Análisis de estrategias de superresolución para la identificación de placas vehiculares en imágenes con resolución no apropiada

Alejandro Ocampo Rojas

Trabajo de grado para optar por el título de Ingeniero Electrónico

Tutor

Ricardo Andrés Velásquez Vélez, PhD

Universidad de Antioquia

Facultad de ingeniería

Ingeniería electrónica

Medellín

2024

Resumen

Este proyecto abordó la identificación de placas vehiculares en imágenes de baja resolución capturadas por cámaras de seguridad no especializadas. Para mejorar la calidad de las imágenes y facilitar su análisis, se implementaron técnicas de superresolución utilizando arquitecturas GAN (SRGAN, ESRGAN y Real-ESRGAN).

Se creó una base de datos con 14,393 imágenes en resoluciones de 128x128 y 32x32 píxeles, etiquetadas usando YOLOv8. Las arquitecturas GAN fueron evaluadas en tres formatos de imagen (JPG, PNG y WEBP), cinco resoluciones y cuatro niveles de compresión para determinar su impacto en el rendimiento.

Los resultados mostraron que ESRGAN superó a SRGAN en calidad y nitidez de imágenes, especialmente en resoluciones bajas y medias, y que el formato PNG mostró mayor resistencia a la compresión. Real-ESRGAN no obtuvo resultados satisfactorios tras 100 épocas de entrenamiento debido a su complejidad y limitaciones de hardware.

Finalmente, se implementó un sistema completo en una SBC VIM3 de Khadas con una webcam Allink 550 1080p, incluyendo captura de video, detección de vehículos y placas con YOLOv8, superresolución, OCR y mejora de contraste. Los experimentos demostraron la viabilidad de aplicar estas técnicas en entornos reales, mejorando la identificación de las placas vehiculares.

Tabla de contenido

1. Introducción

1.1. Planteamiento del problema

En la actualidad, uno de los desafíos más significativos en el control del tráfico y la seguridad vial es la baja resolución de las imágenes captadas por las cámaras de seguridad. Estas cámaras, frecuentemente ubicadas en puntos críticos para la vigilancia urbana, no siempre cuentan con la capacidad técnica para producir imágenes de alta definición. Esta limitación tecnológica resulta en dificultades para la identificación precisa de placas vehiculares, lo cual impacta negativamente en la capacidad de monitoreo y control del tráfico, así como en la efectividad de las medidas de seguridad ciudadana. La identificación incorrecta o fallida de placas puede tener consecuencias graves, desde la incapacidad para rastrear vehículos involucrados en delitos hasta la imposibilidad de aplicar multas por infracciones de tránsito.

Para abordar esta problemática, se han explorado diversas estrategias de superresolución de imágenes. La superresolución es una técnica que permite aumentar la resolución de una imagen, mejorando así su calidad y nitidez. Entre las metodologías más avanzadas para lograr este objetivo se encuentran las arquitecturas basadas en Redes Generativas Adversarias (GAN), una técnica de aprendizaje profundo que ha demostrado ser efectiva en la generación de imágenes de alta calidad a partir de entradas de baja resolución. Las GAN se componen de dos redes neuronales que compiten entre sí: un generador que intenta crear imágenes realistas y un discriminador que trata de diferenciar entre imágenes reales y generadas. Este enfoque adversarial permite al generador mejorar progresivamente la calidad de las imágenes sintetizadas.

Este trabajo de grado se enfoca en la implementación y evaluación de diferentes arquitecturas GAN, específicamente SRGAN, ESRGAN y Real-ESRGAN, con el objetivo de

mejorar la resolución de las imágenes de placas vehiculares captadas por cámaras de seguridad. Se propone un entorno experimental que incluye la creación de una base de datos robusta con imágenes de placas vehiculares en diversas resoluciones y formatos. Además, se emplearán algoritmos de identificación y procesamiento de imágenes, integrando métodos de visión por computadora y aprendizaje profundo para evaluar el desempeño de las arquitecturas GAN en términos de mejora de resolución y efectividad en la identificación de placas.

La implementación de estas técnicas de superresolución basada en arquitecturas GAN ofrece varias ventajas significativas. En primer lugar, las arquitecturas GAN pueden aumentar significativamente la resolución y nitidez de las imágenes, lo cual es crucial para la identificación precisa de placas vehiculares. En segundo lugar, la mejora en la calidad de las imágenes capturadas por las cámaras de seguridad permitirá una detección y rastreo más preciso de las placas vehiculares. Finalmente, un sistema de superresolución eficiente contribuirá a fortalecer las medidas de vigilancia y seguridad en la ciudad, facilitando el monitoreo y control del tráfico y mejorando la efectividad de las medidas de seguridad ciudadana.

1.2. Objetivos

1.2.1. Objetivo general

Desarrollar un sistema de superresolución para mejorar la precisión de sistemas de identificación de placas vehiculares cuyas imágenes son captadas por cámaras de seguridad no especializadas, utilizando arquitecturas GAN y lenguaje de programación Python.

1.2.2. Objetivos específicos

- Construir una base de datos representativa de imágenes de placas vehiculares, que incluya tanto imágenes de baja resolución como de alta resolución.

- Seleccionar una arquitectura GAN que permita generar imágenes de placas en alta resolución a partir de imágenes de baja calidad con altos niveles de exactitud.
- Implementar un algoritmo que permita la identificación de placas vehiculares en imágenes o videos no ideales utilizando la arquitectura GAN seleccionada y lenguaje de programación Python.
- Evaluar cuantitativa y cualitativamente la efectividad del modelo GAN en la precisión del sistema de identificación de placas vehiculares.

1.3. Contribuciones del trabajo

Este trabajo de grado contribuye de manera significativa a la seguridad y el control del tráfico mediante el desarrollo de un sistema de superresolución de imágenes basado en arquitecturas GAN. La investigación no solo implementa y evalúa arquitecturas como SRGAN, ESRGAN y Real-ESRGAN, sino que también proporciona un análisis comparativo de su rendimiento en términos de mejora de resolución y efectividad en la identificación de placas vehiculares. Además, se desarrolla una base de datos robusta con imágenes de placas vehiculares en diversas resoluciones y formatos, enriqueciendo los recursos disponibles para futuras investigaciones en el campo de la visión por computadora y la seguridad vial. La implementación de este sistema en una Single Board Computer (SBC) demuestra su viabilidad en aplicaciones del mundo real, mejorando la calidad de las imágenes capturadas por cámaras de seguridad y facilitando el reconocimiento de placas.

1.4. Contenido del trabajo

El contenido del trabajo se organiza en varias secciones clave que abarcan desde la introducción al problema hasta la evaluación de los resultados obtenidos. En los primeros

capítulos, se describe el contexto y la motivación del estudio, seguido de una revisión exhaustiva de la literatura relacionada con superresolución de imágenes y arquitecturas GAN. Posteriormente, se detallan los métodos y materiales utilizados, incluyendo la construcción de la base de datos y los algoritmos implementados para la identificación de placas vehiculares. La sección de experimentos presenta el diseño experimental y los parámetros de entrenamiento de las diferentes arquitecturas GAN. Finalmente, se discuten los resultados obtenidos, se comparan las diferentes técnicas evaluadas y se concluye con las implicaciones de los hallazgos y posibles direcciones para investigaciones futuras. También se incluye una descripción detallada de la implementación del sistema en una SBC, especificando los componentes utilizados y los pasos seguidos para integrar las tecnologías desarrolladas.

2. Marco teórico

2.1. Arquitectura GAN

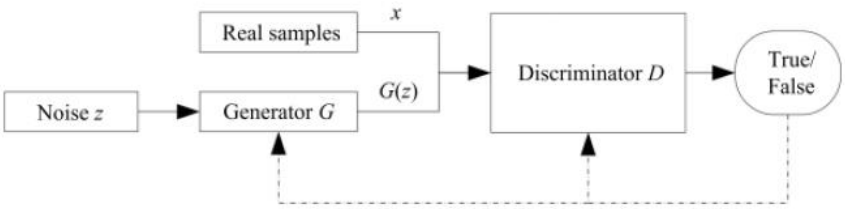
Las redes generativas adversarias se han convertido en el foco de investigación de la inteligencia artificial, comprendiendo un generador y un discriminador entrenados bajo la idea de aprendizaje adversario. El objetivo de las GAN es estimar la distribución potencial de muestras de datos reales y generar nuevas muestras a partir de esa distribución como se indica en [1]. Estas han sido protagonistas debido a sus diversas aplicaciones, desde imágenes hasta NLP, dado que se pueden utilizar para generar nuevos datos y estudiar muestras y ataques adversarios.

Como se evidencia en [1] y se mencionó anteriormente las GAN suelen constar de un generador y un discriminador que aprenden simultáneamente. El generador intenta capturar la distribución potencial de las muestras reales y genera nuevas muestras de datos. El discriminador suele ser un clasificador binario que discrimina las muestras reales de las generadas con la mayor precisión posible. Tanto el generador como el discriminador pueden adoptar la estructura de las redes neuronales profundas [2, 3].

Actualmente, el campo de la visión por computadora y el procesamiento de imágenes está experimentando un auge significativo en la investigación. Se ha logrado crear imágenes fotorrealistas y generar imágenes de alta definición a partir de imágenes de baja resolución, como se menciona en [1]. La idea principal detrás de las Redes Generativas Adversarias (GAN) proviene del equilibrio de Nash en la teoría de juegos, detallado en [4]. Este enfoque supone la existencia de dos participantes en un juego: un generador y un discriminador. El generador busca aprender la distribución de los datos reales, mientras que el discriminador intenta determinar correctamente si los datos de entrada provienen de los datos reales o han sido generados. Para ganar el juego, ambos

participantes deben mejorar continuamente sus capacidades, el generador en la generación de datos realistas y el discriminador en la detección de datos falsos. El objetivo del proceso de optimización es encontrar un equilibrio de Nash entre los dos participantes, donde ambos alcanzan un punto óptimo en sus respectivas tareas.

Figura 1 *Estructura GAN [1]*



En la Figura 1, tomada de [1], se muestra el procedimiento de cálculo y la estructura GAN. Cabe resaltar que un GAN consiste en dos redes neuronales que compiten entre sí en un juego de suma cero como se mencionó anteriormente. Una red, llamada generador (G), crea datos sintéticos intentando pasarlos como datos reales. La otra red, llamada discriminador (D), intenta distinguir entre datos reales y datos generados por el generador. El generador toma una entrada de ruido aleatorio (z) y la transforma en datos sintéticos (G(z)). El discriminador, por otro lado, toma tanto datos reales (x) como datos generados (G(z)) como entrada y trata de distinguir entre ellos, etiquetando los datos reales como "1" (verdadero) y los datos generados como "0" (falso).

El objetivo del generador es engañar al discriminador para que clasifique los datos generados como reales, mientras que el objetivo del discriminador es ser lo más preciso posible en la clasificación. Ambas redes se entrenan en un proceso iterativo y adversarial: mientras el generador intenta mejorar su capacidad para generar datos que engañen al discriminador, este último intenta mejorar su capacidad para distinguir entre datos reales y generados. Cuando el

entrenamiento avanza, el generador se vuelve más habilidoso en la generación de datos que se asemejan a los reales, mientras que el discriminador se vuelve más hábil en distinguirlos.

2.2. Superresolución

La superresolución de imágenes es un proceso que mejora la calidad y resolución de una imagen de baja resolución. Utilizando modelos generativos profundos como redes generativas adversariales (GANs) y técnicas de refinamiento iterativo, se pueden generar imágenes de alta resolución consistentes con las de baja resolución originales. Este proceso es fundamental en aplicaciones como la vigilancia y la fotografía, donde la claridad y los detalles son esenciales [17].

2.3. SRGAN

Como se menciona en [5], los métodos de superresolución se han centrado en minimizar el error medio cuadrático de reconstrucción. Las estimaciones resultantes tienen una elevada relación señal/ruido, pero carecen de detalles de alta frecuencia y son perceptualmente insatisfactorias porque no alcanzan la fidelidad esperada con una mayor resolución. Por lo tanto, se presenta SRGAN, una red generativa adversarial (GAN) para superresolución de imágenes, siendo el primer marco capaz de inferir imágenes naturales fotorrealistas para 4 x factores de ampliación. Para lograrlo en [5] se propone una función de pérdida perceptiva que consta de una pérdida por adversidad y una pérdida de contenido.

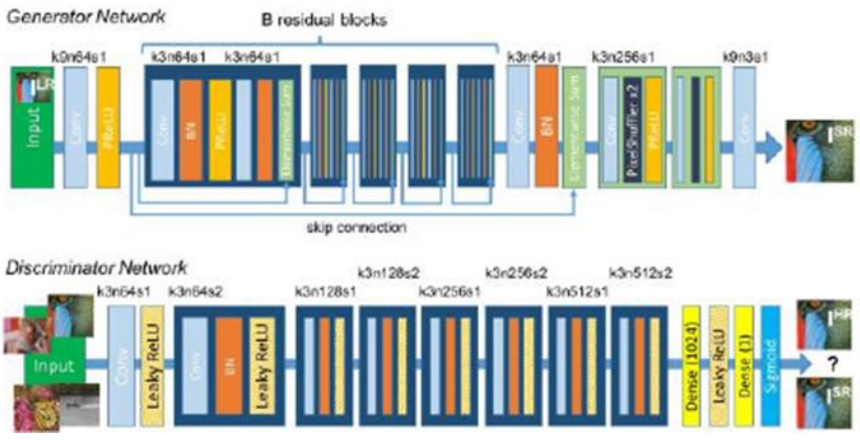
Cabe resaltar que la pérdida adversarial impulsa la solución de generación de imágenes naturales mediante el entrenamiento de una red discriminante para distinguir entre las imágenes generadas y las originales fotorrealistas. Además, se utiliza una pérdida de contenido basada en la similitud en el espacio de píxeles.

La red neuronal profunda es capaz de recuperar texturas fotorrealistas incluso a partir de imágenes fuertemente submuestreadas. Los resultados de la prueba de puntuación media de opinión (MOS) muestran mejoras en la calidad perceptiva con SRGAN. Las puntuaciones MOS obtenidas con SRGAN se acercan más a las de las imágenes de alta resolución que las obtenidas con cualquier otro método más avanzado. Muchos problemas de visión por computadora se han visto influenciados por arquitecturas de redes neuronales convolucionales, especialmente luego del éxito del trabajo de Krizhevsky et al. [6].

Las arquitecturas de redes profundas han demostrado mejorar el rendimiento en Super-Resolución de imagen única. Por ejemplo, como se menciona en [5] Kim et al. [7] presenta una CNN recursiva para alcanzar resultados sobresalientes en este campo. Otra estrategia que se muestra en [5] para facilitar el entrenamiento de redes CNN profundas es el uso de bloques residuales y conexiones saltadas [7, 8, 9].

En [5] se menciona que el aprendizaje de representaciones es beneficioso en términos de precisión y velocidad, lo que mejora significativamente respecto a enfoques anteriores, donde se implementa interpolación cúbica antes de introducir la imagen en la CNN.

Figura 2 *Arquitectura de la red de generadores y discriminadores [5]*



En el núcleo de la red generadora G, como se ilustra en la Figura 2, tomada de [5], se encuentran B bloques residuales con un diseño idéntico. Siguiendo la propuesta de diseño de Gross y Wilber [10], utilizan capas convolucionales de 3x3 con 64 mapas de características, seguidas de capas de normalización por lotes y activación ParametricReLU. Incrementan la resolución de la imagen de entrada con dos capas de convolución sub-píxel, según lo propuesto por Shi et al. [11]. Para distinguir entre imágenes HR reales y muestras SR generadas, entrenan una red discriminadora con ocho capas convolucionales y activación LeakyReLU. Esta red se entrena para resolver un problema de maximización, según lo descrito en el artículo [5], y utiliza convoluciones con paso para reducir la resolución de la imagen. Las características resultantes se procesan con dos capas densas y una función de activación sigmoide para obtener la probabilidad de clasificación.

