# Técnicas de Deep Learning basadas en medidas inerciales para la mejora de las capacidades de teledetección en sensores electro-ópticos embarcados

**Delgado Fernández, Alba [1], Blázquez García, Rodrigo [1] y Burgos García, Mateo [1,*].**

[1] Observatorio Horizonte en Defensa y Seguridad Isdefe-UPM. Av. Complutense, 30. 28040 Madrid. Correos electrónicos: alba.delgado.fernandez@alumnos.upm.es (ADF), rodrigo. blazquez@upm.es (RBG), mateo.burgos@upm.es (MBG)

***** Autor Principal y responsable del trabajo; Correo electrónico: mateo.burgos@upm.es (MBG)

---

**Resumen:** En los últimos años, el uso de plataformas aéreas no tripuladas (*Unmanned Aerial Vehicles*, UAV) ha cobrado especial importancia en aplicaciones de defensa y seguridad. En el ámbito de la teledetección utilizando sensores electro-ópticos embarcados, es primordial conseguir imágenes nítidas para que los algoritmos de detección localicen los blancos de interés con una probabilidad de falsa alarma reducida. Sin embargo, la calidad de las imágenes adquiridas con sensores ópticos embarcados en plataformas móviles se ve degradada por el llamado efecto de emborronado o *blurring* producido por el movimiento y vibración de la plataforma. A pesar del empleo de sistemas de estabilización, existe generalmente un emborronado residual que limita el tamaño mínimo detectable. Por ello, este trabajo analiza técnicas de *Deep Learning* para compensar y aminorar el efecto de emborronado y conseguir mejorar las capacidades operativas de teledetección de los sistemas electro-ópticos embarcados. Se ha propuesto utilizar las medidas de sensores IMU (*Inertial Measurement Unit*), para obtener una estimación de la función de transferencia (*Point Spread Function*) asociada al movimiento y vibración de la cámara que da lugar al emborronado espacialmente variable. Dicha estimación y la imagen original emborronada se utilizan como entradas de una red neuronal convolucional que, tras su entrenamiento con imágenes emborronadas simuladas, permite obtener imágenes más nítidas que las originales. Los resultados obtenidos muestran el gran potencial de las técnicas de *Deep Learning* para llevar a cabo el desemborronado de las imágenes adquiridas mediante cámaras embarcadas en plataformas aéreas, mejorando la calidad de las imágenes y las capacidades de teledetección.

**Palabras clave:** Deep learning, desemborronado, IMU, PSF, teledetección, UAV.

---

## 1. Introducción

En los últimos años los UAVs se han convertido en una de las tecnologías duales con mayor potencial de crecimiento debido a su rápido desarrollo y expansión. Presentan una gran diversidad de aplicaciones relacionadas con la defensa y la seguridad debido a su capacidad de recopilar datos de manera segura y rápida, sus bajos costes operativos en relación con plataformas aéreas pilotadas y a que son idóneos para acceder a zonas de conflicto. En estas aplicaciones, los UAVs suelen embarcar diversidad de sensores entre los que destacan los sensores electro-ópticos, debido a que son capaces de detectar e identificar objetos y son normalmente sensores compactos. Además, estas plataformas suelen contar con sensores IMU (*Inertial Measurement System*), que llevan a cabo medidas de aceleración, rotación y orientación, para conocer los movimientos tridimensionales de la plataforma.

Una problemática que surge a la hora de utilizar los UAVs para tareas de teledetección, sobre todo las plataformas multirotor, es la estabilidad de la plataforma, que impacta sobre la calidad de las imágenes adquiridas debido al efecto de emborronado (*blurring*) que se produce por el movimiento y vibración de las cámaras embarcadas durante su tiempo de exposición. A pesar de disponer de sistemas de estabilización, existe generalmente un emborronado residual de las imágenes que limita el tamaño mínimo detectable y el alcance máximo de estos sistemas para discriminar los blancos de interés.

El objetivo de este artículo es analizar y desarrollar diferentes algoritmos basados en técnicas de *Deep Learning* para compensar y aminorar el efecto de *blurring* en imágenes adquiridas con cámaras embarcadas en plataformas aéreas. De esta forma, se desea evaluar el potencial de estos algoritmos para mejorar las capacidades operativas de teledetección. Para ello, se propone utilizar las medidas realizadas por los sensores IMU para obtener una estimación de la función de transferencia (PSF) asociada al movimiento y vibración de la cámara que da lugar al emborronado, espacialmente variable, y utilizar dicha estimación junto con la imagen adquirida como entradas de una red neuronal convolucional que compense el movimiento de la plataforma.

