Taming Transformers for High-Resolution Image Synthesis

Ronaldo Lopez
FC-UNI
Lima, Peru
ronaldo.lopez.c@uni.pe

Cristhian Sánchez Sauñe
FC-UNI
Lima, Peru
csanchez@uni.pe

Davis Alderete
FC-UNI
Lima, Peru
davis.alderete.v@uni.pe

Abstract

Diseñado para aprender interacciones de largo alcance en datos secuenciales, los Transformers continúan mostrando resultados de vanguardia en una amplia variedad de tareas. A diferencia de las CNN, no contienen sesgos inductivos que prioricen las interacciones locales. Esto los hace expresivos, pero también computacionalmente inviables para secuencias largas, como imágenes de alta resolución. Demostramos cómo la combinación de la efectividad del sesgo inductivo de las CNN con la expresividad de los Transformers les permite modelar y, por lo tanto, sintetizar imágenes de alta resolución. En este trabajo se muestra cómo (i) usar CNN para aprender un vocabulario rico en contexto de los componentes de la imagen y, a su vez, (ii) utilizar Transformers para modelar de manera eficiente su composición dentro de imágenes de alta resolución. El enfoque propuesto se aplica fácilmente a las tareas de síntesis condicional, donde tanto la información no espacial, como las clases de objetos, como la información espacial, como las segmentaciones, pueden controlar la imagen generada. En particular, se presentan los primeros resultados sobre la síntesis guiada semánticamente de imágenes megapíxeles con Transformers y se obtiene el estado del arte entre los modelos autorregresivos en ImageNet condicional de la clase. El código y los modelos previamente entrenados se pueden encontrar en github.com/HiroForYou/Image-Synthesis-with-Transformers

1 Introducción

1.1 Problemática:

En los últimos años, el procesamiento de lenguaje natural (NLP), una rama muy importante de la inteligencia artificial a avanzando en sus modelos a tal punto de poder leer y comprender el lenguaje humano con el fin de interpretarlo y representarlo en forma de texto, imágenes o audio dependiendo de lo que ordenamos. Un Transformer es una arquitectura estándar para las tareas de lenguaje y se adaptan cada vez más en otras áreas como el audio y la visión. Poseen un modelo de aprendizaje profundo y funcionan interaccionando todos los datos, aprendiendo a como modelarlo durante todo el proceso y mostrando los resultados más rápido y sin ninguna intervención humana.

Anteriores investigaciones usaron los Transformers para generar imágenes realizando interacciones en todos los píxeles hasta formar la imagen solicitada, este proceso se puede repetir y el modelo podrá generar diferentes imágenes pero que estén relacionadas con la imagen original como se muestra en la fig 1. Sin embargo, apuntar a imitar la verdadera distribución de imágenes condicional, la síntesis de imágenes diversa, pero de alta fidelidad sigue siendo un gran desafío en la generación de imágenes condicional, especialmente cuando las entradas condicionales provienen de diferentes dominios visuales o incluso dominios heterogéneos. Pero hubo un inconveniente al usar los Transformers, el costo computacional aumenta cuadráticamente y las imágenes de alta resolución utiliza una cantidad

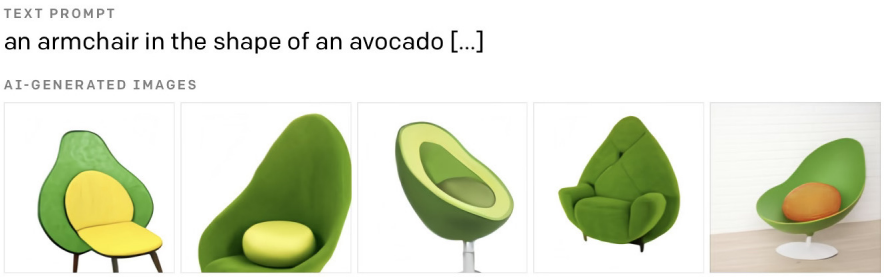

Figure 1: Imágenes de un sillón en forma de aguacate generados por un Transformer

enorme de píxeles que sobrepasan el millón, por lo tanto, la arquitectura de Transformers se hace ineficiente para este tipo de imágenes de alta resolución. Aquí surge nuestra pregunta, ¿Será necesario diseñar otra arquitectura de PNL para un mejor rendimiento generando imágenes de alta resolución o solo bastará implementar otras herramientas a nuestro Transformer para que sea más eficiente?

En este articulo buscaremos una solución de nuestro problema con la segunda alterativa que seria incorporar uno o más modelos a los Transformers implementando así una nueva arquitectura de IA. Uno de los puntos debiles de los Transformers es que no presenta un sesgo inductivo en su modelo que permita priorizar las interacciones locales osea que no lo permite aprender sobre las componentes principales de la imagen evitando asi varias interacciones. Para ello proponemos combinar la efectividad del sesgo inductivo de las redes neuronales convolucionales (CNN) con la expresividad de los Transformers que les permitirá modelar y sintetizar imágenes de alta resolución.

