Image Deblurring using U-Net Architecture

Bomben Filippo
filippo.bomben@studenti.unipd.it

De Laurentis Arianna Pia
ariannapia.delaurentis@studenti.unipd.it

Illiano Alessia
alessia.illiano@studenti.unipd.it

Abstract

Image deblurring is a fundamental task in computer vision that aims to recover sharp images from blurred ones. It is particularly challenging due to the loss of high-frequency information during the blurring process. In this work we propose an efficient deblurring approach based on a U-Net architecture, trained to restore images affected by both Gaussian blur and Motion blur in multiple directions, including horizontal, vertical and diagonal, which operates directly in pixel space and is optimized for perceptual fidelity and structural accuracy. We synthetically blurred 7606 samples from the 'Labeled Faces in the Wild' dataset using parametrized Gaussian and motion kernel, then trained the network end-to-end and evaluated it on a test set using PSNR and SSIM metrics, demonstrating competitive results in both accuracy and visual quality. Our implementation is lightweight and optimized for fast execution. The code is available at https://github.com/bombi00/Image-Deblur-VCS25.

1. Introduction

Image blurring is a common issue that can occur for various reasons, particularly in dynamic scenes. It may result from a variety of complex factors, like camera shake, atmospheric turbulence, low-light conditions and object motion [1]. Blurring can make important details less clear, or even completely lost, diminishing the visual quality and impairing the performance of downstream computer vision tasks that rely on fine visual details, such as object detection and object tracking.

Image deblurring is a fundamental computer vision task that can help in solving these challenges by restoring clarity and sharpness to blurred images. Its process involves understanding the type of blur, their underlying causes, and the mathematical techniques required to develop the effective deblurring algorithm.

To address the problem of image blurring, we propose an efficient deblurring method based on the U-Net architecture, which employs an encoder-decoder structure with skip connections to effectively extract features and recover spatial details. It is trained to restore 7606 images from the 'Labeled Faces in the Wild (LFW)' dataset, which have been synthetically degraded by both Gaussian blur and directional motion blur, applied horizontally, vertically and diagonally. This approach operates directly on pixel data and is optimized to enhance both perceptual quality and structural accuracy, as demonstrated by experimental results showing strong performance in terms of Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM), which highlight the method's effectiveness in restoring blurred facial images within this controlled setting.

2. Related Work

In the field of Computer Vision, image deblurring is a well-established task that has been approached using a variety of methods. Early approaches primarily relied on Deep Convolutional Neural Networks (CNNs), which were employed for other tasks like image super-resolution [2] , involving the reconstruction of high-resolution images from low-resolution inputs. The model introduced in [2] laid the groundwork for image deblurring by demonstrating that deep CNNs could effectively recover details in blurred images.

More recently, advances in deep learning have introduced Generative Adversarial Networks (GANs) for image deblurring, as these models are capable of generating more realistic and visually convincing restorations. A notable example is DeblurGan [4] , which is based on Conditional GANs combined with a Wasserstein GAN with gradient penalty and perceptual loss. This combination enables the generation of sharper and more natural images and successfully restores fine texture details.

Several studies closely related to our approach also utilize the U-Net architecture as a foundational backbone, originally developed for biomedical image segmentation

and later adopted for image deblurring due its ability to preserve semantic information. However, the classical U-Net structure suffers from spatial information loss, which affects the quality of the reconstructed images, and to address this issue various improvements have been proposed. The Dense Block U-Net [6] integrates Residual-in-Residual Dense Blocks (RRDB) to enhance feature reuse and to reduce inference time. Meanwhile, the AMSA U-Net [5] introduces a multi-scale U-shaped architecture capable of focusing on both global and local regions while employing a self-attention mechanism in the decoder to better capture semantic information. Lastly, the [7] improves feature integration across different layers and, by using a Frequency Reconstruction Loss Function based on the Fourier Transform, further enhances the reconstruction of frequency information, leading to superior image quality.

3. Dataset

To train our model we employed the 'Labeled Faces in the Wild (LFW)' dataset, a well-known benchmark primarily designed for facial recognition tasks. Although not conventionally adopted for image deblurring, the dataset offers a controlled environment with minimal scene variability, since all the contained samples consist of close-up portraits of human faces with consistent framing and context, making it particularly suitable for developing and testing deblurring algorithms focused on facial imagery.