Este artículo se organiza de la siguiente forma: El apartado 2 describe la arquitectura de la red neuronal propuesta. El apartado 3 detalla la generación del *dataset* sintético para realizar el entrenamiento de la red y el procesado necesario para obtener las estimaciones de la PSF a partir de las medidas inerciales. El apartado 4 muestra los resultados obtenidos del entrenamiento de la red. Finalmente, el apartado 5 expone las conclusiones del artículo.

## 2. Red neuronal convolucional propuesta

Para llevar a cabo el desemborronado de las imágenes, se propone el uso de una red neuronal convolucional (CNN) de tipo *encoder-decoder*, cuya arquitectura se representa en la Figura 1, basada en [1]. La parte *encoder* se compone de dos capas convolucionales con función de activación ReLU (unidad lineal rectificada), y una operación de *max pooling.* Por otro lado, la capa *decoder*, consiste en una capa de deconvolución (muestreo ascendente del mapa de características, seguido de una convolución), una concatenación con el mapa de características asociado a la etapa *encoder* correspondiente y dos capas convolucionales con ReLU. Por último, en la capa final se lleva a cabo una convolución de 1x1 para asignar el valor de cada píxel de salida a partir de cada vector de características de 64 componentes, quedando finalmente una red con 23 capas convolucionales.

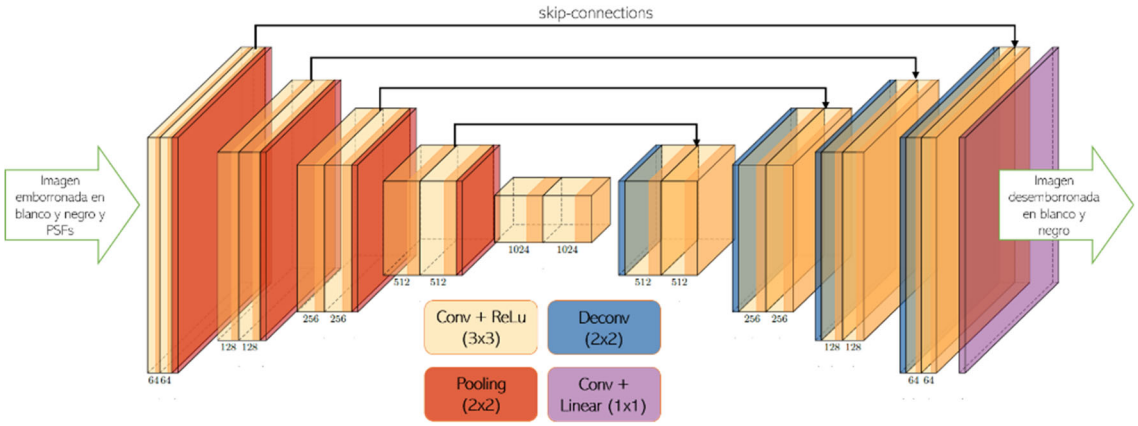



**Figura 1**. Arquitectura de la red neuronal convolucional propuesta tipo *encoder-decoder*.

Las entradas de la red varían según el caso evaluado. Como se describe en el siguiente apartado, el Entrenamiento 1 se realiza considerando solamente las imágenes emborronadas en escala de grises. El Entrenamiento 2 se realiza contando con las imágenes emborronadas y con los campos de *blur* en las direcciones X e Y calculados mediante la aproximación lineal de la función de transferencia a partir de su punto inicial y final. En el Entrenamiento 3, los campos de *blur* se calculan mediante la aproximación lineal considerando todos los puntos de las PSFs estimadas. Por último, el Entrenamiento 4 contará con las imágenes emborronadas y con una imagen en escala de grises que representa las PSFs estimadas (variables espacialmente) en una rejilla de 10x10 celdas.