1.2 Objetivos:

1.2.1 Objetivo general

Estudiar e implementar una GAN compuesta por un auto-encoder convolucional y una red Transformer, con el propósito de generar imágenes de alta resolución con bajo coste computacional.

1.2.2 Objetivos específicos

- O1) Verificar que en nuestro modelo propuesto los Transformers siguen manteniendo su efectiva expresividad y modelado comparándolo con otros modelos con enfoques convolucionales.
- O2) Verificar si la nueva arquitectura tienen un buen rendimiento generando imágenes de alta resolución.

1.2.3 Resultados esperados

Para nuestro primer objetivo, planeamos comparar nuestro arquitectura que se basa principalmente en un Transformer con la arquitectura PixelSNAIL que esta enfocado en una red convolucional que sirve para la generación de imágenes. Esperamos un mejor desempeño por parte de nuestra arquitectura propuesta generando imágenes en un menor tiempo.

Para nuestro segundo objetivo, analizaremos las imágenes que genere nuestra arquitectura y observaremos la calidad y realismo que estos presenten.

1.3 Herramientas y métodos

1.3.1 Herramientas

- PyTorch: Es una librería de Python que permite realizar operaciones matemáticas mediante programación de tensores. Su capacidad de ejecutarse con la GPU lo convierte en una herramienta accesible para crear redes neuronales y cualquier modelo de lenguaje natural.
- Clip: Es una red neuronal que permite interpretar una cadena de texto y representarlo con una imagen. Dicha red neuronal tiene como objetivo abordar estos problemas: está entrenada en

una amplia variedad de imágenes con una amplia variedad de supervisión del lenguaje natural que está abundantemente disponible en Internet. Puede recibir instrucciones en lenguaje natural para realizar una gran variedad de evaluaciones comparativas de clasificación como una cadena de texto. El rendimiento de CLIP es muy bueno para clasificar imágenes,

Conjunto de Datos	Rendimiento de CLIP
ImageNet	76.2%
ImageNet V2	70.1%
ImageNet Rendition	88.9%
ObjectNet	72.3%
ImageNet Sketch	60.2%
ImageNet Adversarial	77.1%

Table 1: Rendimiento de CLIP al momento de clasificar imágenes en base a una cadena de texto

especialmente para imágenes en diferentes posiciones como ObjectNet (72.3%) o imágenes abstractas como ImageNet Rendition e ImageNet Sketch (88.9% y 60.2%). Clip nos será de utilidad para nuestros experimentos ya que nuestra arquitectura no interpreta cadenas de texto solo mejora la calidad de la imagen creada por Clip.

- ImageNet: Es una base de datos donde están disponibles una gran variedad de imágenes en diferentes categorías. Nos sera de utilidad para el entrenamiento de nuestro modelo así como los experimentos correspondientes.

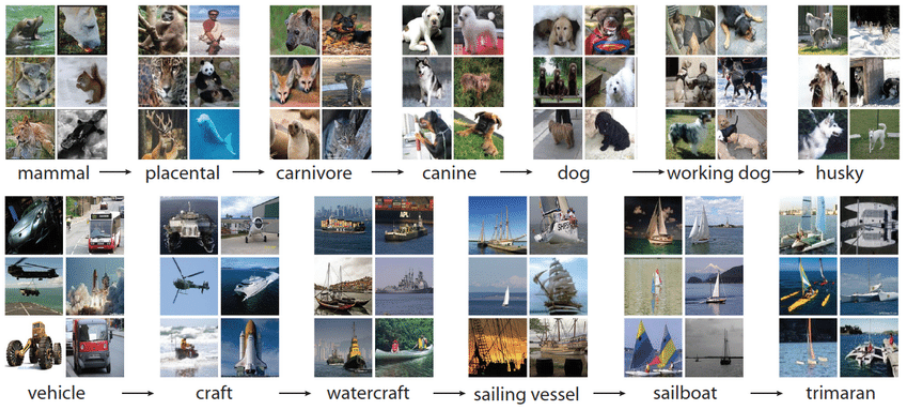

Figure 2: Colección de imágenes de ImageNet clasificados en diferentes categorías

- PixelSNAIL: Es un modelo generativo autorregresivo (ver figura 3) que logra buenos resultados en tareas de estimación que involucran datos de alta dimensión, como imágenes o audio. Dicho modelo combinan convoluciones causales con self-attention.
  Usaremos este modelo para nuestros experimento cuando queramos analizar el rendimiento de los Transformers una vez diseñado nuestro modelo GAN.