In contrast to widely used deblurring benchmarks, like the 'GoPro' dataset which includes a broad diversity of scenes, motions and lighting conditions, the LFW dataset offers a more constrained problem setting, enabling us to concentrate specifically on human faces optimizing deblurring performance, leveraging their uniform appearance and minimizing confounding variability.

3.1. Dataset preprocessing

To facilitate data handling and integration into our training pipeline, we accessed the dataset through the `sklearn.datasets.fetch_lfw_people` utility, which provides a preprocessed and easy loadable version of the original image set. A total of 7606 RGB images were utilized, each originally sized at 125x94 pixels and subsequently uniformly resized to 128x96 pixels to ensure compatibility with the convolutional operations and batch processing procedures within our network architecture. Thanks to this resolution, it is possible to balance computational efficiency with sufficient details for effective deblurring and to ensure compatibility with our convolutional architecture.

To simulate realistic degradation scenarios, we applied two distinct types of blur to each image:

- Gaussian blur, by applying a Gaussian filter using the OpenCV library with a kernel size of 9×9 and a sigma of 2, simulating the out-of-focus blur. These parameters were chosen to introduce a consistent and moderate level of smoothing, mimicking defocus camera lenses;
- Motion blur, to replicate blur caused by movement, by implementing a custom algorithm that randomly shifts pixels in one of the three directions: horizontal, vertical or diagonal. This synthetic motion blur introduces directional artifacts, characteristic of camera shake or subject movement during the image capture.

After degradation, the dataset was partitioned into training and testing sets using an 80/20 splits, resulting in 6084 training images and 1522 test images, ensuring a sufficiently large training set for effective model learning and optimization, while preserving an unbiased test set for subsequent performance evaluation.

4. Method

4.1. U-Net

To address the deblurring task we implemented a simple approach leveraging the U-Net architecture, visible in Figure 1 , which is commonly employed for semantic segmentation tasks, especially in the medical field. Despite not being specifically developed for image deblurring tasks, its architectural structure proves effective in recovering of many spatial details thanks to the skip connections between the two primary components: the encoder and the decoder layers, interspersed by a bottleneck.

4.1.1 Encoder

The encoder is the component used for the feature extraction phase and consists of a sequence of convolutional blocks, each followed by a pooling operation, typically max-pooling. Its main objectives are:

- to progressively reduce the spatial resolution (i.e. width and height) of the input image, effectively decreasing computational complexity while aggregating information at increasingly global scales;
- to increase the depth of the activation maps (i.e. number of channels of the feature map), enabling the network to learn more abstract and complex representations of the input contents;
- to capture the global context of the image, including structural and textural patterns that extend across broad regions of the visual field.

Thus, at each level the encoder will extract features with a progressively higher degree of abstraction, while spatial resolution is correspondingly reduced.

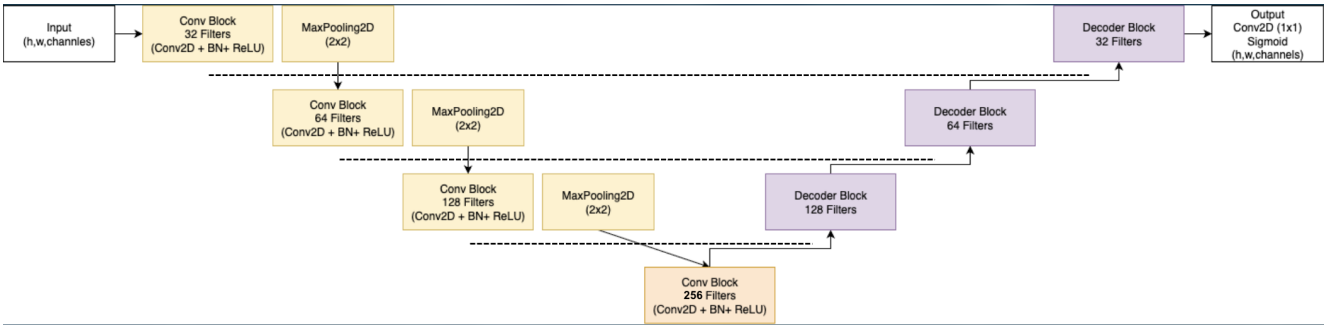

Figure 1. Implemented U-Net architecture.

4.1.2 Bottleneck

The bottleneck represents the deepest part of the U-Net architecture, located between the encoder and the decoder, in which the spatial resolution of the feature maps is at its smallest, while simultaneously the number of feature channels is at its highest. This layer captures the most abstract and high-level features of the input image, effectively summarizing the global context needed for reconstruction. The bottleneck typically consists of one or more convolutional layers, allowing the network to learn complex patterns and representations before beginning the upsampling process in the decoder. It serves as a critical bridge, connecting the encoder's compressed information to the decoder's reconstruction path.