## 3. Generación del *dataset* y estimación de las PSF a partir de medidas inerciales

El entrenamiento de la red neuronal requiere disponer de un *dataset* lo suficientemente grande de imágenes adquiridas mediante UAVs que incluya las imágenes emborronadas, las PSF de emborronado asociadas al movimiento, y las propias imágenes nítidas sin estar afectadas por el movimiento de la cámara. Como la generación de este *dataset* de forma experimental conlleva una gran complejidad, se ha desarrollado un proceso que simula la adquisición de imágenes mediante cámaras embarcadas afectadas por el movimiento y la vibración del UAV. De esta forma, se puede generar un *dataset* sintético con un gran número de imágenes que presentan diferentes tipos de *blurring* asociados a distintas magnitudes de movimiento y vibración de la plataforma. Además, se propone evaluar tres formas (Entrenamientos 2-4) para introducir en la red neuronal la estimación de las PSFs variables espacialmente a través las medidas realizadas por sensores IMU.

### *3.1. Generación del dataset*

Para generar el *dataset,* se ha tomado como base un conjunto de 30k imágenes [2] de las cuales se han utilizado 7.5k para aplicarles emborronados diferentes y aleatorios, repitiendo el proceso 10 veces para cada imagen. Para simular la adquisición de las imágenes afectadas por el movimiento y la vibración de la plataforma, se ha considerado la geometría representada en la Figura 2a con los siguientes parámetros aleatorios: (i) velocidad lineal de vuelo del UAV entre 0 y 15 m/s, (ii) altura de vuelo entre 100 y 500 m, (iii) ángulo de apuntamiento (*tilt*, $\theta$) entre 30 y 60° y (iv) desviación típica de la amplitud de las vibraciones en *roll*, *pitch* y *yaw*, relacionada con la estabilidad de vuelo de la plataforma, entre 0 y 0.4°. Además, se han considerado como parámetros fijos de la cámara un tamaño de imagen de 250x250 píxeles, un *field-of-view* (FOV) de 15° y un tiempo de exposición de 50 ms.

El modelo estocástico de vibración de UAVs utilizado en este trabajo artículo se describe en [3]. A partir de la vibración angular y de la traslación lineal de la plataforma, se realizan transformaciones

a través de proyecciones para obtener las PSFs, analizando cómo los puntos del plano fotografiado son proyectados en el plano de la imagen a lo largo del tiempo de exposición. Para ello, se considera una aproximación de Tierra plana y un modelo de cámara *pin-hole.* Como muestra la Figura 2b, el emborronado ocasionado es espacialmente variable. Por esta razón, para que el *dataset* puedan ser generado de manera eficiente en un tiempo limitado y sin requerir una carga computacional excesiva mediante la convolución de las imágenes originales y las PSFs variables espacialmente utilizando el algoritmo descrito en [4], se divide cada imagen de 250x250 píxeles en 100 parches de 25x25 píxeles, en los que se asume como PSF invariante la asociada a su píxel central.

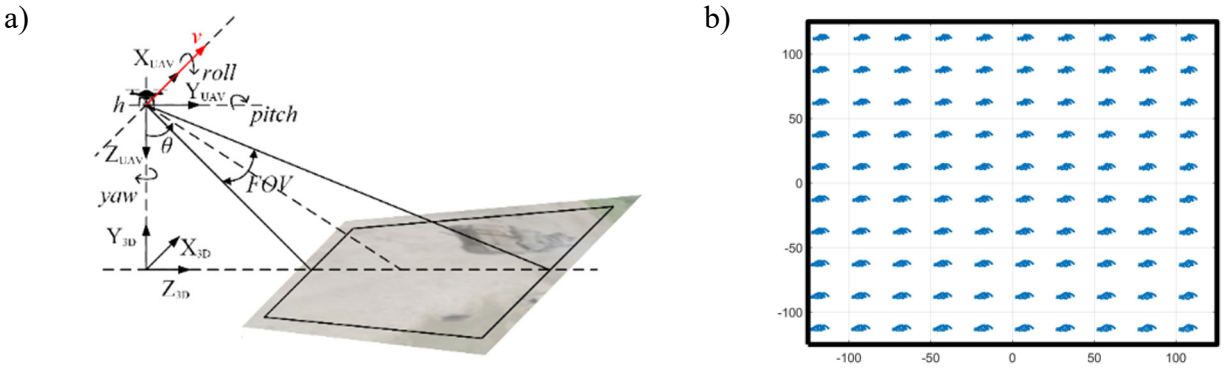



**Figura 2**. a) Geometría para la adquisición de imágenes de la superficie terrestre mediante cámaras embarcadas en UAVs, y b) Ejemplos de PSFs espacialmente variables debidas a la vibración en los tres ejes de rotación (*roll*, *pitch* y *yaw*) y a la traslación lineal del UAV durante el tiempo de exposición.

