

**Sistema embebido de vigilancia que soporta tareas de super resolución sobre rostros humanos**

**AUTORES:**

**Andrés David Gómez Bautista**

**DIRECTOR:**

**Ing. Francisco Carlos Calderón Bocanegra**



**PONTIFICIA UNIVERSIDAD JAVERIANA  
FACULTAD DE INGENIERÍA  
DEPARTAMENTO DE ELECTRÓNICA  
BOGOTÁ D.C Julio 2022**

# Índice general

<b>Introducción</b>	<b>6</b>
<b>1. Planteamiento del problema</b>	<b>7</b>
1.1. Descripción del problema . . . . .	7
1.2. Justificación . . . . .	7
<b>2. Objetivos</b>	<b>8</b>
2.1. Objetivo General . . . . .	8
2.2. Objetivos específicos . . . . .	8
<b>3. Marco Teórico</b>	<b>9</b>
3.1. Conceptos basicos . . . . .	9
3.2. El problema de la resolución . . . . .	10
3.3. Métodos de interpolación clásicos . . . . .	11
3.4. Super resolución . . . . .	12
3.5. Procesamiento de imágenes en el área de la vigilancia. . . . .	13
<b>4. Desarrollo del modelo</b>	<b>14</b>
4.1. Creación del conjunto de datos . . . . .	14
4.1.1. Origen de las imágenes . . . . .	14
4.1.2. Decimación . . . . .	14
4.1.3. Selección de librerías de procesamiento de imágenes . . . . .	15
4.2. Selección del modelo . . . . .	16
4.2.1. Características y arquitectura del modelo . . . . .	16
4.3. Entrenamiento del modelo . . . . .	17
4.3.1. Hiperparámetros del modelo . . . . .	17
4.3.2. Resultados del entrenamiento . . . . .	18
4.4. Construcción de imagen . . . . .	20
<b>5. Desarrollo del sistema embebido</b>	<b>22</b>
5.1. Tarjeta de desarrollo . . . . .	22
5.1.1. Configuración de la cámara . . . . .	22
5.2. Implementación del modelo . . . . .	23
5.2.1. Estructura del sistema . . . . .	24
5.2.2. Comportamiento del sistema . . . . .	25
5.2.3. Operación del sistema . . . . .	28

<b>6. Análisis y resultados</b>	<b>29</b>
6.1. Métricas de comparación . . . . .	29
6.1.1. PSNR y SSIM . . . . .	29
6.1.2. FID . . . . .	30
6.2. Métricas de calidad . . . . .	31
6.2.1. Desenfoque . . . . .	31
6.2.2. SNR . . . . .	32
6.3. Verificación de identidad de rostros . . . . .	32
6.4. Comparación de imágenes . . . . .	33
<b>7. Conclusiones y Recomendaciones</b>	<b>38</b>
7.1. Conclusiones . . . . .	38
<b>8. Anexo: Código de implementación</b>	<b>39</b>
<b>Referencias</b>	<b>44</b>

# Índice de figuras

3.1. Imagen RGB y los canales rojo (a) , verde (b) y azul (c) [propia] . . . . .	10
3.2. Imagen RGB y los canales Y (a) , Cb (b) y Cr (c) [propia] . . . . .	11
4.1. Esquemático pruebas de las librerías [propia] . . . . .	15
4.2. Arquitectura red ESPCN [63] . . . . .	17
4.3. Gráfico de perdidas MSE del entrenamiento [propia] . . . . .	18
4.4. Gráfico de PSNR del entrenamiento [propia] . . . . .	19
4.5. Gráfico de SSIM del entrenamiento [propia] . . . . .	19
4.6. Proceso de construcción una imagen YCbCr por el modelo [propia] . . . . .	20
4.7. Proceso de construcción una imagen RGB por el modelo [propia] . . . . .	20
5.1. Tarjeta de desarrollo Tx2 [propia] . . . . .	23
5.2. Vistas superior (a) y frontal de la cámara (b) [propia] . . . . .	23
5.3. Diagrama de componentes del sistema [propia] . . . . .	24
5.4. Diagrama de flujo implementación del modelo [propia] . . . . .	26
5.5. Imágenes tomadas por la tarjeta con resolución dividida por 2 (a), 4 (b) y 8 (c) [propia] . . . . .	27
5.6. Imágenes RGB generadas por el modelo con factor 2 (a), 4 (b) y 8 (c) [propia] . . . . .	27
5.7. Imágenes Y generadas por el modelo con factor 2 (a), 4 (b) y 8 (c) [propia] . . . . .	27
6.1. Resultados PSNR factor 2 (a), 4 (b) y 8 (c) [propia] . . . . .	29
6.2. Resultados SSIM factor 2 (a), 4 (b) y 8 (c) [propia] . . . . .	30
6.3. Resultados FID factor 2 (a), 4 (b) y 8 (c) [propia] . . . . .	31
6.4. Resultado desenfoque factor 2 (a), 4 (b) y 8 (c) [propia] . . . . .	31
6.5. Resultados SNR factor 2 (a), 4 (b) y 8 (c) [propia] . . . . .	32
6.6. Resultados de la distancia entre rostros por factor 2 (a), 4 (b) y 8 (c) [propia] . . . . .	33
6.7. Comparación de imágenes por cada método factor 2 [propia] . . . . .	34
6.8. Comparación sobre zonas específicas por cada método con factor 2 [propia] . . . . .	34
6.9. Comparación de imágenes por cada método factor 4 [propia] . . . . .	35
6.10. Comparación sobre zonas específicas por cada método con factor 4 [propia] . . . . .	35
6.11. Comparación de imágenes por cada método factor 8 [propia] . . . . .	36
6.12. Comparación sobre zonas específicas por cada método con factor 8 [propia] . . . . .	36

# Índice de tablas

2.1. Tabla de métricas de evaluación del modelo . . . . .	8
4.1. Resultados pruebas sobre librerías. . . . .	16
4.2. Valores de hiperparámetros utilizados para el entrenamiento del modelo. . .	18
5.1. Características de hardware Jetson TX2. . . . .	22
5.2. Tiempos medios en segundos de procesamiento de imagen por el modelo . . .	27
6.1. Resumen resultados métricas factor 2 . . . . .	37
6.2. Resumen resultados métricas factor 4 . . . . .	37
6.3. Resumen resultados métricas factor 8 . . . . .	37

El presente docuemnto es una versión comprimida. El archivo pdf original se puede acceder a través del link: [https://livejaverianaedu-my.sharepoint.com/:b:/g/personal/gomezan\\_javeriana\\_edu\\_co/EZcedQbJb5NAkTzGu9nUVskB5\\_TbG6IWttp8rhtwbKyBuw?e=v3treq](https://livejaverianaedu-my.sharepoint.com/:b:/g/personal/gomezan_javeriana_edu_co/EZcedQbJb5NAkTzGu9nUVskB5_TbG6IWttp8rhtwbKyBuw?e=v3treq)

# Introducción

La detección e identificación de rostros sobre videos o imágenes se ha convertido en un desafío en las áreas de vigilancia y seguridad. Por lo mismo, existe un creciente interés en el incremento de la resolución de dichas imágenes ya esto facilita el posterior tratamiento de las mismas. Este proceso no está exento de dificultades, por lo que existen diferentes métodos y tecnologías orientados a resolver este problema.

El presente documento propone desarrollar un sistema embebido programado con una red neuronal entrenada específicamente para incrementar la resolución espacial de imágenes de rostros de personas capturadas por medio de una cámara. Este dispositivo pretende ser una opción sencilla y económica para tratar las imágenes en tiempo real.

El documento se encuentra dividido en 7 capítulos. A través de los capítulos 1 y 2 pretende abordar el planteamiento del problema y los objetivos del proyecto. El capítulo 3 explica marco teórico que explora los diferentes métodos disponibles en el estado de arte relacionados en super resolución y como esta es abordada en el proyecto. El capítulo 4 explica las actividades de selección, entrenamiento y desarrollo del modelo mientras el capítulo 5 describe la configuración del *hardware* que soporta el modelo. El capítulo 6 documenta el análisis de los resultados obtenidos por la red y finalmente las conclusiones globales del proyecto se encuentran en el capítulo 7.

# Capítulo 1

## Planteamiento del problema

### 1.1. Descripción del problema

A través del desarrollo tecnológico actual, los centros urbanos se encuentran sufriendo un periodo de transformación, donde se desarrollan nuevas formas de automatizar las distintas dimensiones en las que se descompone la urbe. Entre todas estas dimensiones, el campo de la vigilancia y seguridad se convierte en un tema de especial interés, sobre todo la administración de las cámaras de seguridad y el contenido que estas generan [6, 7, 8, 9]. Aunque la existencia de sistemas masivos de monitoreo y grabación compuestos de miles de nodos pueden parecer una obra de ingeniería impresionante, de poco sirven estos, si no son capaces de identificar los protagonistas de los sucesos al otro extremo de la cámara.

Para identificar a un individuo a través del material obtenido de una cámara de seguridad, dicho material debe contener la resolución suficiente para reconocer las características faciales de este individuo. Esto es una tarea bastante complicada, las imágenes obtenidas por las cámaras son distorsionadas a través del efecto de ruido y desenfoque producto del funcionamiento cotidiano de la cámara [10]; además, se debe enfrentar otras fuentes de disturbios provenientes del ambiente, cambios de iluminación y contraste [11]. Todo esto puede alterar drásticamente las características de un rostro, dificultando enormemente su reconocimiento.

### 1.2. Justificación

La construcción de un prototipo capaz de aumentar la resolución de las imágenes de rostros tiene implicaciones importantes en campos tales como: investigación forense, seguridad pública, procesos judiciales, etc. El desarrollo del sistema puede tener gran cantidad de beneficios para los miembros de una comunidad y sus órganos de seguridad.

Dentro del campo de la vigilancia, el sistema representa una herramienta con características muy valiosas: A pesar del efecto del ruido, desenfoque y diversas perturbaciones, puede hacer estimaciones cercanas a la realidad para obtener imágenes de mayor resolución. El modelo de super resolución se encuentra soportado por una red neuronal, esta técnica es más sencilla y económica a soluciones sobre *hardware* que intentan mejorar la resolución propia de la cámara cuando se encuentra capturando imágenes. La red debe ser capaz de procesar la imagen rápidamente en tiempo real, de forma que el aumento de la resolución de las imágenes es transparente para el usuario final. Al final, se obtienen las ventajas de una cámara de alta resolución sobre las comodidades de *hardware* genérico.



# Capítulo 2

## Objetivos

### 2.1. Objetivo General

Diseñar un sistema embebido que por medio de una red neuronal interpole imágenes de rostros de personas en comparación con los métodos clásicos de interpolación.

### 2.2. Objetivos específicos

- Recopilar el conjunto de imágenes de rostros destinado a realizar procesos de entrenamiento, validación y prueba; de por lo menos 5000 muestras.
- Definir y entrenar la topología de red neuronal que permita solucionar un problema de interpolación de imágenes a color.
- Implementar la red neuronal previamente entrenada en un dispositivo embebido de alto nivel.
- Evaluar el desempeño del sistema con respecto a interpoladores clásicos mediante los métodos dispuestos en la tabla 2.1.

Métrica	Librería/Implementación
PSNR	Scikit [1]
SSIM	Scikit [1]
FID	Keras[2] , MLM [3]
Desenfoque	Scikit [1]
SNR	SciPy [4]
Distancia de rostro	Ageitgey [5]

Tabla 2.1: Tabla de métricas de evaluación del modelo

# Capítulo 3

## Marco Teórico

### 3.1. Conceptos basicos

De acuerdo con el estándar 610.4 una imagen se define como una representación en dos dimensiones de una escena [12]. Desglosando su estructura, una imagen es un arreglo bidimensional compuesto por píxeles. El mismo estándar define un píxel como el elemento más pequeño al cual puede ser asignado a una escala de grises. En la práctica un píxel representa un 1 Byte de información y la escala de grises es la representación más sencilla que puede tener una imagen. En ella cada píxel tiene un valor entre 0, el negro absoluto y 255, el blanco absoluto. Los valores entre estos dos extremos son tonalidades intermedias. La debilidad más clara de esta representación es su incapacidad de representar colores, ya que este solo representa la luminosidad, es decir, la cantidad de luz reflejada por la imagen, el blanco refleja la totalidad de la luz mientras el negro la absorbe toda.

La representación de colores en las imágenes se basan en los modelos de color [13] . Estos son representaciones que buscan recrear parte de los colores dentro del espectro de color visible. Existen varios modelos de color populares con diferentes características. Entre ellos se encuentra RGB y YCbCr.

El modelo RGB [14] consta de tres canales de información. Cada canal corresponde a cada uno de los tres colores más sensibles por los conos fotorreceptores del ojo humano: el rojo, el verde y el azul. Este modelo se basa en la síntesis aditiva para representar colores. Los colores rojo, verde y azul se convierten en los colores primarios. El color de un píxel, es entonces, la suma de los valores en cada uno de los canales de información de dicho píxel, que es en esencia, la suma de las longitudes de onda de los colores primarios. El valor de los canales de información de las imágenes RGB se encuentran entre 1 y 255. De forma que un píxel contiene 3 Bytes en total. La figura 3.1 contiene una imagen RGB con sus respectivos canales de color.

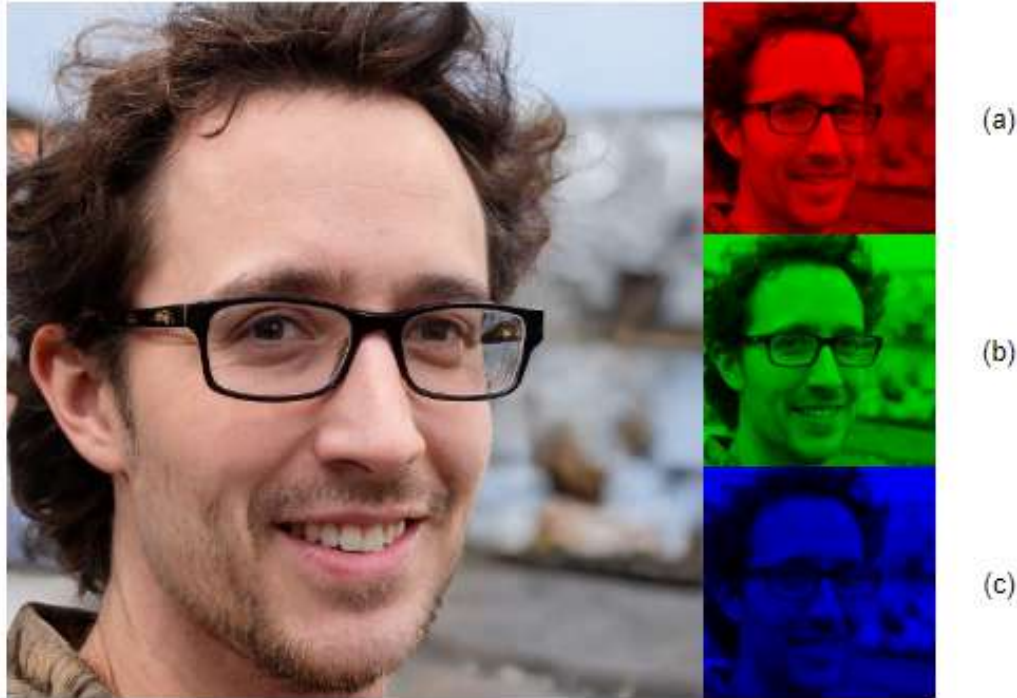


Figura 3.1: Imagen RGB y los canales rojo (a) , verde (b) y azul (c) [propia]

Por otra parte el modelo YCbCr [15] consiste en separar la información de luminancia de la de color. La base este radica en el hecho que el ojo humano es mucho más sensible a la luminosidad que a el color. Este modelo se divide entonces en tres canales de información: En primer lugar se encuentra el canal de Luminancia Y, que almacena toda la información de la luminosidad; mientras la información del color es almacenada en dos componentes de color, crominancia azul (Cb por sus siglas en inglés) y crominancia roja (Cr por sus siglas en inglés). El color es entonces representado como la diferencia de las componentes Cb y Cr con respecto al valor de referencia de azul y rojo, respectivamente. El valor de la componente de luminancia Y se encuentra en un rango nominal de 16 - 235; mientras Cb y Cr están definidas en un rango nominal de 16 - 240. La figura 3.2 contiene una imagen YCbCr con sus respectivos canales de información.

