Análisis y aplicación de la técnica ONE SHOT GAN

Trabajo final para curso de Redes Neuronales Generativas Profundas: Fundamentos y resolución de problemas

Rafael Rosa
MEDIA, CURE
INCO, FING
La Paloma, Uruguay
rafael.rosa.uy@gmail.com

***Resumen*—En este trabajo se analiza la aplicación de la técnica denominada ONE SHOT GAN (OSG) para la creación de imágenes a partir de una única imagen o conjunto de fotogramas de video de entrenamiento, utilizando redes neuronales generativas profundas (GAN). La técnica propone una configuración particular del generador y del discriminador, así como una evaluación particular de las funciones de pérdida. Las métricas de evaluación de resultados evalúan la calidad y diversidad de las imágenes obtenidas. Se aplicó el modelo a cuatro sets de datos y con diferentes parametrizaciones, resultando en trece corridas diferentes. En resumen, la técnica de OSG resulta ser adecuada para situaciones en las que se cuenta con un número limitado de imágenes o videos de entrenamiento. Asimismo, la técnica de OSG es capaz de crear imágenes de alta calidad con buena diversidad. Al momento de escribir este informe, el modelo no se evalúa como apropiado para producir imágenes sintéticas para el entrenamiento de una CNN de detección de tortugas.**

Keywords—ONE SHOT; tortugas; monitoreo; redes neuronales generativas profundas

I. INTRODUCCIÓN

El objetivo de la técnica de ONE SHOT GAN (OSG) es generar imágenes diversas a partir de una única imagen o conjunto de fotogramas de un video utilizando Redes Generativas Adversarias (GAN). La técnica permite la síntesis de imágenes realistas con contenido y paisaje variable, preservando el contexto de la muestra original [1].

De acuerdo con los resultados presentados en [1], la técnica de OSG logra mayor diversidad y calidad de síntesis que otras técnicas de generación de imágenes como los son SinGAN [3] y Fast GAN [2], a partir de muestras únicas de entrenamiento.

Se eligió la aplicación y análisis de esta técnica como primera aproximación al problema de conteo de tortugas marinas registrados en videos obtenidos con vuelos no tripulados sobre la costa de Cerro Verde (Rocha). El desarrollo de un sistema capaz de realizar este conteo es el objetivo principal del trabajo de investigación a realizar en el marco de la presente maestría. Es posible encontrar antecedentes de aplicaciones de este tipo para tortugas en [5] y [6], entre otros estudios publicados. En el caso particular de la aplicación para la costa de Cerro Verde, al momento de realizar este trabajo, se cuenta con pocas imágenes que muestren tortugas marinas. Por lo tanto, OSG resulta ser una técnica atractiva dada su capacidad de generación de imágenes a partir de una única imagen o conjunto de fotogramas de un video como datos de entrenamiento.

II. MARCO TEÓRICO Y BREVE RESEÑA BIBLIOGRÁFICA SOBRE GANS Y CNNS

El aprendizaje profundo (DL) busca descubrir modelos que sean ricos, capaces de jerarquizar características y representen distribuciones de probabilidad de los tipos de datos que se utilizan en aplicaciones de inteligencia artificial (IA) [10].

El aprendizaje supervisado (AS) es el caso más común de DL [9]. Durante el entrenamiento, el modelo produce vectores de valores que corresponden a cada categoría del aprendizaje objetivo. Posteriormente se utiliza una función objetivo para calcular el error entre los valores devueltos por el modelo y los valores del set de datos reales. El aprendizaje no supervisado (ANS) busca que el modelo pueda aprender a través de estudiar un set de datos que no están categorizados como aprendizaje objetivo [11]. Las GAN son un caso de ANS en el que un algoritmo generativo busca generar una distribución $p_g(x)$ que se aproxime a una distribución desconocida de datos $p_{data}(x)$.

La base del aprendizaje de una GAN está en el juego de minimización y maximización, que se genera entre el Generador (G) y el Discriminador (D) y se representa a través de la siguiente ecuación:

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))].$$

El objetivo de optimización de la GAN es maximizar la ecuación para el D y minimizar la ecuación para el G. En esta ecuación E_x es el valor esperado de acuerdo con la distribución de los datos reales ($p_{data}(x)$), E_z es el valor esperado de acuerdo con la distribución del ruido del espacio latente $p_z(z)$. Para el proceso de optimización el D producirá una D(x), la cual representa la probabilidad de que x sea o un dato real o sea el producto de la distribución generada por el G ($p_g(x)$). D(x) será igual a 1 si el D considera que la muestra es real y será 0 si el D considera que la muestra es falsa. En el dominio (0,1), la ecuación será máxima cuando D(x) = 1 y D(G(z) = 0, o sea, cuando el D toma la decisión correcta. G(z) representa una imagen generada por el generador, o sea, una imagen falsa. Para el proceso de optimización de G, el objetivo es que D(G(z)) = 1, es decir que el D no perciba que la muestra G(z)

es falsa. En este caso, el segundo término de la ecuación se minimiza en el dominio (0,1). Es posible demostrar que el criterio de entrenamiento tiene un mínimo global cuando $p_g = p_{data}$, lo cuál haría que el D devuelva una probabilidad de 0.5, mostrando no poder distinguir entre una muestra real y una muestra falsa [10].

En el caso de tratamiento de imágenes, las imágenes que ingresan a un modelo de DL ingresan en forma de distribuciones de probabilidad de pixeles. De esta manera es posible tratar las imágenes con un enfoque probabilístico. Para una imagen x compuesta por n*n píxeles, se estima la distribución p(x) como el producto de las distribuciones sobre cada uno de los pixeles [13]. En el caso de las imágenes a color, la probabilidad de cada píxel está compuesta por la probabilidad en cada uno de los tres canales de color: R, G y B. A cada píxel se le asigna la probabilidad de tener determinado color e intensidad.

La jerarquización de las características de las imágenes permite que los modelos puedan representar el input de manera progresiva. En la primera capa del aprendizaje, los modelos intentan capturar la existencia de bordes en zonas particulares de la imagen. La segunda capa intenta capturar arreglos en estos bordes, la tercera capa intenta unir estos bordes en estructuras similares a objetos reconocibles y las capas subsecuentes detectan objetos como combinaciones de estas estructuras [9].

En cada iteración del proceso de aprendizaje, tanto el G como el D tienen un costo que se genera por el error en el que incurrieron. El objetivo para lograr la optimización es el de minimizar este costo, en el caso de D la evaluación del costo hace que distinga de mejor manera entre las muestras reales y falsas mientras que en el caso de G dicha evaluación hace que genere muestras falsas más próximas a las reales. De acuerdo con [11], en la formulación original de las GANs el costo para el D se define como el Negative log-likelihood que el D asigna a las muestras que recibe en comparación con las muestras reales. El costo del G se define como el opuesto del costo del D. En el caso de problemas de optimización con dos clases (1 y 0), es posible utilizar el caso particular de Binary Cross Entropy como función de pérdida para la optimización del aprendizaje. Según [12], el hecho de que el G y el D necesiten maximizar y minimizar la misma cross entropy respectivamente, genera que el gradiente del G desaparezca cuando el D logra rechazar muestras falsas con alta confianza. La forma de solucionar este problema es cambiar el objetivo de la función de costo. Entonces, en el jugo de minimax, el G minimiza la probabilidad de que el D haga predicciones correctas.

G y D son funciones diferenciables representadas por perceptrones en múltiples capas, siendo $G = G(z, \theta_g)$ y $D = D(z, \theta_d)$ [10]. Los parámetros θ_i representan los pesos que se asignan a cada perceptrón en sus funciones de activación y conectan a las capas entre sí. La evolución del aprendizaje se basa en la actualización de los pesos. El ajuste se realiza a través del proceso de backpropagation del error que se generó durante cada etapa de aprendizaje. El proceso consiste en propagar, para todas las capas de la red neuronal, los gradientes con respecto a los pesos entre los datos generados y los datos objetivo, comenzando por la capa del output y llegando a la capa del input [9]. En este proceso se calculan gradientes que indican al modelo cómo deberán variar los pesos para que los gradientes sean minimizados en próximas etapas de aprendizaje.

De acuerdo con [8], la arquitectura de las CNN combina tres conceptos: local receptive field, pesos compartidos (shared weights) y sub-sampling espacial o temporal. Los local receptive fields cumplen la función de extraer características gruesas de las imágenes (primera etapa de la jerarquización de una imagen) que luego serán combinados por las capas subsecuentes para detectar características más complejas. Las unidades (pixeles) en cada capa se organizan en planos que comparten los pesos y se denominan feature maps, el set de pesos es el núcleo de la operación de convolución entre dos capas. Cada unidad cuenta con un receptive field, el cual es el dominio de cada unidad, situado en la capa anterior del modelo. Cada feature map en una capa utiliza sus pesos compartidos y bias característicos. Con el objetivo de que el aprendizaje sea generalizado y minimizar la influencia de la posición de algunas características dentro de cada imagen, posteriormente a la operación de convolución entre dos capas, las CNNs utilizan la técnica de sub-sampling entre las dos capas siguientes. En este proceso se realiza un promedio de las unidades que vienen de la capa anterior, reduciendo un número determinado de unidades a una única unidad, con un peso y bias asignado y el pasaje por una función no lineal, como forma de conexión entre las capas. Este proceso de convolución y sub-sampling se repite en las CNNs hasta llegar una capa de output que está completamente conectada (full connection), tal como se muestra en el ejemplo de LeNet-5 en [8]. Esta capa de output calcula una penalidad que surge de la distancia entre la unidad que recibió la capa y una unidad de referencia para cada clase. Por lo tanto, la función de pérdida de la CNN debería diseñarse de manera tal que lleve la configuración de la última capa de la red a minimizar esta penalidad. En el caso de LeNet-5, la función de pérdida utilizada es Maximum Likelihood Estimation criterion (MLE) y el cálculo del gradiente de esta función con respecto a todos los pesos en las capas se realiza a través de backpropagation [8].

III. ONE SHOT GAN

A. Descripción de la estructura de OSG y de la evaluación de pérdidas

El modelo de OSG es capaz de generar imágenes a partir de una única imagen o conjunto de fotogramas, a partir de un generador que toma la imagen inicial como dato inicial. Esta característica del modelo se denomina unconditional single-stage GAN. Es decir, un modelo que genera imágenes únicamente con algunos argumentos en el input del modelo y con un generador único que genera las imágenes a partir del ruido inicial del problema. Los desarrolladores de OSG describen que el modelo es capaz de generar imágenes que son significativamente diferentes de la muestra inicial de

entrenamiento, con buena calidad, alta diversidad y preservando su contexto [1].