### *3.2. Estimación de las PSFs*

Para introducir a la red neuronal la estimación de las PSFs determinada a partir de la traslación y de las medidas de las rotaciones angulares obtenidas por la IMU durante el tiempo de exposición se plantean tres posibles estrategias (Entrenamientos 2-4) cuyo funcionamiento se evalúa posteriormente. Además, para simular la medida de las IMUs, se considera que las vibraciones angulares son medidas con una relación señal a ruido variable entre 20 y 40 dB y con una frecuencia de muestreo de 1 kHz.

*a) Estimación de las PSFs mediante aproximaciones lineales (Entrenamientos 2 y 3)*

Los dos primeros métodos se basan en aproximar las PSFs mediante un segmento lineal con una cierta extensión en la dirección X e Y, dando dicho resultado en forma de dos mapas, uno para cada dirección, para todos los píxeles de la imagen. El primer método (Entrenamiento 2) considera únicamente el desplazamiento y rotación total de la plataforma efectuado entre el inicio y final del tiempo de exposición (es decir, el segmento entre el punto inicial y final de las PSFs asociadas a cada píxel de la imagen), mientras que el segundo método (Entrenamiento 3) ajusta a una función lineal los puntos proyectados en el plano de la imagen a lo largo del tiempo de exposición. En ambos casos, es necesario integrar las medidas obtenidas por los giroscopios para determinar la rotación de la cámara. Una vez obtenidas las matrices de rotación se calculan los campos de *blur* en X e Y mediante el desplazamiento de los píxeles en ambos ejes. Para ello, en base a la geometría considerada, se emplea la matriz de homografía plana $H(t)$ dada por la siguiente ecuación:

$$H(t) = K\left[R(t) - \frac{l(t)n^T}{d}\right]K^{-1} \quad (1)$$

donde $R(t)$ es la matriz de rotación calculada a partir de las medidas de la IMU, l(t) es el vector de translación $[v \cdot t, 0,0]^T$ asumiendo únicamente un desplazamiento en la dirección X, $K$ es la matriz intrínseca de la cámara que se obtiene mediante su calibración, $d$ es la distancia al plano de la escena

y $n$ es el vector normal a dicho plano. Para cada píxel, dado por el vector $p_0$, el primer método sólo requiere determinar la matriz de homografía para $t = t_{exposición}$, mientras que el segundo método requiere calcular las matrices de homografía para cada tiempo de muestreo de la IMU durante el tiempo de exposición con el objetivo de estimar los puntos que componen la PSF discreta, $p'(t) = H(t) \cdot p_0$, y realizar un ajuste por un segmento lineal. Por tanto, segunda estrategia requiere una mayor carga computacional. En la Figura 3, se puede observar una rejilla de 100 celdas con las PSFs discretas calculadas para el píxel central de cada región y un ejemplo de las dos aproximaciones consideradas.

*b) Estimación de las PSFs en formato rejilla (Entrenamiento 4)*

En este caso, se genera una imagen en escala de grises que representa la acumulación en cada píxel de los puntos proyectados (mediante las matrices de homografía) asociados a los píxeles centrales de una rejilla de 10x10 obtenidos a partir de las medidas del giroscopio y del desplazamiento durante el tiempo de exposición. Por lo tanto, como se ejemplifica en la Figura 3d, esta imagen representa una superposición de las PSFs discretas estimadas para cada uno de los píxeles centrales de la rejilla.