### 3.2. El problema de la resolución

La alta resolución frecuentemente es un atributo deseable sobre imágenes dado a su capacidad para almacenar más información y por ende conservar figuras con mayor detalle, útil en múltiples campos y aplicaciones. Existen varias formas de incrementar la resolución espacial, disminuir el tamaño de los píxeles es la forma más obvia; sin embargo, existen diversos fenómenos físicos que se vuelven importantes a medida que el tamaño de los píxeles disminuye, por ejemplo, existen limitaciones respecto al tamaño mínimo de píxeles dadas las restricciones de la tecnología CMOS que actualmente domina el campo de la captura digital de imágenes. También vale la pena destacar el ruido de disparo, consecuencia de la poca intensidad lumínica que se obtiene a pequeñas escalas, este logra afectar la imagen de forma severa degradando la calidad global de esta. Otra posible opción es aumentar el número de transistores; no obstante, el aumento del tamaño del chip lleva como consecuencia incremento de la capacitancia y por ende tiempos de retraso mucho mayores [16]. En resumen, obtener



Figura 3.2: Imagen RGB y los canales Y (a) , Cb (b) y Cr (c) [propia]

imágenes de alta resolución mediante técnicas de hardware puede ser un proceso difícil y costoso, dados los fenómenos físicos que se oponen a su desarrollo.

### 3.3. Métodos de interpolación clásicos

Una alternativa es, fuera de enfocarse en la resolución de la adquisición de la imagen, concentrarse en aumentar la resolución de la imagen, después de que esta es tomada. La forma más elemental de conseguir esto es a través de la interpolación [17], esta es una disciplina dentro del campo del análisis numérico que se enfoca en el cálculo de información desconocida a partir de información conocida. Dentro del procesamiento de imágenes, estas técnicas son de interés dado su bajo consumo de recursos de cómputo y baja complejidad algorítmica.

Dentro del campo de los algoritmos de interpolación de imágenes, la interpolación bilineal es un método básico dentro del procesamiento de imágenes. Este se usa como base la interpolación lineal, es decir, se asume que el comportamiento entre dos muestras conocidas se puede describir a través de una función lineal. El método bilineal lleva la interpolación lineal al mundo del procesamiento de imágenes, realizando la interpolación en dos direcciones distintas, donde ambas direcciones son ortogonales entre sí. Esta técnica, pese a ser una alternativa muy elemental, aún es ampliamente utilizada. Los trabajos [18, 19] son propuestas con variantes de la interpolación bilineal más desarrolladas.

Otra técnica bastante estudiada es la interpolación bicúbica, esta técnica es más compleja que la interpolación bilineal. Esta consiste en tomar la imagen como una malla regular de dos dimensiones [17], se toma como referencia cuatro píxeles en las cuatro esquinas del píxel desconocido a interpolar. Las contribuciones de cada uno de estos cuatro píxeles sobre el valor del píxel a desarrollar están definidas por una serie de dieciséis constantes. Estas constantes se calculan a partir de una función específica, su respectiva derivada y las derivadas parciales

en la dirección vertical y horizontal. Este método es conocido por ser usado en diferentes programas de tratamiento digital de imágenes como Adobe Photoshop [20].

La interpolación del vecino más cercano es una técnica más intuitiva a las previamente presentadas. Esta consiste en asignar el valor del píxel conocido más cercano al píxel desconocido que se desea interpolar. El problema se convierte en un problema de minimización, donde se debe encontrar el píxel conocido con la menor distancia a un punto específico [17]. Existen diferentes estrategias para encontrar la menor distancia entre los píxeles desconocidos y el píxel de interés, en resumidas cuentas, se buscan procedimientos con baja complejidad algorítmica para disminuir el costo sobre los recursos de procesamiento.

Finalmente, el último método de interpolación del cual se va a hablar es la interpolación de Lanczos, esta técnica está basada en “splines”, los cuales son polinomios definidos a trozos. En síntesis, el algoritmo es bastante similar a la interpolación polinómica; sin embargo, los “splines” son más eficientes que los polinomios para adoptar el comportamiento de las imágenes, dado que dicho comportamiento es bastante distante del que los polinomios pueden ofrecer [17]. El algoritmo de Lanczos es ampliamente utilizado en áreas diferentes al procesamiento de imágenes, por ejemplo; [21, 22] usan el algoritmo para el análisis espectral de señales; [23, 24] lo emplean para disminuir la carga de procesamiento computacional en cálculos sobre espacios vectoriales.

### 3.4. Super resolución

En términos generales, puede ser difícil conseguir la calidad que requieren ciertos campos o aplicaciones respecto a la calidad de las imágenes obtenidas a través de métodos clásicos de interpolación. Los bordes de estas imágenes tienen un alto grado de desenfoque; así mismo, existe una pérdida de información sobre las componentes en alta frecuencia que estos métodos suelen menospreciar; por último, existe cierta dificultad para procesar adecuadamente estructuras complejas sobre las imágenes [25, 26, 27]. En estos escenarios, la super resolución se impone como una interesante alternativa de investigación. Esta se denomina como un conjunto de técnicas que buscan producir imágenes en alta resolución a partir de la información de una o varias imágenes [28]. Es decir, es posible agregar información sobre los detalles de la imagen a partir de la información contenida a priori de otras imágenes diferentes.

Los métodos de super resolución se pueden dividir en métodos basados en reconstrucción y métodos basados en aprendizaje:

Los métodos basados en reconstrucción se basan en el diseño de premisas que se encargan de estimar los detalles perdidos en la captura de la imagen [28, 29]. Dentro de esta categoría se encuentran los métodos basados en la frecuencia, estos se encargan de mejorar la calidad de la imagen eliminando el solapamiento en el dominio de la frecuencia; y los métodos del dominio del espacio, estos usan conocimiento sobre las características de la imagen para modelar el desplazamiento, ruido y desenfoque presentes para posteriormente contrarrestarlas en la imagen en alta resolución. Los métodos basados en la reconstrucción suelen tener ganancias limitadas de resolución, dado que es difícil modelar las relaciones entre las condiciones de la imagen y su resultado desconocido a priori [30].

Los métodos basados en el aprendizaje se han convertido en un campo activo de investigación debido a los resultados obtenidos para recuperar componentes de alta frecuencia sin incrementar el número de imágenes de entrada [28, 29]. Estos métodos consisten en la creación y entrenamiento de modelos que embeben el análisis estadístico de grandes lotes de información, de forma que, el modelo contiene todo el conocimiento a priori necesario para

realizar el proceso de transformación sobre las imágenes de baja resolución. Actualmente, existen bastantes trabajos de sistemas de vigilancia soportados por super resolución, tantos métodos basados en reconstrucción [31, 32], como métodos basados en el aprendizaje [33, 34, 35].

### **3.5. Procesamiento de imágenes en el área de la vigilancia.**

La aplicación de métodos de procesamiento de imágenes se convierte en una necesidad a medida que las ciudades crecen y se tecnifican. Los gobiernos locales buscan prevenir crímenes en orden de mantener seguros a sus ciudadanos [9]. Específicamente existe bastante interés en aplicaciones de vigilancia sobre rostros humanos. Diferentes trabajos, desde múltiples enfoques, buscan realizar operaciones de procesamiento de imágenes sobre las facciones del rostro humano: los documentos [36, 37], se encargan de realizar la detección de rostros, esta tarea consiste en identificar la cara de una persona en una imagen cualquiera; [38, 39] son trabajos concentrados en el reconocimiento facial, es decir, se encargan de asignar la identidad de una persona dada la imagen de un rostro; [40] se concentra en la captura de rostros, donde, a partir de una métrica específica de evaluación, se debe seleccionar la mejor fotografía de un conjunto de fotografías; finalmente, [41, 42], dada una imagen parcial o distorsionada de un rostro, se encargan de generar las partes faltantes de la imagen dado un modelo computacional definido; este último campo es bastante similar a la super resolución. Varios documentos tales como [43, 44] se concentran en específicamente en el aumento de la resolución de las imágenes de los rostros.

Actualmente, los sistemas de super resolución de rostros humanos tienden a basarse en métodos de aprendizaje y más específicamente en aprendizaje profundo [36, 37, 38, 39]. Las redes neuronales de aprendizaje profundo tienden a tener buenos resultados pese a ser bastante complejas, ya que son muy grandes, por lo que suelen tener bastante latencia [45].

Estas condiciones restringen el potencial en la tecnología en aplicaciones de vigilancia, sobre todo aquellas que se desarrollan en tiempo real. La capacidad de un modelo de operar en tiempo real tiene varios beneficios, entre los que se encuentra, tiempos de respuesta cortos que puede necesitar un equipo humano para reaccionar, es por ello que varios trabajos se enfocan en el desarrollo de redes neuronales livianas, capaces de realizar operaciones más rápidamente y de trabajar sobre piezas de hardware menos complejo [33, 45, 46, 47].

# Capítulo 4

## Desarrollo del modelo

### 4.1. Creación del conjunto de datos

El conjunto de datos es la información con la que se planea entrenar el modelo. En este proyecto, en concreto, consiste en dos grupos de imágenes: El *ground truth*, que corresponde al eje de referencia de resolución con el que se planea medir el desempeño del sistema y un segundo grupo de imágenes que corresponden a las mismas imágenes pertenecientes al *ground truth* pero disminuidas en resolución por un factor de escalado constante. Estas últimas imágenes corresponden a la entrada del modelo, es decir las imágenes a las que se desea incrementar la resolución mientras que el *ground truth* es la imagen ideal a la que el modelo desea alcanzar.

#### 4.1.1. Origen de las imágenes

Las imágenes fueron obtenidas del sitio web [48], estas son generadas por una inteligencia artificial, por lo que no corresponden a personas reales; sin embargo, tienen un acabado muy realista que no se espera que comprometa el desarrollo de la tesis, ni sus resultados.

El *ground truth* se construye a partir de 22000 imágenes, de resolución 1024x1024, las cuales se dividen en dos conjuntos: el conjunto de entrenamiento, que corresponde al 75 % del total y el conjunto de validación que corresponde al 25 % del total. Para ello, se escribe un programa encargado de descargar, ordenar y almacenar las imágenes del sitio. En esta etapa se realiza un pre-procesamiento a la información eliminando imágenes repetidas y cambiando el formato de las imágenes descargadas. Por sugerencia del director de la tesis de grado, las imágenes se manipulan en el formato PNG [49], dado que es un formato que permite conservar la calidad de la imagen, se encuentra ampliamente extendido y es de uso libre.

#### 4.1.2. Decimación

Una vez definido el *ground truth* se procede a crear su homólogo con menor resolución, es decir, con una menor cantidad de píxeles. Para ello se realiza el proceso de decimación sobre el *ground truth*. Este consta del filtrado de las componentes de alta frecuencia de la imagen y el posterior sub-muestreo de la misma. La importancia de este procedimiento radica en dos factores:

- Evitar pérdidas de información: Esto ocurre dado que una imagen es en sí una señal muestreada en dos dimensiones. Cuando se restan muestras de la señal consecutiva

del sub-muestreo de la imagen, se presenta el fenómeno de *aliasing* sobre la imagen resultante. En resumidas cuentas se busca evitar que las componentes de alta y baja frecuencia se cancelen entre sí debido a la disminución de la frecuencia de muestreo.

- Otra consecuencia del fenómeno de *aliasing* es que las componentes de alta frecuencia se van a encontrar embebidas dentro de las componentes de baja frecuencia. Es decir, la información que se desea predecir por la red, se va a encontrar contenida dentro de la misma imagen con la que se entrena. Como consecuencia, el entrenamiento de la red se vuelve deficiente dado que esta no predeciría componentes de alta frecuencia a partir de las componentes de baja frecuencia, sino que se limitaría a extraer las componentes de información de alta frecuencia dentro de las componentes de baja frecuencia.

Para tener una mejor estimación del desempeño del sistema se construyen 3 grupos de imágenes sub-muestreadas por los factores 2, 4 y 8. Estos grupos de imágenes cuenta con exactamente la misma estructura entre el conjunto de entrenamiento (75 % del conjunto inicial) y validación (25 % del conjunto inicial). La función de estos conjuntos es entrenar modelos específicos para entender el desempeño de la red con diferentes factores de sub-muestreo.

#### 4.1.3. Selección de librerías de procesamiento de imágenes

Dado que existe un abanico de librerías de tratamiento de imágenes y en orden de encontrar la mejor opción para el proyecto, se propone un plan de pruebas para evaluar el desempeño de las 3 librerías más importantes del mercado: TensorFlow [50], Pillow [51] y OpenCV [52].

La prueba consiste en tomar las imágenes del conjunto de datos de evaluación del *ground truth* y decimarlas por un factor de 2. El resultado debe interpolarse utilizando el método de vecinos más cercanos, NN por sus siglas en inglés, de tal forma que las imágenes resultantes tienen el mismo tamaño del *ground truth*. Posteriormente se comparan ambos conjuntos de imágenes utilizando el PSNR y el SSIM. Este procedimiento se realiza por cada una de las librerías. Durante las pruebas solo es lícito usar las funciones de dicha librería para las operaciones de carga, sub-muestreo e interpolación de las imágenes. La figura 4.1 contiene el esquemático de las pruebas sobre las librerías.

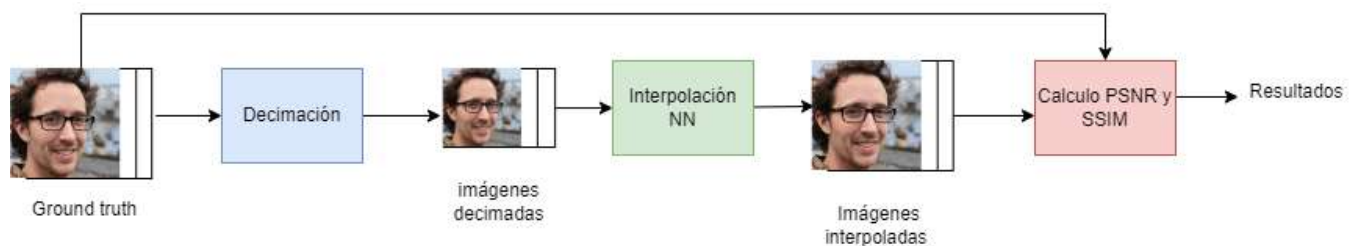


Figura 4.1: Esquemático pruebas de las librerías [propia]

La tabla 4.1 contiene los resultados arrojados por las pruebas. Usando como referencia el SSIM se puede observar que los resultados son idénticos entre las librerías. Empleando el PSNR como referencia, la cual es una métrica más sensible al SSIM, se puede confirmar que los resultados son similares, sobre todo los valores mínimos alcanzados por todas las librerías. El valor máximo de la librería OpenCV se destaca sobre las otras dos. Parece que esta librería logra alcanzar valores más altos, lo que le permite tener un mayor promedio y



mediana. Aunque los resultados no son abruptamente diferentes, la librería escogida para la realización del proyecto es OpenCV, no solo por ser la que mejor se desempeñó durante las pruebas, sino por la facilidad de implementación durante el desarrollo de las mismas.