1.4 Hipótesis:

Los Transformers carecen de los sesgos inductivos de las CNNs, lo que los hace más eficiente en problemas de modelamiento en resoluciones altas.

1.4.1 Justificación:

Los Transformers actúan como una memoria a corto plazo, son muy buenos aprendiendo en el proceso de una tarea, pero no utilizan la información básica que se les proporciona (no aprenden en términos humanos) y las redes convolucionales actúan como una memoria a largo plazo, cuando realizan una tarea primero almacenan la información esencial de la misma y los usa para su solución (se acuerda de lo que considera importante en términos humanos).

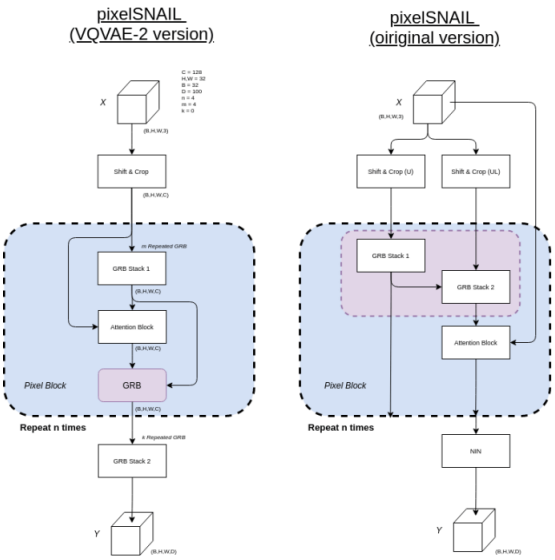

Figure 3: Arquitectura PixelSNAIL

En nuestro trabajo de investigación, si solo implementamos los Transformers podrá generar las imágenes desde cero sin saber las características principales de la imagen (aprenderá algo nuevo sin un concepto previo) y si solo implementamos las redes convolucionales podrá recordar como esta compuesta la imagen y generarla pero no tendrá la potencia suficiente para conectar las componentes de la imagen (sabe los conceptos pero no es eficiente aplicándolo). Si implementamos ambas arquitecturas tendremos un nueva arquitectura capaz de generar imágenes súper realistas.

2 Revisión de la literatura

2.1 Cadenas de búsqueda

Se han buscado trabajos relacionados en páginas como *Scopus* y *Arxiv*. Se realizaron las siguientes consultas:

- TITLE-ABS-KEY ((Convolutional Neural Network OR CNN) AND (Image coloring))
- TITLE-ABS-KEY ((Transformers) AND (Image coloring))
- TITLE-ABS-KEY ((Convolutional Neural Network OR CNN) and (Generation of Images))
- TITLE-ABS-KEY ((General Adversarial Networks OR GAN) and (Generation of Images) and (Image coloring)

Las preguntas sobre las que nos basamos para filtrar nuestros resultados fueron:

- ¿Cómo generar imágenes de alta resolución con un bajo coste computacional?
- ¿Cuál es el mejor enfoque para colorear una imagen de forma realista?

Los resultados se presentan en los siguientes ítems a continuación.

2.2 Resultados: Transformers

La característica definitoria de la arquitectura del Transformer que se presenta en [1] es que modela las interacciones entre sus entradas únicamente a través de la atención que les permite manejar fielmente las interacciones entre las entradas independientemente de su posición relativa entre sí. Aplicado originalmente a las tareas de lenguaje, las entradas al Transformer se daban mediante tokens, pero se pueden utilizar otras señales, como las obtenidas a partir de audio [2] o imágenes [3].

Se sabe que el mecanismo de atención en el que se basan los Transformers tiene una complejidad computacional que aumenta cuadráticamente con la longitud de la secuencia de entrada.

El mecanismo self-attention se puede describir matemáticamente con la siguiente ecuación:

$$Attn(Q, K, V) = softmax\left(\frac{QK^t}{\sqrt{d_k}}\right) V \in \mathbb{R}^{N \times d_v} \tag{1}$$

Donde:

- Tensor Query: $Q \in \mathbb{R}^{Nd_k}$
- Tensor Key: $K \in \mathbb{R}^{Nd_k}$
- Tensor Value: $V \in \mathbb{R}^{Nd_k}$

Se han propuesto diferentes enfoques para reducir el coste computacional para hacer que los Transformers sean factibles para secuencias más largas. Se han encontrado los trabajos de [4] y [5] que restringen los campos de recepción de los módulos de atención, lo que reduce la expresividad y, especialmente para imágenes de alta resolución e introducen supuestos sobre la independencia de los píxeles. También se ha encontrado el trabajo de [6] en el que se retiene el campo receptivo completo pero pueden reducir los costos para una secuencia de longitud n desde de n^2 a $\sqrt{n}$, pero que hace resoluciones superiores a 64 píxeles todavía prohibitivamente caras.