4.1.3 Decoder

The decoder constitutes the complementary phase of spatial reconstruction, and is composed of a series of upsampling operations, often implemented via transposed convolutions or interpolations followed by standard convolutions. Its main functions are:

- restoring of the original spatial resolution of the image, transforming the learned feature maps back to the input size;
- generating a pixel-level output map which, in the case of segmentation, corresponds to assigning each pixel to a specific class;
- integrating both detailed and contextual information by combining the data extracted during the encoding phase through skip connections. This mechanism enables the recovery of spatial details and precise object boundaries.

Through this combination of global context and local details, the decoder is able to generate accurate and detailed segmentation outputs.

4.2. Our method

Our approach involved designing a U-Net architecture that carefully balances efficiency and computational cost. Consequently, the network is relatively shallow, yet it achieves strong performance.

The structure was implemented from scratch using the Keras API built on top of TensorFlow, which provided a higher level of abstraction and facilitated the use of predefined layers such as `Conv2D`, `MaxPooling2D`, and `Conv2DTranspose`.

The fundamental building block of the network consists of a convolutional layer, followed by Batch Normalization, and the application of a ReLU activation function in an element-wise manner.

The convolutional layer is the core component of each block, which applies multiple learnable filters to the input through a convolution with a 3×3 kernel in order to extract spatial features. This operation preserves spatial relationships while reducing the number of parameters compared to fully connected layers, enabling the network to detect patterns such as edges and textures at different levels of abstraction.

After the convolution, Batch Normalization is employed to normalize the pre-activation outputs across each mini-batch. This helps stabilize and accelerate the training process by reducing internal covariate shift, allowing the model to use higher learning rates and typically converges more quickly. Additionally, thanks to its regularizing effect, Batch Normalization also helps reduce overfitting.

Finally, the ReLU (Rectified Linear Unit) activation function introduces non-linearity by setting all negative values to zero while preserving positive values, enabling the network to approximate complex functions and mitigate issues like the vanishing gradient problem, thereby facilitating efficient training of deeper models.

Mathematically, the convolution operation at spatial position (x,y) can be expressed as:

$$O_{x,y} = \sum_{i=-k}^{1} \sum_{j=-k}^{1} K(i,j) I(x+i, y+j)$$

where K is the convolutional kernel of size 3×3, and I is the

input feature map.

The ReLU activation function is defined as:

$$f_{ReLU}(y) = max(0, y)$$

The encoder phase consists of three convolutional layers with 32, 64, and 128 filters respectively, producing corresponding feature maps that are then passed through a pooling layer that, at each block, halves the spatial dimensions of the feature maps, effectively shrinking the image resolution. Max pooling works by selecting the highest activation value within small regions (usually 2×2), which preserves the most important and distinctive features while filtering out less significant information. This process not only reduces the amount of data to be processed in subsequent layers, making the model more efficient, but also helps the network focus on stronger, more relevant signals.

Between the encoder and decoder phases lies the bottleneck, the deepest layer in the architecture where the feature maps have been downsampled to their smallest spatial size (in our case (h/8, w/8)), while a convolutional block with 256 filters is applied to extract the most abstract and high-level features.

The decoder gradually upsamples the feature maps to reconstruct the original spatial resolution: each decoder block takes the current feature map and, using skip connections, concatenates it with the corresponding feature map from the encoder. This mechanism enriches the feature maps by combining high-level, abstract features with low-level, detailed ones.

Finally, a convolution with sigmoid activation produces the output with the same height and width as the input, and c channels suitable for tasks like image segmentation or restoration.

4.3. Other methods

Before adopting our U-Net-based architecture, we experimented with two alternative approaches: the first was implemented by following the methodology proposed in [2], while the second involved the use of Fourier transforms as a deblurring strategy. Unfortunately, neither of these implementations yielded satisfactory results in terms of visual quality or reconstruction performance.

- By implementing the method of [2], which is based on a relatively simple architecture that, according to the original authors, achieved impressive performance on the enhancing resolution task, our results did not meet expectations: by looking at Figure 2 it's possible to notice that, despite following the described setup, the output images remained heavily blurred, as the CNN failed to capture the relevant high-frequency details, effectively worsening the blur instead of restoring the original sharpness.

