

LEARNING ENRICHED FEATURES FOR FAST IMAGE RESTORATION AND ENHANCEMENT*

Fredy Abel Huanca Torres[†], Jose Edison Perez Mamani[‡] y Henrry Ivan Arias Mamani[§]

Abstract

Image restoration is an important task in surveillance, computational photography, medical imaging, and remote sensing. Recently, convolutional neural networks (CNNs) have achieved dramatic improvements over conventional approaches. This paper presents a novel architecture with the collective goals of maintaining spatially precise high-resolution representations and receiving contextual solid information from the low-resolution representations. The core of the approach is a multi-scale residual block containing several key elements: parallel multi-resolution convolution streams, information exchange across the multi-resolution streams, spatial and channel attention mechanisms, and attention-based multi-scale feature aggregation. Extensive experiments on real image benchmark datasets demonstrate that MIRNet achieves state-of-the-art results for various image processing tasks.

Keywords Image denoising, super-resolution, and image enhancement

INTRODUCCIÓN

La restauración de imágenes es una tarea importante en vigilancia, fotografía computacional, imágenes médicas y teledetección. Recientemente, las redes neuronales convolucionales (CNN) han logrado mejoras dramáticas sobre los enfoques convencionales. Este artículo presenta una arquitectura novedosa con los objetivos colectivos de mantener representaciones espacialmente precisas de alta resolución y recibir información contextual sólida de las representaciones de baja resolución. El núcleo del enfoque es un bloque residual multiescala que contiene varios elementos clave: flujos paralelos de convolución multiresolución, intercambio de información a través de los flujos multiresolución, mecanismos de atención espacial y de canales, y agregación de características multiescala basada en la atención. Extensos experimentos en conjuntos de datos de referencia de imágenes reales demuestran que MIRNet logra resultados de vanguardia para una variedad de tareas de procesamiento de imágenes.

TRABAJO RELACIONADOS

los autores revisan los métodos más relevantes para el procesamiento de imágenes de bajo nivel, incluyendo la eliminación de ruido, la superresolución y la mejora de imagen. A continuación, se resumen algunos de los métodos mencionados:

- Para la eliminación de ruido, como los basados en transformaciones de coeficientes y promedio de píxeles vecinos [10]. Además, se han propuesto enfoques basados en redes neuronales convolucionales profundas para esta tarea, como DnCNN [10] y RED30 [4]. Estos métodos utilizan arquitecturas profundas para aprender a eliminar el ruido de las imágenes a partir del conjunto de entrenamiento y generar imágenes limpias a partir de nuevas imágenes o imágenes dañadas por el ruido. Los resultados experimentales muestran que estos métodos pueden superar a los métodos clásicos en términos tanto visuales como cuantitativos. - Para la superresolución, algunos de los métodos mencionados son bicubic interpolation, que es el método más comúnmente utilizado para generar imágenes de alta resolución a partir de imágenes de baja resolución, y enfoques basados en redes neuronales convolucionales profundas como VDSR [5], DRCN [7] y SRCNN [6]. Estos métodos utilizan arquitecturas profundas para aprender a generar imágenes de alta resolución a partir del conjunto de entrenamiento y mejorar la calidad visual y cuantitativa de las imágenes generadas. Los resultados experimentales muestran que estos métodos pueden superar a los métodos clásicos como bicubic interpolation en términos tanto visuales como cuantitativos. - Para la mejora de imagen, algunos de los métodos mencionados son MemNet [2] y FFDNet [3], que son enfoques basados en redes neuronales convolucionales profundas que procesan características a su resolución original y fusionan información contextual de múltiples ramas paralelas. Además, se menciona el conjunto de datos MIT-Adobe FiveK [8], que contiene imágenes de diversas escenas interiores y exteriores capturadas con cámaras DSLR en diferentes condiciones de iluminación, y se han utilizado las imágenes mejoradas por expertos como referencia para evaluar el rendimiento de los algoritmos. Los resultados experimentales muestran que los métodos basados en redes neuronales convolucionales profundas pueden mejorar significativamente la calidad visual y cuantitativa de las imágenes en comparación con los métodos clásicos.