2.4. ESRGAN

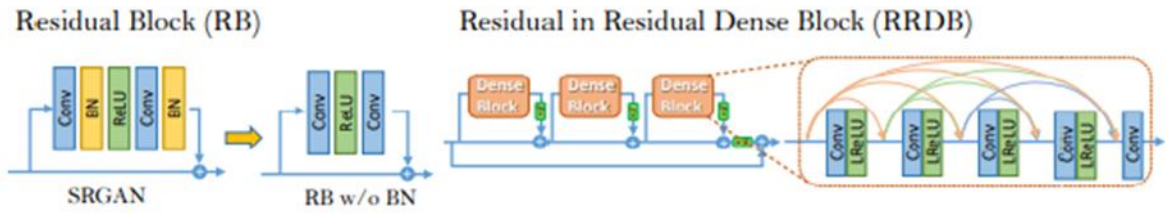
El ESRGAN, mencionado en el documento [12], representa una evolución significativa del SRGAN. Este último, previamente reconocido como un hito en la generación de texturas realistas durante la superresolución de imágenes individuales, exhibía limitaciones en la presencia de artefactos no deseados junto con los detalles enriquecidos. Para superar estas deficiencias, se llevó a cabo un análisis exhaustivo de tres componentes clave del SRGAN: su arquitectura de red, la pérdida por adversarios y la pérdida perceptiva.

Esta arquitectura introduce mejoras significativas en cada componente. En primer lugar, se adopta el bloque denso residual en residual (RRDB) como la estructura básica de la red, eliminando la normalización por lotes para una mayor eficacia.

Además, se implementa la noción de GAN relativista para que el discriminador evalúe la realidad relativa en lugar del valor absoluto. Finalmente, se perfecciona la pérdida perceptual utilizando características previas a la activación, lo que ofrece una supervisión más robusta para la consistencia del brillo y la recuperación de texturas.

Como resultado de estas mejoras, el ESRGAN exhibe una calidad visual mejorada, con texturas más realistas y naturales que el SRGAN. Estas mejoras le valieron el primer lugar en el PIRM2018-SR Challenge [12].

Figura 3 *Izquierda: Eliminación bloque residual SRGAN. Derecha: Uso de RRDB [12]*



Como se puede ver, en la Figura 3, tomada de [12] se muestra en la parte izquierda la eliminación de capas BN, esto ha demostrado aumentar el rendimiento y reducir la complejidad computacional en diferentes tareas incluyendo superresolución y desenfoque. Las capas BN normalizan las características usando la media y la varianza durante el entrenamiento, por lo tanto, cuando las estadísticas de los conjuntos de datos de entrenamiento y de prueba difieren mucho, las capas BN tienden a introducir artefactos desagradables y limitan la capacidad de generalización, por lo tanto, al eliminar las capas BN se puede conseguir un entrenamiento más estable y un rendimiento consistente.

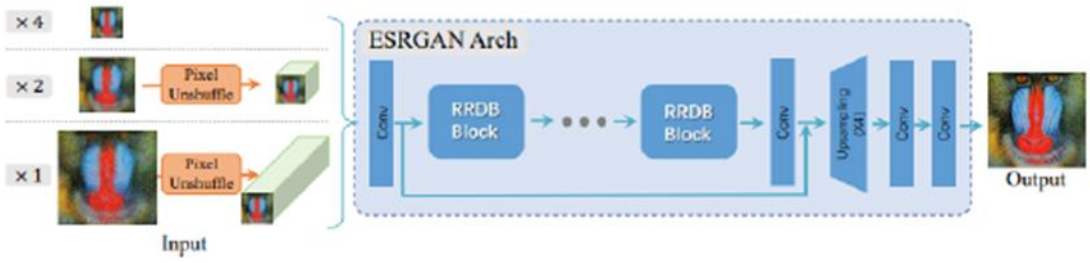
El diseño de la arquitectura se mantiene igual al de la SRGAN y se hace uso de un nuevo bloque básico, la RRDB, como se ve en la Figura 3. En [12] se realiza hincapié que el RRDB

propuesto emplea una estructura más profunda y compleja que el bloque residual original de SRGAN. La RRDB propuesta tiene una estructura residual en residual, donde el aprendizaje residual se utiliza en diferentes niveles.

2.5. Real – ESRGAN

Como se menciona en [13] en Real – ESRGAN se extiende el trabajo de ESRGAN a una aplicación práctica de restauración, que se entrena con datos sintéticos puros, donde se emplea un discriminador U-Net con normalización espectral para aumentar la capacidad del discriminador y estabilizar la dinámica de entrenamiento. Cabe resaltar que al tener un espacio de degradación mayor que el ESRGAN el entrenamiento se convierte en un reto, pues el discriminador requiere una capacidad más potente para discriminar la realidad a partir de resultados de entrenamiento complejo, mientras que la retroalimentación de gradiente del discriminador debe ser más precisa para mejorar los detalles locales. Además, la estructura U-Net y las degradaciones también aumentan la inestabilidad del entrenamiento. Por lo tanto, se emplea una normalización espectral.

Figura 4 *Real-ESRGAN adopta la misma red de generadores que la ESRGAN [13]*



Por otro lado, se adapta el mismo generador del ESRGAN, como se puede ver en la Figura 4, tomada de [13], donde se amplía la arquitectura ESRGAN x4 original para realizar superresolución con un factor de escala x2 y x1. Tal como se indica en [13], dado el objetivo ampliado de Real-ESRGAN para abordar una degradación más significativa que ESRGAN, se ha

encontrado que el diseño original del discriminador ya no es óptimo. Específicamente, el discriminador en Real- ESRGAN necesita tener una capacidad discriminativa mayor para manejar salidas de entrenamiento complejas. En lugar de simplemente discriminar estilos globales, también debe ser capaz de generar gradientes precisos para texturas locales.

2.6. Compresión de imágenes

La compresión de imágenes es crucial en el proyecto de superresolución de imágenes, ya que permite manejar eficientemente los datos visuales, reduciendo el tamaño de los archivos sin sacrificar significativamente la calidad de la imagen. Esto es especialmente importante en aplicaciones donde el almacenamiento y el ancho de banda son limitados, como en sistemas embebidos y transmisiones en tiempo real. A continuación, se detallan los principales formatos de compresión de imágenes utilizados en este proyecto: JPEG, PNG y WebP, explicando sus características y métodos de compresión.

2.6.1. Formato JPG

El formato JPG es uno de los formatos de compresión de imágenes más utilizados debido a su capacidad para reducir significativamente el tamaño de los archivos de imagen con una pérdida aceptable de calidad. JPG utiliza un esquema de compresión con pérdida que aprovecha las limitaciones de la percepción visual humana, eliminando información de la imagen que es menos perceptible para el ojo humano. La compresión JPG es ajustable, permitiendo un equilibrio entre la calidad de la imagen y el tamaño del archivo.

La compresión JPG se basa en la transformada discreta del coseno (DCT) que convierte bloques de píxeles en el dominio de la frecuencia, lo que permite la reducción de componentes de alta frecuencia que son menos visibles. A pesar de su eficacia, JPG introduce artefactos de

compresión conocidos como "bloques" y "anillos", especialmente a niveles de compresión altos [14].

2.6.2. Formato PNG

El formato PNG (Portable Network Graphics) se desarrolló como una mejora sobre el formato GIF y está diseñado para la compresión sin pérdida, lo que significa que no hay pérdida de calidad entre la imagen original y la comprimida. PNG utiliza la compresión DEFLATE, que es una combinación de las técnicas LZ77 y Huffman, permitiendo una alta tasa de compresión sin pérdida de información [15].

PNG es ideal para imágenes que requieren mantener la calidad original, como gráficos con texto, logotipos y dibujos técnicos. A diferencia de JPEG, PNG soporta transparencia a través de un canal alfa, lo que permite la creación de imágenes con áreas transparentes. Sin embargo, el tamaño de los archivos PNG puede ser significativamente mayor en comparación con JPEG, especialmente para fotografías y imágenes con gradientes suaves [15].

2.6.3. Formato WEBP

WebP es un formato de imagen desarrollado por Google que admite tanto compresión con pérdida como sin pérdida. WebP con pérdida utiliza compresión predictiva, similar a la utilizada por el códec de video VP8, donde se predice el valor de los píxeles basándose en los valores de los píxeles vecinos y solo se almacenan las diferencias. WebP sin pérdida utiliza técnicas avanzadas como la codificación de longitud de ejecución y la transformación de color para reducir el tamaño del archivo [16].

Las ventajas del formato WebP incluyen tamaños de archivo más pequeños en comparación con JPEG y PNG, sin comprometer significativamente la calidad de la imagen. WebP también

soporta transparencia y animaciones, haciendo de este formato una opción versátil para su uso en la web. Según Google, las imágenes WebP sin pérdida son aproximadamente un 26% más pequeñas que sus equivalentes en PNG, y las imágenes WebP con pérdida son entre un 25% y un 34% más pequeñas que las imágenes JPEG de calidad comparable [16].

3. Metodología

El presente trabajo se centró en abordar la problemática de la identificación de placas vehiculares en imágenes de baja resolución capturadas por cámaras de seguridad no especializadas. Esta limitación obstaculizaba la identificación efectiva de las placas vehiculares, afectando tanto la seguridad ciudadana como la eficiencia en el control del tráfico. Para solucionar este problema, se propuso la implementación de técnicas de superresolución basadas en arquitecturas GAN (Redes Generativas Adversarias). El objetivo principal fue mejorar la calidad de las imágenes y facilitar su posterior procesamiento para el reconocimiento de placas vehiculares.

La metodología del proyecto se estructuró en tres componentes fundamentales: la creación de una base de datos robusta, la exploración del espacio de diseño (Design Space Exploration) para identificar la mejor arquitectura GAN, y la implementación de un sistema completo que integrara estas tecnologías para mejorar la resolución de las imágenes captadas por cámaras de seguridad.

Primero, se construyó una base de datos utilizando imágenes de alta calidad provenientes del conjunto "license ocr" de Roboflow. Estas imágenes, originalmente de 640 x 640 píxeles, se redujeron a resoluciones específicas de 128 x 128 y 32 x 32 píxeles para adaptar las entradas y salidas requeridas por las arquitecturas GAN.

En la fase de exploración del espacio de diseño, se evaluaron tres arquitecturas GAN: SRGAN, ESRGAN y Real-ESRGAN. Cada arquitectura se entrenó y evaluó utilizando tres formatos de imagen (JPG, PNG y WEBP) en cinco resoluciones (16x16, 32x32, 64x64, 96x96 y 128x128) y 4 niveles de compresión 0%, 10%, 50% y 90% para JPG y WEBP, en el caso de PNG

0%, 1%, 3%, 6%. El objetivo de esta etapa fue determinar cómo cada combinación de formato, resolución y compresión afectaba el rendimiento de las arquitecturas de superresolución.

Finalmente, se implementó un sistema completo en una Single Board Computer (SBC) VIM3 de Khadas, utilizando una webcam Allink 550 1080p para capturar video. El sistema incluyó varias etapas: captura de video, detección de vehículos y placas, división y unificación de imágenes, aplicación de superresolución, OCR y mejora de contraste. Esta implementación demostró la viabilidad de aplicar estas técnicas en entornos reales, mejorando significativamente la calidad de las imágenes y facilitando la identificación de las placas vehiculares.

3.1. Base de datos

Para el desarrollo de este proyecto, se creó una base de datos de imágenes de alta y baja resolución utilizando el conjunto "license ocr" de Roboflow. Inicialmente, las imágenes en la base de datos de Roboflow tenían una resolución de 640 x 640 píxeles. Estas imágenes se redujeron para adaptarse al diseño de la arquitectura GAN, resultando en imágenes de alta resolución con dimensiones de 128 x 128 píxeles y de baja resolución con dimensiones de 32 x 32 píxeles. La base de datos incluyó un total de 14,393 imágenes de alta resolución y 14,393 imágenes de baja resolución. Las imágenes de baja resolución se generaron a partir de las imágenes de alta resolución.

Las imágenes se dividieron en tres conjuntos: entrenamiento, validación y prueba, con una distribución de 94.5% para entrenamiento, 3.4% para prueba y 2.1% para validación. Es importante señalar que la base de datos no estaba etiquetada inicialmente. Para abordar esto, se llevó a cabo un proceso de etiquetado utilizando un modelo de detección de objetos con YOLOv8.

Este modelo se entrenó con las mismas imágenes y se encargó de etiquetar cada carácter de las placas, incluyendo los números del 0 al 9 y las letras de la “a” a la “z”.

Figura 5 *Etiquetado de placas*

Como se observa en la Figura 5, así fue como se etiquetaron las placas, donde la etiqueta de cada placa se agregó a un archivo .txt con el nombre de la placa, es decir, nombre.jpg – nombre.txt, con el fin de tener un registro de cada placa.

3.2. Design space exploration

La exploración del espacio de diseño se enfocó en evaluar tres arquitecturas GAN diferentes: SRGAN, ESRGAN y Real-ESRGAN. Cada una de estas arquitecturas fue entrenada utilizando tres formatos de imagen (JPG, PNG y WEBP). Esto con el fin de que, al momento de evaluar imágenes con un formato determinado, se realice superresolución con el modelo entrenado para ese formato en específico y tener un análisis más detallado y limpio.

Se utilizaron cinco resoluciones de entrada: 16x16, 32x32, 64x64, 96x96 y 128x128 píxeles, y cuatro niveles de compresión: cero, baja, media y alta. Para JPG y WEBP, los niveles de compresión fueron 0%, 10%, 50% y 90%, mientras que para PNG fueron 0%, 1%, 3% y 6%. Este enfoque permitió evaluar el comportamiento de las arquitecturas ante variaciones en la resolución y la compresión, proporcionando un análisis exhaustivo de su rendimiento.

Las imágenes de entrada se dividieron en subimágenes de 32 x 32 píxeles para que pudieran ser procesadas por las arquitecturas GAN. La Tabla 1 muestra cómo se realizaron estas divisiones.

Tabla 1 *Resoluciones*

Resolución de entrada	**División de imagen**	**Resolución de salida**
16 x 16	No hay división (se escala a 32 x 32)	128 x 128
32 x 32	No hay división (escala de 1)	128 x 128
64 x 64	4 de 32 x 32	256 x 256
96 x 96	9 de 32 x 32	384 x 384
128 x 128	16 de 32 x 32	512 x 512

Para las imágenes de 16 x 16, se completaron los píxeles con una banda blanca para alcanzar los 32 x 32 píxeles necesarios. Posteriormente, las subimágenes fueron procesadas y unidas nuevamente para obtener las imágenes de superresolución.

Esta fase de exploración fue crucial para determinar la eficacia de cada arquitectura y formato en mejorar la resolución de las imágenes bajo diferentes condiciones de entrada,

permitiendo así una selección informada de las mejores configuraciones para la implementación final del sistema.

3.2.1. Entrenamiento

El entrenamiento se realizó en el servidor del grupo de investigación SISTEMIC del departamento de ingeniería electrónica y telecomunicaciones, donde se tuvo acceso a GPUs de 12GB y 24GB para el entrenamiento. En el proceso de entrenamiento se consideraron dos aspectos clave: la arquitectura y el formato de las imágenes de entrenamiento. Se utilizaron tres arquitecturas diferentes: SRGAN, ESRGAN y Real-ESRGAN, donde, para cada una de estas arquitecturas, se entrenaron modelos utilizando tres formatos de imagen distintos: JPG, PNG y WEBP. Para las arquitecturas SRGAN y ESRGAN se utilizó una GPU de 12GB y para Real-ESRGAN, como es más robusta, se usó una de 24GB. Para cada resolución y arquitectura se entrenaron 100 épocas, es decir, 9 entrenamientos.

El entrenamiento se realizó utilizando el lenguaje de programación Python. Para SRGAN se utilizó la librería Keras, mientras que para ESRGAN y Real-ESRGAN se empleó PyTorch. Estas librerías fueron seleccionadas por su popularidad en el desarrollo de modelos de aprendizaje profundo.