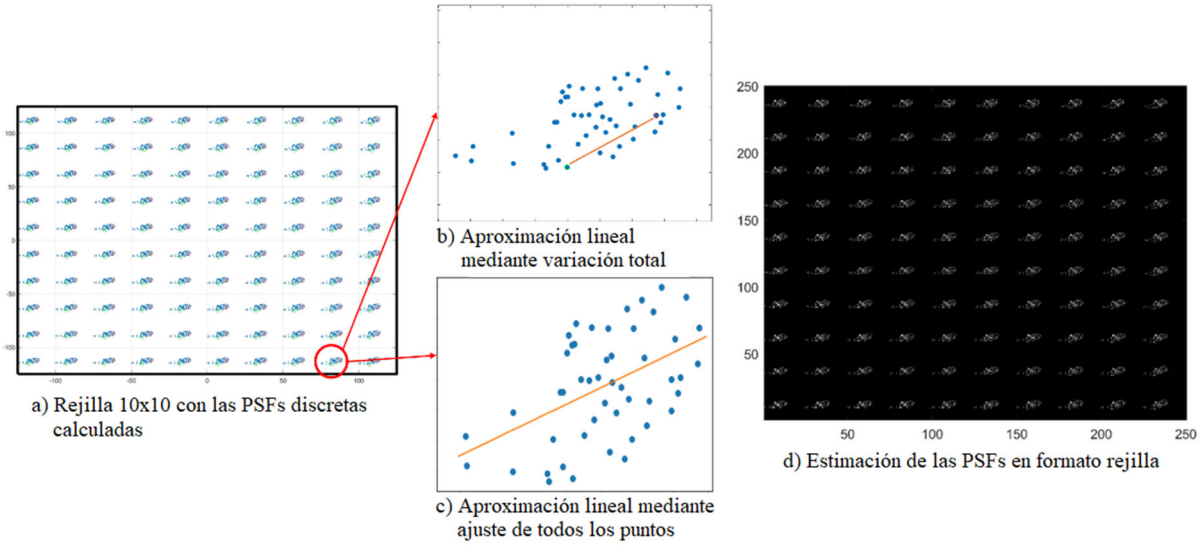



**Figura 3**. (a) Rejilla con las PSFs discretas estimadas, (b) aproximación lineal considerando la rotación y desplazamiento total (segmento entre punto inicial y final); (c) aproximación lineal ajustando todos los puntos, y (d) imagen en escala de grises con la estimación de las PSFs en formato rejilla.

## 4. Resultados

Para llevar a cabo los entrenamientos de la red neuronal en base a las distintas estrategias propuestas, se ha hecho uso de 75k imágenes las cuales se han dividido en tres conjuntos diferentes y aleatorizados (64% para entrenamiento, 16% para validación y 20% para las pruebas tras los entrenamientos). En los entrenamientos se ha seleccionado la métrica MSE (error cuadrático medio) para evaluar la función de pérdida *loss*. Como optimizador de la tasa de pérdidas se ha utilizado el optimizador Adam, junto con una *learning rate* de 0,0005, que se reduce a la mitad cada 10 *epochs*. Por último, se establece que el proceso de entrenamiento se realiza en 40 *epochs* con un *batch-size* de 16, debido a la limitación de los recursos computacionales disponibles.

La Figura 4 representa la evolución de las métricas MSE (*Mean Square Error*) y SSIM (*Structural Similarity Index Measure*) [5] durante los cuatro entrenamientos realizados, que se diferencian en las entradas utilizadas para la red neuronal: (1) imagen en escala de grises; (2) imagen en escala de grises + campos de *blur* en las direcciones X e Y calculados a partir de la aproximación lineal entre el punto

inicial y final de la función de transferencia; (3) imagen en escala de grises + campos de *blur* en las direcciones X e Y calculados a partir de la aproximación lineal utilizando todos los puntos de la función de transferencia, y (4) imagen en escala de grises + estimación de las PSFs en formato rejilla.