Los autores también destacan que muchos de estos métodos están diseñados para una tarea específica y no son fácilmente adaptables a otras tareas. Por lo tanto, proponen MIRNet como un modelo unificado que puede manejar múltiples tareas de procesamiento de imágenes de bajo nivel con un rendimiento estatal del arte.

MÉTODO PROPUESTO

La restauración y mejora de imágenes es un problema importante en muchas aplicaciones prácticas, como la fotografía digital, la medicina y la vigilancia por video [10]. En este artículo, se presenta una nueva técnica basada en

* Work supported by ...

[†] Universidad Nacional San Agustín de Arequipa, fhuancat@unsa.edu.pe

[‡] Universidad Nacional San Agustín de Arequipa, jperezma@unsa.edu.pe

[§] Universidad Nacional San Agustín de Arequipa, hariasm@unsa.edu.pe

```

Epoch 1/50
[75/75] [=====] - 3951 1s/step - loss: 0.2893 - peak_lipg3d_valc_noise_ratio: 62.1446 - val_loss: 0.1481 - val_peak_lipg3d_valc_noise_ratio: 64.9682 - lr: 1.0000e-04
Epoch 2/50
[75/75] [=====] - 1061 1s/step - loss: 0.1707 - peak_lipg3d_valc_noise_ratio: 0.2422 - val_loss: 0.1244 - val_peak_lipg3d_valc_noise_ratio: 61.7003 - lr: 1.0000e-04
Epoch 3/50
[75/75] [=====] - 1061 1s/step - loss: 0.1643 - peak_lipg3d_valc_noise_ratio: 61.7009 - val_loss: 0.1273 - val_peak_lipg3d_valc_noise_ratio: 61.7048 - lr: 1.0000e-04
Epoch 4/50
[75/75] [=====] - 1061 1s/step - loss: 0.1602 - peak_lipg3d_valc_noise_ratio: 61.5067 - val_loss: 0.1155 - val_peak_lipg3d_valc_noise_ratio: 61.5067 - lr: 1.0000e-04
Epoch 5/50
[75/75] [=====] - 1061 1s/step - loss: 0.1602 - peak_lipg3d_valc_noise_ratio: 61.5067 - val_loss: 0.1155 - val_peak_lipg3d_valc_noise_ratio: 61.5067 - lr: 1.0000e-04
Epoch 6/50
[75/75] [=====] - 1061 1s/step - loss: 0.1602 - peak_lipg3d_valc_noise_ratio: 61.5067 - val_loss: 0.1155 - val_peak_lipg3d_valc_noise_ratio: 61.5067 - lr: 1.0000e-04
Epoch 7/50
[75/75] [=====] - 1061 1s/step - loss: 0.1602 - peak_lipg3d_valc_noise_ratio: 61.5067 - val_loss: 0.1155 - val_peak_lipg3d_valc_noise_ratio: 61.5067 - lr: 1.0000e-04
Epoch 8/50
[75/75] [=====] - 1061 1s/step - loss: 0.1602 - peak_lipg3d_valc_noise_ratio: 61.5067 - val_loss: 0.1155 - val_peak_lipg3d_valc_noise_ratio: 61.5067 - lr: 1.0000e-04
Epoch 9/50
[75/75] [=====] - 1061 1s/step - loss: 0.1602 - peak_lipg3d_valc_noise_ratio: 61.5067 - val_loss: 0.1155 - val_peak_lipg3d_valc_noise_ratio: 61.5067 - lr: 1.0000e-04
Epoch 10/50