3.2.2. SRGAN

Se tuvieron 3 puntos clave, la red generadora, discriminadora y la VGG19. La generadora se construyó con bloques residuales y de escalado, que incluyeron capas de convolución, normalización por lotes y activaciones PRelu. Estos bloques transformaron las imágenes de baja resolución en imágenes de alta resolución. Por otro lado, la red discriminadora se diseñó para distinguir entre imágenes generadas y reales, utilizando capas de convolución, utilizando capas de

convolución con activaciones LeakyReLU y normalización por lotes. La salida fue una neurona con activación sigmoide.

El modelo combinado integró la red generadora, la discriminadora y una red VGG19 preentrenada para calcular la pérdida perceptual. En los parámetros de entrenamiento se utilizó un batch size de 1, se realizaron 100 épocas de entrenamiento, optimizador Adam y como función de pérdida para la discriminadora 'Entropía cruzada' y para el modelo combinado la anterior con 'Error cuadrático medio'. La discriminadora se entrenó primero con imágenes reales y generadas, y luego la generadora se entrenó manteniendo fija la discriminadora.

3.2.3. ESRGAN

La arquitectura ESRGAN utilizó bloques residuales densamente conectados en su red generadora. Estos bloques combinaban las ventajas de los bloques residuales y las conexiones densas para mejorar la capacidad de la red para capturar características finas y detalles en las imágenes de alta resolución.

El discriminador adoptó una arquitectura de GAN relativista, que no solo distinguía entre imágenes reales y generadas, sino que también evaluaba la relatividad entre las predicciones de imágenes reales y falsas. Además, se implementó una extracción de características basada en VGG para calcular la pérdida de contenido, lo que mejoró la calidad visual de las imágenes generadas.

Como parámetros de entrenamiento se tuvo una tasa de aprendizaje de 0.0002, un batch size de 4, un optimizador Adam con betas (0.9, 0.999) y 3 funciones: pérdida adversarial (BCEWithLogitsLoss), de contenido (L1Loss) y píxeles (L1Loss). Esta se entrenó durante 100 épocas y se empleó un enfoque de 'warm-up' durante los primeros 500 lotes, donde solo se considera la pérdida de píxeles para estabilizar el entrenamiento inicial del generador.

3.2.4. Real-ESRGAN

Se basa en la arquitectura ESRGAN, específicamente debido a que se usó bloques residuales densamente conectados para transformar imágenes de baja resolución en versiones de alta resolución.

Para el entrenamiento se aplicaron diversas técnicas para mejorar la variabilidad y calidad de las imágenes generadas. Esto incluyó la introducción de ruido gaussiano y poissoniano, así como el uso de diferentes filtros y métodos de escalado como área, bilineal y bicúbico.

El discriminador que se empleó fue una red convolucional estándar, entrenada para distinguir entre imágenes reales de alta resolución y aquellas generadas por el modelo. Se utilizaron dos optimizadores: Adam con una tasa de aprendizaje de 0.0001 y betas de (0.9, 0.99) para los parámetros del generador y discriminador, respectivamente.

Para cada iteración, se calcularon y registraron varias pérdidas. Esto incluyó la pérdida de píxeles entre las imágenes generadas y las reales de alta resolución, la pérdida perceptual y la perdida adversarial para entrenar tanto el generador como el discriminador.

Paramétricamente hablando se configuró el entrenamiento para 100 épocas, donde se aplicaron transformaciones y ruidos aleatorios a las imágenes de baja resolución para mejorar la generalización del modelo.

3.2.5. Métricas

Para evaluar el rendimiento de las imágenes de superresolución generadas por las arquitecturas GAN, se emplearon dos métricas principales: precisión por caracteres y precisión por placas completas. Estas métricas se calcularon utilizando un conjunto de datos de prueba

compuesto por 489 imágenes, las cuales fueron etiquetadas manualmente para asegurar la exactitud de la comparación.

- Precisión por Caracteres: Esta métrica evaluó la exactitud con la que el OCR reconoció cada carácter en las imágenes de superresolución en comparación con las etiquetas manuales de las imágenes originales. Para realizar esta evaluación, se utilizó el modelo YOLOv8, que permitió etiquetar de manera precisa los caracteres de las placas en las imágenes de superresolución. Las etiquetas se almacenaron en archivos .txt con nombres correspondientes a cada imagen. Se compararon los archivos .txt de las etiquetas originales con los de las etiquetas generadas por el modelo para las imágenes de superresolución. La precisión por caracteres se calculó dividiendo el número de caracteres correctamente reconocidos por el número total de caracteres en las etiquetas base y multiplicando el resultado por 100.
- Precisión por Placas Completas: Debido a que la precisión por caracteres no es suficiente para medir el rendimiento global del sistema, también se utilizó la métrica de precisión por placas completas. Esta métrica evaluó la exactitud con la que el OCR reconoció las placas completas en las imágenes de superresolución en comparación con las imágenes originales etiquetadas manualmente. La precisión por placas completas se calculó dividiendo el número de placas correctamente reconocidas por el número total de placas en el conjunto de prueba y multiplicando el resultado por 100. Esta métrica fue crucial para evaluar la capacidad del sistema de superresolución para recuperar imágenes de baja resolución y acercarlas a la calidad de las imágenes originales.

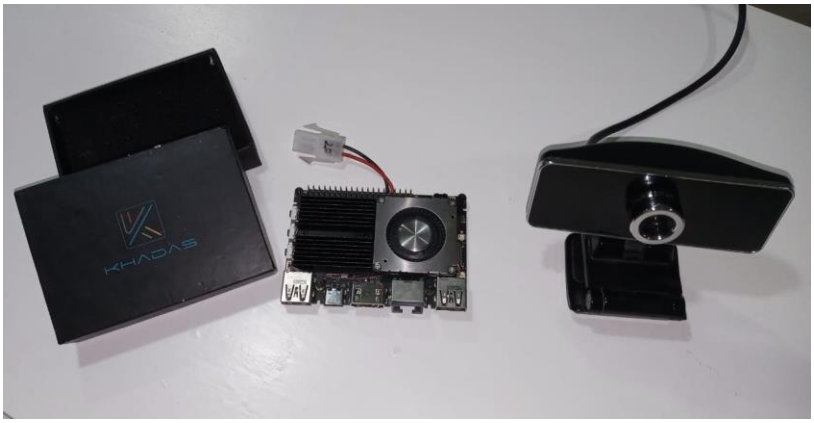
Para asegurar una evaluación justa, se compararon las etiquetas de las imágenes originales de prueba, etiquetadas manualmente, con las etiquetas generadas por el modelo YOLOv8 para las

imágenes de superresolución. El modelo YOLOv8 permitió identificar y extraer de manera precisa los caracteres de las placas en las imágenes de superresolución. Esta comparación entre etiquetas manuales y etiquetas generadas por el modelo permitió observar la capacidad de superresolución de las arquitecturas GAN al partir de imágenes en baja resolución.

3.3. Implementación

Para el desarrollo completo se implementó el sistema de superresolución en una SBC (Single Board Computer), específicamente en una VIM3 de Khadas, y se utilizó una webcam Allink 550 1080p como se puede observar en la Figura 6.

Figura 6 *VIM3 y WebCam*



La arquitectura que se diseñó para implementar el sistema consta de un script, el cual, tiene 6 pasos clave, captura de video con gstreamer (hardware encoder), captura de vehículo y placa (YOLOv8), algoritmo de división de imagen, algoritmo de superresolución, algoritmo de juntar las imágenes, mejora de contraste y algoritmo OCR.

3.3.1. Captura de video

La captura de video se realizó a partir de Gstreamer, siendo esta una herramienta poderosa para la manipulación de flujos de medios, por tal motivo se implementa una tubería específica que realiza varias tareas en secuencia.

Inicialmente, en la tubería se implementó el plugin v4l2src para capturar el video desde el dispositivo '/dev/video0', que era la cámara principal del sistema. Este flujo de video se configuró con un formato JPEG, una resolución de 1280 x 720 píxeles y 25 FPS. El flujo JPEG fue decodificado con 'jpegdec' para que pudiera ser procesado en etapas posteriores. Posteriormente, el flujo de video decodificado se codificó utilizando el codificador de hardware 'amlvenc', reduciendo así la carga sobre la CPU. Como se requirió una salida de video en formato MP4, el video codificado se multiplexó en un contenedor MP4 con 'mp4mux', y finalmente, se guardó el video utilizando 'filesink'.

3.3.2. Captura de vehículo y placa

Para la detección de vehículos y placas se utilizó YOLOv8, siendo fundamental para el reconocimiento efectivo en el sistema. Se emplearon dos modelos: uno para el reconocimiento de vehículos y otro para el reconocimiento de placas vehiculares. El modelo de placas fue entrenado de manera independiente en el servidor de SISTEMIC con una GPU de 12GB, mientras que el modelo de vehículos fue un modelo preentrenado proporcionado por Ultralytics, conocido como yolov8x.pt. Este archivo es un modelo preentrenado de la versión YOLOv8, desarrollado por Ultralytics, utilizado para la detección de objetos. La extensión .pt indica que es un modelo de PyTorch, una biblioteca de aprendizaje profundo utilizada para el desarrollo y entrenamiento de

modelos de inteligencia artificial. Ambos modelos fueron implementados en la versión nano debido a las limitaciones de recursos del sistema embebido.

Inicialmente, se procesó el video capturado en la sección de captura de video fotograma por fotograma. Se estableció un intervalo de procesamiento de fotogramas para asegurar la detección continua de vehículos.

Para el seguimiento de los objetos detectados se empleó la librería ByTracker, que facilitó el seguimiento de los vehículos a lo largo de los diferentes fotogramas del video.

El algoritmo consistió en dos etapas principales de detección utilizando los modelos entrenados:

- Modelo 1: Este modelo, preentrenado por Ultralytics y denominado yolov8x.pt, se utilizó para identificar vehículos en el video. Se extrajeron recortes de los vehículos detectados junto con sus coordenadas y niveles de confianza asociados.
- Modelo 2: Cada recorte obtenido del Modelo 1 se pasó a este segundo modelo, entrenado específicamente para la detección de placas. Los recortes se redimensionaron según las dimensiones actuales del recorte del vehículo obtenidas después de la detección con el Modelo 1. Los tamaños predefinidos utilizados fueron: 16x16, 32x32, 64x64, 96x96 y 128x128. Para cada tamaño de este conjunto se comparó con las dimensiones actuales de la salida del Modelo 1, seleccionando el tamaño más grande de la lista que fuera igual o mayor que las dimensiones actuales del recorte. Esto aseguró que el recorte se redimensionara al tamaño más cercano, pero no menor que sus dimensiones originales.

Durante la ejecución del algoritmo, se guardaron imágenes de fotogramas completos donde se detectaron vehículos, utilizando un nombre de archivo único basado en el tiempo para cada captura exitosa.

3.3.3. Algoritmo división de imagen

Se utilizó un algoritmo desarrollado para dividir las imágenes de una resolución mayor a 32 x 32, con el fin de que se adaptara al "input" de la arquitectura GAN.

Inicialmente, se definió una función para dividir la imagen en fragmentos más pequeños, donde se tomó la imagen y el tamaño que retornó el algoritmo de captura de vehículo y placa, dentro de esta se obtuvo la altura y el ancho de la imagen original, donde luego se iteró sobre la imagen, extrayendo fragmentos de las dimensiones especificadas y agregándolos a la lista. Posteriormente, se definió una función que recorrió todas las carpetas dentro de la carpeta de entrada y procesó las imágenes contenidas en las subcarpetas. Cabe resaltar que para cada subcarpeta se creó una carpeta correspondiente en la ruta de salida.

Finalmente, el script recorrió la estructura de carpetas, procesó las imágenes y guardó los fragmentos resultantes en las ubicaciones correspondiente.

3.3.4. Superresolución

En este punto se recibe la imagen fragmentada en imágenes de 32 x 32, por lo tanto, se realizó un algoritmo que tomó la carpeta donde están dichas imágenes y se le aplicó el modelo que resultó de los entrenamientos de superresolución, para ello se cargó el modelo generador entrenado y se usó para convertir imágenes de baja resolución en versiones de alta resolución, donde se recorrió recursivamente los archivos de la carpeta de entrada, posteriormente se cargó la imagen

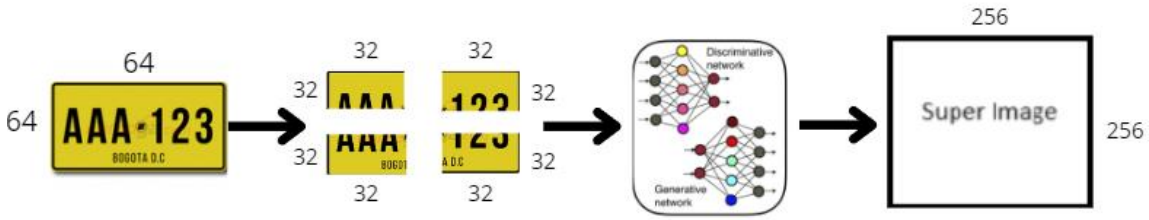
de baja resolución y se convirtió en formato RGB y se normalizó, finalmente, se aplicó el modelo, se normalizó la imagen generada, se convirtió a escala de 0 a 255 y se guardó.

La salida de este paso del sistema fue una carpeta donde los tramos de 32 x 32 salieron de 128 x 128.

3.3.5. Unificar imágenes

La salida de este paso del sistema fue una carpeta donde los tramos de 32 x 32 salieron de 128 x 128. Por consiguiente, se desarrolló un algoritmo que juntó las imágenes de 128 x 128 en el orden correspondiente, es decir, si la imagen inicial fue de 64 x 64, se dividió con 4 de 32 x 32, y posteriormente del algoritmo de superresolución salieron 4 de 128 x 128, dando como resultado una imagen de 256 x 256, como se puede observar en la Figura 7.

Figura 7 *Sistema Superresolución*



En la Figura 7 se puede observar con detalle como fue el procedimiento para integrar resoluciones iniciales que no fueran 32 x 32, donde se divide la imagen antes de ingresar a la arquitectura GAN y se juntan luego de salir de la arquitectura, obteniendo como resultado una imagen de una resolución más alta que 128 x 128.

3.3.6. Mejora de contraste

Para mejorar el rendimiento de la imagen de superresolución se implementó gimp, este software mejoró la imagen a partir de la variación del contraste, donde inicialmente se escribió un script en Scheme que ajustó el contraste de la imagen, luego se guardó este en el directorio de scripts de gimp, y finalmente se ejecuta dicho script y se obtuvo la imagen con la variación de contraste.

3.3.7. OCR

Para el OCR se implementó el mismo modelo YOLOv8 que se usó para etiquetar imágenes, acá se evaluó el rendimiento del sistema completo, con las mismas métricas establecidas: Precisión caracteres correctos y precisión placas completas.

3.3.8. Limitaciones técnicas

En la implementación del sistema de superresolución e identificación de placas vehiculares, no se utilizó la Unidad de Procesamiento Neuronal (NPU) de la VIM3 ni se aplicaron técnicas de cuantización y optimización de modelos. Como resultado, todos los procesos de inferencia se llevaron a cabo utilizando únicamente las capacidades estándar de la CPU, lo que impactó en el tiempo de inferencia del sistema.