La estructura de OSG tiene la particularidad de contar con un discriminador con dos ramas. Cada una de las ramas tiene una función de evaluación particular: una de ellas evalúa la distribución del contenido de la imagen y la otra rama evalúa el paisaje de la imagen. La individualización del contenido se puede entender como el objeto que se extraerá al generar una máscara de la imagen, como podría ser las personas relevantes en la imagen o el "sujeto" de la imagen. El paisaje de la imagen se entiende como todo el contexto en el que se posiciona el contenido. Para el caso de los conjuntos de fotogramas de videos, en cada fotograma el contenido presenta variaciones en su posición y el enfoque que se le da en la imagen. Ambas ramas del discriminador actúan por separado y de esta manera contribuyen a brindar información más específica al generador y a minimizar la posibilidad del overfitting que podría generarse debido al tamaño reducido de la muestra de entrenamiento.

Otra de las particularidades de la estructura de OSG es la técnica de regularización de diversidad del generador (diversity regularization technique). Esto hace que el generador genere imágenes de alta calidad que son significativamente diferentes de las imágenes de entrenamiento. El concepto detrás de esto es hacer que el generador produzca imágenes sin basarse en la distancia entre ellas en el espacio latente (latent space), entendido como el espacio de variables que contienen las características relevantes de la imagen utilizada como input del modelo al ingresar al generador. En OSG se propone estimar la distancia en el espacio de características (feature space), entendido como el espacio en el que se encuentra el output de las capas intermedias del generador. De esta manera el generador capta diferentes características de la imagen en diferentes capas intermedias y genera mayor diversidad de imágenes.

Tal como se muestra en la Figura 1, el discriminador funciona de la siguiente manera:

- Una primera etapa que extrae las características de bajo nivel de la imagen (low level features).
- Una capa intermedia previo a dividir el análisis en dos ramas. Esta capa intermedia se utiliza para aprender las características más relevantes de la imagen.
- Una rama de contenido ($D_{content}$) que sólo evalúa el contenido de la imagen sin importar el paisaje. Se obtiene el contenido de las representaciones intermedias utilizando aggregating spatial information, lo cual se entiende como una técnica que combina características de diferentes regiones o pixeles de la imagen utilizada como input y las compacta en regiones más pequeñas. Esto se hace para captar patrones globales de la imagen inicial. Esta combinación de características se realiza a través de average pooling, es decir que se divide el conjunto de características (feature map) analizadas de la imagen en diferentes subregiones que no se superponen y se evalúa el valor promedio de las características en esta subregión. De esta manera se reduce el tamaño del conjunto de características de la imagen, sin perder el contenido o valor de estas características. Las características pueden ser puntos notorios como bordes o esquinas o aspectos visuales como formas o texturas.
- Una rama de paisaje (D_{layout}) que únicamente evalúa la información espacial. Esta rama trabaja utilizando la técnica de aggregating channels con una convolución de 1 x 1. Esto significa que se aplican los pesos del filtro del discriminador a cada uno de los canales de información de las imágenes de ingreso, manteniendo sus dimensiones y reduciendo los canales de información. Esto permite una mayor combinación de las características de la imagen.

Las funciones de pérdida del discriminador están compuestas por tres partes: la pérdida de la etapa de low level features ($L_{low\text{-}level}$), la pérdida de la rama de contenido ($L_{content}$) y la pérdida de la rama de paisaje (L_{layout}). La decisión del discriminador está compuesta por estos tres términos, otorgándole un peso de 2 al término $L_{low\text{-}level}$ con el objetivo de equilibrar la incidencia de cada uno de los detalles de la imagen en la decisión del discriminador.

El generador trabaja en un único dominio (una única imagen de input), por lo cual se mantiene dentro de este dominio para generar las imágenes. Para la estimación de las pérdidas del generador, se incorpora un término de diversity regulation loss (L_{DR}), el cuál evalúa las distancias entre las imágenes generadas dentro de este dominio (que deberían ser iguales entre todas las imágenes) y también la distancia con respecto a la imagen inicial, la cual debería ser significativamente más grande que las distancias entre imágenes generadas.

B. Descripción de los parámetros de evaluación de resultados

Los parámetros de evaluación de la calidad de las imágenes generadas propuestos por los desarrolladores de OSG son los siguientes:

- FID (SIFID). Para revisar la calidad de las imágenes generadas.
- LPIPS promedio y MS-SSIM entre pares de imágenes. Evalúan la diversidad de las imágenes generadas.
- (Dist. to train). Para verificar que el modelo no reproduce únicamente la imagen de input. Se revisa el LPIPS promedio de una imagen con la imagen más cercana en el conjunto de entrenamiento, aplicándole la misma técnica de aumento de datos (data augmentation) que se utilizó durante el entrenamiento.

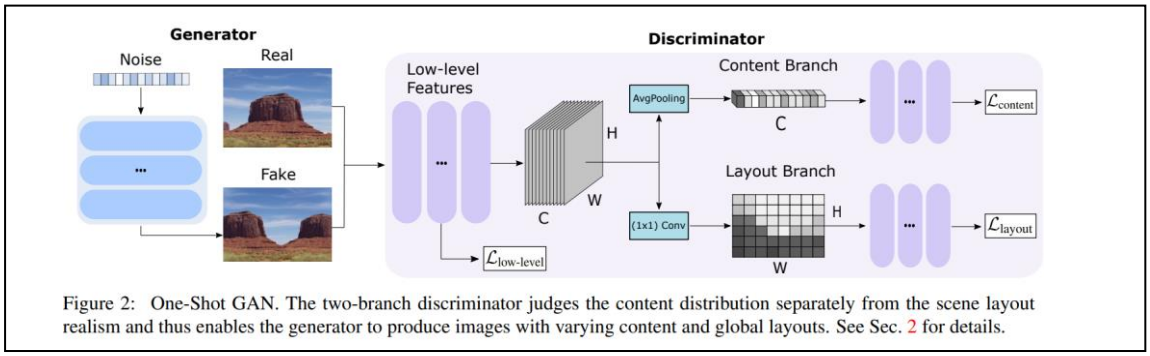


Figure 2: One-Shot GAN. The two-branch discriminator judges the content distribution separately from the scene layout realism and thus enables the generator to produce images with varying content and global layouts. See Sec. 2 for details.

Figura 1. Extracto de [1] en el que se muestra de manera gráfica la estructura de OSG.

Figura 2. Ejemplo de las imágenes utilizadas en los sets de datos. Arriba izquierda: imagen sólo de agua. Arriba centro: imagen de espuma. Arriba a la derecha: imagen de rocas. Abajo izquierda y centro: imagen con tortuga (señalada con círculo amarillo en este informe). Abajo derecha: imagen del set Christoff.

IV. APLICACIÓN DEL MODELO

A. Sets de datos utilizados

Se utilizaron cuatro sets de datos para aplicar el modelo. Tres de estos sets de datos surgen a partir de la elección de una serie de fotogramas de un video base en el que se observa la presencia de tortugas. El cuarto set de datos está compuesto por una única imagen que no está relacionada con los videos de tortugas. Este set de datos se utiliza como referencia para la evaluación de la calidad de generación de imágenes. En la Tabla 1 se describen los cuatro sets de datos.

En la Figura 2 se muestran cinco fotogramas ejemplo incluidos en los tres primeros sets de datos (roca, agua, espuma y tortuga) y la imagen utilizada en el set de datos Christoff.

B. Objetivos de la utilización de los diferentes sets de datos

La utilización de estos diferentes sets de datos tiene diferentes objetivos. Como se mencionó anteriormente, el objetivo principal de este estudio es evaluar la utilización del modelo de OSG para la generación de imágenes sintéticas que puedan ser utilizadas para el entrenamiento de un modelo de CNN de detección de tortugas.

Los tres diferentes sets de datos AA, AB y AC cuentan con diferentes porcentajes de fotogramas conteniendo tortugas. Esa variación se realizó con el objetivo de evaluar la capacidad del modelo de detectar la presencia de tortugas en el set de entrenamiento y subsecuentemente generar imágenes que contengan tortugas. Asimismo, se busca evaluar la utilización de diferentes funciones de pérdida para el modelo. Estas diferentes funciones de pérdida son: Binary Cross Entropy (BCE), Wasserstein Distance (WGAN) y HINGE. Otra evaluación que se realizó fue la aplicación de OSG con diferentes resoluciones de salida. Debido a algunas limitaciones encontradas con el hardware utilizado (GPUs con 12 gb de capacidad), sólo fue posible evaluar resoluciones de salida de 320 x 192 y 512 x 256 pixeles en el caso de los videos de tortugas y de 320 x 256 y 512 x 384 en el caso de la imagen Christoff.

El set de datos denominado Christoff se utilizó para evaluar la performance del modelo al contar con una imagen con mayor contenido y diversidad que los fotogramas de tortugas. Asimismo, se evaluó la capacidad de generación del modelo para diferentes resoluciones de imagen de salida y diferentes épocas de entrenamiento.

C. Parametrización de las diferentes aplicaciones del modelo

A continuación, se muestra las diferentes parametrizaciones que se realizaron en la aplicación del modelo. El objetivo fue aplicar el modelo en diferentes escenarios para poder evaluar su performance. En la Tabla 2 se muestran las diferentes corridas que se realizaron del modelo, mostrando los parámetros que variaron para cada corrida:

Luego de la etapa de entrenamiento en el número de épocas señalados para cada corrida, se aplicó la generación de imágenes para los pesos obtenidos en las siguientes épocas de entrenamiento: 20000, 50000, 100000, 130000, 150000, 200000, 220000 y 250000. Para cada época de entrenamiento, el modelo genera 100 imágenes por defecto. También se calcularon las métricas correspondientes a cada una de estas épocas de generación de imágenes para cada corrida del modelo.

El modelo cuenta con una serie de parámetros adicionales, los cuales no fueron explorados en este análisis. Algunos de estos parámetros que el modelo permite configurar son:

- Utilizar el modelo con o sin máscaras (en este caso se utilizó siempre sin máscaras)
- Utilizar el modelo con differentiable augmentation (fijado como Verdadero)
- Dimensión del vector de ruido (64 por defecto)
- Tasa de aprendizaje del generador y el discriminador (0.0002 por defecto)
- beta1 y beta2 para el optimizador Adam (0.5 y 0.999 por defecto respectivamente)
- Qué seed random utilizar como arranque (22 por defecto)
- Utilizar o no la función de Diversity Regulation (sí utilizar por defecto)
- Lamda para el DR (0.15 por defecto)
- Probabilidad de data augmentation (fijado en 0.7)
- Probabilidad de Feature Augmentation para Contenido y Layout (0.4 por defecto en ambos casos)
- Desactivar el promedio móvil exponencial para los pesos del generador (sí desactivar por defecto)
- Decaimiento de los promedios móviles (0.9999 por defecto)
- Épocas para soft mask warmup de Bernoulli (15000 por defecto)
- Norma para utilizar en Generador y Discriminador (ninguna por defecto)
- Multiplicador de canales para Generador y Discriminador (32 por defecto en ambos casos)

D. Tiempos de cómputo

De acuerdo con el set de datos utilizado, la resolución y la función de pérdida se registraron diferentes tiempos de cómputo en promedio. En la Tabla 3 se muestran los tiempos de cómputo observados y se calculan relaciones rápidas entre una corrida de referencia y otras de características comparables. De esta tabla surge que para los sets de datos de tortugas la aplicación de la función de pérdida BCE es la más rápida y la WGAN es la más lenta. Para el caso del set de datos Christoff no se observa una variabilidad del tiempo de cómputo en función de la función de pérdida. Para sets de datos similares, se observa un aumento significativo del tiempo de cómputo al aumentar la resolución de las imágenes de salida.

