



Warlock-Studio

– Manual Técnico del Usuario y Documentación –

Versión 2.2

Iván Eduardo Chavez Ayub

@Ivan-Ayub97 en GitHub

10 de julio de 2025

Resumen

Este documento es una guía técnica exhaustiva para Warlock-Studio 2.2. La información ha sido validada y enriquecida con un análisis profundo del código fuente para proporcionar detalles precisos sobre su arquitectura, una guía de optimización avanzada y un manual de solución de problemas robusto.

Índice

1	Introducción	3
1.1	Novedades en la Versión 2.2	3
1.2	Características Principales	3
2	Instalación y Arquitectura del Programa	3
2.1	Requisitos del Sistema	4
2.2	Estructura de Archivos y Dependencias	4
3	Guía Detallada de Modelos de IA	4
3.1	Tabla Comparativa de Modelos	4
4	Configuración y Optimización del Rendimiento	5
4.1	Parámetros Críticos de Rendimiento	5
4.2	El Fichero de Preferencias de Usuario	5
5	Guía Avanzada de Solución de Problemas	5
6	Arquitectura y Procesos Avanzados	6
6.1	Motor de Inferencia y Aceleración por Hardware	6
6.2	Sistema de Tiles y Gestión de Memoria	6
6.3	Funcionalidad de Reanudación y Checkpoints	6
6.4	Escritura Asíncrona de Fotogramas	6

1. Introducción

Warlock-Studio es una suite de mejora y escalado de medios impulsada por IA, diseñada para ofrecer resultados de alta calidad a través de una interfaz de usuario accesible. La versión 2.2 introduce mejoras clave en estabilidad, rendimiento y la experiencia del usuario, consolidándose como una herramienta robusta para creadores de contenido y entusiastas.

1.1. Novedades en la Versión 2.2

El análisis del código revela las siguientes mejoras sustanciales:

- **Gestión de Hilos y Cierres Seguros:** Se ha mejorado la gestión de concurrencia mediante Lock y RLock para garantizar que el estado del proceso se actualice de forma segura y se eviten condiciones de carrera al procesar videos. La terminación del proceso (`terminate`) ahora es más robusta.
- **Optimización de Memoria Proactiva:** Durante el procesamiento de videos largos, la aplicación ahora invoca explícitamente al recolector de basura de Python (`gc.collect()`) después de procesar lotes de fotogramas, reduciendo significativamente la probabilidad de errores de memoria insuficiente.
- **Manejo de Errores de GPU:** Se ha añadido un manejador específico para errores de memoria de la GPU durante el escalado de video. Si se detecta un error "out of memory", la aplicación reduce automáticamente el tamaño de los *tiles* y reintenta el proceso.
- **Pantalla de Bienvenida (Splash Screen):** Se ha añadido una pantalla de inicio de 10 segundos que mejora la percepción de la carga inicial de la aplicación.
- **Registro (Logging) y Diagnóstico:** La aplicación crea archivos de registro (`warlock_studio.log` y `error_log.txt`) en la carpeta de **Documentos** del usuario, facilitando el diagnóstico de problemas.

1.2. Características Principales

- **Escalado por IA:** Utiliza modelos de última generación como Real-ESRGAN, BSRGAN y SRVGGNetCompact.
- **Interpolación de Fotogramas:** Aumenta los FPS o crea efectos de cámara lenta fluidos usando modelos RIFE.
- **Reducción de Ruido:** Incluye modelos IRCNN dedicados a la limpieza de imágenes y videos.
- **Aceleración por Hardware:** Usa el motor ONNX Runtime con el proveedor DirectML (`DmlExecutionProvider`) para una aceleración por GPU compatible con DirectX 12.
- **Codificación de Video Avanzada:** Soporte para codificadores acelerados por hardware de NVIDIA (NVENC), AMD (AMF) e Intel (QSV).

2. Instalación y Arquitectura del Programa

2.1. Requisitos del Sistema

Componente	Requisito
Sistema Operativo	Windows 10 (64-bit) o posterior
Memoria RAM	8 GB (Mínimo), 16 GB (Recomendado)
Tarjeta Gráfica (GPU)	Compatible con DirectX 12. Recomendado: 4+ GB VRAM.
Almacenamiento	2 GB de espacio libre. Se recomienda un SSD para un mejor rendimiento.

Cuadro 1: Requisitos de hardware y software para Warlock-Studio 2.2.

2.2. Estructura de Archivos y Dependencias

Warlock-Studio es una aplicación autocontenida. Los siguientes componentes se incluyen en la instalación y no requieren acción por parte del usuario.

- **ffmpeg.exe:** Ubicado en la carpeta **Assets**, es el motor para toda la manipulación, codificación y decodificación de video.
- **exiftool.exe:** También en **Assets**, se utiliza para leer y escribir metadatos (EXIF, XMP), asegurando que la información original del archivo se preserve.
- **Modelos IA:** Los modelos en formato **.onnx** se encuentran en la carpeta **AI-onnx**.
- **Preferencias de Usuario:** Se guarda un archivo

3. Guía Detallada de Modelos de IA

La elección del modelo de IA es el factor más importante para la calidad y el tiempo de procesamiento.

3.1. Tabla Comparativa de Modelos

La siguiente tabla detalla el uso relativo de VRAM de cada modelo.