3.3.9. Condiciones experimentales

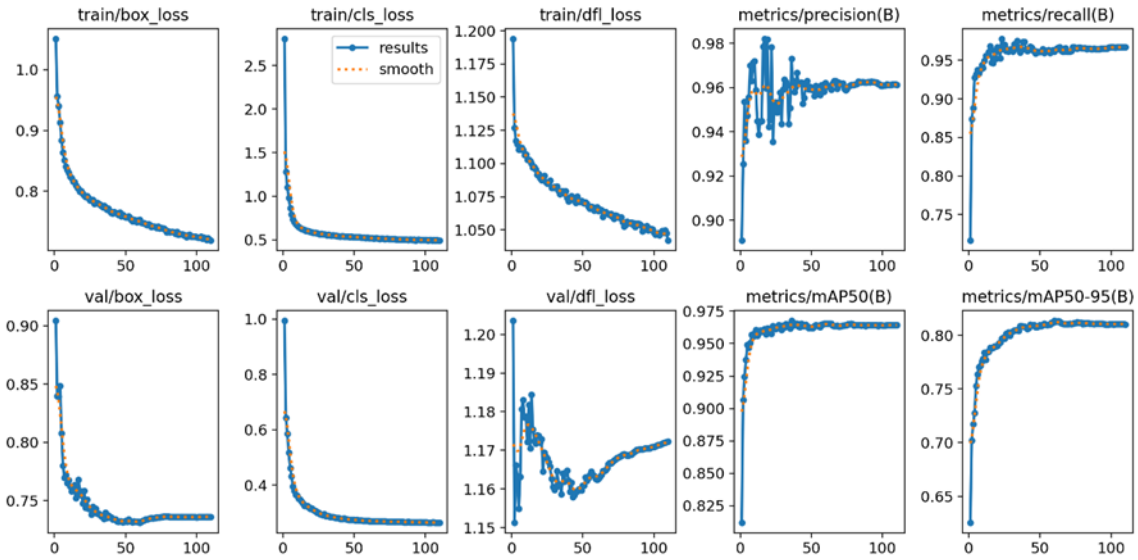
El sistema se probó en un ambiente controlado, con la SBC VIM3 y la WebCam instaladas a una altura de 6 metros. Los experimentos se realizaron bajo diferentes condiciones ambientales: de día, de noche, en días soleados y en días nublados. Esta configuración permitió evaluar la robustez y eficacia del sistema en diversos escenarios. En este análisis se aplicó el modelo con mejor desempeño.

4. Análisis y resultados

4.1. Entrenamiento OCR

Como se mencionó anteriormente se utilizó un modelo YOLOv8 para realizar el etiquetado, el cual, fue parte fundamental para obtener las métricas de precisión mencionadas con anterioridad, se obtuvieron los resultados mostrados en la Figura 8 al momento de entrenar, cabe resaltar que se entrenó en una GPU de 12 GB.

Figura 8 *Métricas de entrenamiento*



La Figura 8 presenta gráficos que detallan el rendimiento del modelo de reconocimiento de caracteres de placas de vehículos durante el entrenamiento y la validación. En la primera fila, se observan las métricas de pérdidas. El gráfico “train/box_loss” muestra una disminución constante en la pérdida de las cajas delimitadoras, indicando una mejora en la precisión del modelo. El gráfico “train/cls_loss” refleja una tendencia decreciente similar en la pérdida de clasificación, sugiriendo que el modelo está aprendiendo a clasificar correctamente los caracteres. La pérdida

del modelo DFL, representada en el gráfico “train/dfl_loss”, también disminuye gradualmente, indicando mejoras en la precisión de las predicciones.

La segunda fila se enfoca en las métricas de validación. El gráfico “val/box_loss” muestra una disminución en la pérdida de las cajas delimitadoras en el conjunto de validación, aunque comienza ligeramente más alta que en el entrenamiento. El gráfico “val/cls_loss” presenta una tendencia similar a la del entrenamiento, con algunas fluctuaciones típicas de la validación. El gráfico “val/dfl_loss” muestra mayores fluctuaciones en la pérdida del modelo DFL, posiblemente debido a la diversidad de los datos de validación.

Los gráficos en la primera fila evalúan las métricas de precisión y recall del modelo. El gráfico “metrics/precision(B)” muestra una mejora significativa en la precisión, estabilizándose en un nivel alto del 96%, indicando un buen equilibrio entre precisión y recall. El gráfico “metrics/recall(B)” refleja una tendencia similar, sugiriendo que el modelo captura una alta proporción de verdaderos positivos. El gráfico “metrics/mAP50(b)” muestra el Mean Average Precision (mAP) con un umbral de IoU de 0.5, que se estabiliza cerca del 95%, indicando alta precisión en la detección de objetos. Finalmente, el gráfico “metrics/mAP50-95(B)” muestra el mAP en un rango de umbrales de IoU (0.5 a 0.95), con una tendencia similar, pero valores más bajos debido al rango de evaluación más estricto.

4.2. Entrenamiento SRGAN

Para SRGAN se entrenó para formatos JPG, PNG y WEBP, los modelos que se obtuvieron fueron los siguientes, mostrados en la Tabla 2.

Tabla 2 *Tamaño modelos SRGAN*

Modelo (100 épocas)	Tamaño
SRGAN_JPG	8.339 KB
SRGAN_PNG	8.339 KB
SRGAN_WEBP	8.339 KB

Como se observa en la Tabla 2 , se realizó un entrenamiento para cada formato, donde se ve que no hay una variación alta en el tamaño del modelo, lo cual es importante, debido a que el sistema final será una implementación en un sistema embebido, entonces conviene rotundamente que no tenga mucho peso dadas las limitaciones en memoria que tienen los sistemas embebidos en este caso la VIM3, estos entrenamientos se realizaron con el fin de evaluar el comportamiento de SRGAN en dichos formatos, para ello, cada formato se analizó con diferentes niveles de compresión con el objetivo de mirar que tanto varía su precisión con estos niveles.

4.3. Entrenamiento ESRGAN

Para ESRGAN se entrenó para formatos JPG, PNG y WEBP, los modelos que se obtuvieron fueron los siguientes, mostrados en la Tabla 3.

Tabla 3 *Tamaño modelos ESRGAN*

Modelo (100 épocas)	Tamaño
ESRGAN_JPG	150.873 KB
ESRGAN_PNG	150.873 KB
ESRGAN_WEBP	150.873 KB

Como se observa en la Tabla 3, se observa que los tamaños de los modelos para cada formato son del mismo tamaño. Cabe resaltar que son aproximadamente 19 veces más pesados que los modelos de SRGAN, esto es coherente debido a que ESRGAN es una arquitectura más robusta que SRGAN, por lo tanto, se toma más tiempo en cuestión de entrenamiento. Este dato es factor clave, debido a que finalmente la idea es la implementación final de un sistema desplegado en una Single Board Computer, lo que es un sistema embebido.

4.4. Entrenamiento Real-ESRGAN

En el contexto de la superresolución de placas de vehículos, el modelo Real-ESRGAN no logró obtener resultados satisfactorios tras 100 épocas de entrenamiento. Esto puede atribuirse a varias razones relacionadas con la complejidad y robustez de su arquitectura.

Real-ESRGAN se basa en la arquitectura ESRGAN, pero introduce mejoras adicionales que aumentan su robustez. Estas mejoras incluyen el uso de redes generativas y discriminatorias más avanzadas, diseñadas para mejorar la calidad de las imágenes generadas, pero también incrementan la complejidad del modelo. La red generativa, en particular, suele tener una estructura más profunda con más capas convolucionales. Aunque permite capturar más detalles, también requiere un tiempo de entrenamiento considerablemente mayor.

Además, Real-ESRGAN utiliza pérdidas perceptuales y de emparejamiento de características para mejorar la calidad visual de las imágenes generadas. Estas pérdidas son más complejas de optimizar y pueden necesitar más épocas de entrenamiento para alcanzar una buena convergencia. La arquitectura también emplea técnicas avanzadas de regularización y normalización, como la regularización de pesos y la normalización de batch, que son cruciales para la estabilidad del entrenamiento, pero pueden hacer que la convergencia sea más lenta.

El hecho de no alcanzar la convergencia con 100 épocas puede deberse a razones específicas de la arquitectura y del proceso del entrenamiento. Primero, dada la complejidad del modelo, 100 épocas pueden no ser suficientes para que el modelo aprenda las características detalladas de las placas de vehículos. Modelos más robustos como Real-ESRGAN requieren más iteraciones para ajustar los parámetros y minimizar las pérdidas adecuadamente. Segundo, durante el entrenamiento, Real-ESRGAN puede necesitar un mayor poder de procesamiento y memoria, pues inicialmente se probó con la GPU de 12GB y no fue posible entrenar, posteriormente con una de 24 GB, con este si fue posible, sin embargo, es probable que este limitado aún y resulta en un entrenamiento poco eficiente. Tercero, la selección de optimizadores y tasas de aprendizaje adecuados es crítica. En modelos complejos, tasas de aprendizaje demasiado altas pueden causar inestabilidad, mientras que tasas muy bajas pueden ralentizar el proceso de convergencia significativamente.

La intención de implementar este modelo en una Single Board Computer (SBC) introduce limitaciones adicionales, Las SBCs, como en este caso la VIM3, tienen capacidades de procesamiento y memoria limitadas. Un modelo tan robusto y pesado como Real-ESRGAN puede no ser práctico para este entorno debido al alto consumo de recursos y el tiempo de procesamiento requerido. Integrar un modelo que no haya sido completamente optimizado y que, además, no haya alcanzado resultados satisfactorios en un entorno de entrenamiento controlado, no sería eficiente ni viable para aplicaciones en una SBC.

A pesar de estas limitaciones, es importante señalar que, si se dispusiera de más tiempo de entrenamiento, es muy probable que Real-ESRGAN pudiera alcanzar buenos resultados. Sin embargo, debido a las restricciones impuestas por el uso de una SBC, es necesario priorizar modelos más ligeros y funcionales y tiempo de entrenamiento. Por lo tanto, aunque Real-ESRGAN

tiene un gran potencial, en este caso específico, modelos como SRGAN o ESRGAN, que han demostrado ser mucho más viables, resultan ser más convenientes y prácticos para los objetivos de este trabajo y las limitaciones de hardware disponibles.

4.5. Análisis cualitativo de superresolución

En esta sección se presenta un análisis cualitativo de las imágenes de superresolución de placas vehiculares utilizando las arquitecturas GAN SRGAN y ESRGAN, evaluadas en tres formatos de imagen: JPG, WEBP y PNG. Este análisis tiene como objetivo comparar visualmente la calidad de las imágenes bajo diferentes niveles de compresión y resoluciones de entrada para cada formato, determinando así la efectividad de las arquitecturas en mejorar la legibilidad de las placas vehiculares. Las Tablas 4, 5 y 6 muestran imágenes originales y sus correspondientes versiones superresolucionadas para cada formato y nivel de compresión. En estas tablas, la columna "R" representa la resolución de entrada de las imágenes, y está indexada de la siguiente manera: 1 para 16x16, 2 para 32x32, 3 para 64x64, 4 para 96x96 y 5 para 128x128.

4.5.1. JPG

Para JPG se analizaron cinco resoluciones de entrada y cinco resoluciones de salida que corresponden a las resoluciones de la Tabla 1, además por cada resolución se establecieron análisis con cuatro niveles de compresión, respectivamente en JPG fueron 0%, 10%, 50% y 90%, siendo cero, baja, media y alta compresión respectivamente, como se observa en la Tabla 4.

Tabla 4 *Resultados imágenes JPG*

	Compresión cero			Compresión baja			Compresión media			Compresión alta		
R	**Original**	**SRGAN**	**ESRGAN**	**Original**	**SRGAN**	**ESRGAN**	**Original**	**SRGAN**	**ESRGAN**	**Original**	**SRGAN**	**ESRGAN**
1												
2												
3												
4												
5												

En las imágenes con compresión cero, se observa en la Tabla 4 que, a medida que aumenta la resolución de entrada, la legibilidad de los caracteres mejora significativamente tanto en las imágenes originales como en las procesadas por SRGAN y ESRGAN. En las resoluciones más bajas (16x16 y 32x32), las imágenes originales son apenas legibles; sin embargo, la aplicación de SRGAN y ESRGAN resulta en una mejora notable en la claridad de los caracteres. ESRGAN, en particular, ofrece una mayor nitidez y contraste en comparación con SRGAN. A resoluciones más altas (64x64, 96x96 y 128x128), las imágenes originales ya presentan una legibilidad considerable, y tanto SRGAN como ESRGAN logran mantener o mejorar ligeramente la nitidez y el contraste.

En el caso de la compresión baja, se percibe una disminución en la calidad de las imágenes originales debido a la aparición de artefactos de compresión, como se indica en la Tabla 4. En las resoluciones más bajas (16x16 y 32x32), SRGAN y ESRGAN mejoran significativamente la legibilidad de los caracteres al reducir los artefactos de compresión y aumentar la nitidez. ESRGAN proporciona una imagen más equilibrada y clara en comparación con SRGAN. En resoluciones medias y altas (64x64, 96x96 y 128x128), la compresión baja tiene un impacto menor, y ambas arquitecturas GAN logran mantener una alta legibilidad, con ESRGAN ofreciendo una ligera ventaja en términos de claridad y reducción de artefactos.

La compresión media introduce un nivel significativo de degradación en las imágenes originales, especialmente en resoluciones más bajas, como se evidencia en la Tabla 4. Las imágenes originales muestran una notable pérdida de detalles y la aparición de artefactos de compresión. SRGAN y ESRGAN logran mejorar la calidad visual al reducir estos artefactos y aumentar la nitidez de los caracteres. Nuevamente, ESRGAN supera a SRGAN en términos de claridad y definición de los caracteres. A resoluciones más altas, aunque la compresión media sigue

afectando la calidad, las arquitecturas GAN logran restaurar gran parte de la legibilidad, con ESRGAN proporcionando los mejores resultados.

La compresión alta tiene el mayor impacto negativo en la calidad de las imágenes originales, introduciendo una gran cantidad de artefactos de compresión y pérdida de detalles, como se observa en la Tabla 4. En resoluciones bajas (16x16 y 32x32), las imágenes originales son prácticamente ilegibles. SRGAN y ESRGAN logran recuperar cierta legibilidad, aunque con limitaciones evidentes. ESRGAN, aunque proporciona una mejora notable respecto a las imágenes originales, aún muestra dificultades en la restauración completa de los detalles. En resoluciones más altas (64x64, 96x96 y 128x128), ambas arquitecturas GAN logran una mejor recuperación de la calidad, pero la compresión alta sigue afectando significativamente la claridad, con ESRGAN mostrando un rendimiento superior en comparación con SRGAN.

Es importante destacar que las imágenes procesadas por SRGAN tienden a mostrar un tono gris en el fondo, lo cual es consistente en todas las condiciones de compresión. Esta tonalidad gris puede afectar la percepción de claridad y contraste en comparación con ESRGAN, que no presenta este problema, resultando en una apariencia más nítida y equilibrada en general.

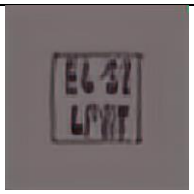
4.5.2. PNG

Para el análisis de imágenes en formato PNG, se evaluaron cinco resoluciones de entrada y sus correspondientes resoluciones de salida, conforme a lo descrito en la Tabla 1.

La compresión en formato PNG no provoca pérdida de información debido a su naturaleza de compresión sin pérdida. Esta característica asegura que todos los datos originales de la imagen se mantengan intactos, permitiendo que la imagen sea reconstruida de manera idéntica a su estado previo a la compresión. Independientemente del nivel de compresión aplicado, la calidad de la

imagen no se degrada. Esto implica que la imagen comprimida conserva una apariencia idéntica a la original, lo cual es fundamental para aplicaciones que requieren mantener la máxima calidad de imagen, como es el caso del reconocimiento de placas vehiculares, este análisis se puede ver en la Tabla 5.

Tabla 5 *Resultados imágenes PNG*

R	Original	SRGAN	ESRGAN
1			
2			
3			
4			
5			

En la Tabla 5 se presentan los resultados cualitativos de las imágenes en formato PNG procesadas por las arquitecturas SRGAN y ESRGAN, comparadas con las imágenes originales. Es importante recordar que, debido a la naturaleza de compresión sin pérdida del formato PNG, no se realizaron análisis con variaciones de compresión, ya que estas no afectan la calidad de la imagen.

Al observar las imágenes procesadas, se puede notar que, para las resoluciones más bajas (16x16 y 32x32), las imágenes originales presentan una legibilidad limitada. Sin embargo, al aplicar las arquitecturas GAN, se logra una mejora significativa en la claridad de los caracteres. En particular, ESRGAN ofrece una mayor nitidez y contraste comparado con SRGAN, como se puede ver claramente en la Tabla 5. A pesar de esto, es evidente que las imágenes procesadas por SRGAN presentan un tono gris en el fondo, lo cual puede afectar la percepción de claridad.