Nombre	Cantidad de fotogramas	Resolución fotogramas	Fotogramas sólo agua (%)	Fotogramas espuma (%)	Fotogramas rocas (%)	Fotogramas tortugas (%)
AA	72	3840 x 2160	15	14	14	58
AB	72	3840 x 2160	4	4	4	88
AC	72	3840 x 2160	0	0	0	100
Christoff	1	4320 x 3240	-	-	-	-

Tabla 1. Diferentes sets de datos utilizados en el análisis.

Nombre corrida	Dataset	Resolución	Épocas	Loss function
AA_330_BCE	AA	320 x 192	150,000	BCE
AA_330_HINGE	AA	320 x 192	150,000	HINGE
AA_330_WGAN	AA	320 x 192	150,000	WGAN
AC_330_BCE	AC	320 x 192	250,000	BCE
AA_500_BCE	AA	512 x 256	150,000	BCE
AA_500_HINGE	AA	512 x 256	150,000	HINGE
AA_500_WGAN	AA	512 x 256	150,000	WGAN
AB_500_BCE	AB	512 x 256	240,000	BCE
AC_500_BCE	AC	512 x 256	250,000	BCE
Christoff_330_BCE	Christoff	320 x 256	150,000	BCE
Christoff_500_BCE	Christoff	512 x 384	250,000	BCE
Christoff_500_HINGE	Christoff	512 x 384	250,000	HINGE
Christoff_500_WGAN	Christoff	512 x 384	250,000	WGAN

Tabla 2. Resumen de corridas realizadas para el análisis de OSG.

Nombre corrida	Tiempo de cómputo (seg por época)	Relación con referencia
AA_330_BCE	0.4	1.0
AA_330_HINGE	1.3	3.3
AA_330_WGAN	1.7	4.3
AC_330_BCE	0.4	1.0
AA_500_BCE	1.6	1.0
AA_500_HINGE	1.7	1.1
AA_500_WGAN	2.5	1.6
AB_500_BCE	1.16	0.7
AC_500_BCE	1.7	1.1
Christoff_330_BCE	0.44	1.0
Christoff_500_BCE	1.15	2.6
Christoff_500_HINGE	1.13	2.6
Christoff_500_WGAN	1.16	2.6

Tabla 3. Tiempos de cómputo observados en las diferentes corridas. Cada color muestra un grupo de corridas entre las cuales se comparan los tiempos de cómputo.

E. Funciones de pérdida

Los modelos de GAN utilizan dos funciones de pérdida (FP), una función para el Generador y una función para el Discriminador. Tanto el Generador como el Discriminador son entrenados de manera simultánea y actualizan sus costos de manera independiente.

El modelo de OSG brinda la posibilidad de graficar las FP para cada una de las corridas. Devuelve la información diferenciando la evolución de las FP de la siguiente manera: pérdida para el Generador (G), pérdida para el Discriminador frente imágenes falsas (Dfake) y pérdida para el Discriminador frente a imágenes reales (Dreal).

En la Figura 3 se muestran las gráficas que se obtuvieron para analizar la evolución de las FP para cada corrida.

Para las corridas que se realizaron aplicando WGAN como FP, se observa que estas funciones tuvieron un comportamiento inesperado y que no tiene a una convergencia ni minimización de estas. Esto es coincidente con los resultados obtenidos para las corridas aplicadas con WGAN. En estas corridas se obtuvieron imágenes que no tienen un mínimo de calidad ni estructura aceptables.

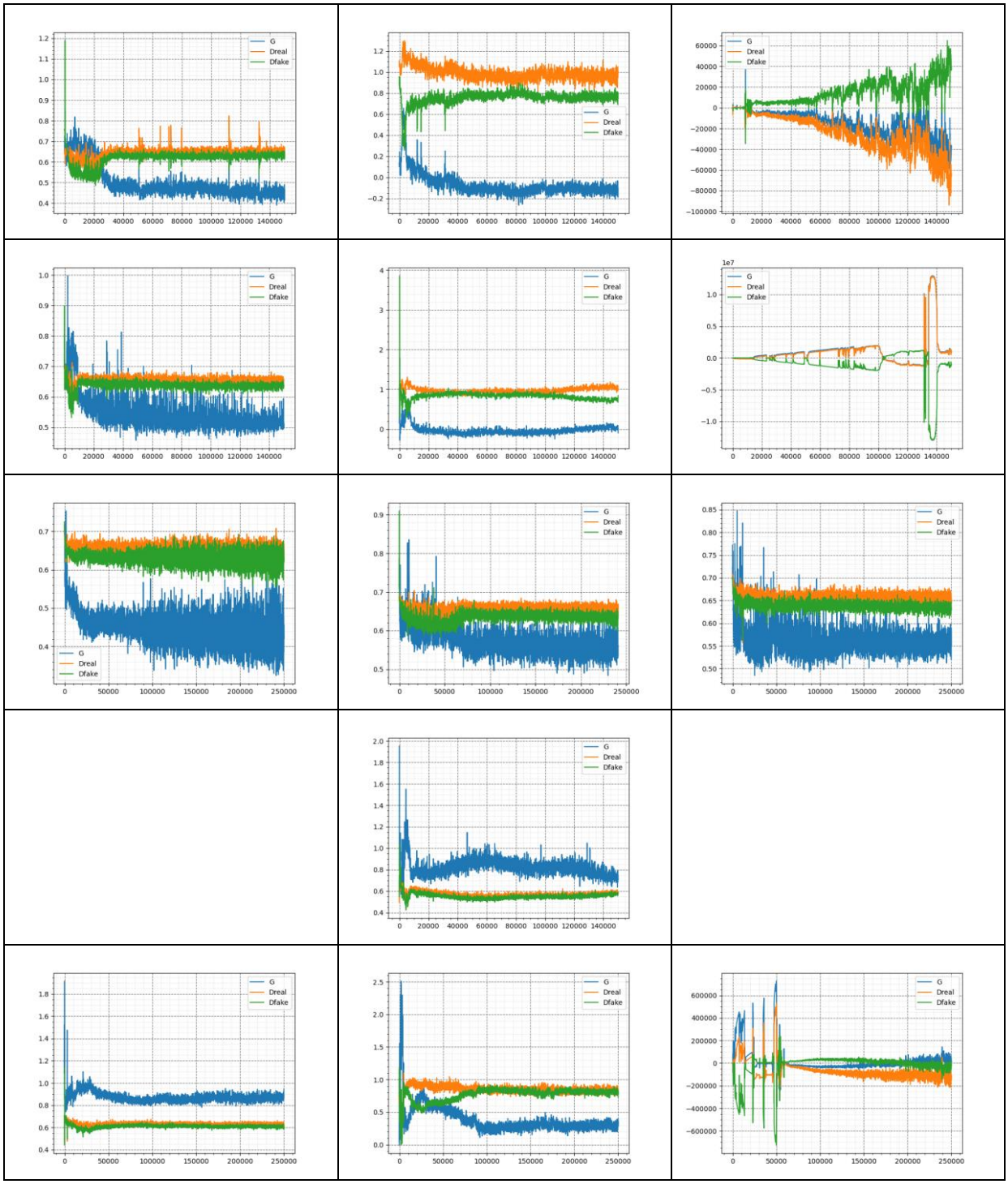
Para las corridas realizadas con el set de datos AA y la FP HINGE, se observa que la FP tiene un valor absoluto superior a la FP BCE. En ambas corridas con FP HINGE se observa que no hay una convergencia tan ajustada entre la FP Dreal y Dfake como sí lo hay en las corridas con la FP BCE. Los valores de FP HINGE tienden a valores asintóticos (con una amplitud de 0.1 aproximadamente) pero no se superponen entre sí. En el caso de la corrida con el set de datos Christoff, la FP HINGE sí muestra una superposición en los valores entre Dreal y Dfake para las épocas posteriores. Para la corrida AA_500_BCE, la FP Dfake tiene una caída a partir de la época 110000 mientras que la Dreal tiene un aumento en el mismo rango de épocas. Este resultado es interpretable a través de una evaluación cualitativa de las imágenes obtenidas, en principio se entiende esto como un indicio de que el entrenamiento del D mejora a partir de esa época. En el caso de la FP del G, en los tres casos se nota una tendencia a valores asintóticos, dando indicios de que a partir de cierta época la FP del D no va a disminuir más su valor absoluto. Estos umbrales son las épocas 100000 y 12000 en el caso de las corridas AA_330_HINGE y AA_500_HINGE respectivamente y la época 200000 en el caso de la corrida Christoff_500_HINGE.

En el caso de las corridas realizadas con el set de datos AA y la FP BCE, se observa que la FP del G tiende a disminuir su valor absoluto con el aumento de las épocas. Las FP del D tienen valores similares durante todas las épocas y tienen un comportamiento asintótico. En el caso de la corrida AC_330_BCE se observa que todas las FP aumentan su amplitud con el aumento de las épocas. En particular, la FP del G comienza a tener oscilaciones y un aumento de su amplitud notorio a partir de la época 75000 aproximadamente.

Para las corridas AB_500_BCE y AC_500_BCE se observa que las FP del D decrecen con el aumento de las épocas. Para el G, además de un decrecimiento del valor absoluto con las épocas, se observa una disminución en la amplitud de las oscilaciones del valor de la FP. Esto podría ser un indicio de una convergencia con el aumento de épocas.

En el caso de las corridas realizadas con el set de datos Christoff y resolución alta (Christoff_500_BCE y Christoff_500_HINGE), se observa una caída en el valor absoluto de las FP consistente con el aumento de épocas. Este resultado es esperable y muestra una mejora en el entrenamiento de la GAN. Para la corrida Christoff_330_BCE se observa un aumento (disminución) en las FP del G (D) entre las épocas 40000 y 80000. Luego de este rango de épocas, la FP para el G comienza a bajar consistentemente con el aumento de épocas y la FP del D comienza a aumentar. El resultado para el D no es el esperable dado que sería esperable que disminuyan.

Como resultado general, se observó un comportamiento esperable para las FP en la mayoría de los casos. En algunos casos se observaron aumentos en el valor de las FP al aumentar las épocas, lo cual no sería esperable si las componentes de la red (G y D) estuvieran mejorando su entrenamiento. Las oscilaciones con amplitud alta en los valores de las FP tampoco parecen ser un indicador de buen entrenamiento. En el caso de la corrida AC_330_BCE, estas oscilaciones son notorias. A primera vista, la FP HINGE mostró menor amplitud en las oscilaciones de su valor absoluto, lo cuál es un indicio de mayor capacidad de minimizar los gradientes al pasar las épocas.