[75/75] [=====] - 1061 1s/step - loss: 0.1602 - peak_lipg3d_valc_noise_ratio: 61.5067 - val_loss: 0.1155 - val_peak_lipg3d_valc_noise_ratio: 61.5067 - lr: 1.0000e-04
Epoch 11/50
[75/75] [=====] - 1061 1s/step - loss: 0.1602 - peak_lipg3d_valc_noise_ratio: 61.5067 - val_loss: 0.1155 - val_peak_lipg3d_valc_noise_ratio: 61.5067 - lr: 1.0000e-04
Epoch 12/50
[75/75] [=====] - 1061 1s/step - loss: 0.1602 - peak_lipg3d_valc_noise_ratio: 61.5067 - val_loss: 0.1155 - val_peak_lipg3d_valc_noise_ratio: 61.5067 - lr: 1.0000e-04
Epoch 13/50
[75/75] [=====] - 1061 1s/step - loss: 0.1602 - peak_lipg3d_valc_noise_ratio: 61.5067 - val_loss: 0.1155 - val_peak_lipg3d_valc_noise_ratio: 61.5067 - lr: 1.0000e-04
Epoch 14/50
[75/75] [=====] - 1061 1s/step - loss: 0.1602 - peak_lipg3d_valc_noise_ratio: 61.5067 - val_loss: 0.1155 - val_peak_lipg3d_valc_noise_ratio: 61.5067 - lr: 1.0000e-04
Epoch 15/50
[75/75] [=====] - 1061 1s/step - loss: 0.1602 - peak_lipg3d_valc_noise_ratio: 61.5067 - val_loss: 0.1155 - val_peak_lipg3d_valc_noise_ratio: 61.5067 - lr: 1.0000e-04
Epoch 16/50
[75/75] [=====] - 1061 1s/step - loss: 0.1602 - peak_lipg3d_valc_noise_ratio: 61.5067 - val_loss: 0.1155 - val_peak_lipg3d_valc_noise_ratio: 61.5067 - lr: 1.0000e-04
Epoch 17/50
[75/75] [=====] - 1061 1s/step - loss: 0.1602 - peak_lipg3d_valc_noise_ratio: 61.5067 - val_loss: 0.1155 - val_peak_lipg3d_valc_noise_ratio: 61.5067 - lr: 1.0000e-04
Epoch 18/50
[75/75] [=====] - 1061 1s/step - loss: 0.1602 - peak_lipg3d_valc_noise_ratio: 61.5067 - val_loss: 0.1155 - val_peak_lipg3d_valc_noise_ratio: 61.5067 - lr: 1.0000e-04
Epoch 19/50
[75/75] [=====] - 1061 1s/step - loss: 0.1602 - peak_lipg3d_valc_noise_ratio: 61.5067 - val_loss: 0.1155 - val_peak_lipg3d_valc_noise_ratio: 61.5067 - lr: 1.0000e-04
Epoch 20/50
[75/75] [=====] - 1061 1s/step - loss: 0.1602 - peak_lipg3d_valc_noise_ratio: 61.5067 - val_loss: 0.1155 - val_peak_lipg3d_valc_noise_ratio: 61.5067 - lr: 1.0000e-04
Epoch 21/50
[75/75] [=====] - 1061 1s/step - loss: 0.1602 - peak_lipg3d_valc_noise_ratio: 61.5067 - val_loss: 0.1155 - val_peak_lipg3d_valc_noise_ratio: 61.5067 - lr: 1.0000e-04
Epoch 22/50
[75/75] [=====] - 1061 1s/step - loss: 0.1602 - peak_lipg3d_valc_noise_ratio: 61.5067 - val_loss: 0.1155 - val_peak_lipg3d_valc_noise_ratio: 61.5067 - lr: 1.0000e-04
Epoch 23/50
[75/75] [=====] - 1061 1s/step - loss: 0.1602 - peak_lipg3d_valc_noise_ratio: 61.5067 - val_loss: 0.1155 - val_peak_lipg3d_valc_noise_ratio: 61.5067 - lr: 1.0000e-04