Para las resoluciones más altas (64x64, 96x96 y 128x128), las imágenes originales ya son bastante legibles. Al aplicar SRGAN y ESRGAN, ambas arquitecturas logran mantener o incluso mejorar ligeramente la nitidez y el contraste de las imágenes. Sin embargo, ESRGAN sigue mostrando una ventaja en términos de definición y claridad de los caracteres, proporcionando resultados más limpios y definidos. La presencia del tono gris en las imágenes procesadas por SRGAN es nuevamente evidente en estas resoluciones, afectando potencialmente la percepción de la calidad visual en comparación con ESRGAN.

4.5.3. WEBP

Para el formato WEBP, se analizaron cinco resoluciones de entrada y cinco resoluciones de salida, conforme a lo descrito en la Tabla 1. Además, para cada resolución se realizaron análisis con cuatro niveles de compresión en el formato WEBP: 0%, 10%, 50% y 90%, representando cero, baja, media y alta compresión respectivamente como se observa en la Tabla 6.

Tabla 6 *Resultados imágenes WEBP*

	Compresión cero			Compresión baja			Compresión media			Compresión alta		
R	**Original**	**SRGAN**	**ESRGAN**	**Original**	**SRGAN**	**ESRGAN**	**Original**	**SRGAN**	**ESRGAN**	**Original**	**SRGAN**	**ESRGAN**
1												
2												
3												
4												
5												

En las imágenes con compresión cero, se observa en la Tabla 6 que, a medida que aumenta la resolución de entrada, la legibilidad de los caracteres mejora significativamente tanto en las imágenes originales como en las procesadas por SRGAN y ESRGAN. En las resoluciones más bajas (16x16 y 32x32), las imágenes originales son apenas legibles; sin embargo, la aplicación de SRGAN y ESRGAN resulta en una mejora notable en la claridad de los caracteres. ESRGAN, en particular, ofrece una mayor nitidez y contraste en comparación con SRGAN, proporcionando caracteres más definidos. A resoluciones más altas (64x64, 96x96 y 128x128), las imágenes originales ya presentan una legibilidad considerable, y tanto SRGAN como ESRGAN logran mantener o incluso mejorar ligeramente la nitidez y el contraste, aunque las diferencias se vuelven menos evidentes debido a la alta calidad de las imágenes originales.

En el caso de la compresión baja, se percibe una disminución en la calidad de las imágenes originales debido a la aparición de artefactos de compresión, como se indica en la Tabla 6. En las resoluciones más bajas (16x16 y 32x32), SRGAN y ESRGAN mejoran de manera significativa la legibilidad de los caracteres al reducir los artefactos de compresión y aumentar la nitidez. ESRGAN, en particular, proporciona una imagen más equilibrada y clara en comparación con SRGAN. Cabe destacar que las imágenes procesadas por SRGAN tienden a mostrar un tono gris en el fondo, lo cual puede afectar la percepción de claridad en algunos casos. En resoluciones medias y altas (64x64, 96x96 y 128x128), la compresión baja tiene un impacto menor, y ambas arquitecturas GAN logran mantener una alta legibilidad, con ESRGAN ofreciendo una ligera ventaja en términos de claridad y reducción de artefactos.

La compresión media introduce un nivel significativo de degradación en las imágenes originales, especialmente en resoluciones más bajas, como se evidencia en la Tabla 6. Las imágenes originales muestran una notable pérdida de detalles y la aparición de artefactos de

compresión. SRGAN y ESRGAN logran mejorar la calidad visual al reducir estos artefactos y aumentar la nitidez de los caracteres. Nuevamente, ESRGAN supera a SRGAN en términos de claridad y definición de los caracteres. Sin embargo, las imágenes procesadas por SRGAN presentan un tono gris en el fondo que no está presente en las imágenes procesadas por ESRGAN, lo que podría influir en la percepción de calidad. A resoluciones más altas, aunque la compresión media sigue afectando la calidad, las arquitecturas GAN logran restaurar gran parte de la legibilidad, con ESRGAN proporcionando los mejores resultados.

La compresión alta tiene el mayor impacto negativo en la calidad de las imágenes originales, introduciendo una gran cantidad de artefactos de compresión y pérdida de detalles, como se observa en la Tabla 6. En resoluciones bajas (16x16 y 32x32), las imágenes originales son prácticamente ilegibles. SRGAN y ESRGAN logran recuperar cierta legibilidad, aunque con limitaciones evidentes. ESRGAN, aunque proporciona una mejora notable respecto a las imágenes originales, aún muestra dificultades en la restauración completa de los detalles. Es importante mencionar que las imágenes procesadas por SRGAN presentan un tono gris en el fondo, lo cual puede comprometer la claridad general. En resoluciones más altas (64x64, 96x96 y 128x128), ambas arquitecturas GAN logran una mejor recuperación de la calidad, pero la compresión alta sigue afectando significativamente la claridad, con ESRGAN mostrando un rendimiento superior en comparación con SRGAN.

En todas las resoluciones y niveles de compresión, se observa consistentemente que las imágenes procesadas por SRGAN tienden a mostrar un tono gris en el fondo. Esta característica es evidente en la Tabla 6 y podría afectar la percepción de claridad y calidad general de las imágenes superresolucionadas. Este fenómeno no se presenta en las imágenes procesadas por

ESRGAN, lo que refuerza la ventaja de esta arquitectura en términos de calidad visual y fidelidad en la restauración de los detalles.

4.6. Resultados cualitativos

Basado en el análisis cualitativo realizado en la sección 5.5 del documento, se puede finiquitar que la arquitectura ESRGAN muestra un desempeño superior en comparación con SRGAN en términos de calidad de imagen en la superresolución de placas vehiculares. Esta postura se fundamenta en diversas observaciones detalladas a través de diferentes formatos de imagen (JPG, PNG y WEBP), niveles de compresión y resoluciones.

En primer lugar, ESRGAN ha demostrado una mayor nitidez y claridad en las imágenes procesadas, especialmente en las resoluciones más bajas (16x16 y 32x32). En estas resoluciones, las imágenes originales presentaban una legibilidad deficiente; sin embargo, la aplicación de ESRGAN resultó en una mejora significativa en la claridad de los caracteres, superando notablemente a SRGAN. ESRGAN ofreció una definición y contraste superiores, proporcionando caracteres más nítidos y claros, mientras que SRGAN tendía a mostrar un tono gris en el fondo que afectaba la percepción general de calidad.

En el contexto de compresiones bajas y medias, tanto en los formatos JPG como WEBP, ESRGAN mostró una mayor eficacia en la reducción de artefactos de compresión en comparación con SRGAN. Las imágenes procesadas por ESRGAN mantenían una claridad y calidad visual superiores. Incluso en niveles de compresión alta, donde ambos modelos enfrentaron desafíos significativos, ESRGAN demostró una mejor capacidad para mantener la calidad general de la imagen y minimizar los artefactos de compresión.

Asimismo, ESRGAN mostró un rendimiento más consistente a través de todas las resoluciones evaluadas (16x16, 32x32, 64x64, 96x96 y 128x128). En resoluciones más altas, donde las imágenes originales ya presentaban una alta calidad, ESRGAN mantuvo una ventaja en términos de claridad y definición sobre SRGAN.

En cuanto al desempeño de los formatos, el análisis reveló que el formato PNG, debido a su naturaleza de compresión sin pérdida, no mostró degradación alguna en la calidad de imagen, independientemente del nivel de compresión aplicado. Esto permitió que las imágenes PNG se mantuvieran idénticas a las originales, proporcionando una base sólida para el proceso de superresolución. Por otro lado, los formatos JPG y WEBP, aunque generalmente bien soportados por ESRGAN, presentaron una mayor susceptibilidad a la degradación por compresión.

4.7. Análisis cuantitativo de superresolución

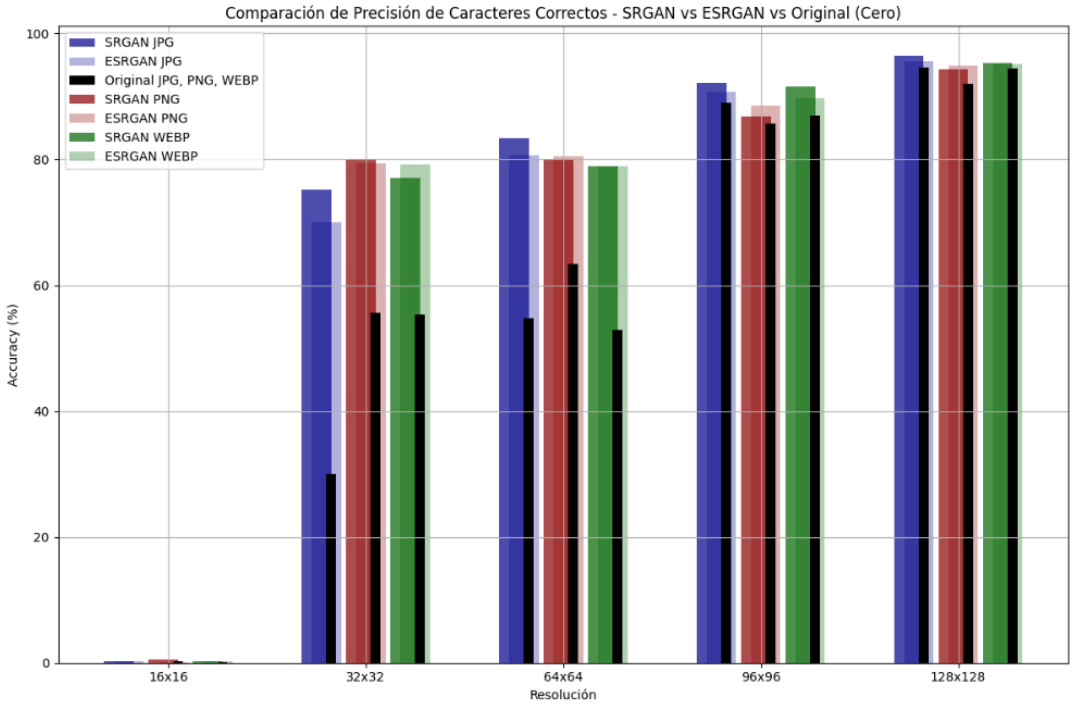
En esta sección, se presenta un análisis cuantitativo del rendimiento de las arquitecturas SRGAN y ESRGAN en la identificación de caracteres correctos y placas completas. Este análisis se basa en gráficas que comparan la precisión de estas arquitecturas en tres formatos de imagen diferentes (JPG, PNG y WEBP), evaluadas bajo diversas resoluciones y niveles de compresión.

Las gráficas muestran la precisión alcanzada por SRGAN y ESRGAN, así como la precisión de la imagen original sin procesar, tomada como referencia base. Las barras en las gráficas se diferencian por colores y opacidad: barras oscuras para SRGAN (azul para JPG, rojo para PNG y verde para WEBP) y barras claras para ESRGAN (azul claro para JPG, rojo claro para PNG y verde claro para WEBP). Las barras negras indican la precisión de la imagen original sin procesar en los tres formatos mencionados. Las gráficas se organizan según los niveles de compresión (cero, baja, media y alta), resoluciones (16x16, 32x32, 64x64, 96x96 y 128x128) y

métricas de precisión (caracteres correctos y placas completas), permitiendo una comparación detallada del rendimiento de cada arquitectura en diferentes escenarios.

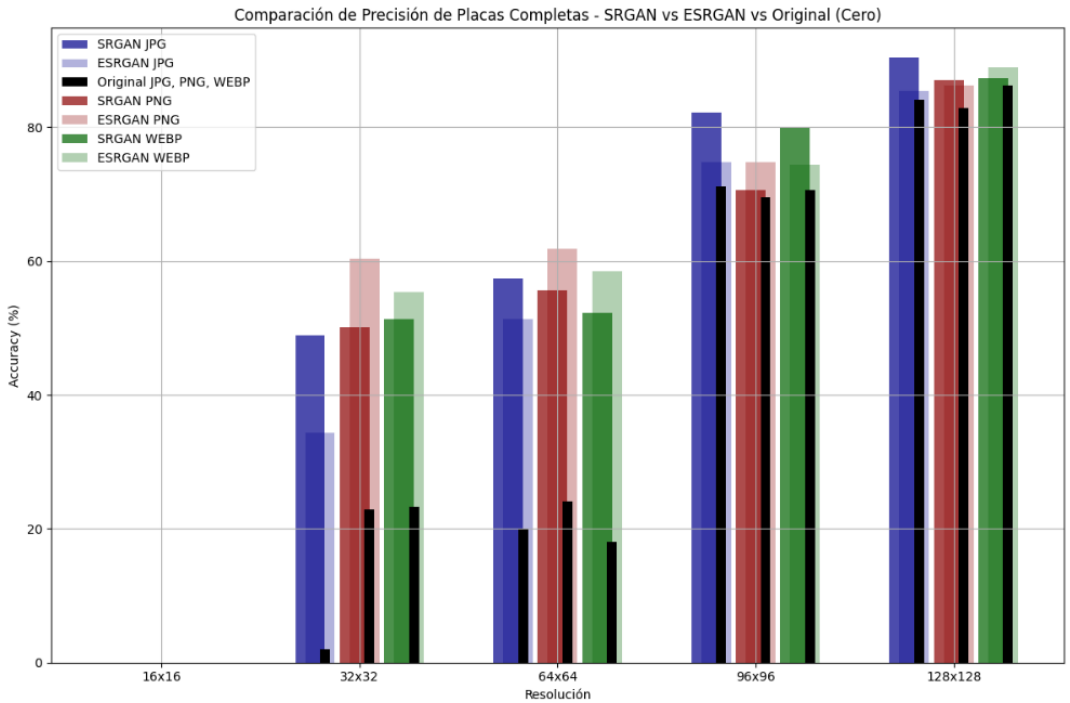
La Figura 9 muestra la precisión de caracteres correctos bajo compresión cero. Observamos que a medida que aumenta la resolución, la precisión mejora significativamente para todas las arquitecturas y formatos. En resoluciones bajas (16x16 y 32x32), tanto SRGAN como ESRGAN muestran una mejora considerable respecto a la imagen original (barra negra). ESRGAN generalmente supera a SRGAN en estas resoluciones, especialmente en PNG y WEBP. A partir de 64x64, ambas arquitecturas alcanzan una alta precisión, con diferencias mínimas entre ellas. Sin embargo, SRGAN muestra un rendimiento ligeramente superior en JPG en resoluciones de 96x96 y 128x128, mientras que ESRGAN mantiene una precisión constante y alta en todos los formatos y resoluciones.

Figura 9 *Precisión de caracteres correctos (Compresión Cero)*



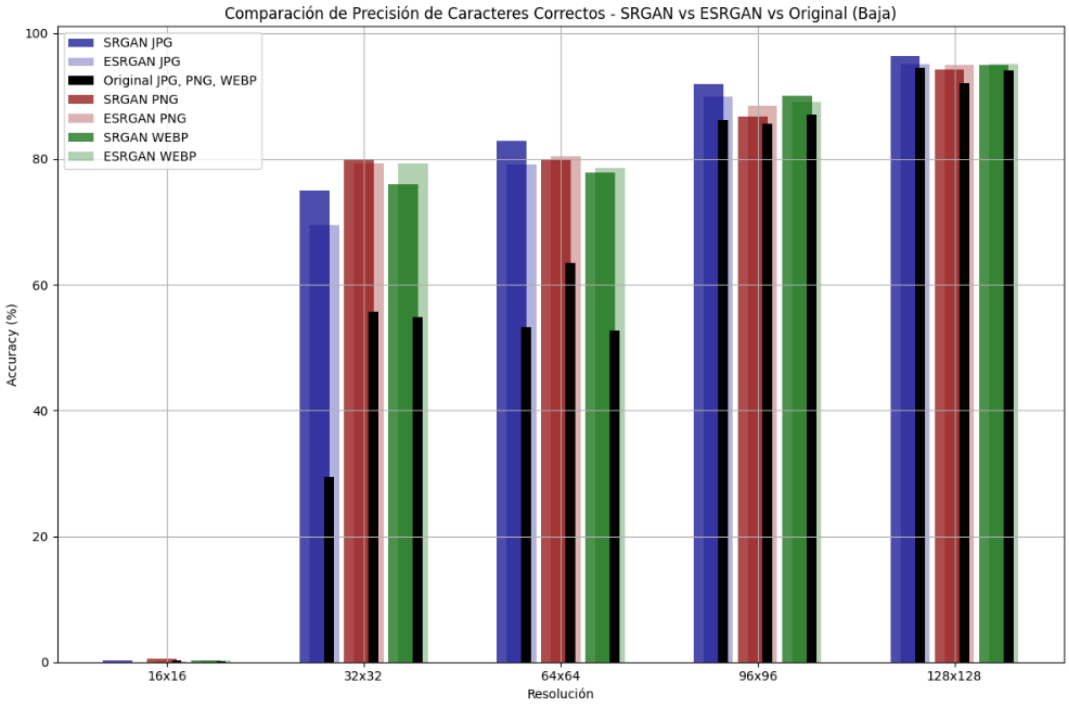
Por otro lado, en la Figura 10, se observa un patrón similar al de la Figura 9, pero para la precisión de placas completas. Nuevamente, ESRGAN tiende a superar a SRGAN en las resoluciones más bajas, con una diferencia más marcada en PNG y WEBP. A medida que aumenta la resolución, ambas arquitecturas alcanzan una alta precisión, pero SRGAN muestra un desempeño ligeramente mejor en JPG en resoluciones más altas (96x96 y 128x128). En resoluciones de 64x64 y superiores, ambas arquitecturas logran reducir considerablemente los errores, alcanzando niveles de precisión muy altos. Sin embargo, la ventaja de ESRGAN en resoluciones bajas (32x32) en formatos PNG y WEBP es significativa y muestra su capacidad para manejar mejor las imágenes con mayor pérdida de información. En general, ESRGAN parece ser más robusta en condiciones de compresión cero para la precisión de placas completas en todos los formatos, especialmente en aquellos más propensos a artefactos de compresión.