Figurara 3. Funciones de pérdida para las corridas analizadas en este estudio. La referencia de a qué corrida corresponde cada gráfica se presenta a continuación:

AA_330_BCE	AA_330_HINGE	AA_330_WGAN
AA_500_BCE	AA_500_HINGE	AA_500_WGAN
AC_330_BCE	AB_500_BCE	AC_500_BCE
	Christoff_330_BCE	
Christoff_500_BCE	Christoff_500_HINGE	Christoff_500_WGAN

F. Análisis de métricas

Aplicando las herramientas provistas por el modelo, se computaron las métricas mencionadas en la sección III.B para todas las corridas del modelo. A continuación, se presenta un análisis de las siguientes métricas:

- SIFID. Representa la calidad de las imágenes generadas. Cuanto menor es el valor absoluto de SIFID, se interpreta como una imagen de mayor calidad.
- Dist. to train (DTT). Representa la diversidad de las imágenes generadas. Cuanto mayor es el valor absoluto de (Dist. to train), se interpreta como una imagen de mayor diversidad.

La comparación de las métricas entre diferentes corridas brinda la oportunidad de interpretar algunos resultados, como la influencia de la función de pérdida en la performance del modelo, o la influencia de la resolución de las imágenes de salida. También permite comparar resultados entre diferentes sets de datos.

Para el caso de las corridas realizadas con la función de pérdida WGAN, no se obtuvieron resultados aceptables. Es decir que las imágenes generadas no llegan a representar una imagen de una calidad mínima. Por lo tanto, las corridas realizadas con WGAN no son incluidas en este análisis.

Se compararon las métricas obtenidas para los siguientes dúos o tríos de corridas:

- AA_330_BCE vs. AA_330_HINGE. Se busca evaluar la influencia de la función de pérdida en los resultados.
- AA_500_BCE vs. AA_500_HINGE. Idem anterior.
- AA_500_BCE vs. AB_500_BCE vs. AC_500_BCE. Se busca evaluar cómo responden las métricas para diferentes sets de datos.
- AC_330_BCE vs. AC_500_BCE. Se busca evaluar la influencia de la resolución en los resultados.
- Christoff_500_BCE vs. Christoff_500_HINGE. Se busca evaluar la influencia de la función de pérdida en los resultados para el caso del set de datos más regular (una única imagen).
- Christoff_330_BCE vs. Christoff_500_BCE. Se busca evaluar la influencia de la resolución en los resultados.

De acuerdo con lo que los autores del modelo señalan, es esperable que las imágenes generadas aplicando un conjunto de pesos para épocas tempranas tengan una mayor diversidad. Por otra parte, las imágenes generadas deberían tener mayor calidad al ser generadas aplicando los pesos de épocas más tardías. En el caso de los resultados obtenidos con los sets de datos de videos de tortugas, la tendencia en la métrica SIFID no parece ser clara.

Análisis de calidad de imagen. En el caso del SIFID, se notó que su valor comienza a tener un comportamiento asintótico al superar las 200000 épocas. En varios casos se notó que el SIFID alcanza un mínimo (máxima calidad de imagen) entre los valores de 130000 y 200000 épocas. Este fue el caso de las corridas Christoff_500_BCE y Christoff_500_HINGE, en ambos casos el SIFID mínimo se dio en la época 200000 (ver Figura 4.5). En el caso de la corrida AC_500_BCE también se encontró un SIFID mínimo en la época 200000 mientras que en la corrida AC_330_BCE el mínimo fue en la época 220000 (ver Figura 4.4).

Para el caso de la comparación de métricas entre las corridas AC_330_BCE y AC_500_BCE, se observó que hasta la época 200000 inclusive, la corrida con mayor resolución de imagen (AC_500_BCE) mantenía un SIFID menor (mayor calidad). Sin embargo, a partir de esa época, la corrida con menor resolución comienza a tener un SIFID menor. Este resultado no parece intuitivo en primera instancia.

La comparación de las corridas realizadas con el set de datos AA no devolvió resultados que permitan interpretar una tendencia en las métricas. Tanto para las corridas con menor resolución (AA_330) como para las corridas con mayor resolución (AA_500) se observaron valores de SIFID sin la tendencia esperable. En el caso de las corridas con menor resolución, el valor de SIFID alcanza un máximo en la época 100000, un mínimo en la época 20000. Las corridas con la función de pérdida HINGE tienen menor SIFID que las corridas con BCE a partir de la época 100000 (ver Figura 4.1). En el caso de las corridas con mayor resolución, el SIFID alcanza un mínimo en la época 100000 y las corridas realizadas con la función de pérdida BCE tienen menor SIFID (mayor calidad), ver Figura 4.2.

Al comparar el comportamiento de la métrica de calidad de imagen para diferentes sets de datos de videos de tortugas (AA, AB y AC) se observó que el set de datos AC muestra valores de SIFID significativamente menores que los otros dos sets de datos y con una tendencia decreciente hasta la época 200000 y luego comportamiento asintótico. Los otros dos sets de datos no mostraron un comportamiento con una tendencia clara de SIFID (ver Figura XX).

En el caso de la comparación entre las corridas Christoff_500_BCE vs. Christoff_500_HINGE y Christoff_500_BCE vs. Christoff_330_BCE se observó un menor SIFID siempre para el caso de Christoff_500_BCE. Esto se interpreta como que la calidad de imagen es mayor para corridas con mayor resolución y para corridas que aplican la función de pérdida BCE (ver Figuras 4.5 y 4.6).

Análisis de diversidad de imágenes. Para la métrica DTT los resultados sí mostraron una tendencia similar a la señalada por parte de los autores del modelo.

En el caso de la comparación de función de pérdida, se observó, para ambas resoluciones de imagen estudiadas, una mayor diversidad para la función BCE hasta la época 100000. Luego de esta época, la función de pérdida HINGE parece devolver mayor diversidad (ver Figuras 5.1 y 5.2).

Para la comparación de las corridas con diferentes sets de datos de videos de tortugas, el set de datos AA muestra un valor mayor de DTT en todas las épocas comparado con los sets de datos AB y AC (ver Figura 5.3). Esto es consistente con el hecho de que el set de datos AA es el más diverso en cuanto a sus contenidos (mayor porcentaje de fotogramas con rocas o espuma). Al comparar corridas con el mismo set de datos (AC) pero diferentes resoluciones, se observó mayor diversidad para el caso del set de datos con menor resolución (ver Figura 5.4).

Las corridas con el set de datos Christoff resultaron en una tendencia clara con respecto a DTT, es decir, mayor DTT a menor época. La corrida con mayor resolución y función de pérdida BCE (Christoff_500_BCE) fue la que mostró mayor DTT (mayor diversidad) al compararla con las otras dos corridas del mismo set de datos (ver Figura 5.5 y Figura 5.6).

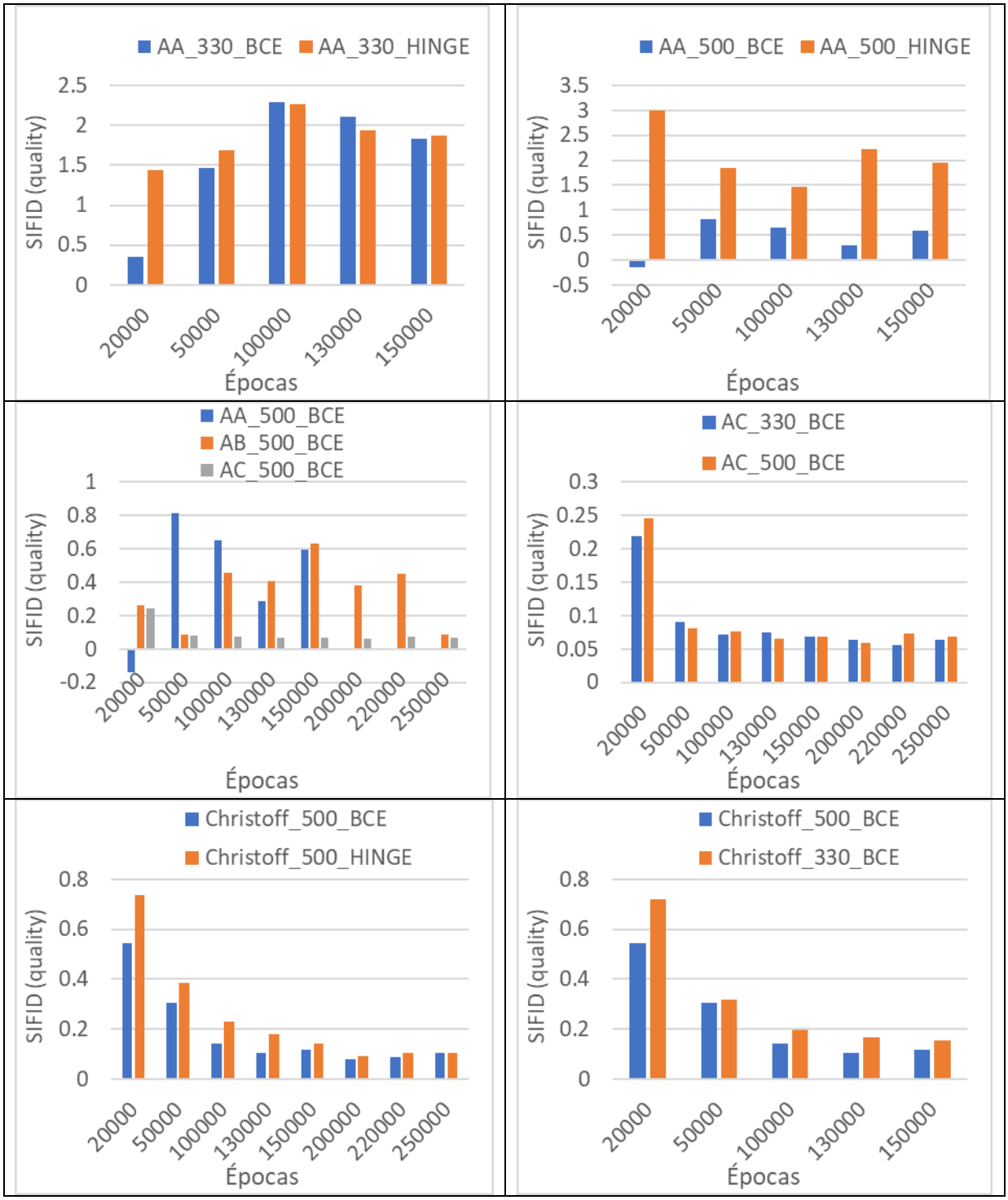


Figura 4. Comparación de la métrica SIFID para diferentes dúos o tríos de corridas. De izquierda a derecha y de arriba abajo se numeran las gráficas como 4.1 (AA_330_BCE vs. AA_330_HINGE), 4.2 (AA_500_BCE vs. AA_500_HINGE), 4.3 (AA_500_BCE vs. AB_500_BCE vs. AC_500_BCE), 4.4 (AC_330_BCE vs. AC_500_BCE), 4.5 (Christoff_500_BCE vs. Christoff_500_HINGE) y 5.6 (Christoff_330_BCE vs. Christoff_500_BCE).