	Tensorflow		Pillow		OpenCV	
	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR
Promedio	0.95	37.13	0.95	37.09	0.95	37.18
Des. Estándar	0.01	1.84	0.01	1.83	0.01	1.86
Mediana	0.95	37.30	0.95	37.26	0.95	37.34
Máximo	0.98	42.99	0.98	42.86	0.98	43.15
Mínimo	0.90	27.5	0.90	27.50	0.90	27.51

Tabla 4.1: Resultados pruebas sobre librerías.

## 4.2. Selección del modelo

Nunca fue la ambición de este proyecto diseñar una red neuronal. Esto sobrepasa los objetivos planteados dada la complejidad y el tiempo necesario para ello. Para mayor información sobre los objetivos específicos, revisar la sección 2.2. Se pretende elegir una red neuronal disponible del estado del arte en super resolución y afines que estén pensadas para aplicaciones en tiempo real o diseñadas para trabajar con tiempos de latencia bajos y además deben ser ligeras para poder ser implementada sobre *hardware* genérico.

A partir de la investigación del marco teórico se presumen como modelos candidatos iniciales [29, 33, 45, 46, 47, 53, 54]. De estas redes se seleccionan aquellas que cuenten con una implementación base, como anexo o adjunto, con la cual se puede trabajar.

Para facilitar la búsqueda y selección de la red los requerimientos del modelo son priorizados de la siguiente manera: El requerimiento más importante es el desempeño de la red en el área de super resolución, el segundo es la velocidad con la cual esta trabaja y finalmente la de menor importancia es que sean *hardware friendly*, es decir, su capacidad de ser desplegada sobre *hardware* genérico.

Investigando trabajos relacionados y teniendo en cuenta los candidatos iniciales, las implementaciones encontradas son divididas en: *hardware friendly*, [55, 56, 57]; redes de una sola imagen, [58, 59] y redes de video [60, 61, 62]. Las redes de una sola imagen son preferidas sobre las redes de video, puesto que estas últimas requieren de que las imágenes a su entrada hagan parte de una secuencia mientras que para las primeras cada imagen es independiente una de otra. Con esto en mente y con base en la jerarquía de requerimientos previamente planteada, se seleccionan 3 redes: DCSCN [58]; ESPCN [59] y LESRCNN [56]

Se procede a revisar la documentación asociada al trabajo realizado sobre cada red y el código de la implementación relacionada; Sin embargo, el factor que determinó la red escogida fue la calidad del código encontrado. La red ESPCN se destaca por tener código más limpio, legible y más fácil de entender con respecto a las 3 redes seleccionadas, por lo que es la escogida para ser implementada.

### 4.2.1. Características y arquitectura del modelo

La red ESPCN, siglas en ingles de *Efficient Sub-Pixel Convolutional Neural Network*, esta basada en el artículo [63] publicado en el 2016. Esta red está concebida para ser muy rápida

usando como base *hardware* con características limitadas. El texto original clama que esta es la primer red neuronal convolucional capaz de realizar operaciones de super resolución sobre videos de 1080p empleando una sola K2 GPU [63]. Esta red soporta aplicaciones de tiempo real de una única imagen o sobre videos.

La arquitectura de la red se puede encontrar en la figura 4.2. Esta se encuentra pensada para ser muy rápida y tener un bajo costo computacional. Su arquitectura puede descomponerse en una serie de capas convolucionales y una capa final de convolución a nivel de sub-píxel. Las responsabilidades de estas son:

- Capas convolucionales: Son las capas iniciales, funcionan como filtros, encargadas de extraer características de la imagen de entrada.
- Capa de convolución de sub-píxel: Esta capa es la capa final de la red. Esta se encarga de incrementar la resolución de la imagen a partir de las características extraídas por las capas convolucionales. En esencia esta es una serie de filtros que incrementan progresivamente la resolución hasta obtener la imagen de super resolución a la salida.

El que esta sea la capa final disminuye el costo computacional del procedimiento en comparación de que el incremento de la resolución se realizara en capas anteriores dentro de la red.

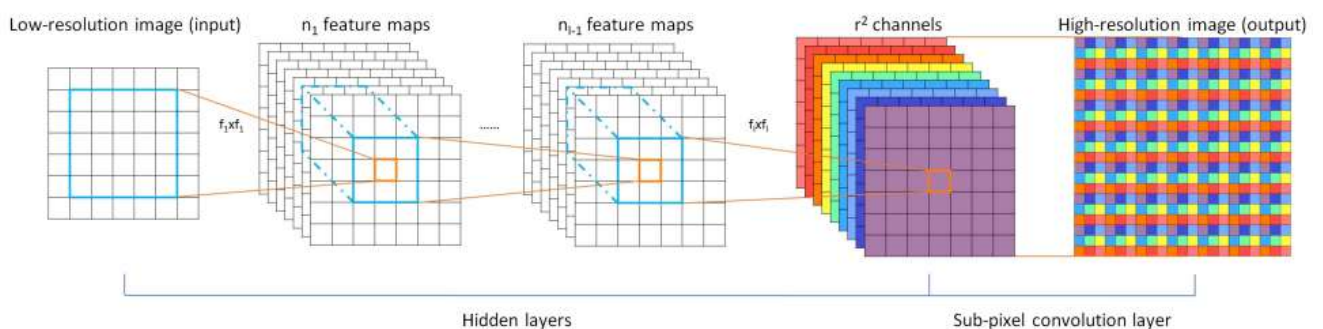


Figura 4.2: Arquitectura red ESPCN [63]

## 4.3. Entrenamiento del modelo

### 4.3.1. Hiperparámetros del modelo

Tanto el entrenamiento como el modelo están basados en la librería de aprendizaje autónomo de Pytorch [65]. Se emplea el optimizador de Adam [64] bajo una estrategia de mini-lotes como base entrenamiento. La función de costo es el error cuadrático medio (MSE por sus siglas en inglés) y la tasa de aprendizaje manejada es de  $1e-3$ . Con respecto a la implementación original [59] se incrementó el tamaño de lote sustancialmente a 1000 con el fin de sacar mayor provecho de los recursos de hardware disponible. El entrenamiento se realiza en un par de tarjetas gráfica nvidia GeForce GTX 1080 Ti. El valor de otros hiperparámetros se asignó como el defecto manejado por la librería. El resumen de los hiperparámetros se encuentra en la tabla 4.2.

Hiperparámetro	Valor
Tasa de aprendizaje	1e-3
Tamaño del lote	1000
Epsilon	1e-8
Beta 1	0.9
Beta 2	0.999

Tabla 4.2: Valores de hiperparámetros utilizados para el entrenamiento del modelo.

### 4.3.2. Resultados del entrenamiento

Para determinar el estado del entrenamiento se emplean una serie de métricas. La principal es la función de pérdidas, la tabla 4.3 presenta dicha función para el entrenamiento del modelo por factor 4.

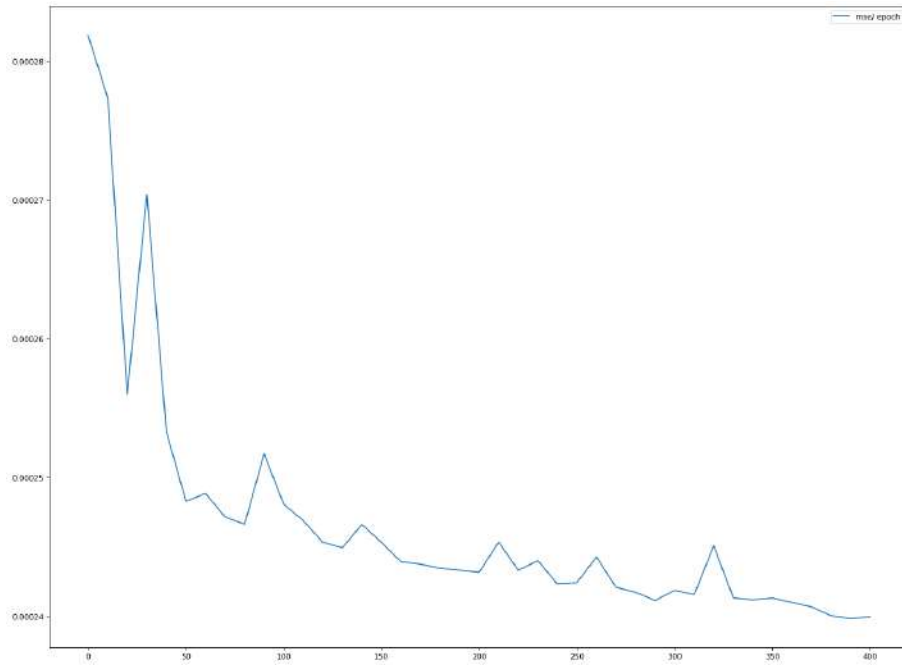


Figura 4.3: Gráfico de pérdidas MSE del entrenamiento [propia]

Como métricas complementarias se emplea el SSIM y el PSNR. Estas métricas permiten comparar las imágenes a la salida del modelo con el *ground truth*, valores altos de ambas métricas están relacionadas con imágenes de mejor calidad. Las figuras 4.4 y 4.5 representan los valores de estas métricas por cada época de entrenamiento. Las mediciones se realizan sobre el conjunto de validación para analizar la capacidad de generalización de la red y evitar fenómenos de sub-entrenamiento o sobre-entrenamiento.

Se puede observar como en las tres figuras mencionadas anteriormente, se llega al equilibrio en las 370 y 400 épocas. Durante esta región de entrenamiento, el modelo alcanza su desempeño máximo y es la zona ideal para seleccionar los pesos de la ESPCN dado que como se emplea el conjunto de validación para medir el PSNR y SSIM se puede concluir que el modelo conserva un nivel de generalización amplio.

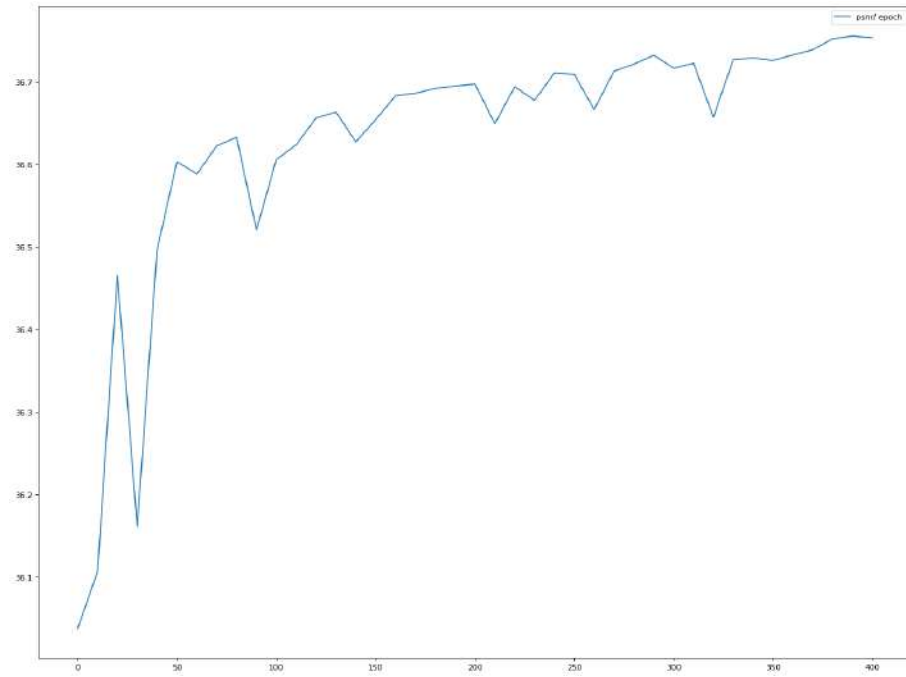


Figura 4.4: Gráfico de PSNR del entrenamiento [propia]

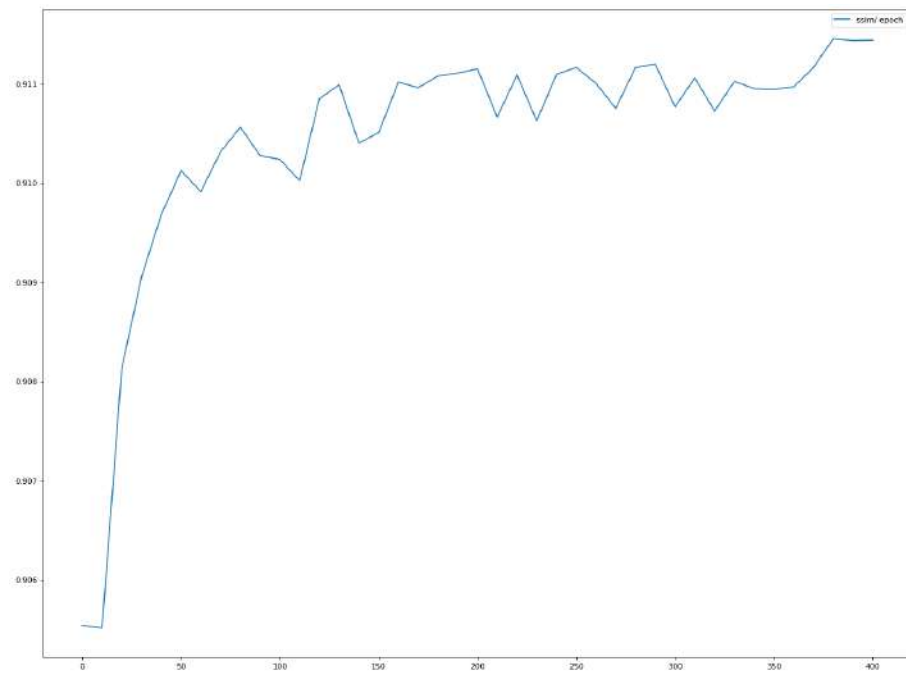


Figura 4.5: Gráfico de SSIM del entrenamiento [propia]

## 4.4. Construcción de imagen

Un aspecto importante a resaltar sobre la implementación de la ESPCN [59] es que este utiliza el modelo de color YCbCr para el procesamiento de la imagen. De forma más concreta, utiliza únicamente el canal Y para el entrenamiento y predicción de imagen en super resolución. La razón de ello es que el ojo humano es más sensible sobre la luminancia de una imagen que la información de color; además, manejar un solo canal le permite tener tiempos de latencia menores a comparación de utilizar 3. Dado que el modelo maneja únicamente el canal Y para construir la imagen de super resolución, son necesarios canales Cb y Cr con la misma resolución que el canal Y a la salida del modelo. La implementación [59] resuelve este problema interpolando la imagen de menor resolución y tomando las componentes Cb y Cr de este. Finalmente, se unen las componentes de color al canal Y para la construcción de la imagen final.

Como propuesta de mejora a la implementación original, se propone utilizar como método de interpolación a Lanczos ya que este es un método bastante robusto que suele tener muy buenos resultados. La imagen 4.6 ilustra el proceso de construcción de la imagen de super resolución utilizando el modelo YCbCr.