Figura 10 *Precisión de placas completas (Compresión cero)*



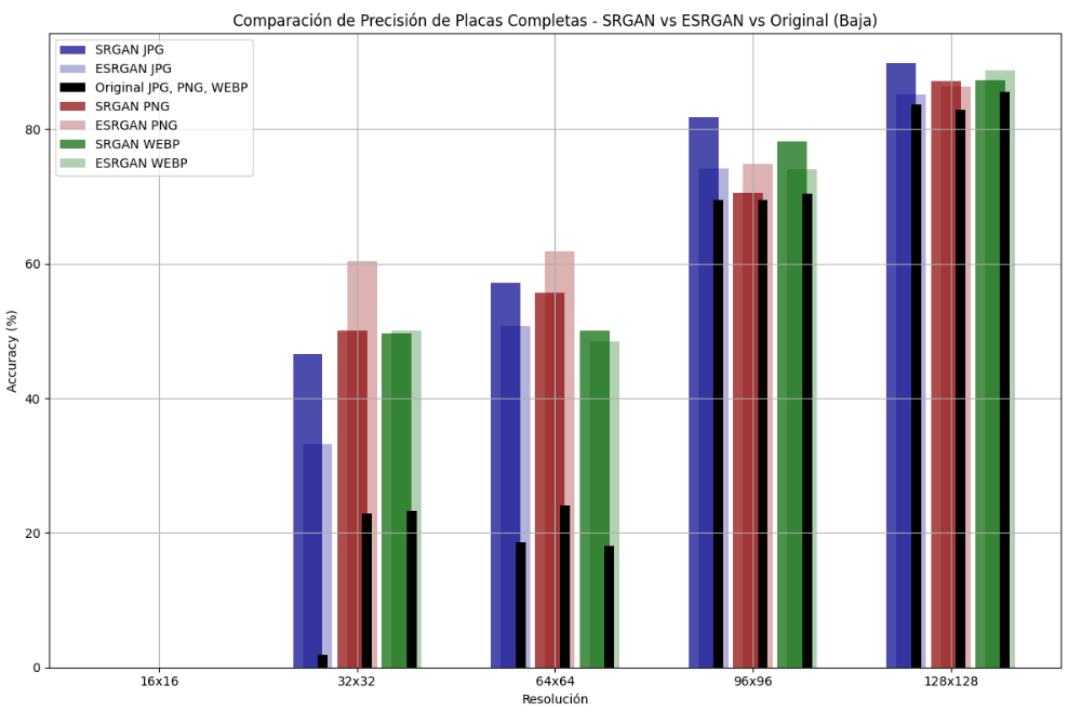
La Figura 11 presenta la precisión de caracteres correctos bajo compresión baja. La calidad de las imágenes originales se ve afectada por la compresión, lo que se refleja en una menor precisión de la barra negra. Tanto SRGAN como ESRGAN logran mejorar la precisión significativamente. En resoluciones bajas (16x16 y 32x32), ESRGAN nuevamente supera a SRGAN en PNG y WEBP, mientras que SRGAN muestra un mejor rendimiento en JPG. En resoluciones más altas, como 96x96 y 128x128, ambas arquitecturas alcanzan niveles similares de precisión, con una ligera ventaja para ESRGAN en general. La diferencia de rendimiento entre SRGAN y ESRGAN se reduce en resoluciones más altas, lo que indica que ambas arquitecturas pueden manejar eficazmente las imágenes con mayor detalle. Sin embargo, ESRGAN sigue teniendo una ligera ventaja en términos de consistencia y capacidad para manejar artefactos de compresión en todos los formatos.

Figura 11 *Precisión de caracteres correctos (Compresión Baja)*



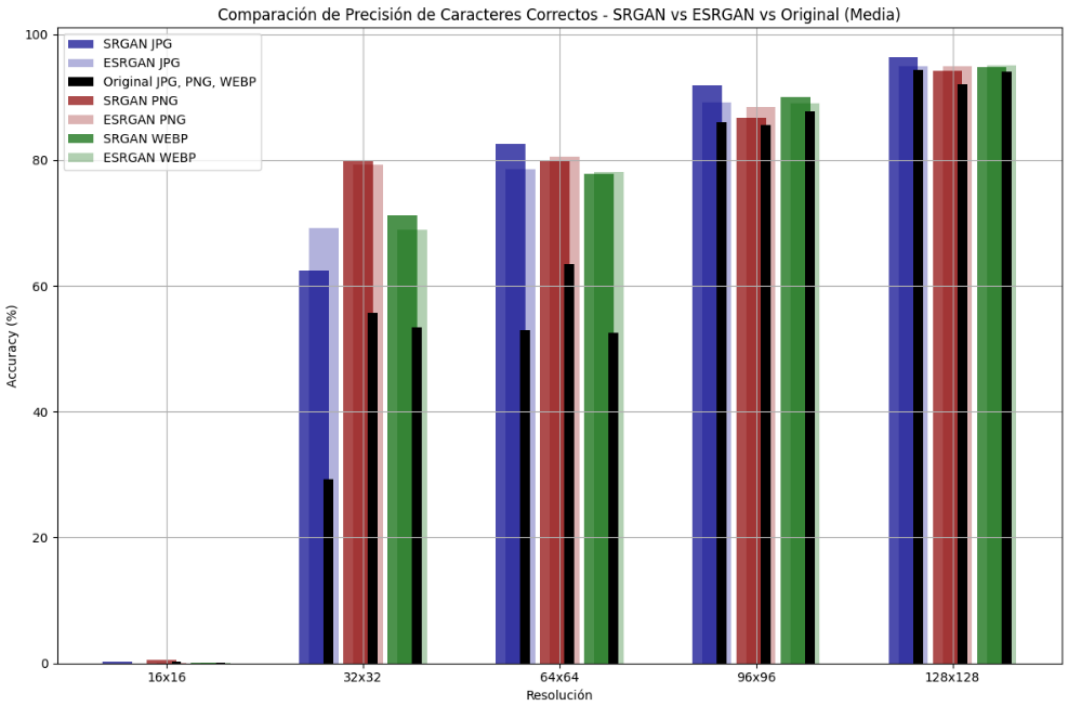
La precisión de placas completas bajo compresión baja, mostrada en la Figura 12, refleja que ESRGAN continúa mostrando un rendimiento superior en resoluciones bajas y medias (32x32 y 64x64), especialmente en PNG y WEBP. En JPG, SRGAN y ESRGAN tienen un rendimiento comparable. En resoluciones altas (96x96 y 128x128), ambas arquitecturas logran una alta precisión, con ESRGAN manteniendo una ligera ventaja en todos los formatos. La capacidad de ESRGAN para mantener una alta precisión en resoluciones bajas y medias sugiere que es más efectiva en la reducción de artefactos de compresión que pueden afectar la identificación de placas completas. Además, la consistencia de ESRGAN en diferentes formatos y niveles de compresión baja refuerza su robustez en aplicaciones prácticas donde la calidad de la imagen puede variar significativamente.

Figura 12 *Precisión de placas completas (Compresión Baja)*



Con compresión media, la Figura 13 revela una disminución significativa en la precisión de la imagen original debido a los artefactos de compresión. Tanto SRGAN como ESRGAN logran mejorar la precisión en todos los formatos y resoluciones. ESRGAN muestra una ventaja notable en resoluciones bajas y medias (32x32 y 64x64) en PNG y WEBP, mientras que SRGAN tiene un mejor rendimiento en JPG en algunas resoluciones. En resoluciones altas, ambas arquitecturas alcanzan una precisión similar, aunque ESRGAN mantiene una ligera ventaja general. La diferencia de rendimiento entre las dos arquitecturas se hace más evidente en condiciones de compresión media, donde ESRGAN demuestra ser más eficiente en la reconstrucción de detalles finos y la reducción de artefactos. Esta capacidad es crucial en aplicaciones donde la compresión media es común y puede afectar significativamente la calidad de la imagen.

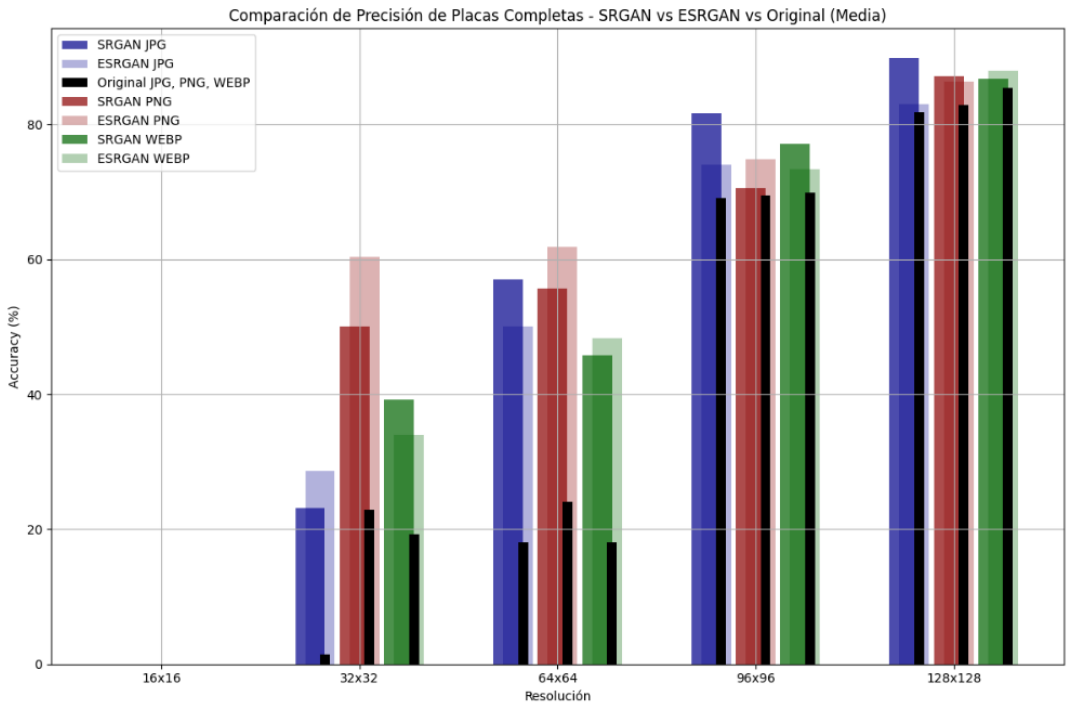
Figura 13 *Precisión caracteres correctos (Compresión Media)*



La Figura 14 presenta la precisión de placas completas bajo compresión media. Se observa que ESRGAN supera consistentemente a SRGAN en resoluciones bajas y medias, especialmente

en PNG y WEBP. En JPG, SRGAN muestra un rendimiento competitivo en resoluciones más altas. A 96x96 y 128x128, ambas arquitecturas alcanzan una precisión alta, con ESRGAN manteniendo una ligera ventaja en la mayoría de los formatos. La superioridad de ESRGAN en condiciones de compresión media, especialmente en formatos como PNG y WEBP, destaca su capacidad para preservar detalles importantes en la imagen y reducir artefactos que podrían afectar la identificación de placas completas. La consistencia de ESRGAN en diferentes resoluciones y formatos refuerza su eficacia en escenarios donde la calidad de la imagen es crucial para el rendimiento del sistema.

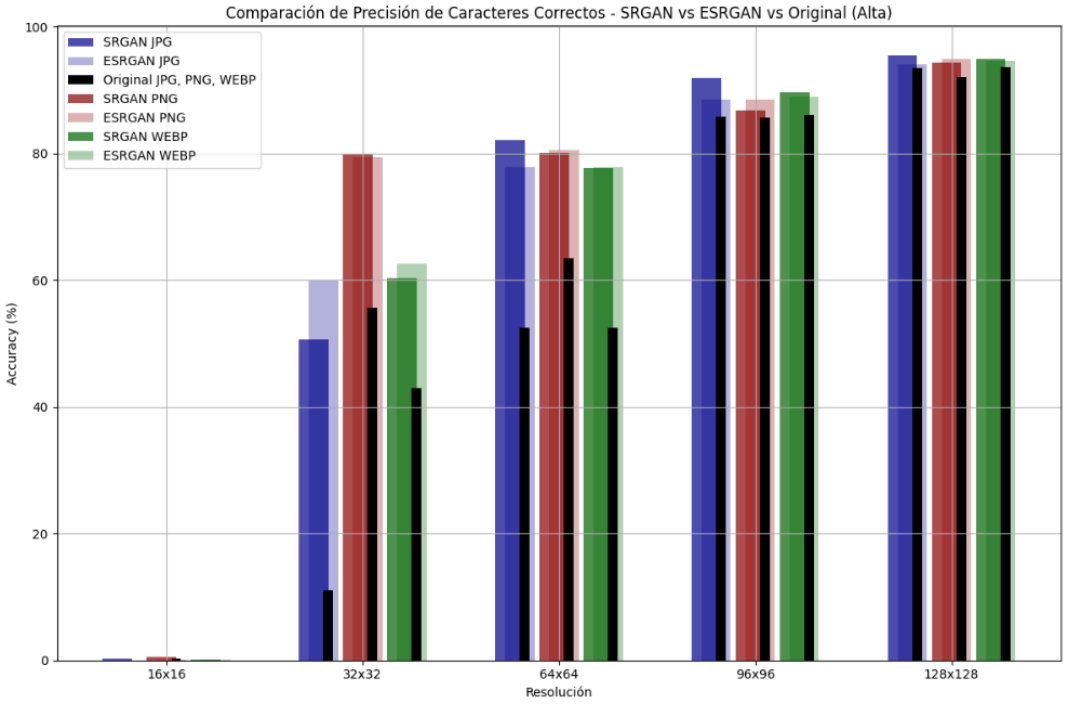
Figura 14 *Precisión placas completas (Compresión Media)*



Bajo condiciones de compresión alta vistas en la figura 15, la precisión de caracteres correctos se ve severamente afectada en la imagen original. Tanto SRGAN como ESRGAN logran mejoras significativas. ESRGAN muestra una ventaja clara en resoluciones bajas y medias (32x32 y 64x64) en WEBP. En resoluciones altas (96x96 y 128x128), ambas arquitecturas logran niveles

similares de precisión, aunque ESRGAN generalmente mantiene una ventaja en términos de reducción de artefactos de compresión. La capacidad de ESRGAN para manejar mejor las condiciones de compresión alta sugiere que es más robusta en escenarios donde la calidad de la imagen está severamente comprometida. Esta robustez es esencial en aplicaciones donde las imágenes comprimidas de baja calidad deben ser mejoradas para su análisis.