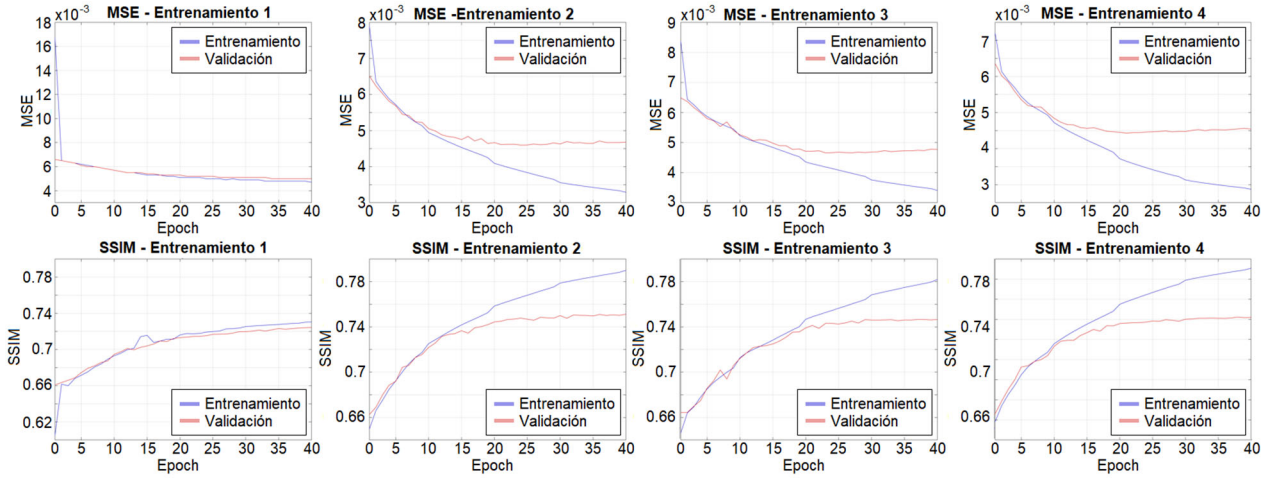



**Figura 4**. Resultados de las métricas MSE y SSIM para los 4 entrenamientos

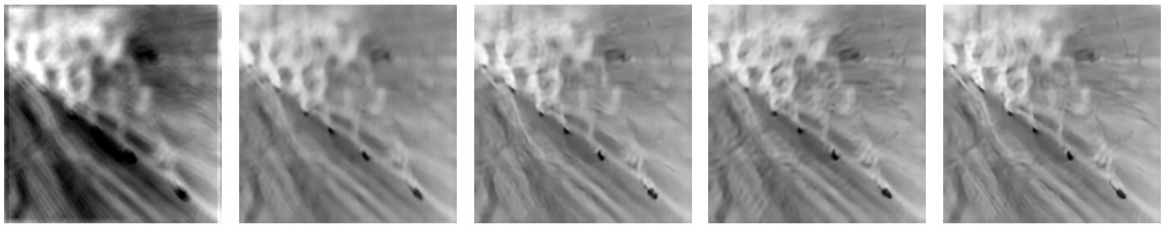
Cabe destacar que en los entrenamientos 2, 3 y 4 se produce cierto *overfitting* ya que las curvas de validación y entrenamiento divergen a partir de cierto *epoch*, aunque no emporan los resultados de validación. Como puede observarse, los valores obtenidos en el entrenamiento con el cuarto modelo son los mejores. La Tabla 1 muestra los resultados obtenidos del valor medio de las métricas MSE y SSIM al utilizar las redes neuronales entrenadas para las distintas estrategias sobre el conjunto de imágenes de prueba. De manera similar, la estrategia 4 presenta los mejores resultados.

| | **Entrenamiento 1** | **Entrenamiento 2** | **Entrenamiento 3** | **Entrenamiento 4** |
|---|---|---|---|---|
| **MSE** | 0.005 | 0.0047 | 0.0048 | 0.0045 |
| **SSIM** | 0.72 | 0.7501 | 0.7428 | 0.76 |

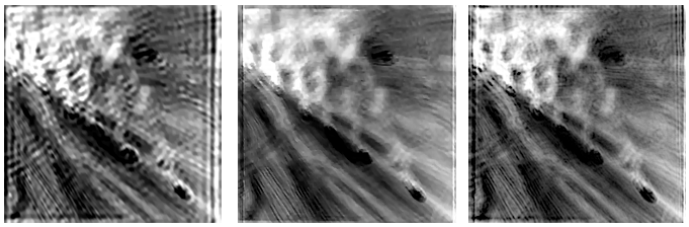
**Tabla 1**. Valores medios de las métricas MSE y SSIM obtenidos al aplicar las redes neuronales entrenadas al conjunto de imágenes de prueba.