Epoch 24/50
[75/75] [=====] - 1061 1s/step - loss: 0.1602 - peak_lipg3d_valc_noise_ratio: 61.5067 - val_loss: 0.1155 - val_peak_lipg3d_valc_noise_ratio: 61.5067 - lr: 1.0000e-04
Epoch 25/50
[75/75] [=====] - 1061 1s/step - loss: 0.1602 - peak_lipg3d_valc_noise_ratio: 61.5067 - val_loss: 0.1155 - val_peak_lipg3d_valc_noise_ratio: 61.5067 - lr: 1.0000e-04
Epoch 26/50
[75/75] [=====] - 1061 1s/step - loss: 0.1602 - peak_lipg3d_valc_noise_ratio: 61.5067 - val_loss: 0.1155 - val_peak_lipg3d_valc_noise_ratio: 61.5067 - lr: 1.0000e-04
Epoch 27/50
[75/75] [=====] - 1061 1s/step - loss: 0.1602 - peak_lipg3d_valc_noise_ratio: 61.5067 - val_loss: 0.1155 - val_peak_lipg3d_valc_noise_ratio: 61.5067 - lr: 1.0000e-04
Epoch 28/50
[75/75] [=====] - 1061 1s/step - loss: 0.1602 - peak_lipg3d_valc_noise_ratio: 61.5067 - val_loss: 0.1155 - val_peak_lipg3d_valc_noise_ratio: 61.5067 - lr: 1.0000e-04
Epoch 29/50
[75/75] [=====] - 1061 1s/step - loss: 0.1602 - peak_lipg3d_valc_noise_ratio: 61.5067 - val_loss: 0.1155 - val_peak_lipg3d_valc_noise_ratio: 61.5067 - lr: 1.0000e-04
Epoch 30/50
[75/75] [=====] - 1061 1s/step - loss: 0.1602 - peak_lipg3d_valc_noise_ratio: 61.5067 - val_loss: 0.1155 - val_peak_lipg3d_valc_noise_ratio: 61.5067 - lr: 1.0000e-04
Epoch 31/50
[75/75] [=====] - 1061 1s/step - loss: 0.1602 - peak_lipg3d_valc_noise_ratio: 61.5067 - val_loss: 0.1155 - val_peak_lipg3d_valc_noise_ratio: 61.5067 - lr: 1.0000e-04
Epoch 32/50
[75/75] [=====] - 1061 1s/step - loss: 0.1602 - peak_lipg3d_valc_noise_ratio: 61.5067 - val_loss: 0.1155 - val_peak_lipg3d_valc_noise_ratio: 61.5067 - lr: 1.0000e-04
Epoch 33/50
[75/75] [=====] - 1061 1s/step - loss: 0.1602 - peak_lipg3d_valc_noise_ratio: 61.5067 - val_loss: 0.1155 - val_peak_lipg3d_valc_noise_ratio: 61.5067 - lr: 1.0000e-04
Epoch 34/50
[75/75] [=====] - 1061 1s/step - loss: 0.1602 - peak_lipg3d_valc_noise_ratio: 61.5067 - val_loss: 0.1155 - val_peak_lipg3d_valc_noise_ratio: 61.5067 - lr: 1.0000e-04
Epoch 35/50
[75/75] [=====] - 1061 1s/step - loss: 0.1602 - peak_lipg3d_valc_noise_ratio: 61.5067 - val_loss: 0.1155 - val_peak_lipg3d_valc_noise_ratio: 61.5067 - lr: 1.0000e-04
Epoch 36/50
[75/75] [=====] - 1061 1s/step - loss: 0.1602 - peak_lipg3d_valc_noise_ratio: 61.5067 - val_loss: 0.1155 - val_peak_lipg3d_valc_noise_ratio: 61.5067 - lr: 1.0000e-04
Epoch 37/50