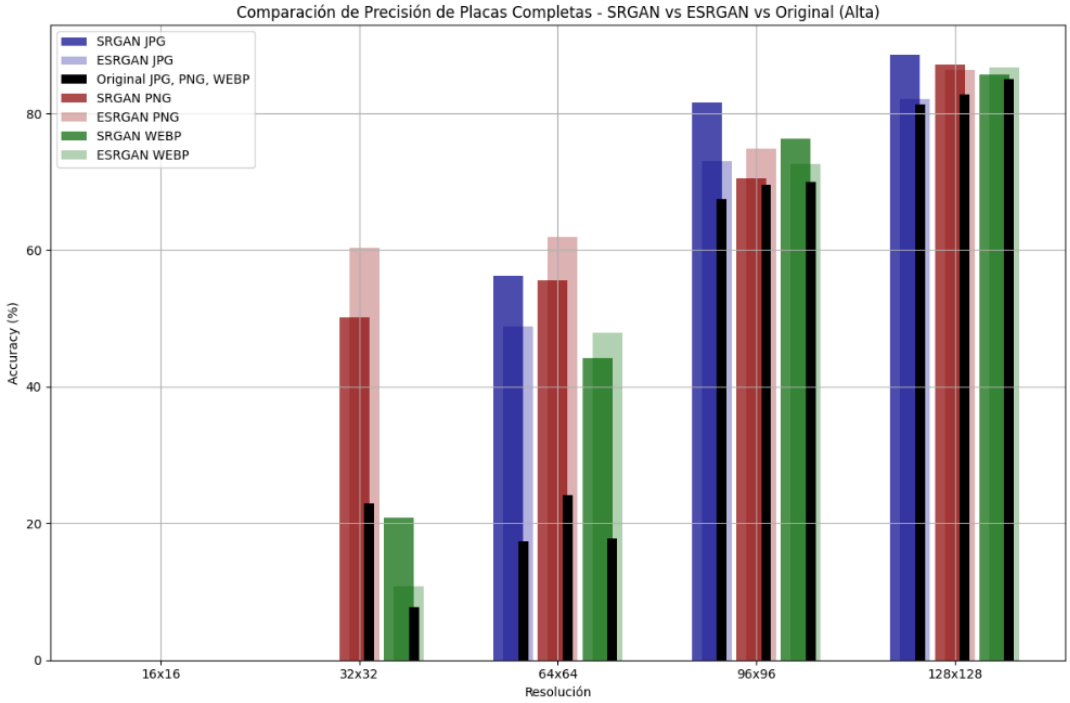
Figura 15 *Precisión caracteres correctos (Compresión Alta)*



Finalmente, la Figura 16 muestra la precisión de placas completas bajo compresión alta. ESRGAN continúa mostrando un rendimiento superior en resoluciones bajas y medias (32x32 y 64x64) en PNG y WEBP. En JPG, SRGAN muestra un rendimiento competitivo, especialmente en resoluciones más altas. A 96x96 y 128x128, ambas arquitecturas logran una alta precisión, con ESRGAN manteniendo una ligera ventaja en la mayoría de los formatos. La capacidad de ESRGAN para mantener una alta precisión en condiciones de compresión alta, especialmente en

formatos más susceptibles a la pérdida de calidad, como PNG y WEBP, refuerza su eficacia en la mejora de la calidad de la imagen y la reducción de artefactos.

Figura 16 *Precisión placas completas (Compresión Alta)*



4.8. Resultados cuantitativos

El análisis cuantitativo basado en las figuras presentadas demuestra que ESRGAN generalmente supera a SRGAN en PNG y WEBP, especialmente en resoluciones bajas y medias, mientras que SRGAN muestra un rendimiento más competitivo en JPG, especialmente en resoluciones altas. A medida que aumenta el nivel de compresión, ESRGAN tiende a mantener una ventaja en términos de precisión de caracteres correctos y placas completas debido a su capacidad para reducir artefactos de compresión más eficazmente. En resoluciones altas (96x96 y 128x128), ambas arquitecturas logran una alta precisión, aunque ESRGAN mantiene una ligera ventaja general. La precisión de caracteres correctos y placas completas sigue un patrón similar en términos de rendimiento de las arquitecturas, con ESRGAN mostrando una ventaja consistente en

ambos aspectos, especialmente en condiciones de compresión media y alta. Este análisis detallado confirma que, en general, ESRGAN ofrece un mejor rendimiento en la mayoría de los escenarios evaluados, aunque SRGAN sigue siendo competitivo en ciertos contextos, especialmente en el formato JPG.

4.9. Resultados generales

Para la implementación de la superresolución en la identificación de placas vehiculares, es fundamental seleccionar tanto el modelo de superresolución como el formato de imagen adecuados. Tras un análisis exhaustivo tanto cualitativo como cuantitativo, se recomienda utilizar el modelo ESRGAN en conjunto con el formato de imagen PNG.

El modelo ESRGAN destaca por su rendimiento superior en una variedad de condiciones. Tanto en resoluciones bajas como medias, ESRGAN muestra una mejora significativa en la nitidez y claridad de los caracteres y placas vehiculares comparado con SRGAN. Esta capacidad es particularmente crítica en aplicaciones donde la legibilidad de cada detalle es esencial. Además, ESRGAN ha demostrado ser más efectivo en la reducción de artefactos de compresión, lo que resulta en imágenes más limpias y definidas. Esto es especialmente beneficioso en escenarios con alta compresión, donde las imágenes suelen estar más degradadas y la precisión en la identificación se ve comprometida. Asimismo, ESRGAN mantiene una alta precisión en diversos formatos de imagen y resoluciones, desde 16x16 hasta 128x128, lo que demuestra su adaptabilidad y robustez.

En cuanto al formato de imagen, PNG es la opción más adecuada debido a su capacidad de compresión sin pérdida. Esta característica asegura que no se pierda calidad de imagen durante el proceso de compresión, lo cual es crucial para aplicaciones de reconocimiento de caracteres y placas vehiculares, donde cada detalle cuenta. El análisis cuantitativo indica que, en resoluciones

bajas y medias, ESRGAN muestra un rendimiento particularmente bueno con imágenes PNG. Esto sugiere que el formato PNG permite a ESRGAN explotar al máximo su capacidad de mejora de la resolución. Además, aunque PNG es menos susceptible a artefactos de compresión en comparación con otros formatos como JPG y WEBP, ESRGAN aún logra mejorar significativamente la calidad de las imágenes PNG, asegurando una mayor precisión en la identificación de caracteres y placas. Por último, la compatibilidad y flexibilidad del formato PNG con diversos dispositivos y software facilita su integración en sistemas de vigilancia y control de tráfico.

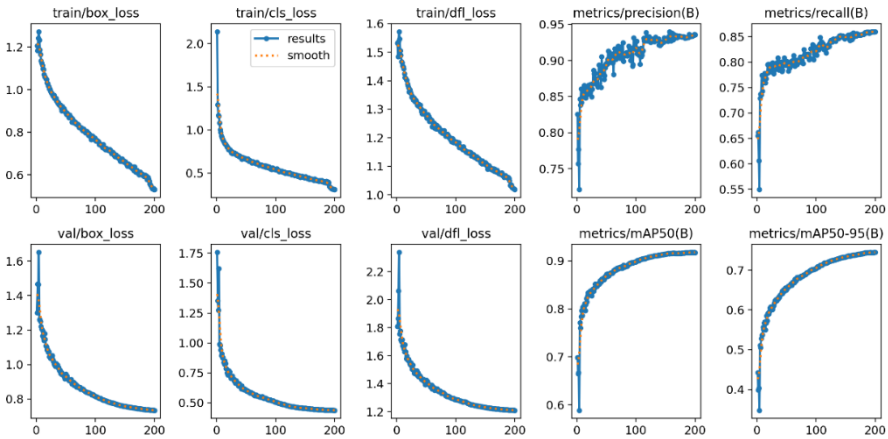
4.10. Implementación

Para el cumplimiento oportuno del desarrollo de la implementación se parte de 3 modelos claves, un modelo pre entrenado por parte de Ultralytics que detecta vehículos, otro modelo entrenado para placas de vehículos y el último que fue escogido en la exploración del espacio de diseño, es decir, ESRGAN_PNG.

4.10.1. Modelo para placas

Para realizar el entrenamiento de las placas se utilizó una GPU de 12GB, donde se obtuvo un modelo con un peso de 6.151 KB y los resultados mostrados en la Figura 17.

Figura 17 *Resultados detección de placas*



La figura 17 presenta diversas gráficas que ilustran el rendimiento y las métricas del modelo YOLOv8 durante las fases de entrenamiento y validación para la detección de placas. Las pérdidas de las cajas delimitadoras (train/box_loss y val/box_loss) y las pérdidas de clasificación (train/cls_loss y val/cls_loss) muestran una disminución constante a lo largo de las épocas, lo que indica una mejora en la capacidad del modelo para predecir las posiciones de las cajas delimitadoras y en la correcta identificación de las clases (placas) sin evidencias de sobreajuste significativo.

Asimismo, la disminución consistente de la pérdida de la función de distribución (train/dfl_loss y val/dfl_loss) sugiere que el modelo está afinando su capacidad para predecir distribuciones precisas de los valores de salida, contribuyendo a la precisión global de las predicciones. Las métricas de precisión y recuperación (metrics/precision(B) y metrics/recall(B)) evidencian una mejora continua, alcanzando valores elevados hacia el final del entrenamiento, lo que sugiere una creciente precisión en la detección de placas y en la identificación correcta de las instancias presentes en las imágenes.

Por último, las métricas de precisión media promedio (metrics/mAP50(B) y metrics/mAP50-95(B)) también muestran una tendencia de mejora sostenida, estabilizándose en un rango elevado. Esto refleja que el modelo mantiene una alta precisión y consistencia en la detección de placas a diferentes umbrales de intersección sobre unión (IoU).

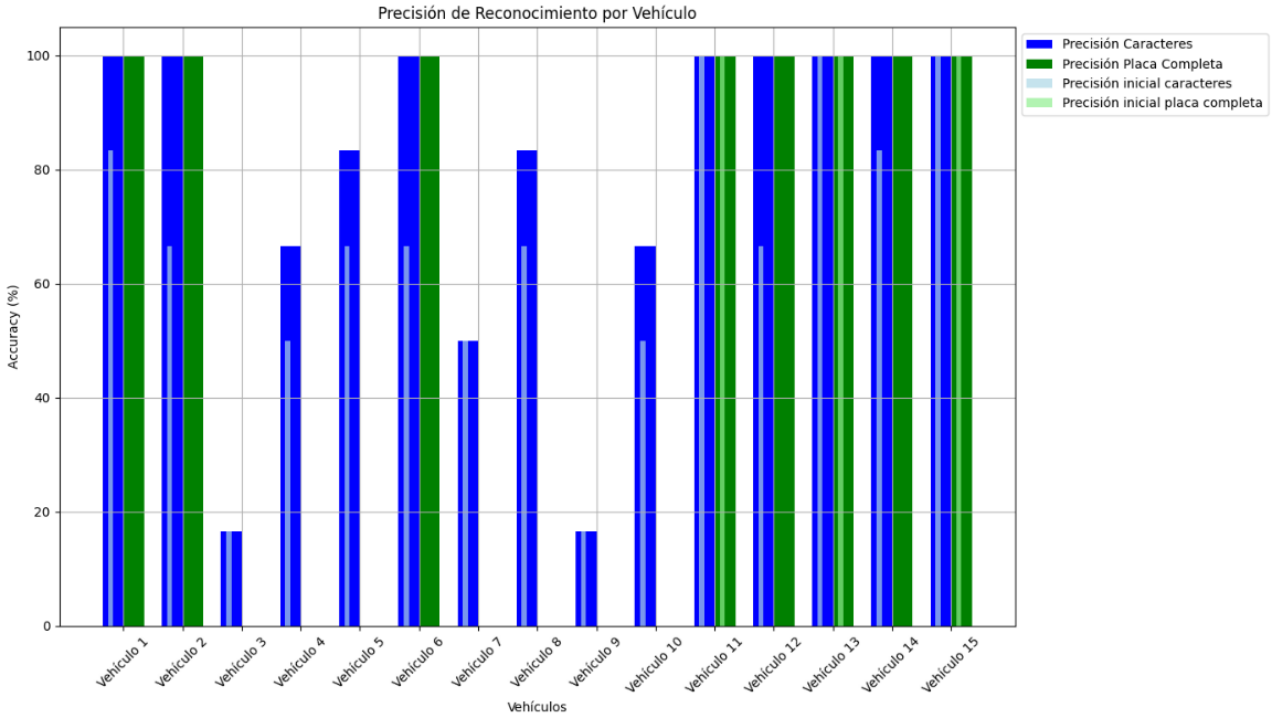
4.10.2. Experimento en ambiente controlado

El experimento se llevó a cabo en un ambiente controlado con la cámara ubicada a una altura de 6 metros. Este montaje tuvo la finalidad de simular una cámara no especializada ubicada a una distancia considerable de la calle y los vehículos, emulando un escenario típico de vigilancia

urbana. Durante el experimento, se evaluó el rendimiento de la implementación utilizando dos métricas principales: precisión por caracteres y precisión de placa completa.

Cabe resaltar que los vehículos fueron detectados a partir de videos de 7 segundos de duración. El estudio se desarrolló utilizando un total de 15 vehículos. Las precisiones obtenidas para cada vehículo fueron registradas y analizadas para determinar la eficacia de los modelos implementados. Los resultados se resumen en términos de precisión tanto para caracteres individuales como para la identificación completa de las placas vehiculares como se observa en la figura.

Figura 18 *Precisión de caracteres y placas*



La Figura 18 presenta la precisión de reconocimiento de caracteres y placas completas para un conjunto de 15 vehículos, utilizando el sistema implementado. En esta, se comparan cuatro métricas de precisión: precisión de caracteres correctos (representada en azul oscuro), precisión de

placas completas (representada en verde oscuro), precisión inicial de caracteres (representada en azul claro) y precisión inicial de placas completas (representada en verde claro).

Para cada vehículo, se han medido y representado las precisiones alcanzadas, permitiendo una evaluación detallada del rendimiento del sistema. Los resultados muestran una variabilidad considerable en las precisiones, con algunos vehículos alcanzando una precisión del 100% en ambas métricas después de aplicar la superresolución, mientras que otros presentan valores significativamente más bajos.

En particular, se observa que la precisión de caracteres correctos es más alta en general comparada con la precisión de placas completas. Por ejemplo, los vehículos 1, 2, 6, 11, 12, 13, 14, y 15 lograron una precisión del 100% en caracteres correctos y en placas completas. Por otro lado, vehículos como el 3, 4, 5, 7, 8, 9, 10 presentan una notable disminución en la precisión, especialmente en la métrica de placas completas, dado que, si no se reconocen todos los caracteres, la precisión de placa completa será 0%.

Las barras intermedias, azul claro y verde claro, representan la precisión inicial de caracteres y placas completas, respectivamente, antes de aplicar la superresolución. Estas barras proporcionan una visión de la mejora obtenida gracias al uso de técnicas de superresolución. La precisión inicial de caracteres y placas completas muestra valores generalmente más bajos que los obtenidos después de aplicar la superresolución, evidenciando la efectividad de esta técnica para mejorar el rendimiento del sistema de reconocimiento.

El análisis de la precisión inicial revela que, sin la aplicación de superresolución, el sistema presenta una menor capacidad para identificar correctamente tanto los caracteres individuales como las placas completas. Esto se debe a la baja resolución de las imágenes capturadas en

condiciones adversas, como poca iluminación o excesiva luz solar directa. La aplicación de superresolución, por lo tanto, juega un papel crucial en la mejora de la calidad de las imágenes, permitiendo un reconocimiento más preciso.

En la parte final de la gráfica se incluyen las barras que representan los promedios de precisión para todas las muestras analizadas. El promedio de precisión para caracteres correctos fue de 78.81%, mientras que para placas completas fue de 53.33%. Estos promedios, representados con barras rayadas, proporcionan una visión general del rendimiento del sistema a lo largo de todos los vehículos evaluados.