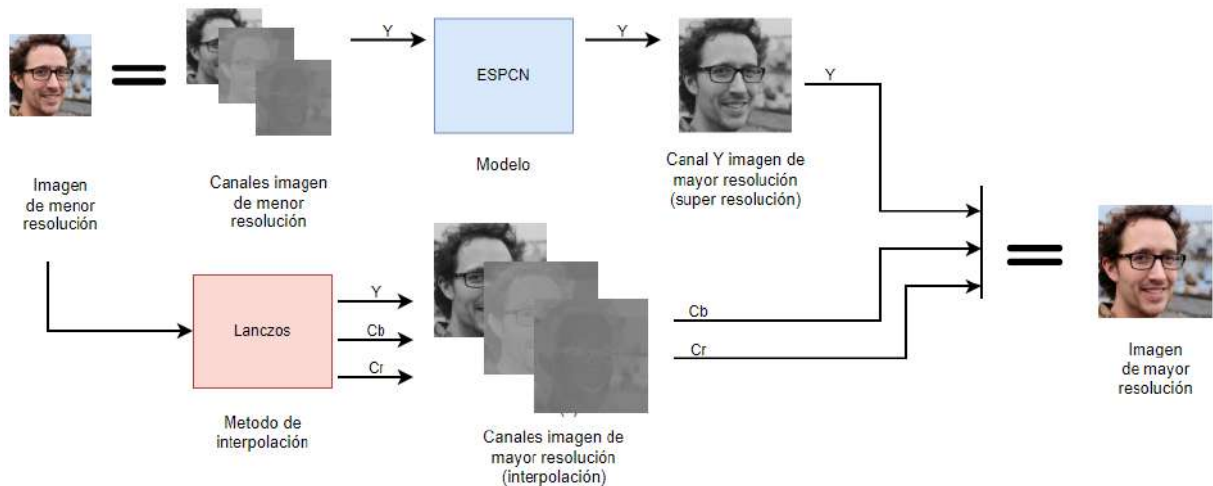


Figura 4.6: Proceso de construcción una imagen YCbCr por el modelo [propia]

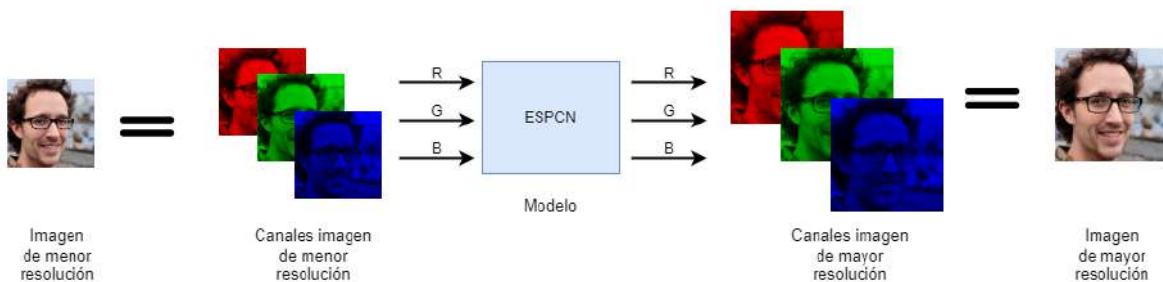


Figura 4.7: Proceso de construcción una imagen RGB por el modelo [propia]

Utilizar únicamente el canal Y de información tiene ciertas desventajas. Entre ellas que no se puede aplicar la super resolución a toda la imagen, comprometiendo los resultados de esta, y ya que es difícil conocer que proporción de los resultados son producto de la

interpolación o de la ESPCN, el modelo pierde cierto grado de analizabilidad. Por ello se propone utilizar este sobre imágenes RGB, es decir, el modelo sigue siendo entrenando sobre el canal Y pero, ahora se aplica la super resolución a cada uno de los canales de información RGB. Esto es posible dado que estos canales son similares al canal de luminancia Y, ya que estos también almacenan la información de luminosidad de la imagen. La obvia desventaja potencial es el incremento en los tiempos de procesamiento a causa del incremento de la información a tratar. La imagen 4.7 ilustra el proceso de construcción de la imagen de super resolución utilizando el modelo RGB.

# Capítulo 5

## Desarrollo del sistema embebido

### 5.1. Tarjeta de desarrollo

El sistema embebido elegido es la tarjeta Jetson TX2 [66]. Este módulo es una solución enfocada en inteligencia artificial fabricada por NVIDIA [67], comúnmente es utilizada en aplicaciones como robótica y dispositivos de borde. La tarjeta está construida en una arquitectura GPU Nvidia Pascal que goza de 2 CPUs Denver de 64 bits junto a un co-procesador Quad-Core A57. La tabla 5.1 tiene un resumen de las características de *hardware* de la tarjeta.

Característica	Valor
Arquitectura	Pascal
CPU	X2 Denver 64 bits
Co-procesador	Quad-Core A57
Ram	8 GB
Tipo Ram	128 bit DDR4
Memoria	32 GB
Tipo memoria	Flash

Tabla 5.1: Características de hardware Jetson TX2.

Como se puede observar, la tarjeta cuenta con recursos físicos más que suficientes y está pensada para desplegar soluciones de inteligencia artificial. Esta es idónea para el proyecto dado que, aunque la ESPCN fue concebida para consumir bajo recurso de cómputo, dicho consumo sigue siendo relativamente alto ya que el modelo esta pensado para trabajar sobre GPUs. Esto tiene sentido porque es difícil alcanzar un alto desempeño disminuyendo los recursos físicos que soportan el modelo. Recordar que el requerimiento más importante de la selección del modelo fue el desempeño de este. Para más información respecto a la selección del modelo revisar la sección 4.2.

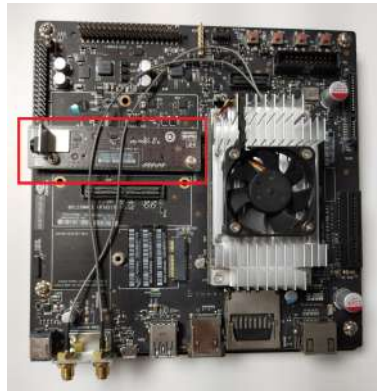
#### 5.1.1. Configuración de la cámara

La Jetson TX2 cuenta con un módulo de cámara 5 MP [66] enfocada en aplicaciones de visión embebida. Esta es una cámara digital regida por las especificaciones de la *MIPI Alliance*, siglas en inglés de *Mobile Industry Processor Interface Alliance* bajo el estándar CSI, *Camera Serial Interface*. Esta cámara está pensada para ser muy sencilla, la captura de las imágenes se basan en la técnica de foco fijo, por lo que todas las capturas siempre



Figura 5.1: Tarjeta de desarrollo Tx2 [propia]

tienen el mismo enfoque sin importar la distancia entre la cámara y el objeto a capturar. Al igual que el resto de la tarjeta, está pensado para soportar operaciones de visión artificial y se consideran que sus características son suficientes para el desarrollo del proyecto. La figura 5.2 permite observar la ubicación de la cámara dentro de la TX2. Dentro de la figura 5.2(a), en el cuadro resaltado en rojo, se puede apreciar el módulo que soporta la cámara, mientras la figura 5.2(b) permite ver la lente del dispositivo.



((a)) Vista superior



((b)) Vista frontal

Figura 5.2: Vistas superior (a) y frontal de la cámara (b) [propia]

## 5.2. Implementación del modelo

La lógica del sistema se puede dividir en: estructura y comportamiento. La estructura del sistema es una descripción estática orientada a explicar su composición básica, mientras, el comportamiento es una descripción funcional donde se explica como el sistema realiza sus funciones.



### 5.2.1. Estructura del sistema

La figura 5.3 permite observar el diagrama de componentes que describe al sistema. En esencia se busca mejorar la calidad del código a través de la modularización de este. Se crean 4 componentes distintos para cumplir responsabilidades específicas, de forma que, el programa principal *SR* se encarga de consumir recursos que brindan estos 4 componentes. De esta forma se reduce el acoplamiento del sistema logrando un código más limpio y reutilizable.

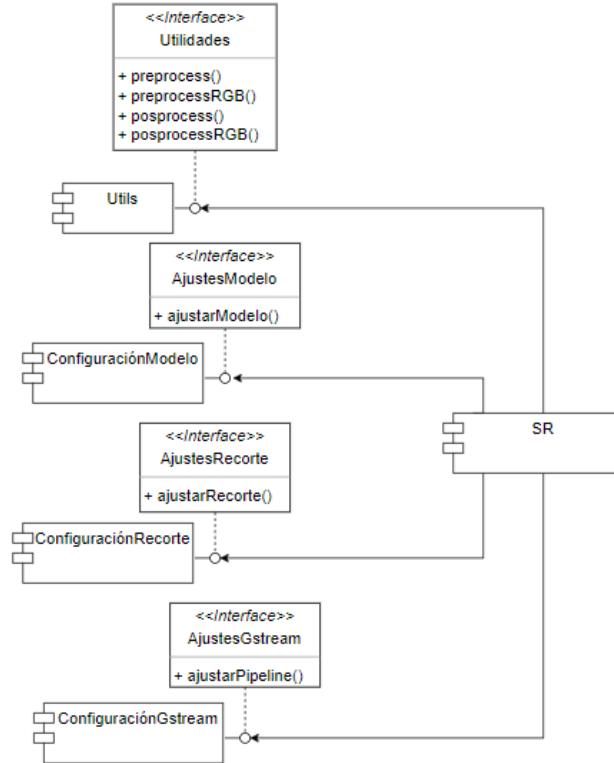


Figura 5.3: Diagrama de componentes del sistema [propia]

Las responsabilidades de los 4 componentes que brindan servicios a *SR* son:

- **Utils:** Esta contiene funciones de procesamiento de imágenes como transformaciones de formato RGB a YCbCr y a la inversa, la normalización de la información de los píxeles, manipulación de las dimensiones de las imágenes, etc; sin embargo, todas estas funciones son presentadas como 4 servicios únicos: el pre-procesamiento y pos-procesamiento de las imágenes, en RGB o solo en la componente Y. El pre-procesamiento busca preparar las imágenes para la entrada al modelo mientras que el pos-procesamiento construyen las imágenes finales a la salida de este.
- **configuracionModelo:** Dentro de este componente se encuentra la arquitectura de la ESPCN y la lógica detrás su configuración. Este permite instanciar el modelo, cargar los pesos de este y su asignación a un procesador.
- **configuracionRecorte:** Este componente embebe toda la lógica de la manipulación de la región de interés (ROI por sus siglas en inglés). Esta consiste en la región de la fotografía tomada por la cámara que contiene el rostro de una persona. El ROI es definido por el clasificador en cascada Haar de cv2 [69], que es un modelo de inteligencia artificial

entrenado para detectar rostros de personas en posición frontal. Una vez definido el ROI este debe ser procesado para asegurar que la totalidad del rostro sea capturada de forma adecuada. Para ello se ha definido como resolución de salida máxima del sistema 1280x960, dado que es la resolución dentro de SVGA [70] que es similar a las imágenes del *ground truth* con el que fue entrenado el sistema (1024x1024) y es lo suficientemente grande para permitir sub-muestreo con factor de 8.

- `configuracionGstream`: Este funciona como una capa de abstracción entre la librería de Gstreamer [68] y el sistema. Esta librería se encarga de la comunicación con la cámara física y la captura de imágenes. De forma que el componente facilita la configuración del pipeline que se encuentra detrás de la librería.

### 5.2.2. Comportamiento del sistema

La implementación de la ESPCN básicamente consiste en tomar imágenes, aumentar su resolución y posteriormente visualizar el resultado del proceso. La figura 5.4 contiene el resumen de este proceso. Los pasos descritos a continuación se repiten mientras el proceso asociado siga ejecutándose. Cerrar la ventana de visualización o cerrar dicho proceso finaliza con el flujo de operación.

- **Inicialización** : Se inicializan variables relacionadas con la captura de las imágenes y se instancian objetos importantes como la ESPCN y el modelo de clasificación en cascada encargado de detectar rostros.
- **Capturar imagen**: La cámara del sistema embebido toma una fotografía inicial.
- **Detección de rostro**: El modelo de clasificación en cascada se encarga de tomar la fotografía inicial y determinar si existe la vista frontal de un rostro humano. En caso de encontrarlo este modelo ubica una región de interés ROI alrededor del rostro encontrado.
- **Tratamiento del ROI**: El ROI debe ser procesado para asegurar que este contenga todo el rostro de la persona enfrente de la cámara y respetar el estándar SVGA escogido (1280x960).
- **Modelo**: Este bloque embebe el proceso de super resolución. El pre-procesamiento de la imagen se encarga de preparar la imagen para el modelo. Se incluyen actividades tales como la normalización de los píxeles y la descomposición de la imagen en sus canales de información. La ESPCN toma los canales a su entrada y los trata para aumentar su resolución espacial. Finalmente, en el bloque de pos-procesamiento, se unen los canales de información y se trata los datos obtenidos para construir la imagen final.
- **Visualizar imagen**: Se visualiza en pantalla la imagen de interés. Si el clasificador en cascada halló un rostro, se presenta la imagen del rostro en super resolución. De lo contrario se visualiza la imagen inicial tomada por la cámara.

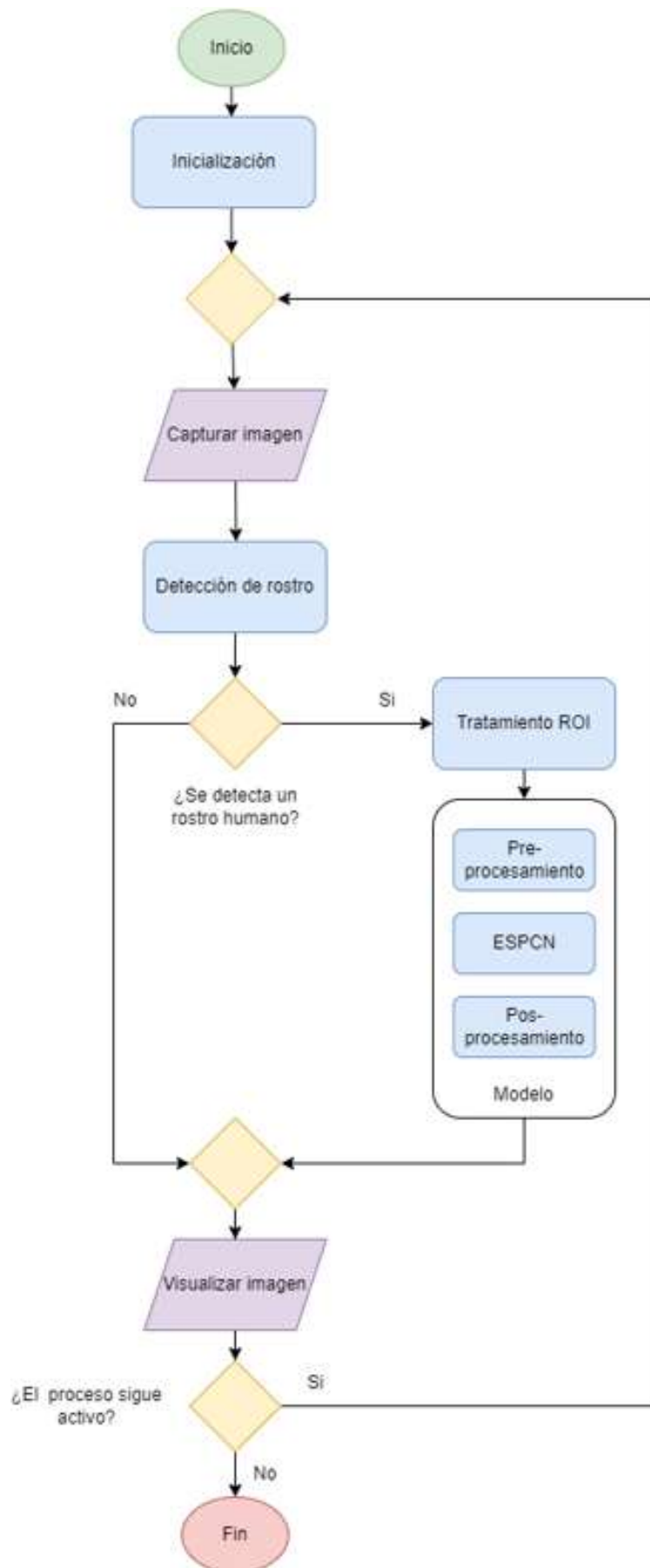


Figura 5.4: Diagrama de flujo implementación del modelo [propia]