La Figura 5 representa los resultados obtenidos para una imagen simulada con las redes neuronales entrenadas en base a las cuatro estrategias descritas y con tres algoritmos de deconvolución ciega (sin utilizar las medidas de la IMU) propuestos en la literatura [6-8]. Para realizar una comparativa de los resultados, se incluyen las métricas MSE, SSIM, PSNR (*Peak Signal-to-Noise Ratio*) y *Perceptual Blur* [9], el tiempo de ejecución necesario para realizar el proceso de desemborronado. Puede comprobarse, que la red entrenada considerando la estrategia 4 presenta mejores métricas de calidad, un tiempo de ejecución menor y una cierta mejora visual, mientras que los algoritmos de deconvolución ciega evaluados no permiten obtener resultados satisfactorios cuando el fondo es uniforme y la escena presenta pocos detalles.

Por último, se han realizado medidas experimentales preliminares utilizando la cámara, previamente calibrada, y la IMU de un móvil. La Figura 6 representa un ejemplo de resultado donde puede observarse la imagen emborronada adquirida por el movimiento de la cámara, la estimación de las PSFs en forma de rejilla y el resultado de desemborronar la imagen con la red neuronal entrenada siguiendo la estrategia 4. Aunque en estas pruebas se obtiene una mejora visual de la nitidez de la imagen, es necesario verificar los resultados con imágenes adquiridas mediante cámaras embarcadas en UAVs y determinar cuantitativamente la mejora que se obtiene para aplicaciones de teledetección.



| | Emborronada | Entrenamiento 1 | Entrenamiento 2 | Entrenamiento 3 | Entrenamiento 4 |
|---|---|---|---|---|---|
| MSE | 0.0018 | 0.0018 | 0.0010 | 0.0012 | 0.0010 |
| SSIM | 0.54 | 0.6288 | 0.656 | 0.65 | 0.67 |
| PSNR | 27.55 | 29.27 | 29.5 | 29.3 | 29.8 |
| *Blur* [9] | 0.67 | 0.64 | 0.64 | 0.55 | 0.53 |
| Tiempo de ejecución | - | 0.033 s | 0.195 s | 31 s | 0.05 s |



| | Algoritmo [6] | Algoritmo [7] | Algoritmo [8] |
|---|---|---|---|
| MSE | 0.12 | 0.12 | 0.14 |
| SSIM | 0.1 | 0.1 | 0.08 |
| PSNR | 9.18 | 9.64 | 9.4 |
| *Blur* [9] | 0.32 | 0.38 | 0.32 |
| Tiempo de ejecución | 20 s | 250 s | 134 s |

**Figura 5.** Resultados obtenidos al aplicar las redes neuronales entrenadas y tres algoritmos de deconvolución ciega sobre una imagen emborronada simulada con el modelo de vibración y movimiento de UAVs.

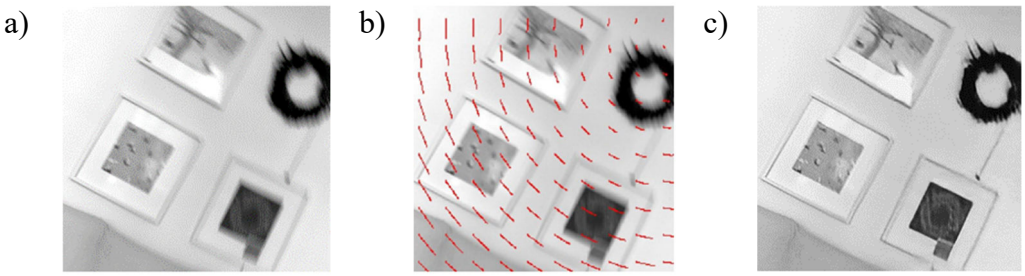



**Figura 6.** Resultados experimentales del procesado de imagen propuesto: a) Imagen emborronada adquirida con un móvil en movimiento, b) Estimaciones de las PSFs superpuestas a la imagen adquirida y c) Imagen resultante del proceso de desemborronado utilizando la red neuronal entrenada siguiendo la estrategia 4.

## 5. Conclusiones

El uso de plataformas aéreas no tripuladas se ha incrementado en los últimos años y se han ido introduciendo nuevos tipos cada vez más compactos, que en ocasiones conllevan una pérdida de estabilidad en condiciones atmosféricas adversas o debido a turbulencias. En el ámbito de la teledetección, es importante conseguir imágenes nítidas para mejorar la detección e identificación de los blancos minimizando el número de falsas alarmas. Por esta razón, el *blurring* ocasionado por el movimiento y vibración de la plataforma es una problemática importante que limita las capacidades operativas de los sensores electro-ópticos embarcados en UAVs.