La Figura 18 evidencia que, aunque el sistema de superresolución e identificación vehicular es capaz de alcanzar altas precisiones en ciertos casos, existe una variabilidad significativa en su rendimiento. Esta variabilidad surge principalmente debido a las condiciones ambientales durante la captura de los videos. En condiciones nocturnas, dado que la cámara utilizada no es especializada y carece de visión nocturna, la precisión del sistema se ve considerablemente afectada. Asimismo, en días con mucha luz solar directa hacia el lente de la cámara, el rendimiento del sistema se ve comprometido. Por otro lado, el sistema funciona mejor en días templados con iluminación uniforme.

4.10.3. Tiempo de inferencia

Durante la implementación del sistema de superresolución e identificación de placas vehiculares, se evaluó el tiempo de inferencia utilizando videos de 7 segundos de duración. El procesamiento completo de estos videos requirió un tiempo total de 117,719 ms (aproximadamente 117.7 segundos o 1.96 minutos). Este tiempo se desglosa de la siguiente manera:

El procesamiento de los modelos de detección de vehículos y placas tomó 85,804 ms (aproximadamente 85.8 segundos o 1.43 minutos). La división de las imágenes en resoluciones específicas (64x64, 96x96 y 128x128) requirió 551 ms (aproximadamente 0.55 segundos o 0.009 minutos). La aplicación del modelo de superresolución tardó 18,071 ms (aproximadamente 18.1 segundos o 0.30 minutos). La unión de las imágenes en tramos de 128x128 demandó 488 ms (aproximadamente 0.49 segundos o 0.008 minutos). El aumento de contraste de las imágenes tomó 765 ms (aproximadamente 0.76 segundos o 0.013 minutos). Finalmente, el reconocimiento óptico de caracteres (OCR) utilizando el modelo de caracteres de placas vehiculares requirió 12,000 ms (aproximadamente 12 segundos o 0.20 minutos).

Es relevante mencionar que el sistema no procesa cada fotograma del video, sino que selecciona fotogramas a intervalos de tiempo específicos. En el código de procesamiento, se establece un intervalo de tiempo de 0.09 segundos entre fotogramas procesados, lo que permite realizar inferencias periódicas a lo largo del video. Este enfoque asegura que solo se procesan aquellos fotogramas que se ajustan a este intervalo de tiempo, optimizando así el uso de recursos y tiempo de procesamiento.

Además, el sistema está diseñado para procesar únicamente los fotogramas en los que el modelo YOLOv8 detecta la presencia de una placa vehicular. El primer modelo YOLO se utiliza para detectar vehículos en el fotograma, y solo si se encuentra un vehículo, el segundo modelo YOLO se aplica para detectar la placa dentro del recorte del vehículo. Esto significa que el procesamiento intensivo se realiza únicamente en los fotogramas que contienen posibles placas vehiculares.

El sistema no está optimizado para aplicaciones en tiempo real debido al tiempo de inferencia. Para que el sistema sea viable en escenarios de tráfico real, se requiere una reducción significativa en el tiempo de procesamiento. Por ejemplo, considerando un vehículo que se desplaza a una velocidad promedio de 30 km/h, el tiempo requerido para recorrer 100 metros es de aproximadamente 12 segundos. Comparado con el tiempo total de inferencia del sistema, este tiempo es considerablemente más corto, lo que hace que el sistema actual no sea adecuado para aplicaciones en tiempo real.

Sin embargo, es importante destacar que la implementación actual captura videos y luego realiza el procesamiento, lo que permite su uso en análisis post-procesamiento de tráfico. Este enfoque es útil para estudiar videos específicos y obtener datos detallados sobre el comportamiento del tráfico y la identificación de placas vehiculares en situaciones controladas.

Para mejorar la viabilidad del sistema y permitir su aplicación en tiempo real, se pueden considerar varias optimizaciones. En primer lugar, la implementación de una Unidad de Procesamiento Neuronal (NPU) podría acelerar significativamente el tiempo de inferencia. Las NPUs están diseñadas específicamente para acelerar las operaciones de redes neuronales, lo que puede reducir drásticamente el tiempo de procesamiento. Además, la aplicación de técnicas de cuantización puede disminuir el tamaño del modelo y mejorar la velocidad de inferencia sin sacrificar demasiado la precisión, por último, optimizar los modelos para la ejecución en dispositivos específicos también puede contribuir a mejorar los tiempos de procesamiento.

Conclusiones

El desarrollo de un sistema de superresolución utilizando arquitecturas GAN (SRGAN y ESRGAN) ha demostrado ser efectivo para mejorar la calidad de las imágenes de placas vehiculares capturadas por cámaras de seguridad no especializadas. La arquitectura ESRGAN, en particular, ha mostrado un rendimiento superior en la generación de imágenes de alta calidad, facilitando el posterior procesamiento y reconocimiento de caracteres y placas completas.

Se creó una base de datos representativa con imágenes de alta y baja resolución, utilizando el conjunto de datos "license ocr" de Roboflow. Esta base de datos fue fundamental para entrenar y evaluar los modelos de superresolución. La recopilación de imágenes de diferentes resoluciones y condiciones de iluminación permitió desarrollar un sistema robusto y adaptable a diversas situaciones.

Tras un exhaustivo proceso de evaluación, se determinó que ESRGAN es la arquitectura más adecuada para la tarea, superando consistentemente a SRGAN en términos de calidad de imagen y reducción de artefactos de compresión. La superioridad de ESRGAN se observó en diversos formatos y niveles de compresión, excepto en el formato JPG, donde SRGAN demostró un mejor desempeño.

Se implementaron tanto ESRGAN como SRGAN en Python, evaluando su desempeño en la tarea específica de superresolución de imágenes de placas vehiculares. Durante esta fase, se observó que ESRGAN ofrece mejores resultados en términos de calidad de imagen y reducción de artefactos, lo que respalda su selección como la arquitectura preferida.

Se llevaron a cabo evaluaciones cualitativas y cuantitativas para determinar la eficacia de los modelos ESRGAN y SRGAN. Utilizando métricas de precisión tanto para caracteres

individuales como para placas completas, se obtuvo una precisión de caracteres de 73.53% con ESRGAN y 67% con SRGAN, mientras que la precisión de placas completas fue de 56.6% y 51.1%, respectivamente. Estos resultados validan la superioridad de ESRGAN sobre SRGAN en varios niveles de compresión y resolución, resaltando su desempeño en el formato PNG por su capacidad de compresión sin pérdida. El análisis cualitativo reveló que ESRGAN produce imágenes más claras y detalladas, mejorando significativamente la legibilidad de las placas en variadas condiciones de resolución y compresión. Se observaron mejoras notables en nitidez y reducción de artefactos especialmente en PNG, lo que subraya la ventaja de este formato en la preservación de detalles. En consecuencia, ESRGAN se destaca como la opción más efectiva para la superresolución de imágenes de placas vehiculares, especialmente cuando se utiliza PNG.

En la implementación final del sistema de superresolución e identificación de placas vehiculares, se observó un comportamiento que confirma la eficacia de los modelos en mejorar la calidad de las imágenes de placas vehiculares. La implementación mostró que, aunque el sistema logra una mejora significativa en la precisión de reconocimiento, el tiempo de inferencia fue considerablemente alto debido a las limitaciones del hardware utilizado y la ausencia de optimizaciones avanzadas como el uso de NPU y técnicas de cuantización. Esto implica que, bajo las condiciones actuales, el sistema no es viable para aplicaciones en tiempo real. Sin embargo, es útil para el análisis de videos pregrabados, permitiendo estudios detallados del tráfico. El análisis detallado del rendimiento del sistema proporciona una base sólida para futuras optimizaciones. La implementación de mejoras como la NPU y la cuantización permitirá reducir los tiempos de inferencia y hacer posible su aplicación en tiempo real, mejorando así la viabilidad y efectividad del sistema en escenarios prácticos.

Departamento de Ingeniería Electrónica y de Telecomunicaciones

UNIVERSIDAD DE ANTIOQUIA
Facultad de Ingeniería

ANÁLISIS DE ESTRATEGIAS DE SUPERRESOLUCIÓN PARA LA IDENTIFICACIÓN DE PLACAS VEHICULARES EN IMÁGENES CON RESOLUCIÓN NO APROPIADA

PRACTICANTE: Alejandro Ocampo Rojas
ASESORES: Ricardo Andrés Velásquez Vélez

PROGRAMA: Ingeniería Electrónica
Semestre de la práctica: 2024-1

Objetivos

- Construir una base de datos representativa de placas vehiculares, que incluya tanto imágenes de baja resolución como de alta resolución.
- Seleccionar una arquitectura GAN que permita generar imágenes de placas en alta resolución a partir de imágenes de baja calidad con altos niveles de exactitud.
- Implementar un algoritmo que permita la identificación de placas vehiculares en imágenes o videos no ideales utilizando la arquitectura GAN seleccionada y lenguaje de programación Python.
- Evaluar cuantitativa y cualitativamente la efectividad del modelo GAN en la precisión del sistema de identificación de placas vehiculares.

Introducción

Planteamiento del problema

Actualmente, uno de los mayores retos en el control del tráfico y la seguridad vial es la baja resolución de las imágenes capatadas por las cámaras de seguridad. Esto dificulta la identificación precisa de las placas vehiculares, afectando la vigilancia y la aplicación de multas. Para solucionar este problema, se han explorado técnicas de superresolución de imágenes mediante Redes Generativas Adversarias (GAN). Este proyecto se enfoca en implementar y evaluar diferentes arquitecturas GAN, como SRGAN, ESRGAN y Real-ESRGAN, para mejorar la calidad de las imágenes de placas vehiculares. La mejora en la resolución permitirá una identificación más precisa, fortaleciendo las medidas de seguridad y el monitoreo del tráfico urbano.

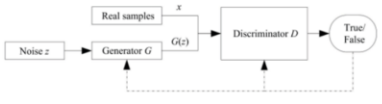


Figura 1. *Arquitectura GAN*

Metodología

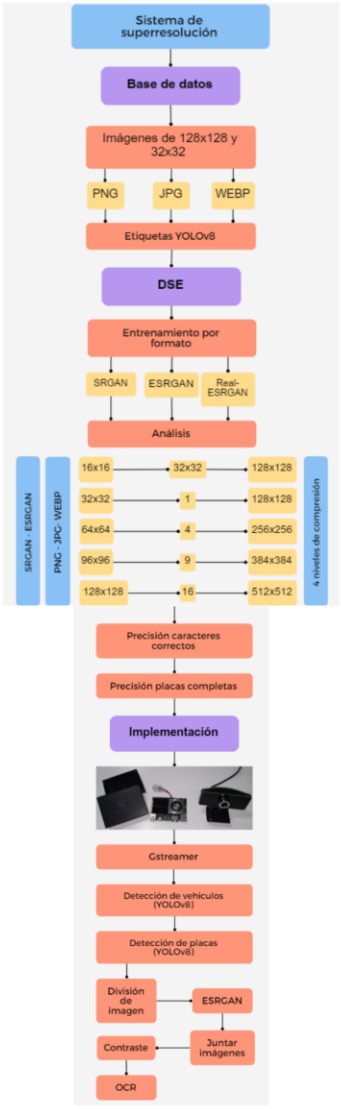


Figura 2. *Metodología*

Resultados

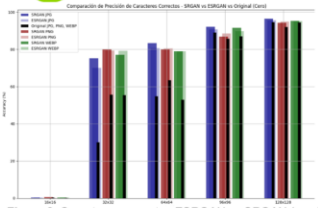


Figura 3. *Caracteres correctos ESRGAN vs SRGAN (cero)*

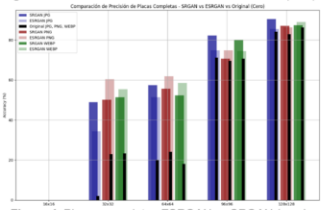


Figura 4. *Placas completas ESRGAN vs SRGAN (cero)*

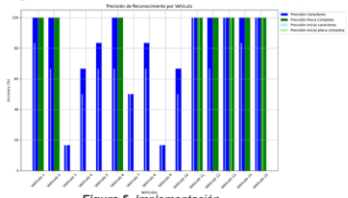


Figura 5. *Implementación*

Conclusiones

- Las arquitecturas GAN (SRGAN y ESRGAN) mejoraron la calidad de imágenes de placas vehiculares. ESRGAN destacó por su superior rendimiento en generar imágenes de alta calidad, facilitando el reconocimiento de caracteres y placas completas.
- ESRGAN logró una precisión del 73.53% en caracteres correctos y 56.6% en placas completas, superando a SRGAN, que obtuvo un 67% y 51.1% respectivamente, confirmando la superioridad de ESRGAN en compresión y resolución.
- ESRGAN produce imágenes más claras y detalladas, mejorando la legibilidad de las placas, especialmente en formato PNG debido a su compresión sin pérdida y menor cantidad de artefactos.
- La implementación mejoró la calidad y precisión de las imágenes de placas, pero el tiempo de inferencia es alto debido a limitaciones de hardware. No es viable en tiempo real, pero es útil para videos pregrabados. Optimizar con NPU y cuantización podría permitir su uso en tiempo real.

DATOS DE CONTACTO DEL AUTOR:

+57 3025618625 alejandro.ocampo2@udea.edu.co https://www.linkedin.com/in/alejandro-ocampo-rojas-a1051b215/

Referencias Bibliográficas

[1] Wang, K., Gou, C., Duan, Y., Lin, Y., Zheng, X., & Wang, F.-Y. (2017). Generative adversarial networks: introduction and outlook. IEEE/CAA Journal of Automatica Sinica, 4(4), 588-598. doi:10.1109/JAS.2017.7510583.

[2] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. New York, USA: MIT Press.

[3] Radford, A., Metz, L., & Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv:1511.06434.

[4] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. In Advances in Neural Information Processing Systems 27, Montreal, Quebec, Canada, (pp. 2672−2680).

[5] Ledig, C., et al. (2017). Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, (pp. 105-114). doi:10.1109/CVPR.2017.19.

[6] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Advances in Neural Information Processing Systems (NIPS), (pp. 1097–1105).

[7] Kim, J., Lee, J. K., & Lee, K. M. (2016). Deeply-recursive convolutional network for image super-resolution. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[8] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (pp. 770–778).

[9] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Identity mappings in deep residual networks. In European Conference on Computer Vision (ECCV), (pp. 630–645). Springer.

[10] Gross, S., & Wilber, M. (2016). Training and investigating residual nets [Online]. Retrieved from http://torch.ch/blog/2016/02/04/resnets.html.

[11] Shi, W., Caballero, J., Huszar, F., Totz, J., Aitken, A. P., Bishop, R., Rueckert, D., & Wang, Z. (2016). Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (pp. 1874–1883).

[12] Wang, X., Yu, K., Wu, S., Gu, J., Liu, Y., Dong, C., ... & Loy, C. C. (2018). Esrgan: Enhanced super-resolution generative adversarial networks. In Proceedings of the European conference on computer vision (ECCV) workshops, (pp. 0-0).

[13] Wang, X., Xie, L., Dong, C., & Shan, Y. (2021). Real-ESRGAN: Training Real-World Blind Super-Resolution with Pure Synthetic Data. arXiv e-prints. doi:10.48550/arXiv.2107.10833.

[14] Cayuela García, J. (2008). Multiresolución de Harten aplicada a la compresión de imágenes digitales: comparación, en bits, con los formatos standard JPEG y PNG.

[15] Rosario, Y. (2023). Formatos de imagen para el diseño web: una revisión. *A3manos*, *10*(20), 34-50.

[16] Singh, S. (2023). *A Comparative Evaluation of Next-Generation Image Formats on Low-Cost Mobile Hardware*. Tech. Rep. 2, New York University Abu Dhabi, Abu Dhabi, UAE.

[17] Saharia, C., Ho, J., Chan, W., Salimans, T., Fleet, D. J., & Norouzi, M. (2023). Image Super-Resolution via Iterative Refinement. IEEE Transactions on Pattern Analysis and Machine Intelligence, 45(4), 4713-4726. doi: 10.1109/TPAMI.2022.3204461.