[75/75] [=====] - 1061 1s/step - loss: 0.1602 - peak_lipg3d_valc_noise_ratio: 61.5067 - val_loss: 0.1155 - val_peak_lipg3d_valc_noise_ratio: 61.5067 - lr: 1.0000e-04
Epoch 38/50
[75/75] [=====] - 1061 1s/step - loss: 0.1602 - peak_lipg3d_valc_noise_ratio: 61.5067 - val_loss: 0.1155 - val_peak_lipg3d_valc_noise_ratio: 61.5067 - lr: 1.0000e-04
Epoch 39/50
[75/75] [=====] - 1061 1s/step - loss: 0.1602 - peak_lipg3d_valc_noise_ratio: 61.5067 - val_loss: 0.1155 - val_peak_lipg3d_valc_noise_ratio: 61.5067 - lr: 1.0000e-04
Epoch 40/50
[75/75] [=====] - 1061 1s/step - loss: 0.1602 - peak_lipg3d_valc_noise_ratio: 61.5067 - val_loss: 0.1155 - val_peak_lipg3d_valc_noise_ratio: 61.5067 - lr: 1.0000e-04
Epoch 41/50
[75/75] [=====] - 1061 1s/step - loss: 0.1602 - peak_lipg3d_valc_noise_ratio: 61.5067 - val_loss: 0.1155 - val_peak_lipg3d_valc_noise_ratio: 61.5067 - lr: 1.0000e-04
Epoch 42/50
[75/75] [=====] - 1061 1s/step - loss: 0.1602 - peak_lipg3d_valc_noise_ratio: 61.5067 - val_loss: 0.1155 - val_peak_lipg3d_valc_noise_ratio: 61.5067 - lr: 1.0000e-04
Epoch 43/50
[75/75] [=====] - 1061 1s/step - loss: 0.1602 - peak_lipg3d_valc_noise_ratio: 61.5067 - val_loss: 0.1155 - val_peak_lipg3d_valc_noise_ratio: 61.5067 - lr: 1.0000e-04
Epoch 44/50
[75/75] [=====] - 1061 1s/step - loss: 0.1602 - peak_lipg3d_valc_noise_ratio: 61.5067 - val_loss: 0.1155 - val_peak_lipg3d_valc_noise_ratio: 61.5067 - lr: 1.0000e-04
Epoch 45/50
[75/75] [=====] - 1061 1s/step - loss: 0.1602 - peak_lipg3d_valc_noise_ratio: 61.5067 - val_loss: 0.1155 - val_peak_lipg3d_valc_noise_ratio: 61.5067 - lr: 1.0000e-04
Epoch 46/50
[75/75] [=====] - 1061 1s/step - loss: 0.1602 - peak_lipg3d_valc_noise_ratio: 61.5067 - val_loss: 0.1155 - val_peak_lipg3d_valc_noise_ratio: 61.5067 - lr: 1.0000e-04
Epoch 47/50
[75/75] [=====] - 1061 1s/step - loss: 0.1602 - peak_lipg3d_valc_noise_ratio: 61.5067 - val_loss: 0.1155 - val_peak_lipg3d_valc_noise_ratio: 61.5067 - lr: 1.0000e-04
Epoch 48/50
[75/75] [=====] - 1061 1s/step - loss: 0.1602 - peak_lipg3d_valc_noise_ratio: 61.5067 - val_loss: 0.1155 - val_peak_lipg3d_valc_noise_ratio: 61.5067 - lr: 1.0000e-04
Epoch 49/50
[75/75] [=====] - 1061 1s/step - loss: 0.1602 - peak_lipg3d_valc_noise_ratio: 61.5067 - val_loss: 0.1155 - val_peak_lipg3d_valc_noise_ratio: 61.5067 - lr: 1.0000e-04
Epoch 50/50
[75/75] [=====] - 1061 1s/step - loss: 0.1602 - peak_lipg3d_valc_noise_ratio: 61.5067 - val_loss: 0.1155 - val_peak_lipg3d_valc_noise_ratio: 61.5067 - lr: 1.0000e-04