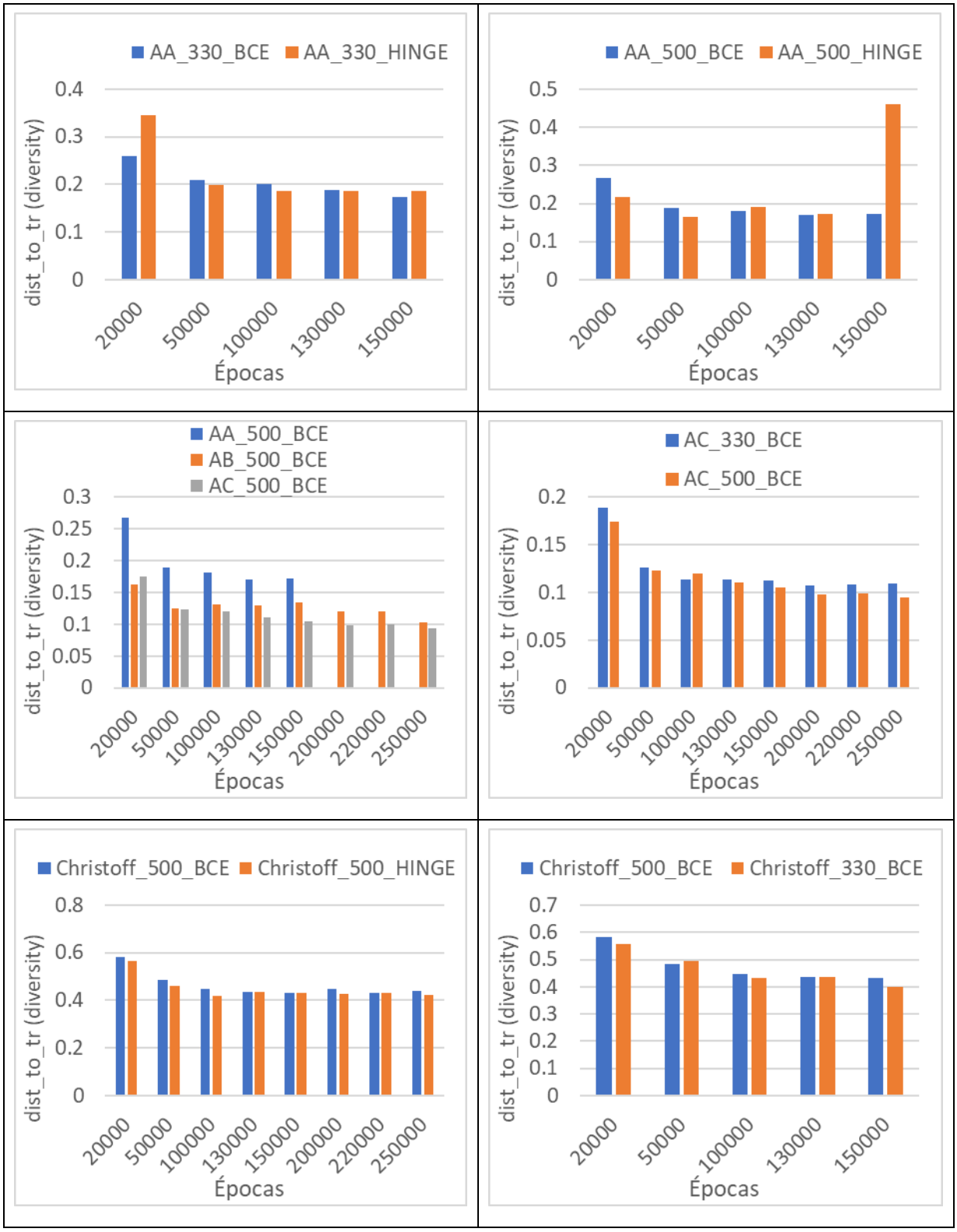


Figura 5. Comparación de la métrica Distance_to_train para diferentes dúos o tríos de corridas. De izquierda a derecha y de arriba abajo se numeran las gráficas como 5.1 (AA_330_BCE vs. AA_330_HINGE), 5.2 (AA_500_BCE vs. AA_500_HINGE), 5.3 (AA_500_BCE vs. AB_500_BCE vs. AC_500_BCE), 5.4 (AC_330_BCE vs. AC_500_BCE), 5.5 (Christoff_500_BCE vs. Christoff_500_HINGE) y 5.6 (Christoff_330_BCE vs. Christoff_500_BCE).

G. Análisis cualitativo de los resultados

En el contexto de la generación de imágenes con GAN, resulta importante realizar un análisis cualitativo de los resultados obtenidos. El análisis de las imágenes generadas en este estudio de OSG, se focalizará en tres aspectos: la capacidad del modelo de generar imágenes que contengan tortugas, la calidad de las imágenes generadas en general y la diversidad de las imágenes generadas en general. En esta aplicación del modelo, se generaron 100 imágenes para cada uno del conjunto de pesos de las épocas 20000, 50000, 10000, 130000, 150000, 200000, 220000 y 250000. Algunas corridas sólo se realizaron hasta la época 150000.

La Figura 6.1 muestra tres imágenes generadas para la corrida AA_500_BCE utilizando los pesos de la época 150000. Estas imágenes presentan buena calidad y cierta diversidad al mostrar imágenes de espuma, rocas y sólo agua. Para este set de datos, ninguna imagen generada mostró tortugas.

La Figura 6.2 muestra tres imágenes para la corrida AC_500_BCE en la época 250000. Se observa una muy buena calidad de imágenes y la presencia de una tortuga en las tres. Sin embargo, la tortuga está posicionada en la misa zona de la imagen, lo cual denota una baja diversidad con respecto a la misma. El resplandor sobre el agua sí se muestra variado para cada una de las tres muestras.

La Figura 6.3 muestra tres imágenes para la corrida AC_500_BCE en la época 120000. Se observa una calidad aceptable y la presencia de tortugas en la zona baja de la imagen. También se observa que son más de una tortuga. Es decir que el modelo generó imágenes que tienen cierta variabilidad con respecto a las imágenes de input (más de una tortuga) y también presentan diversidad en el reflejo del Sol sobre el agua. Sin embargo, las tortugas están posicionadas en la misma zona de la imagen en todos los casos.

La Figura 6.4 muestra tres imágenes para la corrida AC_500_BCE en la época 100000. En este caso se observaron resultados similares a las 120000 épocas. La diferencia principal es que las tortugas casi no se perciben y que en set de 100 imágenes generadas, hay pocas imágenes que contienen tortugas. En las imágenes generadas con pesos correspondientes a épocas anteriores a 100000, no se observaron tortugas.

La Figura 6.5 muestra tres imágenes para la corrida AB_500_BCE en la época 150000. Se observa una calidad de imagen media y cierta diversidad con respecto al reflejo del Sol sobre el agua. Para este set de datos ninguna de las imágenes generadas contiene tortugas.

Las Figuras 6.6 y 6.7 muestran las imágenes generadas al aplicar el modelo con función de pérdida WGAN. Para el set de datos AA y para el set de datos Christoff.

La Figura 6.8 muestra cuatro imágenes para la corrida Christoff_500_BCE en la época 250000. Se observa que el modelo es capaz de replicar la imagen inicial de buena manera. El sujeto (persona y perro) quedan fijos en todas las imágenes y el contexto varía. Algunas variaciones son quitar la camioneta, cambiar la forma de la casa en dos maneras diferentes, quitar la casa y dejar toda la imagen con fondo de naturaleza, variar el reflejo del Sol a través de los árboles del fondo

La Figura 6.9 muestra cuatro imágenes para la corrida Christoff_500_BCE en la época 200000. Esta época es la que, según el análisis de métricas, devolvió las imágenes de mayor calidad y buena diversidad de esta corrida. En este caso también se observa buena calidad de imágenes y cierta diversidad en el contexto de la misma. Los sujetos (persona y perro) quedaron fijos.

La Figura 6.10 muestra cuatro imágenes para la corrida Christoff_500_HINGE en la época 250000. Se observa muy buena calidad de imagen y menor diversidad. Al igual que las otras corridas analizadas, la diversidad se genera en el contexto de la imagen al variar o quitar la casa y la camioneta y al variar el paisaje.

La Figura 6.11 muestra dos imágenes para la corrida Christoff_500_BCE en la época 20000. En este caso se observa una calidad de imagen mucho menor pero sí se observa que el modelo intenta darle diversidad al sujeto. Esto lo hace cambiando de lugar y reiterando la presencia de la persona y el perro en las imágenes que genera.

La Figura 6.12 muestra tres imágenes para la corrida Christoff_500_BCE en la época 50000. Se observa al sujeto permanecer quieto pero el contexto sí varía fuertemente. Por ejemplo, la casa desaparece.

La Figura 6.13 muestra cuatro imágenes para la corrida Christoff_500_HINGE en la época 50000. En este caso se genera un equilibrio un poco más interesante con respecto al balance entre calidad de imagen y diversidad. Se ve al sujeto cambiar de lugar, repetirse o desaparecer de la imagen. Al mismo tiempo se generan imágenes con cierta calidad, aunque no es una calidad suficiente como para utilizar esas imágenes para entrenar otro modelo.

En resumen, el análisis cualitativo permitió verificar que para las épocas más altas de las corridas del modelo con diferentes sets de datos fue posible generar imágenes de alta calidad. En el caso de los sets de datos con tortugas, sólo el set de datos AC logró entrenar al modelo para que genere imágenes con tortugas. La aparición de las tortugas estuvo relacionada con las épocas de entrenamiento más altas (a partir de 10000 épocas). Sin embargo, no fue posible generar imágenes de tortugas que cumplan los requisitos de calidad, presencia de tortugas y diversidad. Con respecto a la diversidad, las imágenes generadas que contienen tortugas las posicionaron en zonas fijas de la imagen, sin variar las mismas por toda la imagen. El set de datos Christoff devolvió resultados con más aspectos a analizar, dado que la imagen del input contiene más aspectos para que el modelo analice y reproduzca. Se observó la capacidad del modelo de reproducir la imagen inicial con buena calidad y de variar el contexto de maneras diversas. En el caso del sujeto de la foto (persona y perro), el modelo sólo los hizo variar para épocas bajas de entrenamiento. Estas variaciones fueron moviendo a los sujetos de lugar o reiterándolos en la imagen. Sin embargo, para las imágenes que variaron los sujetos la calidad no fue tan buena (por ser épocas más tempranas). La variación de los sujetos también fue limitada, por ejemplo, ninguna imagen generada

rotó los sujetos (los hizo mirar para otro lado) o los hizo interactuar de otra manera entre sí (el perro siempre quedó de espaldas a la persona).

V. PRÓXIMOS PASOS EN LA APLICACIÓN DEL MODELO

A. Aplicación del modelo para imágenes de salida con mayor resolución

Se buscará aplicar el modelo utilizando el GPU NVIDIA A100 disponible en el Cluster-UY con 40 gb de memoria. Al momento de escribir este informe no fue posible compatibilizar las versiones de Pytorch y Cuda utilizadas en el ambiente de OSG con la utilización de este GPU.