Figura 5.5: Imágenes tomadas por la tarjeta con resolución dividida por 2 (a), 4 (b) y 8 (c) [propia]

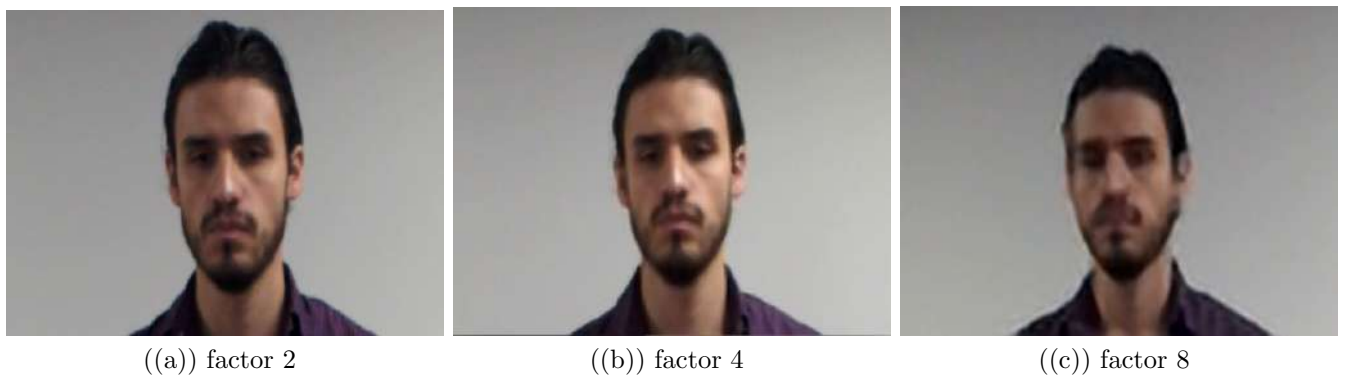


Figura 5.6: Imágenes RGB generadas por el modelo con factor 2 (a), 4 (b) y 8 (c) [propia]

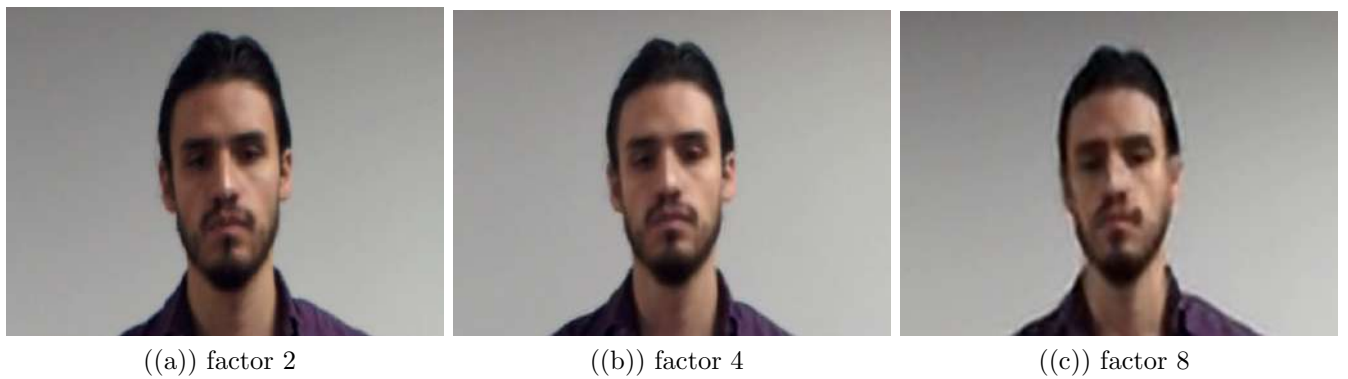


Figura 5.7: Imágenes Y generadas por el modelo con factor 2 (a), 4 (b) y 8 (c) [propia]

Imagen\Factor	2	4	8
RGB	5,72	4,47	4,5
Y	5,60	5,18	4,60

Tabla 5.2: Tiempos medios en segundos de procesamiento de imagen por el modelo

### 5.2.3. Operación del sistema

Para observar el sistema en funcionamiento se configura la cámara para capturar imágenes a tamaños específicos. En escancia se consideran tres tamaños: 640x480, 320x240 y 160x120. Estos son el tamaño máximo fijado (1280x960) dividido por 2, 4 y 8. De forma que se incrementa la resolución de cada una de estas imágenes por los factores 2, 4 y 8 respectivamente. La figura 5.5 permite ver imágenes iniciales del sistema sin tratamiento, sobre un rostro que no ha sido empleado como parte del entrenamiento, en cada uno de los tamaños descritos, mientras las figuras 5.6 y 5.7 cuenta con las imágenes obtenidas del proceso de super resolución a partir imágenes con los mismos tamaños de la figura 5.5.

Con el ánimo de entender los tiempos de procesamiento del sistema, se midieron los tiempos que tardaba el modelo en realizar la predicción de una imagen sobre los tres canales RGB y sobre el canal de luminancia Y. Este tiempo de procesamiento es el más importante dado que a comparación suya, los tiempos de pre-procesamiento, pos-procesamiento, detección de rostro y ajuste del ROI son despreciables. La tabla 5.2 permite observar el promedio en segundos de 10 muestras tomadas del procesamiento de una sola imagen RGB o Y. Se puede observar como los tiempos de procesamiento no son tan diferentes entre sí, pese a que en una imagen RGB se deben procesar 3 canales de información, mientras que en Y solo debe procesarse uno.

# Capítulo 6

## Análisis y resultados

Para analizar el desempeño del modelo se tienen una serie de métricas encargadas de cuantificar diferentes aspectos de las imágenes a la salida de la ESPCN. Como referencia se comparan los resultados de esta con métodos de interpolación clásicos: vecinos más cercanos, bilineal, bicúbica y Lánczos. Dentro del análisis del modelo se incluyen imágenes en YCbCr donde solo se trata el canal Y e imágenes RGB donde el tratamiento se extiende a los tres canales de información. Para más información respecto a la construcción de las imágenes por el modelo revisar la sección 4.4.

### 6.1. Métricas de comparación

Estas métricas se enfocan en comparar las imágenes del *ground truth* con respecto a las imágenes producidas con los diferentes métodos empleados.

#### 6.1.1. PSNR y SSIM

El PSNR ( *peak signal-to-noise ratio*) y SSIM ( *Structural similarity index*) son métricas ampliamente utilizadas por la literatura de super resolución. Estas métricas comparan dos imágenes píxel a píxel, de forma que, permiten medir la similitud global de dichas imágenes. Las figuras 6.1 y 6.2 contienen los resultados de PSNR y SSIM respectivamente.

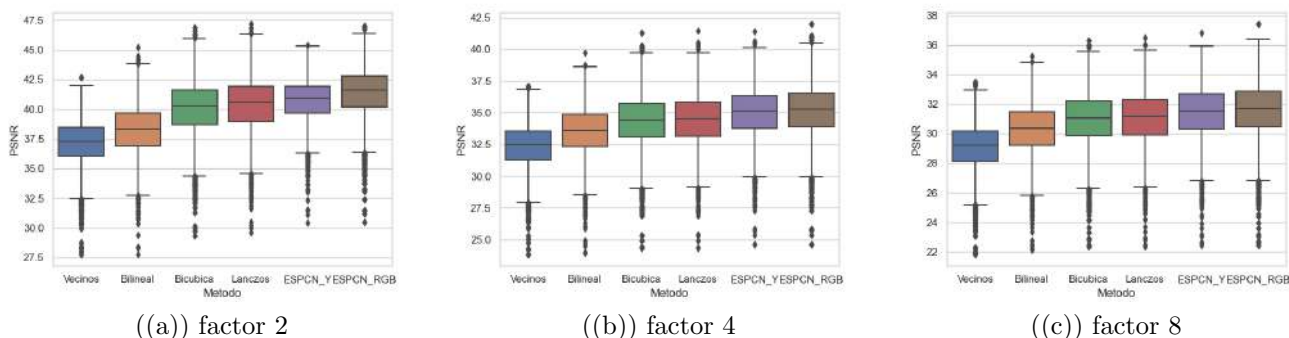


Figura 6.1: Resultados PSNR factor 2 (a), 4 (b) y 8 (c) [propia]

Los resultados de ambas métricas parecen ser congruentes entre sí. En primer lugar, la ESPCN es el método que mejores resultados obtiene en ambas sin importar el factor.

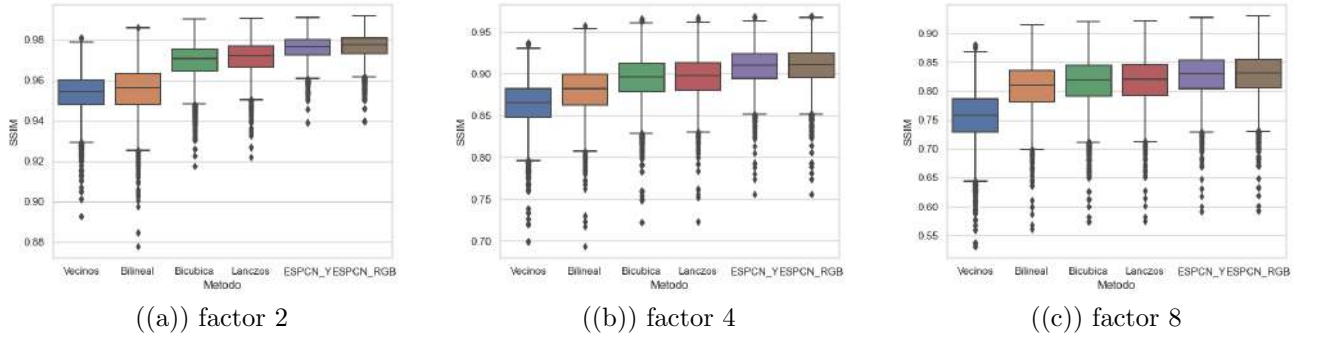


Figura 6.2: Resultados SSIM factor 2 (a), 4 (b) y 8 (c) [propia]

El método de interpolación con mejor desempeño es Lanczos y el de menor desempeño es vecinos más cercanos. Los métodos bilineal y bicubico tienen resultados intermedios pero bicubico tiene mejor desempeño.

A medida que el factor de sub-muestreo aumenta los resultados del modelo se acercan más a los resultados por los métodos de interpolación, se puede observar como el rango entre el valor menor y el mayor incrementan, hasta alcanzar rangos similares a los de Lanczos. Además la diferencia entre el promedio de este último método y el promedio de la ESPCN es menor a medida que el factor incrementa.

Un aspecto a resaltar es el hecho de que las imágenes  $ESPCN_Y$  utilizan dos canales provenientes del método de Lanczos, por lo que comparando ambos resultados se puede concluir que el tratamiento de un solo canal es suficiente para incrementar significativamente los resultados de la imagen. Esta mejora crece tratando los 3 canales de información como se pueden ver en los resultados de las imágenes  $ESPCN_{RGB}$ .

### 6.1.2. FID

El FID ( *Fréchet inception distance*) compara la distribución de dos grupos diferentes de imágenes, en este caso el *ground truth* y las imágenes producto de un método específico. Entre más pequeño sea el FID, las distribuciones entre ambos grupos de imágenes son más similares. Esta métrica es muy interesante dado que también es sensible a ruido y otras formas de perturbación sobre las imágenes. La figura 6.3 ilustra los resultados obtenidos dados los factores de sub-muestreo 2, 4 y 8.

El desempeño de la ESPCN supera a los métodos de interpolación en la mayoría de los casos. Cuando el factor es igual a 2 6.3(a), las imágenes provenientes del modelo tienen resultados excelentes, la diferencia con los métodos de interpolación es abrupta. Cuando el factor aumenta a 4 6.3(b), la diferencia disminuye bastante, de hecho, se aproximan a los resultados por vecinos más cercanos, que es el método con mejor FID. La figura 6.3(c) permite observar como este último método sobrepasa los resultados de la ESPCN, pese a que el modelo supera con una marcada diferencia a los métodos: bilineal, bicubica y Lanczos. Resulta interesante que la distribución de la interpolación de vecinos más cercanos sea tan similar a la del *ground truth*, dado que es el método más sencillo y con resultados más pobres con respecto al PSNR y SSIM.

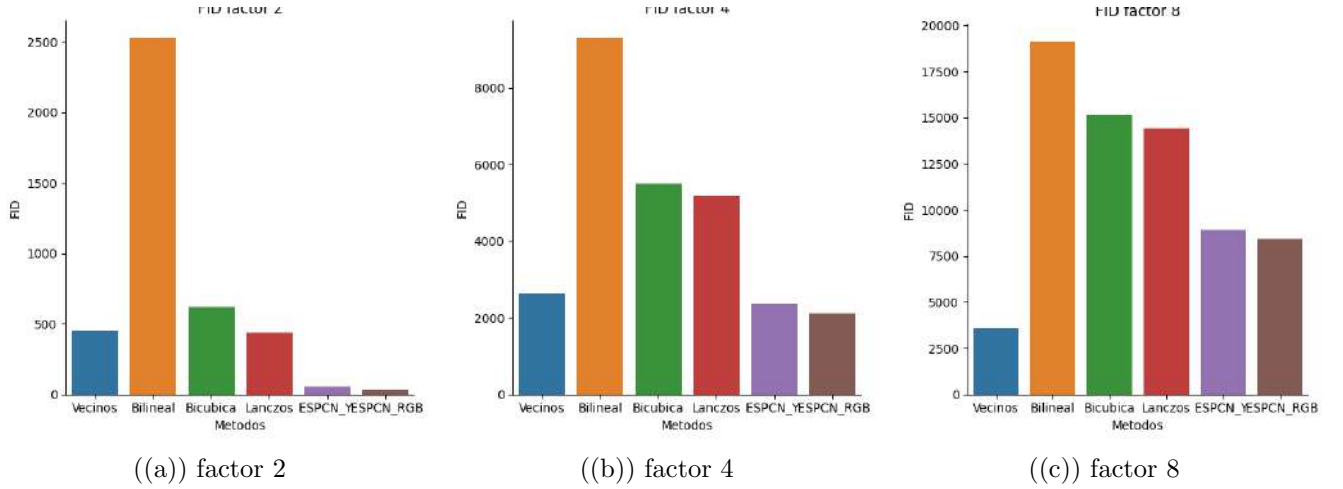


Figura 6.3: Resultados FID factor 2 (a), 4 (b) y 8 (c) [propia]

## 6.2. Métricas de calidad

Las métricas de calidad, a diferencia de las métricas previamente hechas, no usan como referencia el *ground truth*. Estas buscan medir parámetros específicos para entender las características de las imágenes a la salida del modelo. Como marco de referencia, los resultados se comparan con los de los métodos de interpolación.

### 6.2.1. Desenfoque

El desenfoque permite medir el nivel de "suavizado" que tiene una imagen. Altos niveles de desenfoque o *blur* están fuertemente relacionados con carencia de información de alta frecuencia. La figura 6.4 contiene los resultados de las pruebas dados diferentes factores de sub-muestreo.