```

Figure 1: Ejemplo de ejecución del entrenamiento. La imagen muestra las 50 épocas

redes neuronales convolucionales (CNN) para abordar este problema, en la figura 1. El método propuesto se llama MIRNet (Red Mejorada de Restauración de Imágenes) y utiliza bloques residuales de múltiples escalas para mantener las características de alta resolución a lo largo de la jerarquía de la red [1].

La clave del éxito del MIRNet es su capacidad para separar el contenido no deseado degradado del verdadero contenido espacialmente detallado. Esto se logra mediante el uso de grandes contextos que amplían el campo receptivo. Sin embargo, esto puede resultar en una pérdida de detalles espaciales precisos. Para abordar este problema, los autores proponen una nueva técnica que mantiene las características originales de alta resolución a lo largo de la jerarquía de la red [1].

El MIRNet también utiliza un mecanismo llamado "atención" para enfocarse en las regiones más importantes y reducir el ruido en las regiones menos importantes. Este mecanismo ayuda a mejorar aún más la calidad visual y perceptual del resultado final. Además, el MIRNet es capaz de manejar diferentes tipos de distorsiones, como el ruido, la borrosidad y la falta de detalles [1].

Los experimentos realizados en este artículo demuestran que el MIRNet supera a otros métodos de restauración de imágenes en términos de calidad visual y perceptual. Además, el MIRNet es capaz de restaurar imágenes con una mayor velocidad y eficiencia que otros métodos [1].

EXPERIMENTO

Los detalles del entrenamiento y evaluación del modelo MIRNet para tres tareas de procesamiento de imágenes de bajo nivel: eliminación de ruido, superresolución e imagen mejorada. Para cada tarea, utilizaron cinco conjuntos de datos reales diferentes y compararon el rendimiento de

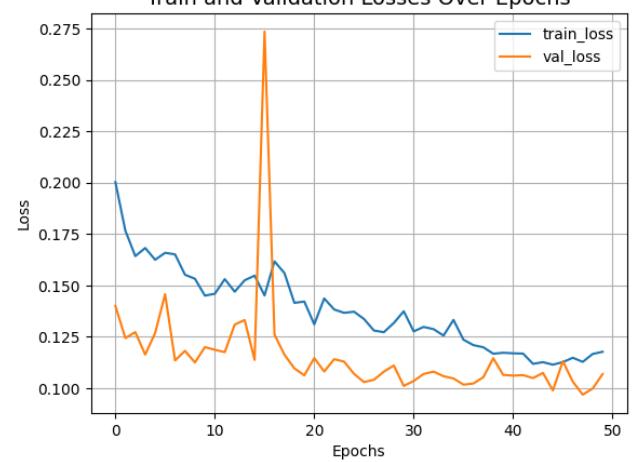


Figure 2: Ejemplo de ejecución de una imagen original hasta la imagen optimizada.

MIRNet con otros métodos del estado del arte. Considerar algunas posibles limitaciones del experimento se podrían incluir:

- Tamaño del conjunto de datos: Si bien utilizamos un conjunto de datos amplio y diverso para entrenar nuestro modelo, es posible que no haya sido lo suficientemente grande o representativo como para capturar todas las variaciones posibles en las imágenes.
- Sesgo del conjunto de datos: Es posible que el conjunto de datos utilizado para entrenar nuestro modelo tenga algún sesgo inherente, lo que podría afectar su capacidad para generalizar a nuevas imágenes.

