., Bin Sulong, G., & Md. Johar, Md. G. (2013). Computer Forensics and Image Deblurring: An Inclusive Investigation. *International Journal of Modern Education and Computer Science*, *5*(11), 42–48. https://doi.org/10.5815/ijmecs.2013.11.06

[10]. Truong, K. (2020, June 23). *This Image of a White Barack Obama is AI's Racial Bias Problem in a Nutshell*. VICE. https://www.vice.com/en/article/7kpxyy/this-image-of-a-white-barack-obama-is-ais-racial-bias-problem-in-a-nutshell

[11]. *Self-Supervised Photo Upsampling via Latent Space Exploration of Generative Models*. PULSE. (n.d.). http://pulse.cs.duke.edu/

[12]. https://colab.research.google.com/github/tg-bomze/Face-Depixelizer/blob/master/Face_Depixelizer_Eng.ipynb

[13]. The United States Government. (2022, December 23). *Barack Obama*. The White House. https://www.whitehouse.gov/about-the-white-house/presidents/barack-obama/

[14]. Luu, C. (2023, March 13). *Michelle Yeoh is officially the first Asian best actress Oscar winner*. InStyle. https://www.instyle.com/michelle-yeoh-first-asian-best-actress-oscars-2023-speech-7153424