Todas las corridas del modelo se realizaron en el GPU NVIDIA P100, el cual cuenta con una memoria de 12 gb. Para este GPU no fue posible correr el modelo con resoluciones de salida mayores a 550. Incluso bajando el tamaño del lote al mínimo posible, el GPU no tuvo suficiente capacidad como para realizar el proceso de entrenamiento.

B. Selección de dataset y parametrización definitivos

Para la generación de imágenes sintéticas aplicables al entrenamiento de una red CNN de detección de tortugas, es necesario seleccionar un set de datos y parametrización acordes. De los resultados obtenidos en este análisis no surge claramente que OSG sea una alternativa viable para dicho objetivo. Tal como se señaló anteriormente, la aplicación de OSG a los tres sets de datos de videos de tortugas propuestos no arrojó resultados satisfactorios en ninguno de los casos.

Resta analizar las posibles parametrizaciones que ofrece el modelo, aunque en principio no parece que estas parametrizaciones vayan a producir resultados significativamente mejores.

RECONOCIMIENTOS

En [5], los desarrolladores del modelo solicitan citar la utilización del código de OSG, de la siguiente manera:

```
@article{sushko2021generating,
  title={Generating Novel Scene Compositions from Single Images and Videos},
  author={Sushko, Vadim and Zhang, Dan and Gall, Juergen and Khoreva, Anna},
  journal={arXiv preprint:2103.13389},
  year={2021}

@article{sushko2022one,
  title={One-Shot Synthesis of Images and Segmentation Masks},
  author={Sushko, Vadim and Zhang, Dan and Gall, Juergen and Khoreva, Anna},
  journal={arXiv preprint:2209.07547},
  year={2022}
```

REFERENCIAS BIBLIOGRÁFICAS

[1] V. Sushko, D. Zhang, G. Jürgen, A. Khoreva, “Generating Novel Scene Compositions from Single Images and Videos”. In arXiv preprint:2103.13389, 2021.

[2] B. Liu, Y. Zhu, K. Song, A. El-gammal, “Towards faster and stabilized {gan} training for high-fidelity few-shot image synthesis”. In International Conference on Learning Representations, 2021.

[3] T. Rott Shaham, T. Dekel, T. Michaeli., “Singan: Learning a generative model from a single natural image” in International Conference on Computer Vision (ICCV), 2019.

[4] G. Velez-Rubio, Videos tomados durante campañas de monitoreo en el verano del 2021 en la zona de Cerro Verde (Rocha, Uruguay). Material confidencial y sin publicar.

[5] P.C. Gray, A.B. Fleishman, D. Klein, M.W. McKown, V.S. Bezy, K.J. Lohman, D.W. Johnston, “A convolutional neural network for detecting sea turtles in drone imagery”. In Methods on Ecology and Evolution;10:345–355, 2018

[6] S.T. Sykora-Bodie, V. Best, D.W. Johnston, E. Newton, K.J. Lohman, D.W. Johnston, “Quantifying Nearshore Sea Turtle Densities: Applications of Unmanned Aerial Systems for Population Assessments”. In Scientific REPOrTS | 7: 17690 | DOI:10.1038/s41598-017-17719-x, 2017

[7] Repositorio de Github en el que se tiene acceso al código del modelo y detalles sobre la ejecución del mismo: https://github.com/boschresearch/one-shot-synthesis

[8] LeCun Y. , L. Bottou, Y. Bengio, & P. Haffner, “Gradient-based learning applied to document recognition”. In Proceedings of the IEEE 86 (11), 2278-2324, 1998

[9] LeCun Y. , Y. Bengio, & G. Hinton, “Deep learning”. In NATURE. Vol. 521: 436. doi:10.1038/nature14539, 2015

[10] Goodfellow I., Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair, S. A. Courville, & Y. Bengio, “Generative adversarial nets”. In arXiv:1406.2661 [stat.ML], 2014

[11] Goodfellow I., Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair, S. A. Courville, & Y. Bengio, “Generative adversarial networks”. In Communications of the ACM| 139: 144 | DOI:10.1145/3422622, 2020

[12] Goodfellow I, “NIPS 2016 Tutorial: Generative Adversial Networks”. In arXiv:1701.00160v4 [cs.LG], 2017

[13] Van den Oord, A., Kalchbrenner, N., Kavukcuoglu, K., “Pixel recurrent neural networks”. In Proceedings of the 33 rd International Conference on Machine Learning, New York, NY, USA, 2016. JMLR: W&CP volume 48, 2016

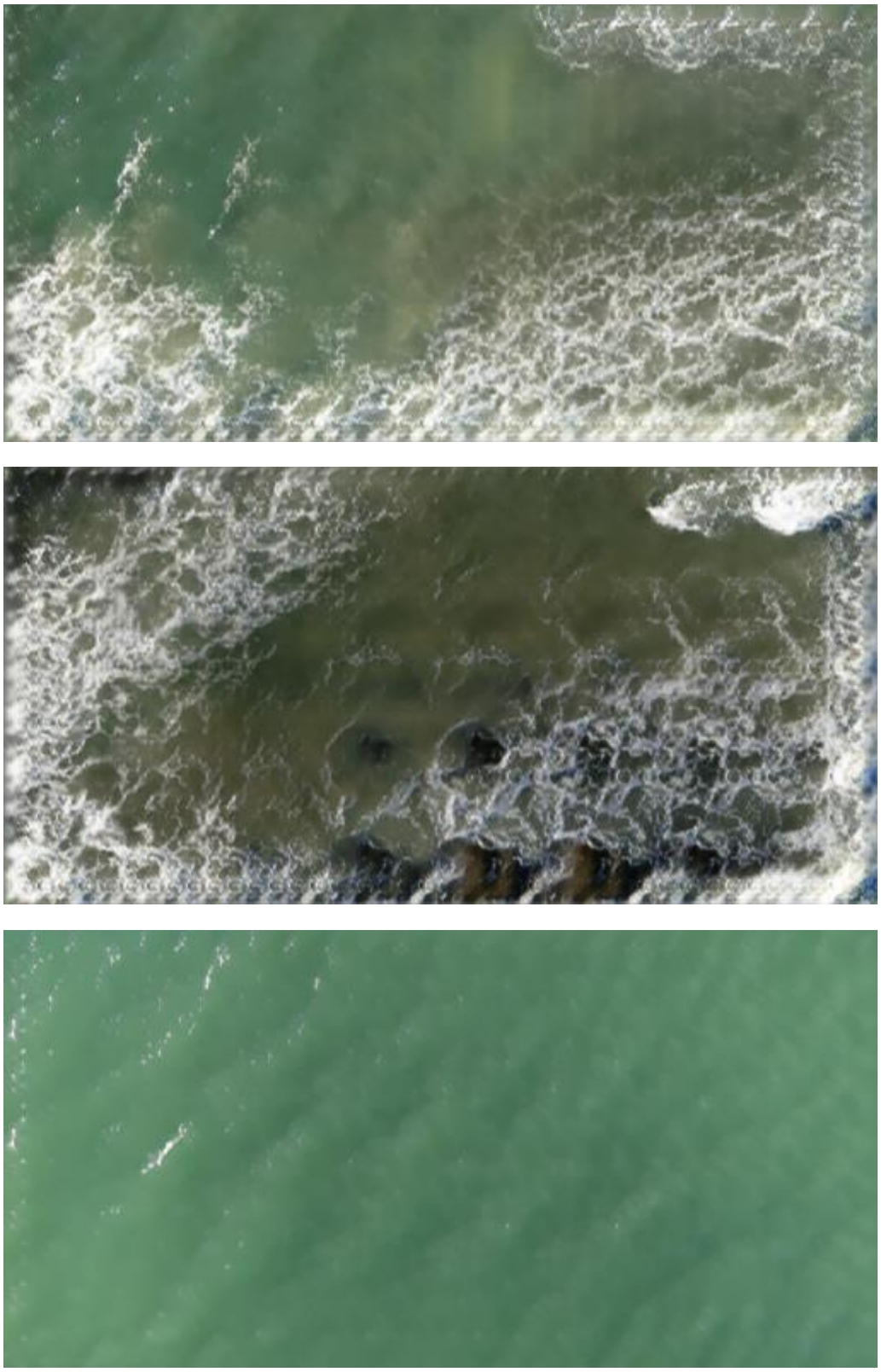


Figura 6.1. Muestra de imágenes generadas en la corrida AA_500_BCE con los pesos de la época 150000

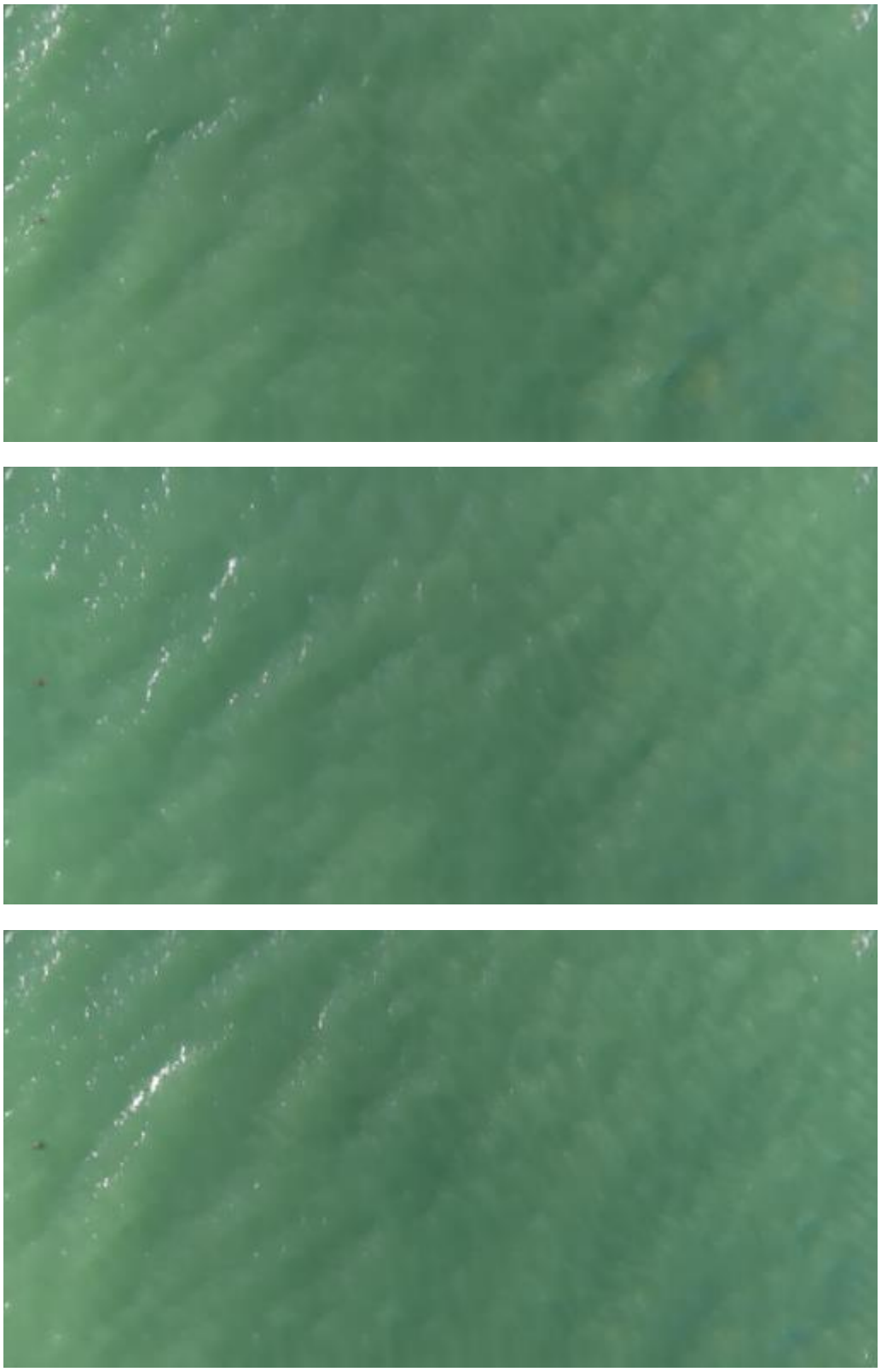


Figura 6.2. Muestra de imágenes generadas en la corrida AC_500_BCE con los pesos de la época 250000