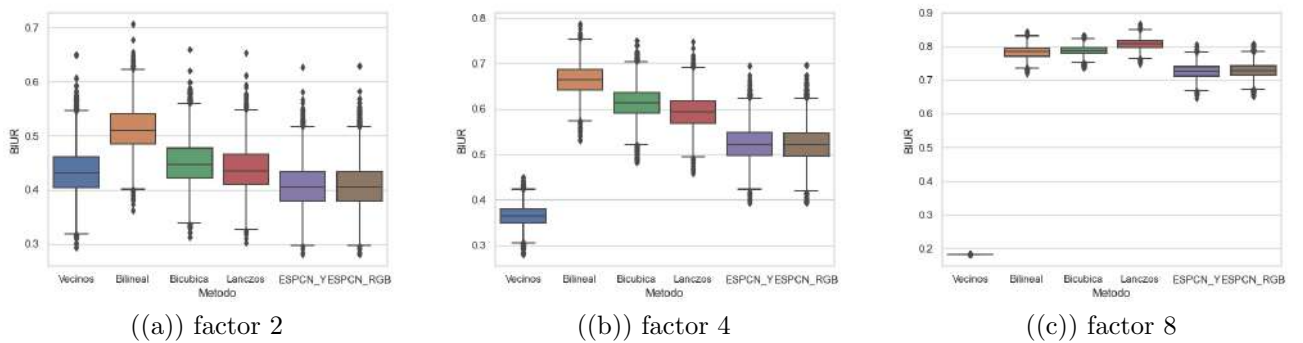


Figura 6.4: Resultado desenfoque factor 2 (a), 4 (b) y 8 (c) [propia]

Los niveles de desenfoque de la ESPCN son bajos en general. Esta logra superar los resultados de los métodos bilineal, bicúbico y Lanczos sin problema. Vecinos más cercanos, aunque en las pruebas con factor 2 6.4(a), obtiene resultados similares a los demás métodos, durante las pruebas 6.4(b) y 6.4(c), este logra conseguir niveles muy bajos de desenfoque con referencia a los demás métodos, incluso a los del modelo.



Comparando los resultados de  $ESPCN_Y$  y  $ESPCN_{RGB}$ , no parece existir una diferencia apreciable entre estos dos.

### 6.2.2. SNR

[4] define la relación señal a ruido (SNR por sus siglas en inglés) como la relación que existe entre la media de los píxeles y la desviación estándar de estos sobre una misma imagen. Esta es una métrica que busca calcular el grado en que los valores de los píxeles son diferentes unos a otros, asociando la concentración de valores atípicos con la presencia de ruido. La figura 6.5 permite observar los resultados obtenidos por los métodos utilizados por los factores 2, 4 y 8.

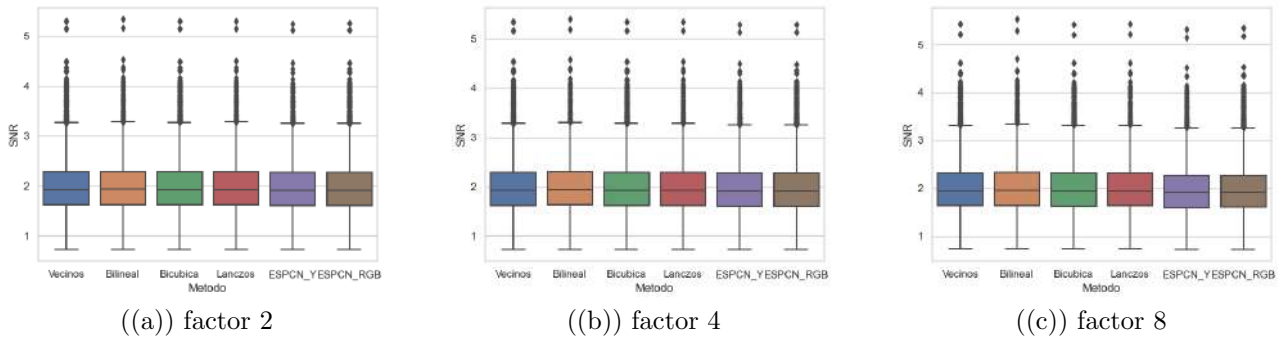


Figura 6.5: Resultados SNR factor 2 (a), 4 (b) y 8 (c) [propia]

Los resultados de las figuras 6.5(a), 6.5(b) y 6.5(c) no parecen tener diferencias significativas. Tampoco se observan diferencias importantes entre los métodos de interpolación y la ESPCN. Parece que la relación entre la media y la desviación estándar es aproximadamente la misma en todos los métodos. Esto resulta lógico dado que todos los métodos apuntan a tener los mismos valores sobre los píxeles, más específicamente los valores del *ground truth* por lo que tiene sentido que los resultados de esta métrica sean tan similares entre sí.

### 6.3. Verificación de identidad de rostros

Se desea evaluar si la identidad del rostro por los distintos métodos puede ser fácilmente asociada a la imagen original del *ground truth*. Para ello se hace uso de la distancia entre rostros entre estos dos grupos de imágenes. Esta métrica se basa en el modelado de rasgos faciales y la comparación entre dichos modelos. Distancias largas corresponden a rostros muy diferentes y distancias cortas a rostros muy similares. Idealmente, dos imágenes con rostros de la misma persona tendrían una distancia de 0. La figura 6.6 contiene los resultados obtenidos de la distancia entre los rostros por los factores 2, 4 y 8.

De acuerdo a los resultados obtenidos, se puede observar como factores más bajos están relacionados con distancias más bajas, es decir, es más sencillo verificar la identidad de la persona cuando el sub-muestro de la imagen ha sido menor. La figura 6.6(a) no permite observar diferencias significativas entre los métodos. Cuando el factor es 4, se observa como los métodos de vecinos más cercanos y bilineal empiezan a distanciarse de los demás, aun así la media de todos los métodos continúan por debajo de 0,05. Finalmente, la figura 6.6(c)

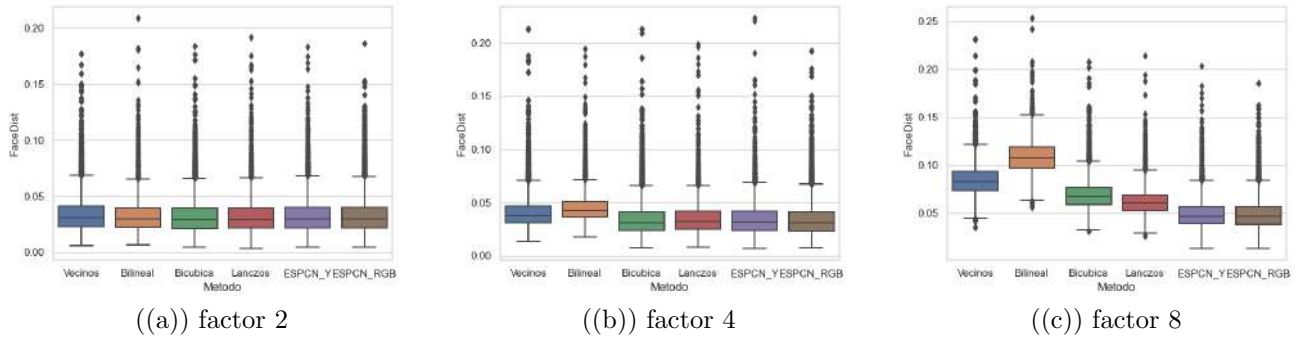


Figura 6.6: Resultados de la distancia entre rostros por factor 2 (a), 4 (b) y 8 (c) [propia]

permite observar una mayor variabilidad de los resultados. La media de todos los métodos de interpolación se encuentran por encima de los 0,05. El peor método es bilineal cuya media sobrepasa el umbral de los 0,01. La ESPCN tiene los mejores resultados comparados con los métodos de interpolación, parece que la información de los rasgos de las personas que esta logra conservar facilita la operación del software de reconocimiento facial. Esto tiene importantes implicaciones para potenciales aplicaciones en verificación de identidad, dado que este software puede determinar la identidad de un rostro 8 veces más pequeño a el *ground truth* con la ayuda de la ESPCN.

## 6.4. Comparación de imágenes

Las figuras 6.7 , 6.9 y 6.11 ilustran la comparación entre la construcción de una imagen tomada del conjunto de validación por cada uno de los métodos con factor 2, 4 y 8 respectivamente contra el *ground truth*. Dado que puede ser difícil comparar los resultados visualmente, las figuras 6.8 , 6.10 y 6.12 permiten amplificar rasgos específicos de las imágenes previas. Se puede observar que todos los métodos tienen facilidad en recrear la textura de la piel, incluso si esta tiene manchas o imperfecciones. Se cree que esto es dado que este patrón tiene poca información de alta frecuencia, por lo que puede ser fácilmente replicable hasta por los métodos de interpolación. Lo mismo sucede con pautas de arrugas; aunque si se puede apreciar como las arrugas más tenues pierden definición con mayor frecuencia en los métodos de interpolación. Por otra parte, los ojos de los rostros parecen ser el punto más difícil de tratar por todos los métodos, este es el punto donde es más fácil comparar el desempeño de estos. Finalmente, el patrón de barba o vello facial, parece encontrarse en un punto medio donde es más difícil que la piel, pero es más sencillo que los ojos.

Comparando todos los métodos, se puede apreciar como las imágenes de la ESPCN tienen siempre el mejor acabado confirmando los resultados obtenidos en las métricas. Comparando entre si el  $ESPCN_{RGB}$  Y  $ESPCN_Y$ , los resultados son bastante similares, pero el  $ESPCN_{RGB}$  parece tener un mejor acabado. El método de vecinos más cercanos tiene un acabado pixelado, probablemente el peor acabado, seguido por la interpolación bilineal que se caracteriza por sus imágenes con alto desenfoque. Los mejores métodos de interpolación son el bicúbico y Lanczos pero Lanczos tiene detalles más definidos.

Como resumen de todas las métricas empleadas, las tablas 6.1, 6.2 y 6.3 contienen los resultados de los factores 2, 4 y 8. Los valores del PSNR, SSIM, desenfoque o blur, SNR y la distancia del rostro son el valor promedio obtenido por todas las imágenes evaluadas.

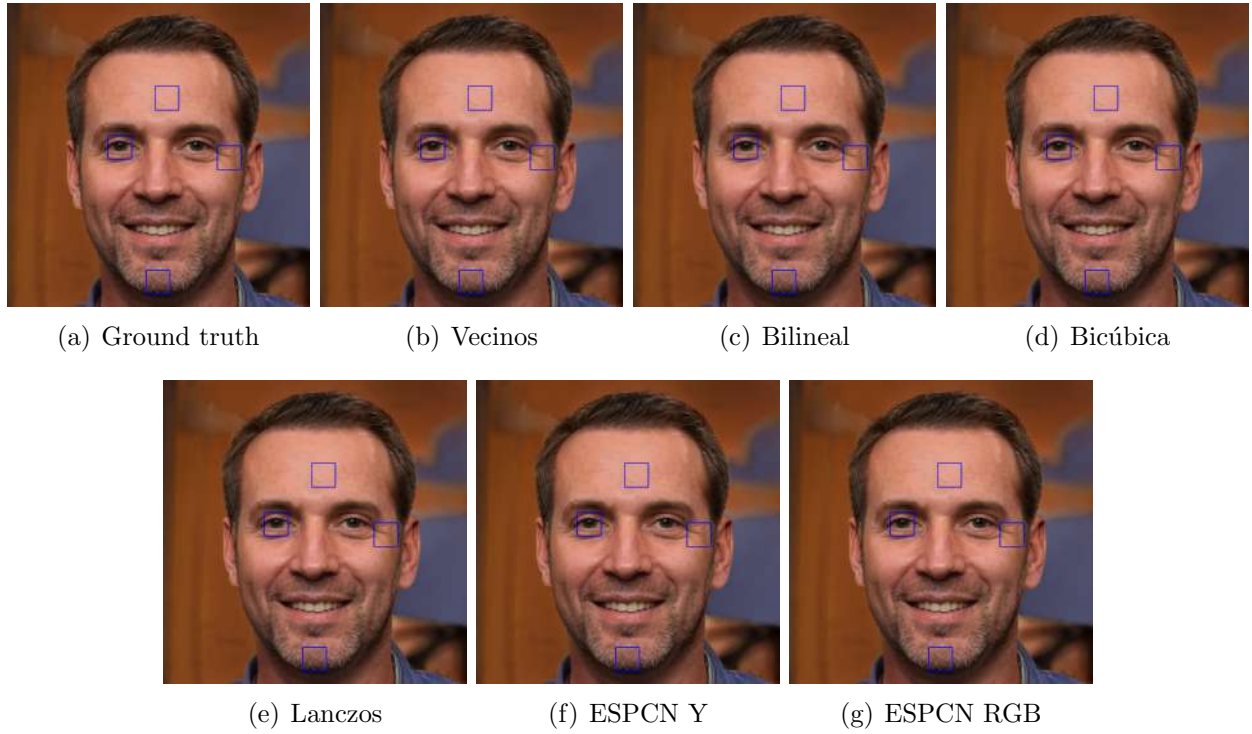


Figura 6.7: Comparación de imágenes por cada método factor 2 [propia]



Figura 6.8: Comparación sobre zonas específicas por cada método con factor 2 [propia]

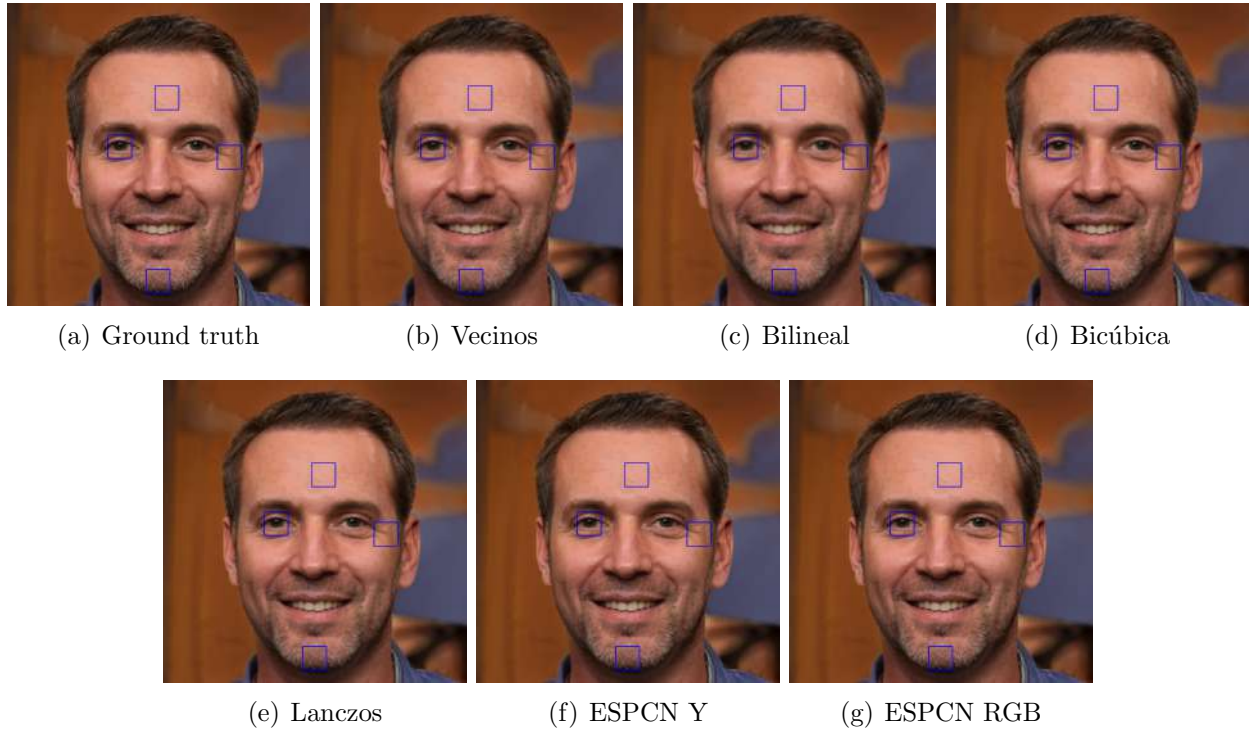


Figura 6.9: Comparación de imágenes por cada método factor 4 [propia]