El trabajo realizado en este artículo trata de reducir las degradaciones ocasionadas por la vibración y traslación de las plataformas aéreas durante el vuelo utilizando técnicas de *Deep Learning* a partir de las imágenes emborronadas y las estimaciones de las PSFs que se obtienen mediante las medidas inerciales de la IMU. Para ello se ha generado un *dataset* basado en un modelo que simula las vibraciones residuales de los UAVs en los tres ejes de rotación. Con estos datos se han realizado cuatro variaciones de entrenamiento sobre la red CNN planteada. El primero entrena la red considerando sólo las imágenes emborronadas sin información adicional; el segundo introduce adicionalmente una estimación de la PSF para cada píxel como una aproximación lineal a partir de sus puntos inicial y final durante el tiempo de exposición; el tercero introduce de nuevo las estimaciones de las PSFs aproximadas linealmente, ajustando sus puntos discretos a un segmente lineal; y el cuarto introduce la estimación de las PSFs en formato rejilla mediante una imagen en escala de grises. Una vez entrenadas las cuatro variantes, se concluye que la variante que consigue en media menores pérdidas y un mejor SSIM es Entrenamiento 4, que además requiere un menor tiempo de ejecución pudiendo ser aplicado en tiempo real. Por otra parte, también se compara el rendimiento de la red neuronal con algoritmos basados en deconvolución ciega, consiguiendo una mayor disminución del *blurring* con las técnicas planteadas y unos menores tiempos de ejecución.

Por último, se realizan pruebas experimentales preliminares con una cámara de un móvil en movimiento durante el tiempo de exposición, mostrando una mejora visual de la nitidez de las imágenes obtenidas tras aplicar la red neuronal del Entrenamiento 4. Aunque es necesario verificar los resultados obtenidos con imágenes adquiridas mediante cámaras embarcadas en UAVs y analizar cuantitativamente la mejora que se consigue y la robustez de la red neuronal entrenada, los resultados obtenidos ponen de manifiesto el gran potencial de las técnicas de *Deep Learning* para llevar a cabo el desemborronado de las imágenes adquiridas mediante cámaras embarcadas en plataformas aéreas, mejorando la calidad de las imágenes y las capacidades de teledetección.

**Referencias**


1. Mustaniemi J, Kannala J, Särkkä S, Matas J, Heikkila J. Gyroscope-aided motion deblurring with deep networks. En: *IEEE Winter Conf. on Applications of Computer Vision*. **2019**. p. 1914–22.
2. Flickr Image dataset [Internet]. [Accedido 03/08/2022]. Disponible en: https://www.kaggle.com/datasets/hsankesara/flickr-image-dataset
3. Cerrada Ramírez A, Blázquez García R, Burgos García M. Análisis y modelado de sistemas de estabilización de sensores electro-ópticos embarcados en vehículos aéreos no tripulados. En: *VIII Congreso Nacional de I+D en Defensa y Seguridad*. **2020**.
4. Chan SH. Constructing a sparse convolution matrix for shift varying image restoration problems. En: *2010 IEEE International Conference on Image Processing*. **2010**. p. 3601–3604.
5. Wang Z, Bovik AC, Sheikh HR, Simoncelli EP. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*. **2004**;13(4):600–12.
6. Krishnan D, Tay T, Fergus R. Blind deconvolution using a normalized sparsity measure. En: *CVPR 2011*. **2011**. p. 233–40.
7. Xu L, Zheng S, Jia J. Unnatural L0 Sparse Representation for Natural Image Deblurring. En: *2013 IEEE Conference on Computer Vision and Pattern Recognition*. **2013**. p. 1107–14.
8. Whyte O, Sivic J, Zisserman A, Ponce J. Non-uniform deblurring for shaken images. En: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. **2010**. p. 491–8.
9. Crete F, Dolmiere T, Ladret P, Nicolas M. The blur effect: perception and estimation with a new no-reference perceptual blur metric. En: *Human Vision and Electronic Imaging XII*. **2007**.