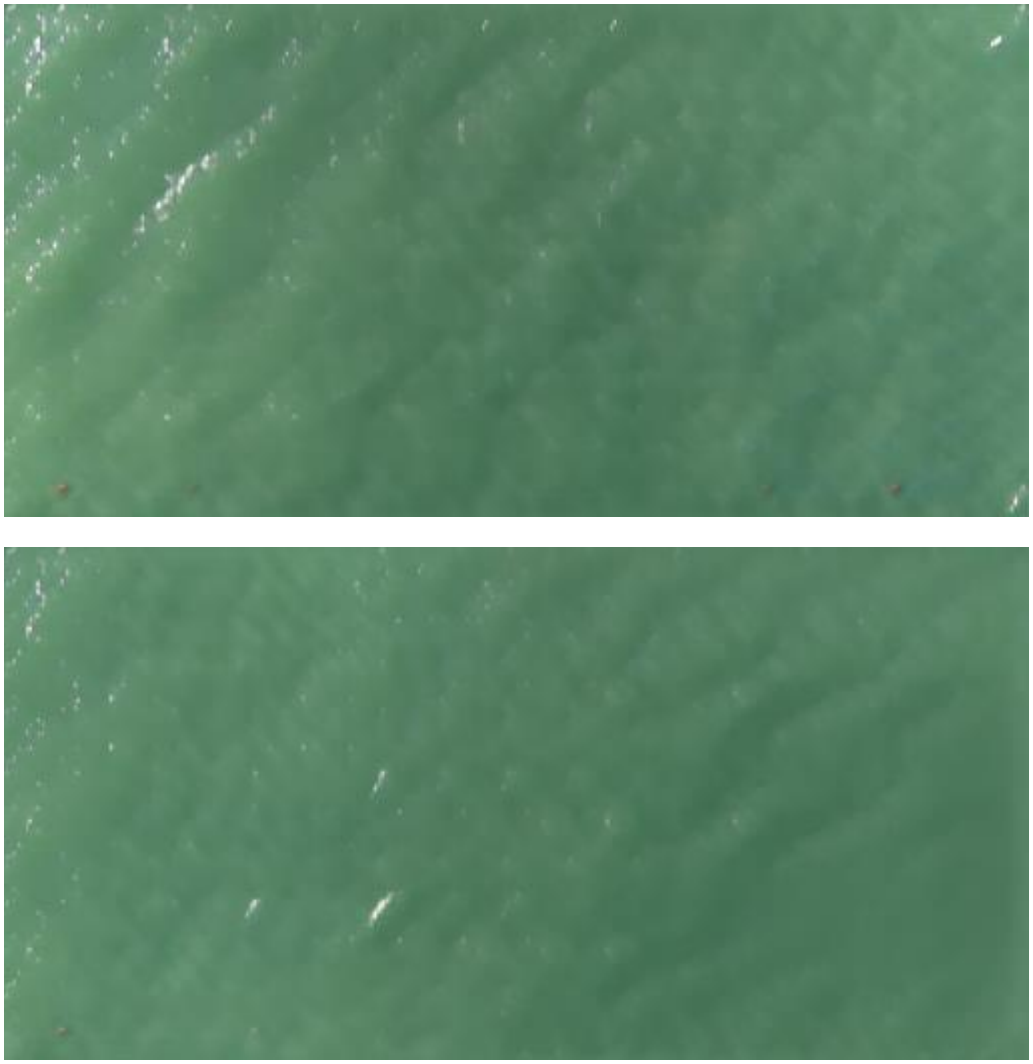


Figura 6.3. Muestra de imágenes generadas en la corrida AC_500_BCE con los pesos de la época 120000

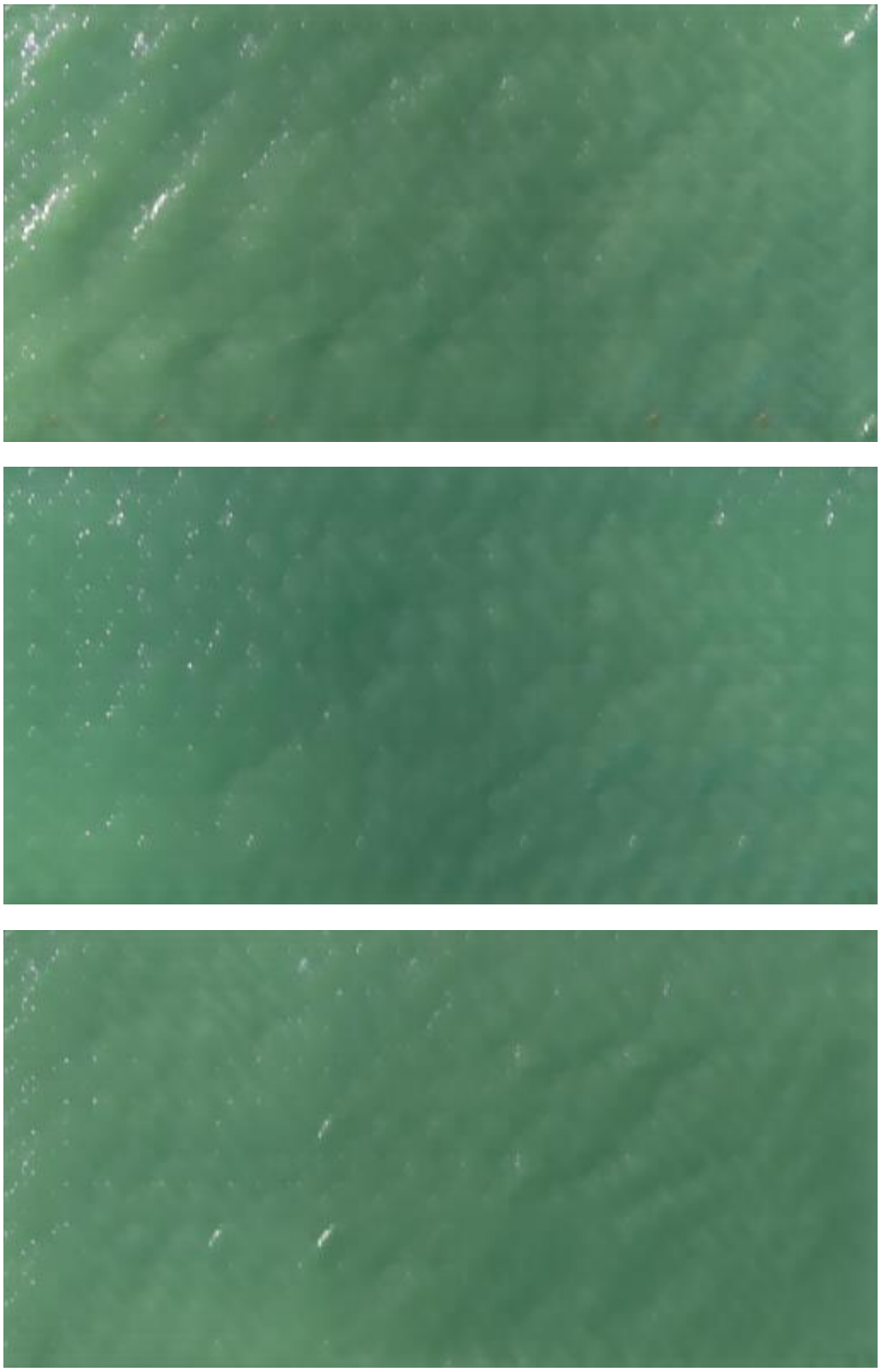


Figura 6.4. Muestra de imágenes generadas en la corrida AC_500_BCE con los pesos de la época 100000