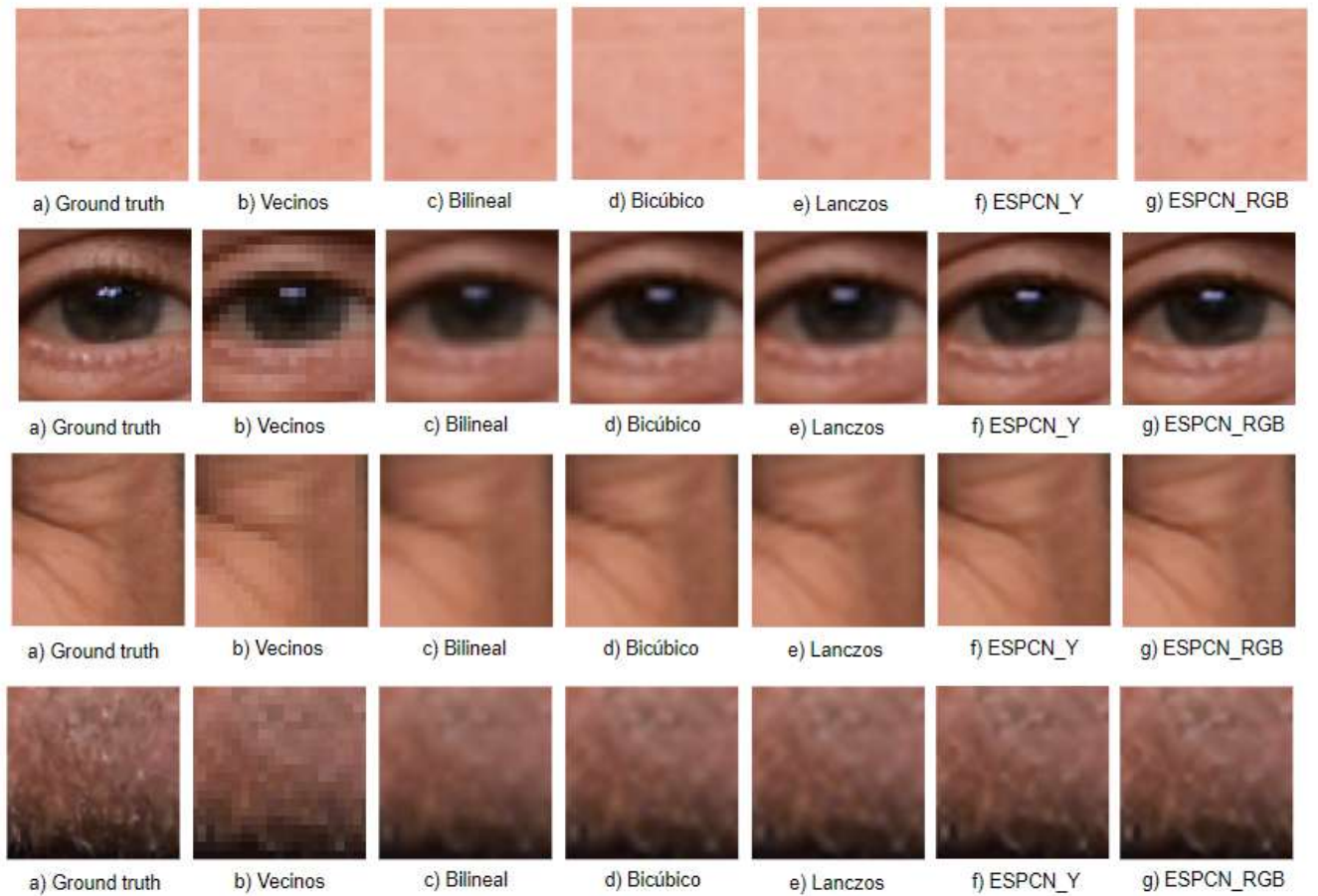


Figura 6.10: Comparación sobre zonas específicas por cada método con factor 4 [propia]



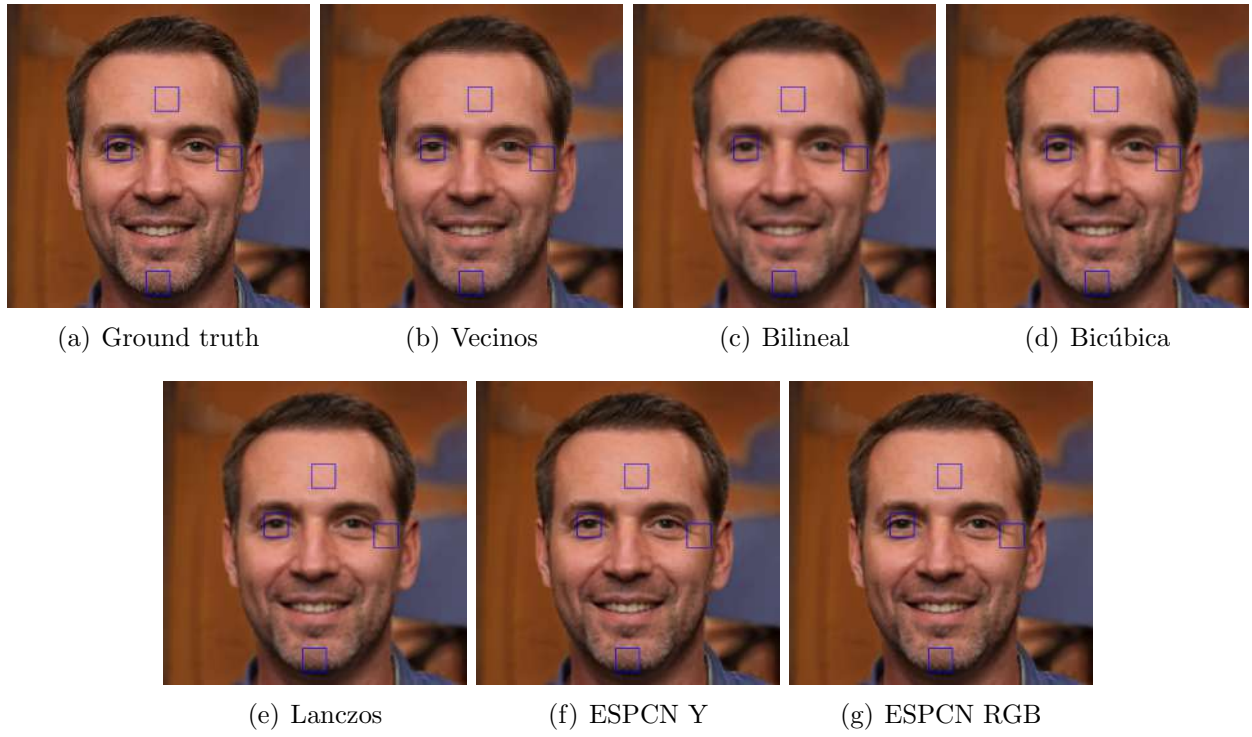


Figura 6.11: Comparación de imágenes por cada método factor 8 [propia]

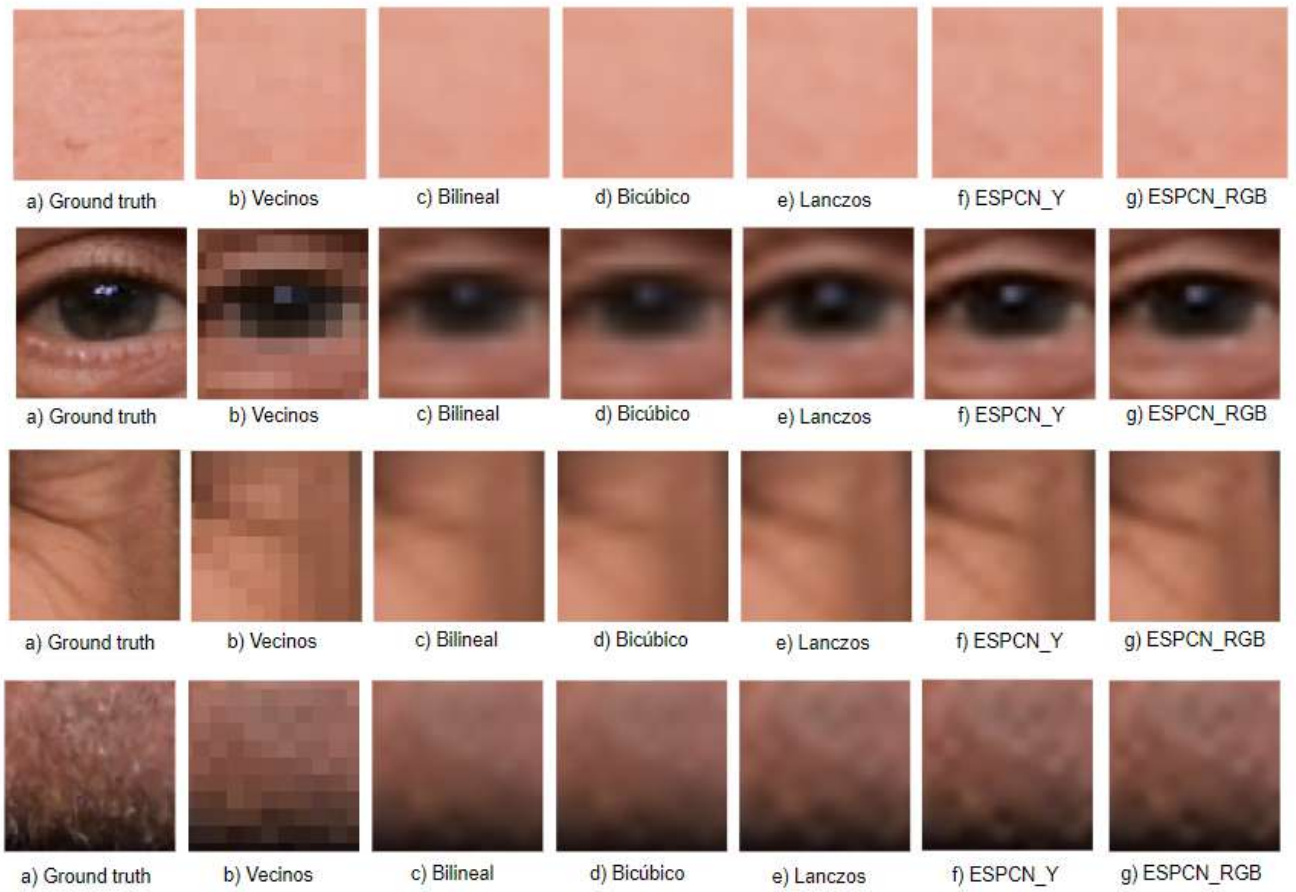


Figura 6.12: Comparación sobre zonas específicas por cada método con factor 8 [propia]

Métrica \ método	Factor 2					
	Vecinos	Bilineal	Bicúbica	Lanczos	ESPCN Y	ESPCN RGB
PSNR	37,16	38,21	40,07	40,38	40,70	41,38
SSIM	0,95	0,95	0,97	0,97	0,98	0,98
FID	450,48	2523,69	621,10	442,13	51,53	33,96
Blur	0,43	0,51	0,45	0,44	0,41	0,41
SNR	1,98	1,99	1,98	1,98	1,97	1,97
Face dist.	0,035	0,034	0,033	0,033	0,034	0,034

Tabla 6.1: Resumen resultados métricas factor 2

Métrica \ método	Factor 4					
	Vecinos	Bilineal	Bicúbica	Lanczos	ESPCN Y	ESPCN RGB
PSNR	32,36	33,5	34,34	34,43	34,97	35,15
SSIM	0,86	0,88	0,89	0,90	0,91	0,91
FID	2647,25	9295,36	5495,17	5193,59	2372,12	2121,90
Blur	0,36	0,66	0,61	0,59	0,52	0,52
SNR	1,99	2,00	1,99	1,99	1,98	1,98
Face dist.	0,041	0,046	0,035	0,036	0,035	0,035

Tabla 6.2: Resumen resultados métricas factor 4

Métrica \ método	Factor 8					
	Vecinos	Bilineal	Bicúbica	Lanczos	ESPCN Y	ESPCN RGB
PSNR	29,1	30,27	30,97	31,10	31,42	31,60
SSIM	0,76	0,81	0,82	0,82	0,83	0,83
FID	3600,99	19123,29	15120,29	14413,11	8893,36	8443,4
Blur	0,18	0,78	0,79	0,81	0,73	0,73
SNR	2,00	2,02	2,00	2,00	1,97	1,97
Face dist.	0,084	0,11	0,069	0,062	0,050	0,049

Tabla 6.3: Resumen resultados métricas factor 8

# Capítulo 7

## Conclusiones y Recomendaciones

### 7.1. Conclusiones

Los resultados obtenidos son prometedores, en términos generales el modelo ESPCN ha demostrado ser una técnica superior a los métodos de interpolación clásicos. La inteligencia artificial es capaz de tomar información a priori sobre un vasto conjunto de información y aplicarla para aumentar la resolución espacial de imágenes de rostros. Las desventajas de esta radican en el tiempo y recursos necesarios para el entrenamiento y el hecho de que este método solo sea capaz de predecir imágenes de rostros de personas.

Los aportes del trabajo radican en la implementación del modelo sobre un sistema embebido para realizar el proceso de super resolución en tiempo real; además, se realizó un esfuerzo importante en la construcción del conjunto de datos, evitando el *aliasing* y sus consecuencias perjudiciales sobre el entrenamiento, lo cual es un aspecto infra-valorada en trabajos similares. Resulta interesante que el modelo pueda ser entrenado únicamente sobre la componente de luminancia Y y al aplicarse sobre tres canales de información RGB se consiga tener un mejor desempeño sin gastar más recursos durante su ejecución o su entrenamiento. Finalmente, dados los resultados del modelo durante la verificación de identidad de rostros, se comprobó como la ESPCN puede ser una herramienta valiosa para la identificación de personas sobre imágenes, útil en aplicaciones donde la resolución de la imagen dificulten esta tarea, como que los individuos se encuentren a varios metros de distancia, entre otros.

Trabajos posteriores pueden enfocarse en la puesta en pruebas de lanzamiento del producto en contextos asociados a la vigilancia. Es necesario más información sobre los límites y capacidades de la tecnología en escenarios reales. En primer lugar, es necesario preparar el sistema para lograr trabajar con rostros desde diferentes ángulos y vistas; lo que significa que el entrenamiento debe realizarse con un conjunto de datos compuesto por imágenes desde diferentes perspectivas de caras de personas, además que el modelo detrás de la detección del rostro también debe soportar estos distintos escenarios. Actualmente, se desconoce una gran cantidad variables del sistema, como el rango de distancia de operación del sujeto de observación o los resultados sobre las imágenes cuando existen diferentes perturbaciones ambientales como ruido, contaminación lumínica, etc.

El mundo evoluciona a pasos agigantados todos los días y constantemente desafía la forma en que vivimos. La tecnificación de la vigilancia permite a los gobiernos y comunidades tener más control respecto a sus individuos, detectar amenazas de forma temprana, reducir riesgos, etc. Se espera que con el tiempo, el desarrollo de dispositivos similares se sumen a los esfuerzos para velar por la seguridad de las personas y la protección de la sociedad.

## Capítulo 8

### Anexo: Código de implementación

A continuación se agrega el link del repositorio de Github donde se encuentra todo el código empleado para el desarrollo del proyecto.