  This degradation in visual quality may be attributed to a flaw in our implementation or possibly to missing details in the original paper regarding preprocessing, normalization, or training dynamics.

- We explored a frequency-domain approach based on the Fourier Transform by manipulating the high-frequency components of the blurred image. By looking at the results in the Figure 3 we can deduce that this method is computationally very fast and produces results that, while not optimal, are reasonably satisfactory given the low execution time. However, the output images often exhibited poorly defined or segmented edges, especially in regions with fine details.

  A major limitation of Fourier-based deblurring is that it requires prior knowledge of the blur kernel, and this stems from the fact that the blurred image B is modeled as the convolution of the original image I with a blur kernel K:

  $$B = I * K$$

  In the frequency domain, this relationship becomes a multiplication:

  $$\mathcal{F}(B) = \mathcal{F}(I) * \mathcal{F}(K)$$

  Therefore, to recover the original image I, the inverse operation must be applied:

  $$I = \mathcal{F}^{-1}(\frac{\mathcal{F}(B)}{\mathcal{F}(K)})$$

  However, this direct division can be numerically unstable, especially when the Fourier Transform of the kernel $\mathcal{F}(K)$ has values close to zero, which may amplify noise present in the blurred image. To mitigate this issue and stabilize the process, we applied the Wiener filter, which balances the recovery of the image with the suppression of noise.

5. Experiments

To evaluate our model, we adopted four metrics:

- **Mean Squared Error (MSE) Loss**, which measures the average of the squared differences between predicted and ground truth pixel values. It penalizes larger errors more strongly.

- **Mean Absolute Error (MAE)**, that computes the average of the absolute differences between the predicted and true pixel intensities. Unlike MSE, it is less sensitive to outliers.

Figure 2. Results after applying first approach of deblurring CNN.

Figure 3. Results of Fourier based deblurring approach.

- **Structural Similarity Index Measure (SSIM)**, a perceptual metric that quantifies image similarity by considering three key components: luminance, contrast, and structural similarity. The SSIM between two image patches x and y is defined as:

$$\text{SSIM}(x, y) = l(x, y) \cdot c(x, y) \cdot s(x, y)$$

where:

  - **Luminance:** $l(x, y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1}$
  - **Contrast:** $c(x, y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}$
  - **Structure:** $s(x, y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3}$

  Here, μ_x and μ_y denote the means, σ_x and σ_y the standard deviations, and σ_{xy} the covariance of the image patches x and y. C_1, C_2, and C_3 are small constants to avoid division by zero.

- **Peak Signal-to-Noise Ratio (PSNR)**, expressed in decibels (dB), which measures the ratio between the maximum possible pixel value and the power of the reconstruction error. It is widely used for assessing the quality of lossy image compression or reconstruction.

We trained our model for 100 epochs using a batch size of 12, relatively small given the limited number of samples, and a learning rate of 1e-4. We also employed the `Adam` optimizer, along with `ReduceLROnPlateau` (with a minimum learning rate of 1e-7) and `EarlyStopping` to prevent overfitting and to optimize training efficiency.

The initial model was significantly deeper and more complex, requiring several hours for a full training cycle.

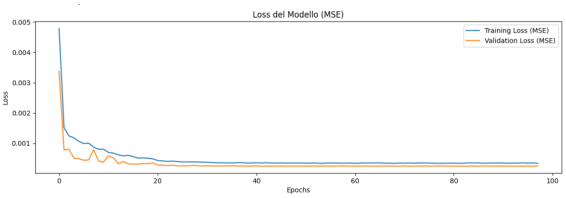

Figure 4. Loss of Gaussian Blur.

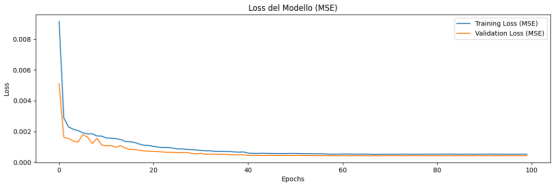

Figure 5. Loss of Motion Blur.

However, since the performance was already satisfactory, we progressively simplified the architecture, ultimately arriving at the final version presented in this work.

Figure 4 and Figure 5 report the MSE loss curves for the Gaussian blur and the motion blur cases, which have slight difference for the varying complexity of the blur types. In both cases, the training and validation losses decrease rapidly in the early epochs and then stabilize, indicating effective convergence without overfitting, and the validation loss remains close to the training one throughout, confirming a good generalization.