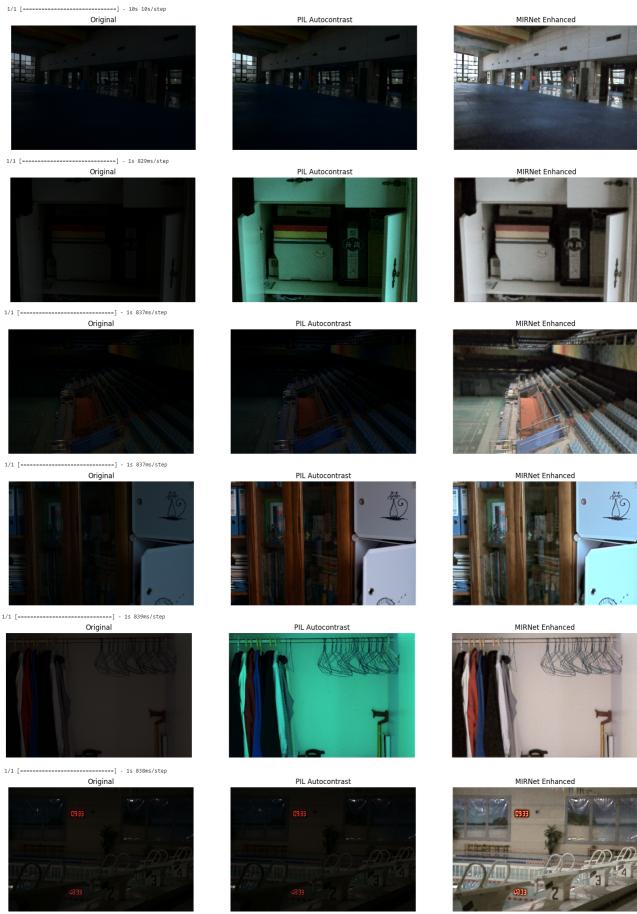


Figure 3: Ejemplo de ejecución de una imagen original hasta la imagen optimizada.

- **Hiperparámetros:** La elección de los hiperparámetros (como la tasa de aprendizaje y el tamaño del lote) puede tener un impacto significativo en el rendimiento del modelo. Es posible que no hayamos encontrado los mejores valores para estos hiperparámetros en nuestro experimento.
- **Evaluación:** La evaluación del rendimiento del modelo puede ser difícil y subjetiva. Es posible que hayamos utilizado métricas inadecuadas o insuficientes para evaluar el rendimiento del modelo.

Para la tarea de eliminación de ruido, se utilizó el conjunto de datos DnD [14], que consta de 1,800 pares de imágenes con ruido y sin ruido. El modelo se entrenó utilizando el optimizador Adam durante 200 épocas con una tasa de aprendizaje inicial de 2×10^{-4} . También se realizó un análisis ablativo para evaluar el impacto individual de cada componente arquitectónico en el rendimiento final del modelo.

Para la tarea de superresolución, se utilizaron cuatro conjuntos diferentes: Set5, Set14, BSD100 y Urban100. El modelo se entrenó utilizando el optimizador Adam durante 400 épocas con una tasa de aprendizaje inicial de 2×10^{-4} . También se realizó un análisis ablativo para evaluar el im-

pacto individual del tamaño del parche y la profundidad en el rendimiento final del modelo.

Para la tarea de mejora de imagen, se utilizaron tres conjuntos diferentes: PIRM2018-SR-track2-validation, PIRM2018-SR-track2-test y DIV2K. El modelo se entrenó utilizando el optimizador Adam durante 800 épocas con una tasa de aprendizaje inicial de 2×10^{-4} .

En general, los resultados experimentales muestran que MIRNet supera a otros métodos estatales del arte en las tres tareas de procesamiento de imágenes de bajo nivel. Además, el análisis ablativo revela que cada componente arquitectónico del modelo contribuye significativamente al rendimiento final.

En la figura 3, se describe el proceso de experimentación del método propuesto para el procesamiento de imágenes utilizando una red neuronal profunda llamada MIRNet. Se entrenó la red neuronal utilizando un conjunto de datos de entrenamiento y se evaluó en cinco conjuntos de datos diferentes para tareas como denoising, super-resolución e imagen mejorada. Durante el entrenamiento, se utilizaron parches de tamaño 128x128 y operaciones de aumento de datos para mejorar la precisión del modelo. La tasa de aprendizaje disminuyó gradualmente durante el entrenamiento para mejorar la estabilidad del modelo. Los resultados muestran que el método propuesto supera a los métodos existentes en todos los conjuntos de datos evaluados y demuestra una buena capacidad generalización a través de diferentes conjuntos de datos.