- <https://github.com/gomezan/SRrostros>



# Bibliografía

- [1] SCIKIT, «scikit-image 0.19.0.dev0 docs — skimage v0.19.0.dev0 docs», [Online] Available: <https://scikit-image.org/docs/dev/index.html>
- [2] K. TEAM,, «Keras documentation: InceptionV3», [Online] Available: <https://keras.io/api/applications/inceptionv3/>
- [3] J. BROWNLEE,, «How to Implement the Frechet Inception Distance (FID) for Evaluating GANs», *Machine Learning Mastery*, 29 de agosto de 2019, [Online] Available: <https://machinelearningmastery.com/how-to-implement-the-frechet-inception-distance-fid-from-scratch/>
- [4] SCIPY,, «SciPy.org — SciPy.org», [Online] Available: <https://www.scipy.org/>
- [5] A. GEITGEY, «Face Recognition», [Online] Available: [https://github.com/ageitgey/face\\_recognition](https://github.com/ageitgey/face_recognition)
- [6] D. NAGOTHU, R. XU, S. Y. NIKOUEI y Y. CHEN, «A Microservice-enabled Architecture for Smart Surveillance using Blockchain Technology», *IEEE International Smart Cities Conference (ISC2)*, sep. 2018, pp. 1-4.
- [7] G. BALDONI, M. MELITA, S. MICALIZZI, C. RAMETTA, G. SCHEMBRA, y A. VASSALLO, «A dynamic, plug-and-play and efficient video surveillance platform for smart cities, » *en 2017 14th IEEE Annual Consumer Communications Networking Conference (CCNC)*, ene. 2017, pp. 611-612.
- [8] D. EIGENRAAM y L. J. M. ROTHKRANTZ,, «A smart surveillance system of distributed smart multi cameras modelled as agents, » *en 2016 Smart Cities Symposium Prague (SCSP)*, may 2016, pp. 1-6.
- [9] J. VAN HEEK, K. AMING, y M. ZIEFLE,, «“How fear of crime affects needs for privacy and safety”: Acceptance of surveillance technologies in smart cities» *en 2016 5th International Conference on Smart Cities and Green ICT Systems (SMARTGREENS)*, abr. 2016, pp. 1-12.
- [10] A. KHAN, M. A. KHAN, F. OBAID, S. JADOON, M. A. KHAN, y M. SIKANDAR,, «A novel multi-frame super resolution algorithm for surveillance camera image reconstruction» *en 2015 First International Conference on Anti-Cybercrime (ICACC)*, nov. 2015, pp. 1-6.
- [11] F. MOKHAYERI, E. GRANGER, y G.-A. BILODEAU,, «Synthetic face generation under various operational conditions in video surveillance» *en 2015 IEEE International Conference on Image Processing (ICIP)*, sep. 2015, pp. 4052-4056.

- [12] «IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries» *IEEE Std 610*, pp. 1-217, ene. 1991.
- [13] P. RONCAGLIOLO, «Procesamiento digital de imágenes».[Online] Available: [http://www2.elo.utfsm.cl/~elo328/pdf1dpp/PDI13\\_Color\\_1dpp.pdf](http://www2.elo.utfsm.cl/~elo328/pdf1dpp/PDI13_Color_1dpp.pdf)
- [14] J. IGUAL GARCÍA,, «Modelo de color RGB: representación de un color mediante la mezcla por adición de los tres colores luz primarios rojo, verde y azul,» abr. 2011.
- [15] J. A. MARCIAL BASILIO, G. AGUILAR TORRES, G. SÁNCHEZ PÉREZ, K. TOSCANO MEDINA, Y H. M. PÉREZ MEANA, «Novedosa técnica para la detección de imágenes pornográficas empleando modelos de color HSV y YCbCr, » *Revista Facultad de Ingeniería Universidad de Antioquia*, .<sup>o</sup> 64, pp. 79-90, sep. 2012.
- [16] SUNG CHEOL PARK, MIN KYU PARK, Y MOON GI KANG,, «Super-resolution image reconstruction: a technical overview» *IEEE Signal Process. Mag.*, vol. 20, n.o 3, pp. 21-36, may 2003.
- [17] R. GARCIA Y M. MONTOLIO,, «La interpolación aplicada al procesamiento de imágenes digitales», p. 74.
- [18] L. JING, S. XIONG, Y W. SHIHONG,, «An Improved Bilinear Interpolation Algorithm of Converting Standard-Definition Television Images to High-Definition Television Images» en *2009 WASE International Conference on Information Engineering*, jul. 2009, vol. 2, pp. 441-444.
- [19] H. KIM, S. PARK, J. WANG, Y. KIM, Y J. JEONG,, «Advanced Bilinear Image Interpolation Based on Edge Features» en *2009 First International Conference on Advances in Multimedia*, jul. 2009, pp. 33-36.
- [20] PHOTOSHOP, «Software de fotografía y diseño — Comprar Adobe Photoshop oficial».[Online] Available: <https://www.adobe.com/la/products/photoshop.html>
- [21] H. ZHANG, G. PENG, Y L. LIU,, «Low complexity signal detector based on Lanczos method for large-scale MIMO systems» en *2016 6th International Conference on Electronics Information and Emergency Communication (ICEIEC)*, jun. 2016, pp. 6-9.
- [22] M. R. ZUNOUBI, J.-M. JIN, K. C. DONEPUDI, Y W. C. CHEW,, «A spectral Lanczos decomposition method for solving 3-D low-frequency electromagnetic diffusion by the finite-element method» en *2016 6th International Conference on Electronics Information and Emergency Communication (ICEIEC)*, IEEE Trans. Antennas Propag., vol. 47, n.o 2, pp. 242-248, feb. 1999.
- [23] J. CHEN Y Y. SAAD,, «Lanczos Vectors versus Singular Vectors for Effective Dimension Reduction» *IEEE Trans. Knowl. Data Eng.*, vol. 21, n.o 8, pp. 1091-1103, ago. 2009.
- [24] W. YING,, «An Improved Block Lanczos Algorithm to Solve Large and Sparse Matrixes on GPUs» en *2013 Ninth International Conference on Computational Intelligence and Security*, dic. 2013, pp. 464-468.
- [25] Y. MENGBEI, W. HONGJUAN, L. MENG YANG, Y L. PEI,, «Overview of Research on Image Super-Resolution Reconstruction» en *2021 IEEE International Conference on Information Communication and Software Engineering (ICICSE)*, mar. 2021, pp. 131-135.

- [26] X. NIU,, «An Overview of Image Super-Resolution Reconstruction Algorithm» *en 2018 11th International Symposium on Computational Intelligence and Design (ISCID)*, dic. 2018, vol. 02, pp. 16-18.
- [27] E. CASTRO, M. NAKANO, G. SÁNCHEZ, Y H. PÉREZ,, «Improvement of Image Super-resolution Algorithms using Iterative Back Projection», *IEEE Lat. Am. Trans.*, vol. 15, n.o 11, pp. 2214-2219, nov. 2017.
- [28] L. ZIWEI, W. CHENG DONG, C. DONGYUE, Q. YUANCHEN, Y W. CHUNPING,, «Overview on image super resolution reconstruction», *en The 26th Chinese Control and Decision Conference (2014 CCDC)*, may 2014, pp. 2009-2014.
- [29] P. SHAMSOLMOALI, «Deep learning approaches for real-time image super-resolution, » p. 2.
- [30] T. CUI, L. TANG, J. NAN, Y Z. LI,, «Space Target Super-resolution Based on Low-complex Convolutional Networks, » *en 2019 IEEE International Conference on Signal, Information and Data Processing (ICSIDP)*, dic. 2019, pp. 1-5.
- [31] H. HUANG, X. FAN, C. QI, Y S.-H. ZHU,, «A learning-based POCS algorithm for face image super-resolution reconstruction, » *en 2005 International Conference on Machine Learning and Cybernetics*, ago. 2005, vol. 8, pp. 5071-5076 Vol. 8.
- [32] X. YANG, W. WU, K. LIU, P. W. KIM, A. K. SANGAIAH, Y G. JEON,, «Long-Distance Object Recognition With Image Super Resolution: A Comparative Study, » *IEEE Access*, vol. 6, pp. 13429-13438, 2018.
- [33] X. HU, X. LIU, Z. WANG, X. LI, W. PENG, Y G. CHENG, «RTSRGAN: Real-Time Super-Resolution Generative Adversarial Networks,» *en 2019 Seventh International Conference on Advanced Cloud and Big Data (CBD)*, sep. 2019, pp. 321-326.
- [34] Y. LEE, J. YUN, Y. HONG, J. LEE, Y M. JEON, «Accurate License Plate Recognition and Super-Resolution Using a Generative Adversarial Networks on Traffic Surveillance Videom,» *en 2018 IEEE International Conference on Consumer Electronics - Asia (ICCE-Asia)*, jun. 2018, pp. 1-4.
- [35] S. Y. KIM Y P. BINDU «Realizing Real-Time Deep Learning-Based Super-Resolution Applications on Integrated GPUs, » *en 2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, dic. 2016, pp. 693-696.
- [36] Z. BOJKOVIC Y A. SAMCOVIC, «Face Detection Approach in Neural Network Based Method for Video Surveillance,» *en 2006 8th Seminar on Neural Network Applications in Electrical Engineering*, sep. 2006, pp. 44-47.
- [37] H. QEZAVATI, B. MAJIDI, Y M. T. MANZURI, «Partially Covered Face Detection in Presence of Headscarf for Surveillance Applications,» *en 2019 4th International Conference on Pattern Recognition and Image Analysis (IPRIA)*, mar. 2019, pp. 195-199.
- [38] Y. WANG, T. BAO, C. DING, Y M. ZHU, «Face recognition in real-world surveillance videos with deep learning method,» *en 2017 2nd International Conference on Image, Vision and Computing (ICIVC)*, jun. 2017, pp. 239-243.

- [39] J. HARIKRISHNAN, A. SUDARSAN, A. SADASHIV, Y R. A. S. AJAI, «Vision-Face Recognition Attendance Monitoring System for Surveillance using Deep Learning Technology and Computer Vision,» *en 2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN)*, mar. 2019, pp. 1-5.
- [40] Q. ZHAO Y S. WANG, «Real-time Face Tracking in Surveillance Videos on Chips for Valuable Face Capturing,» *en 2020 International Conference on Artificial Intelligence and Computer Engineering (ICAICE)*, oct. 2020, pp. 281-284.
- [41] A. MAKHFOUDI, S. ALMAADEED, A. BOURIDANE, G. SEXTON, Y R. JIANG, «Visualization of faces from surveillance videos via face hallucination,» *en 2014 International Conference on Control, Decision and Information Technologies (CoDIT)*, nov. 2014, pp. 701-705.
- [42] Z.-L. CHEN, Q.-H. HE, W.-F. PANG, Y Y.-X. LI, «Frontal Face Generation from Multiple Pose-Variant Faces with CGAN in Real-World Surveillance Scene,» *en 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, abr. 2018, pp. 1308-1312.
- [43] W. W. W. ZOU Y P. C. YUEN, «Learning the Relationship Between High and Low Resolution Images in Kernel Space for Face Super Resolution,» *en 2010 20th International Conference on Pattern Recognition*, ago. 2010, pp. 1152-1155.
- [44] P. H. HENNINGS-YEOMANS, B. V. K. V. KUMAR, Y S. BAKER, «Robust low-resolution face identification and verification using high-resolution features,» *en 2009 16th IEEE International Conference on Image Processing (ICIP)*, nov. 2009, pp. 33-36.
- [45] Z. GONG, F. YU, Y Y. TANG, «Real-time Video Image Super-Resolution Network for Mobile Terminals,» *en 2019 3rd International Conference on Circuits, System and Simulation (ICCSS)*, jun. 2019, pp. 201-205.
- [46] T. MANABE, Y. SHIBATA, Y K. OGURI, «FPGA implementation of a real-time super-resolution system using a convolutional neural network,» *en 2016 International Conference on Field-Programmable Technology (FPT)*, dic. 2016, pp. 249-252.
- [47] Z. HE, H. HUANG, M. JIANG, Y. BAI, Y G. LUO, «FPGA-Based Real-Time Super-Resolution System for Ultra High Definition Videos,» *en 2018 IEEE 26th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*, br. 2018, pp. 181-188.
- [48] THIS PERSON DOES NOT EXIST, «This Person Does Not Exist».[Online] Available: <https://thispersondoesnotexist.com/>
- [49] G. ROELOFS, «PNG: The Definitive Guide,» O'Reilly Media, 1999.[Online] Available: <https://oers.taiwanmooc.org/handle/123456789/130174>
- [50] TENSORFLOW.IMAGE, «Module: tf.image — TensorFlow Core v2.8.0,» TensorFlow.[Online] Available: [https://www.tensorflow.org/api\\_docs/python/tf/image](https://www.tensorflow.org/api_docs/python/tf/image)
- [51] PILLOW, « Pillow: Python Imaging Library (Fork) ,».[Online] Available: <https://python-pillow.org>

- [52] OPENCV, « OpenCV ,».[Online] Available: <https://opencv.org/>
- [53] Y. KIM, J.-S. CHOI, Y M. KIM, «A Real-Time Convolutional Neural Network for Super-Resolution on FPGA With Applications to 4K UHD 60 fps Video Services, » *IEEE Trans. Circuits Syst. Video Technol*, vol. 29, n.o 8, pp. 2521-2534, ago. 2019.
- [54] X. LI, Y. WU, W. ZHANG, R. WANG, Y F. HOU, «Deep learning methods in real-time image super-resolution: a survey,» *Real-Time Image Process*, vol. 17, n.o 6, pp. 1885-1909, dic. 2020.
- [55] VEGA, «HUAWEI Noah's Ark Lab, 2022».[Online] Available: [https://github.com/huawei-noah/vega/blob/f18a6ab03b8d9ee5b3bfaf5fa36e4b77365fa712/docs/en/algorithms/esr\\_ea.md/](https://github.com/huawei-noah/vega/blob/f18a6ab03b8d9ee5b3bfaf5fa36e4b77365fa712/docs/en/algorithms/esr_ea.md/)
- [56] C. TIAN, «LESRCNN,».[Online] Available: <https://github.com/hellloxiaotian/LESRCNN>
- [57] C. ZHANG, « zoom-learn-zoom ,».[Online] Available: <https://github.com/ceciliavision/zoom-learn-zoom>
- [58] J. YAMANAKA, « Fast and Accurate Image Super Resolution by Deep CNN with Skip Connection and Network in Network,».[Online] Available: <https://github.com/jiny2001/dcscn-super-resolution>
- [59] J. YEO, « ESPCN,».[Online] Available: <https://github.com/yjn870/ESPCN-pytorch>
- [60] Q. DAI, « VSRNet,».[Online] Available: [https://github.com/usstdqq/vsrnet\\_pytorch](https://github.com/usstdqq/vsrnet_pytorch)
- [61] H. RAZA, «Fast-SRGAN ,».[Online] Available: <https://github.com/HasnainRaz/Fast-SRGAN>
- [62] X. TAO, «Detail-revealing Deep Video Super-resolution ,».[Online] Available: [https://github.com/jiangsutx/SPMC\\_VideoSR](https://github.com/jiangsutx/SPMC_VideoSR)
- [63] W. SHI ET AL, « Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network,» *arXiv*, 23 de septiembre de 2016.
- [64] D. P. KINGMA Y J. BA, «Adam: A Method for Stochastic Optimization» *arXiv*, 29 de enero de 2017.
- [65] PYTORCH, «PyTorch,».[Online] Available: <https://www.pytorch.org>
- [66] JETSON TX2, «Harness AI at the Edge with the Jetson TX2 Developer Kit, ».[Online] Available: <https://developer.nvidia.com/embedded/jetson-tx2-developer-kit>
- [67] NVIDIA, «Líder en Computación de Inteligencia Artificial — NVIDIA».[Online] Available: <https://www.nvidia.com/es-la>
- [68] GSTREAMER, «gststreamer open source multimedia framework», [Online] Available: <https://gststreamer.freedesktop.org/bindings/python.html>
- [69] FACE DETECTION, «OpenCV: Cascade Classifier», [Online] Available: [https://docs.opencv.org/3.4/db/d28/tutorial\\_cascade\\_classifier.html](https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html)
- [70] VGA, «Video Graphics Array», p. 9.