Modelo	Función Principal	Escala	Peso VRAM	Caso de Uso Recomendado
<i>Modelos de Reducción de Ruido (Denoising)</i>				
IRCNN_Mx1	Denoise	x1	4.0	Ruido moderado.
IRCNN_Lx1	Denoise	x1	4.0	Ruido intenso.
<i>Modelos de Escalado - Alta Calidad (Lentos)</i>				
BSRGANx4	Upscale	x4	0.6	Fotografías realistas. Excelente detalle fino.
BSRGANx2	Upscale	x2	0.7	Similar a x4 pero para un escalado menor.
RealESRGANx4	Upscale	x4	0.6	Uso general, bueno para texturas.
RealESRNetx4	Upscale	x4	2.2	Alternativa a RealESRGAN, puede ser más rápido.
<i>Modelos de Escalado - Alta Velocidad (Ligeros)</i>				

Modelo	Función Principal	Escala	Peso VRAM	Caso de Uso Recomendado
RealESR_Gx4	Upscale	x4	2.2	Escalado rápido, ideal para videos.
RealESR_AnimeX4	Upscale	x4	2.2	Optimizado para anime y dibujos animados.
Modelos de Interpolación de Fotogramas (Solo Video)				
RIFE	Interpolar	N/A	N/A	Máxima calidad de interpolación.
RIFE_Lite	Interpolar	N/A	N/A	Versión más rápida, ideal para GPUs con < 4 GB VRAM.

Cuadro 2: Guía de selección de modelos de IA y su impacto en VRAM.

4. Configuración y Optimización del Rendimiento

4.1. Parámetros Críticos de Rendimiento


- **Input Resolution %:** El ajuste más efectivo para la velocidad. Reduce la resolución antes de procesarla con IA. Un valor entre **50 % y 75 %** suele ser ideal.
- **GPU VRAM Limiter (GB):** Defina la VRAM de su GPU. Se usa para calcular el tamaño de los *tiles* de procesamiento y evitar errores de memoria.
- **AI Multithreading:** Solo para videos. Procesa varios fotogramas en paralelo. Acelera el proceso pero consume más VRAM y CPU.
- **AI Blending:** Combina la imagen original con la procesada. Útil para reducir artefactos cuando se usa un *Input Resolution* bajo.

4.2. El Fichero de Preferencias de Usuario


El archivo


Warlock-Studio2.2UserPreference.json guardasuconfiguracin.

5. Guía Avanzada de Solución de Problemas

 Advertencia

La causa **Número 1** de errores son los **caracteres especiales** en rutas y nombres de archivo. Evite usar: ' , " , @ , # , \$, % , & , * , [,] , ? , etc..

 **Error: "FFmpeg encoding failed: Invalid argument"**
Causa: Nombre de archivo o ruta no válida. **Solución:** Renombre el archivo y/o la carpeta eliminando caracteres especiales.

 **Error: "out of memory" o cierre inesperado**
Causa: La GPU se quedó sin memoria de video (VRAM). **Solución:**

1. Reduzca el **VRAM Limiter** a un valor igual o inferior al de su GPU.
2. Baje el **Input Resolution %** a 75 % o menos.
3. Para videos, disminuya los hilos de **AI Multithreading** o apáguelo.

4. La aplicación intentará recuperarse de este error automáticamente.

Error: "cannot convert float NaN to integer"

Causa: Timeout del driver de la GPU, a menudo por sobrecarga o sobrecalentamiento. **Solución:** Reinicie el proceso **sin borrar la carpeta de fotogramas**. El programa reanudará el trabajo donde se quedó.

🔊 **Problema: Video de salida sin audio**

Causa: El video original no tenía audio, se usó un modo *Slowmotion* o el códec de audio era incompatible. **Solución:** El programa intenta copiar el audio, si falla, lo re-codifica a AAC. Si todo falla, guarda el video sin audio. Usar `.mkv` en la salida puede ayudar.

❓ **Problema: La aplicación no se abre**

Causa: Configuración corrupta, falta de permisos o error del entorno. **Solución:**

1. Elimine el archivo

6. Arquitectura y Procesos Avanzados

6.1. Motor de Inferencia y Aceleración por Hardware

Warlock-Studio utiliza **ONNX Runtime** con el proveedor **DirectML** (`DmlExecutionProvider`). Este traduce las operaciones de IA a llamadas de **DirectX 12**, garantizando una amplia compatibilidad con GPUs de NVIDIA, AMD e Intel.

6.2. Sistema de Tiles y Gestión de Memoria

Para manejar archivos de alta resolución, la aplicación divide cada fotograma en fragmentos (*tiles*). El tamaño de estos se calcula dinámicamente usando el **VRAM Limiter**. Además, se invoca al recolector de basura de Python (`gc.collect()`) para forzar la liberación de memoria y garantizar estabilidad.

6.3. Funcionalidad de Reanudación y Checkpoints

Si un proceso de video se interrumpe, los fotogramas procesados se guardan. Al reiniciar la tarea, la función `check_video_upscaling_resume` detecta estos archivos y continúa el trabajo desde donde falló, ahorrando tiempo.

6.4. Escritura Asíncrona de Fotogramas

Durante el escalado de video, los fotogramas procesados se envían a un hilo de escritura separado. Esto permite a la GPU procesar el siguiente lote sin esperar a que la escritura en disco (más lenta) termine, maximizando el rendimiento.