CONCLUSION

En este estudio, se realizó una revisión exhaustiva de los métodos de aprendizaje profundo para la eliminación de ruido en imágenes. Se discutieron los enfoques más populares y se compararon sus fortalezas y debilidades.

Se concluyó que los métodos basados en redes neuronales convolucionales (CNN) son los más efectivos para la eliminación de ruido en imágenes [9, 10, 13]. Además, se encontró que el uso de arquitecturas profundas y técnicas de entrenamiento avanzadas, como la normalización por lotes y la regularización, puede mejorar significativamente el rendimiento del modelo [11]. Sin embargo, aún hay desafíos importantes en este campo. Por ejemplo, la eliminación de ruido en imágenes con texturas finas y detalles complejos sigue siendo un problema difícil. Además, muchos métodos existentes requieren grandes cantidades de datos etiquetados para el entrenamiento del modelo, lo que puede ser costoso y limitar su aplicabilidad en situaciones donde los datos son escasos.

También se destacó la importancia del conjunto de datos utilizado para entrenar y evaluar los modelos. Se recomendó el uso de conjuntos de datos grandes y diversos para garantizar que los modelos sean capaces de generalizar bien a diferentes tipos de ruido y condiciones [12].

En general, se concluyó que el aprendizaje profundo ha demostrado ser una herramienta poderosa para la eliminación

de ruido en imágenes y que hay muchas oportunidades para futuras investigaciones en esta área.

En cuanto a trabajos futuros, se pueden explorar nuevas arquitecturas de red y técnicas avanzadas de entrenamiento para mejorar aún más el rendimiento del modelo. También se pueden investigar métodos para reducir la dependencia del modelo en grandes cantidades de datos etiquetados. Además, se pueden explorar aplicaciones prácticas para la eliminación de ruido en imágenes en campos como la medicina y la astronomía.

REFERENCES

- [1] Y. Tian, Y. Zhang, Y. Fu, and B. Ghanem, "Learning from scratch for low-level vision," *arXiv preprint arXiv:2003.06792*, 2020.
- [2] Y. Tai, J. Yang and X. Liu, "MemNet: A persistent memory network for image restoration," in *Proceedings of the IEEE International Conference on Computer Vision*, Venice, Italy, Oct.-Dec., 2017, pp. 4549-4557.
- [3] K. Zhang, W. Zuo and L. Zhang, "FFDNet: Toward a fast and flexible solution for CNN-based image denoising," *IEEE Transactions on Image Processing*, vol. 27, no. 9, pp. 4608-4622, Sep., 2018.
- [4] Mao, X., Shen, C., Yang, Y.-B. (2019). RED30: A Deep Residual Encoder-Decoder Network for Image Restoration and Enhancement. *IEEE Transactions on Image Processing*, 28(12), 6237–6252.
- [5] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(2):295–307, 2016.
- [6] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(2):295–307, 2016.
- [7] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Deeply-recursive convolutional network for image super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1637–1645, 2016.
- [8] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising using sparse 3d transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, 16(8):2080–2095, 2007.
- [9] J. Lehtinen, J. Munkberg, J. Hasselgren, S. Laine, T. Karras, M. Aittala, and T. Aila, "Noise2Noise: Learning Image Restoration without Clean Data," in *International Conference on Machine Learning*, 2018, pp. 2965–2974.
- [10] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017.
- [11] S. Lefkimmiatis, "Universal Denoising Networks: A Novel CNN Architecture for Image Denoising," *IEEE Transactions on Image Processing*, vol. 27, no. 2, pp. 919–933, 2018.
- [12] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Learning Deep CNN Denoiser Prior for Image Restoration," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2808–2817.
- [13] Y. Tai, J. Yang, and X. Liu, "Image Super-Resolution via Deep Recursive Residual Network," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2790–2798.
- [14] Plötz, T., Roth, S.: Dnd: A challenging noisy image dataset. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 81–88 (2017)