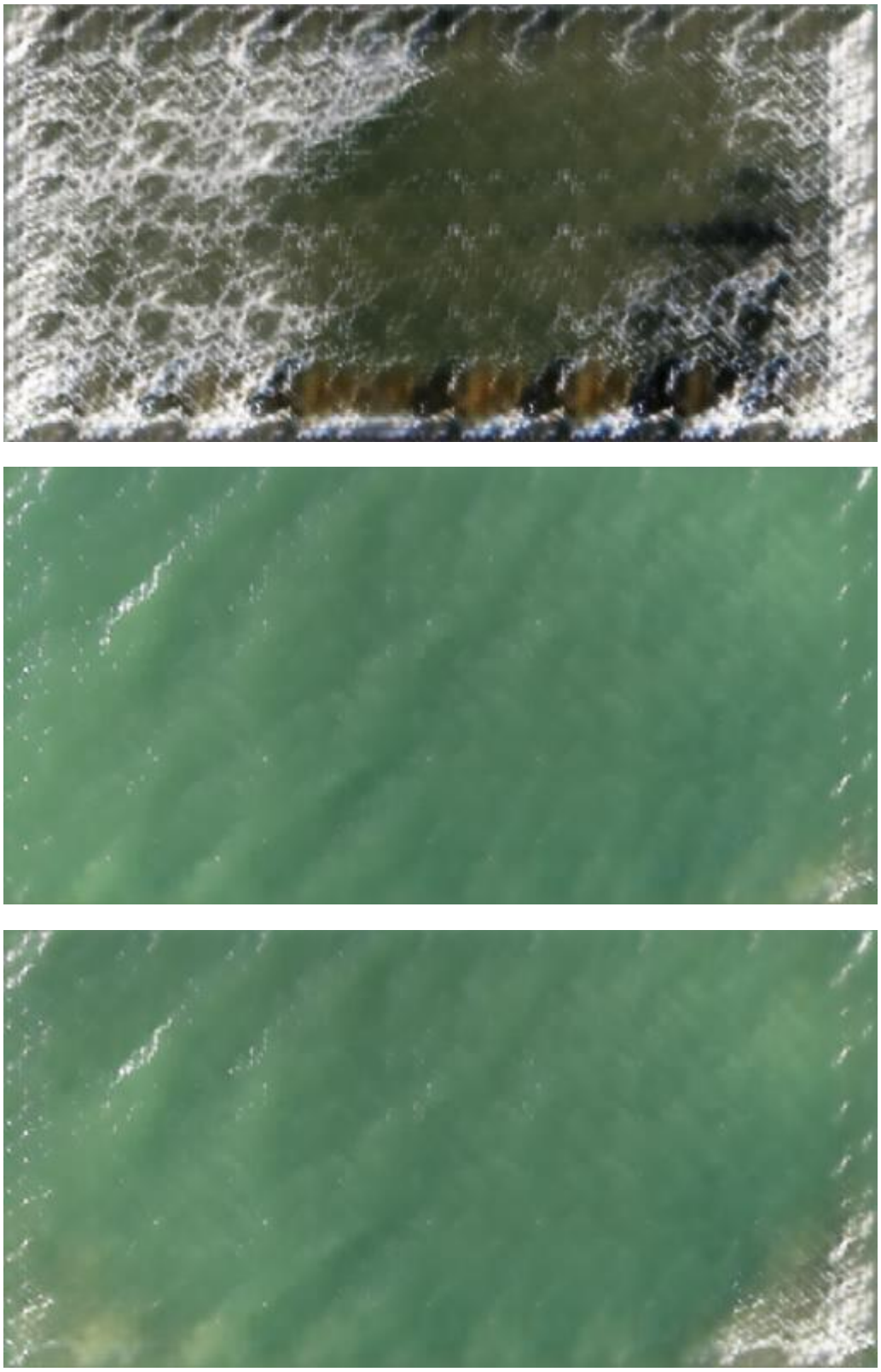


Figura 6.5. Muestra de imágenes generadas en la corrida AB_500_BCE con los pesos de la época 150000

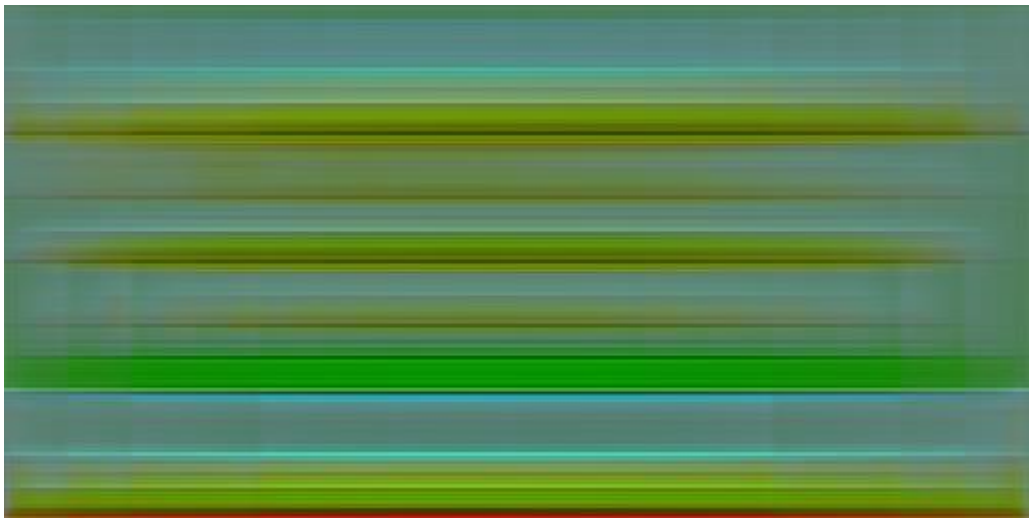


Figura 6.6. Muestra de imagen generada en la corrida AA_500_WGAN con los pesos de la época 150000

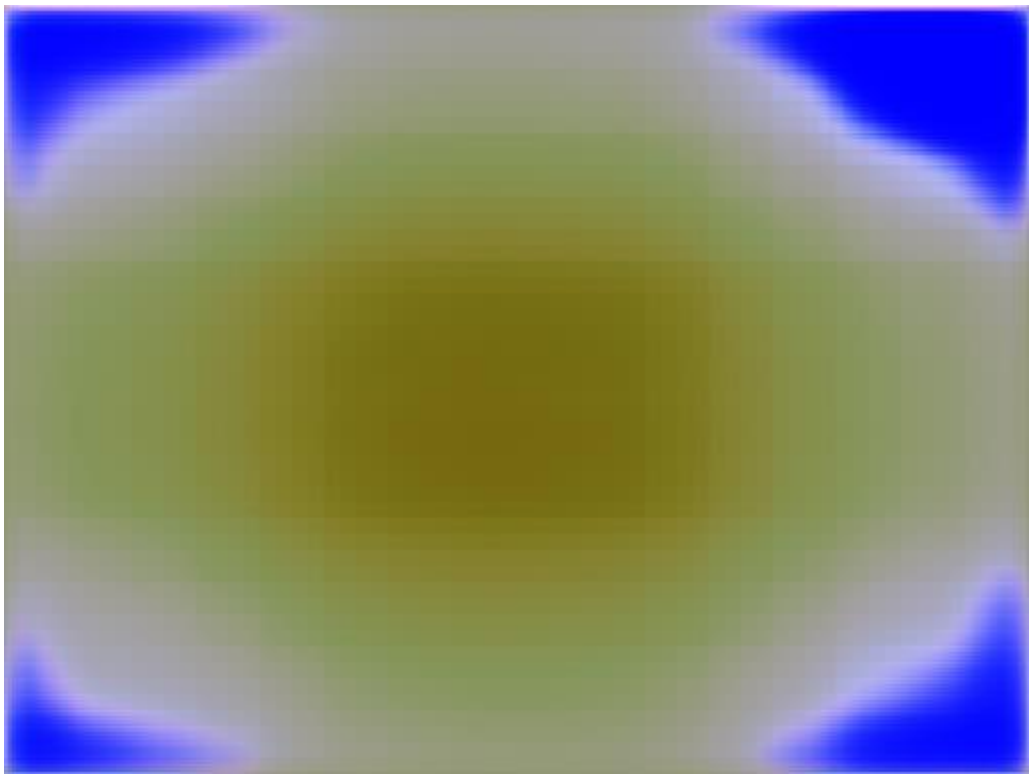


Figura 6.7. Muestra de imagen generada en la corrida Christoff_500_WGAN con los pesos de la época 150000

Figura 6.8. Muestra de imágenes generadas en la corrida Christoff_500_BCE con los pesos de la época 250000

Figura 6.9. Muestra de imágenes generadas en la corrida Christoff_500_BCE con los pesos de la época 200000

Figura 6.10. Muestra de imágenes generadas en la corrida Christoff_500_HINGE con los pesos de la época 250000

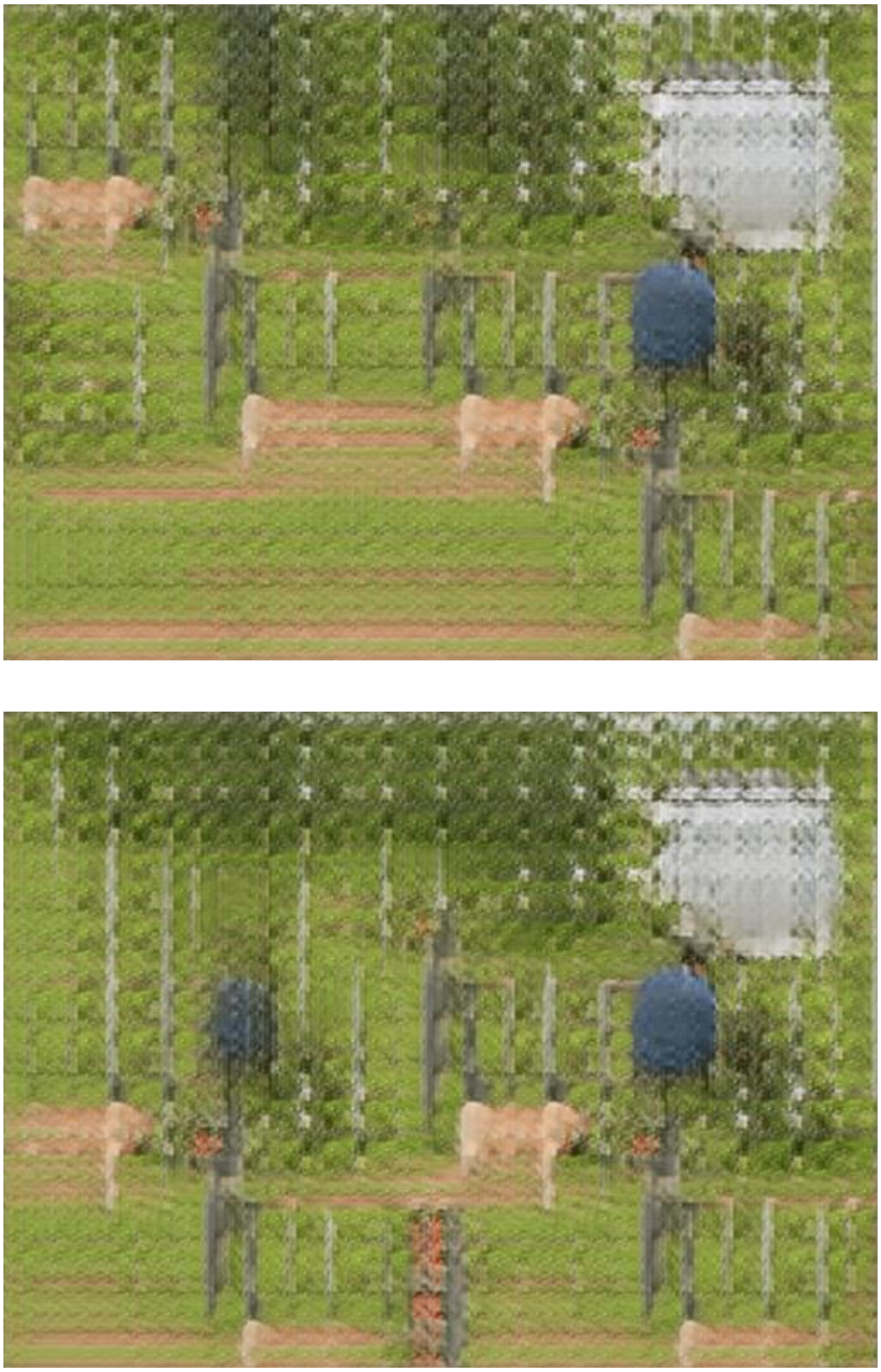


Figura 6.11. Muestra de imágenes generadas en la corrida Christoff_500_BCE con los pesos de la época 20000

Figura 6.12. Muestra de imágenes generadas en la corrida Christoff_500_BCE con los pesos de la época 50000

Figura 6.13. Muestra de imágenes generadas en la corrida Christoff_500_HINGE con los pesos de la época 50000