As also discussed in [3], SSIM alone is not a fully reliable metric for image quality assessment. For this reason, we complemented the quantitative evaluation with qualitative visual inspection.

In the following figures we display the performance un-

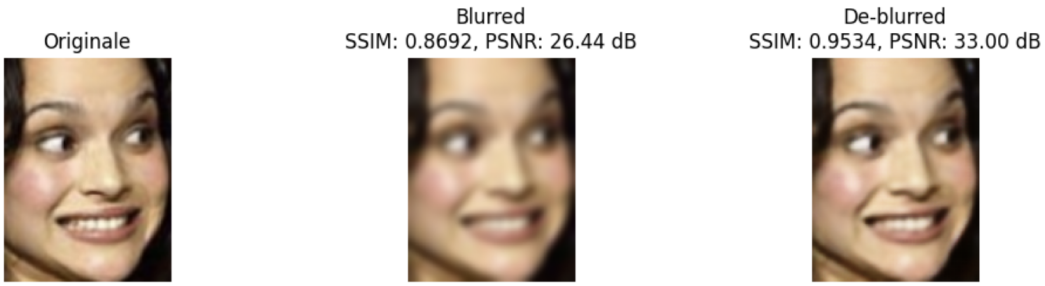

Figure 6. Results of U-Net based architecture on Gaussian Blur.

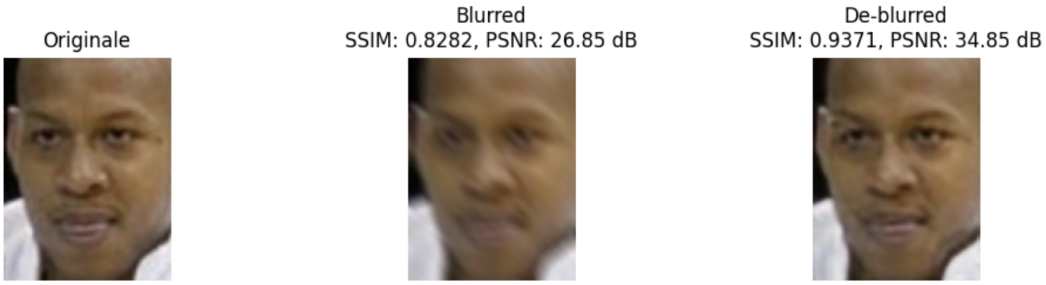

Figure 7. Results of U-Net based architecture on Motion Blur.

der Gaussian Blur conditions in Figure 6 and the corresponding results for Motion Blur in Figure 7, showing highly satisfactory results.

6. Conclusion

In this paper we have presented an efficient image deblurring approach based on the U-Net architecture, that aims to restore images representing human faces that have been degraded by both Gaussian and directional motion blur. As mentioned in the previous sections, the model benefits from the U-Net's encoder-decoder structure with skip connections, which enables the recovery of fine spatial details and ensures the preservation structural accuracy. By evaluating the results using PSNR and SSIM metrics, we confirmed that the proposed approach achieves great performance in terms of both quantitative accuracy and perceptual quality. Future work may focus on extending this framework to more diverse and challenging datasets, as well as incorporating advanced architectural components such as attention mechanisms or residual dense blocks, to further enhance restoration quality. Overall, this study reinforces the potential of U-Net based models as effective tools for image deblurring, particularly in applications involving facial imagery, and paves the way for further research in this direction.

References

[1] Palla Bhargava Rao Birru Shiva Shankar Akhilesh Pandey Ch. M. Shruthi, Vemullapalli Ramachandra Anirudh. Deep learning based automated image deblurring, 2023.

[2] Kaiming He Xiaoou Tang Chao Dong, Chen Change Loy. Image super-resolution using deep convolutional networks, 2015.

[3] Tomas Akenine-Möller Jim Nilsson. Understanding ssim, 2020.

[4] Mykola Mykhailych Dmytro Mishkin Jii Matas Orest Kupyn, Volodymyr Budzan. Deblurgan: Blind motion deblurring using conditional adversarial networks, 2018.

[5] Yingying Wang. Amsa-unet: An asymmetric multiple scales u-net based on self-attention for deblurring, 2024.

[6] Yawei Li Yinan Mao Lei He Zhoufeng Liu Yujie Wu, Hong Zhang. Dense block u-net for dynamic scene deblurring, 2020.

[7] Haizhen Wang Zuozheng Lian. An image deblurring method using improved u-net model based on multilayer fusion and attention mechanism, 2023.