2.3 Resultados: Redes Convolucionales

La estructura bidimensional de las imágenes sugiere que las interacciones locales son particularmente importantes. Las CNN explotan esta estructura al restringir las interacciones entre las variables de entrada a una vecindad local definida por el tamaño del kernel del kernel convolucional. Por lo tanto, la aplicación de un kernel da como resultado costos que escalan linealmente con la longitud total de la secuencia (el número de píxeles en el caso de las imágenes) y cuadráticamente con el tamaño del kernel, que, en las arquitecturas CNN modernas, a menudo se fija en una pequeña constante como 3 × 3. Este sesgo inductivo hacia interacciones locales conduce a cálculos eficientes, pero la amplia gama de capas especializadas que se introducen en las CNN para manejar diferentes tareas de síntesis (como las presentadas en el trabajo de [7]) sugiere que este sesgo es a menudo demasiado restrictivo.

Se han utilizado arquitecturas convolucionales para el modelado autorregresivo de imágenes, como el trabajo de [8], pero para imágenes de baja resolución, trabajos anteriores [4, 6] demostraron que los Transformers superan consistentemente a sus homólogos convolucionales.

2.4 Resultados: Enfoques de generación de imágenes

Entre los enfoques más cercanos al propuesto se encuentran los enfoques de dos etapas que, en primer lugar, aprenden una codificación de datos y, después, las salas aprenden, en una segunda etapa, un modelo probabilístico de esta codificación. El trabajo de [9] demostró evidencia tanto teórica como empírica sobre las ventajas de aprender primero una representación de datos con un Autoencoder Variacional (VAE) [10, 11] , y luego aprender nuevamente su distribución con un VAE. Para mejorar la eficiencia del entrenamiento de las Redes Generativas Adversarias (GAN) [12], [13] entrena una GAN para que aprenda representaciones de un autoencoder. En el trabajo de [14] se presenta el *Vector Quantised Variational Autoencoder* (VQVAE), un enfoque para aprender representaciones discretas de imágenes, y modela su distribución de forma autorregresiva con una arquitectura convolucional. [15] amplía este enfoque para utilizar una jerarquía de representaciones aprendidas. Sin embargo, estos métodos todavía se basan en la estimación de densidad convolucional, lo que dificulta la captura de interacciones de largo alcance en imágenes de alta resolución.

El trabajo de [3] utiliza un VQVAE para codificar imágenes hasta una resolución de 192 × 192. En un esfuerzo por mantener la representación discreta aprendida tan espacialmente invariante como sea posible con respecto a los píxeles, se emplea un VQVAE poco profundo con un pequeño campo receptivo. Por el contrario, mostraremos que una primera etapa poderosa, que capture la mayor cantidad de contexto posible en la representación aprendida, es fundamental para permitir una síntesis de imágenes de alta resolución eficiente con Transformers.

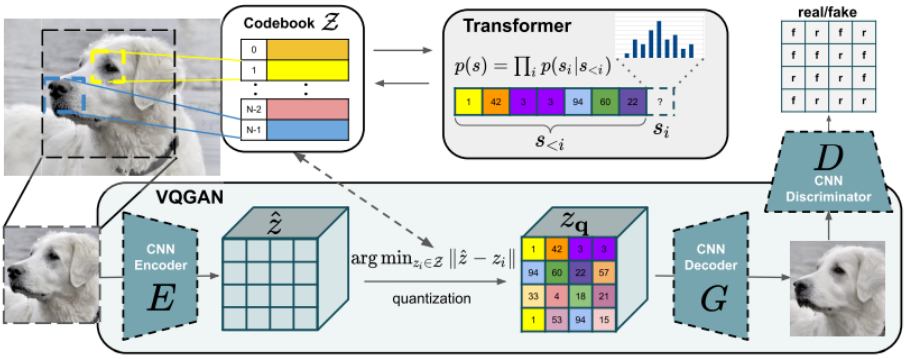

Figure 4: El enfoque propuesto utiliza un VQGAN convolucional para aprender un libro de códigos de partes visuales ricas en contexto, cuya composición se modela posteriormente con una arquitectura de Transformer autorregresivo. Un libro de códigos discreto proporciona la interfaz entre estas arquitecturas y un discriminador basado en parches permite una fuerte compresión al tiempo que conserva una alta calidad de percepción. Este método presenta la eficiencia de los enfoques convolucionales para la síntesis de imágenes de alta resolución basada en Transformers.

3 Metodología

El objetivo principal es aprovechar las prometedoras capacidades de aprendizaje de los modelos Transformers [1] e introducirlos en la síntesis de imágenes de alta resolución hasta el rango de megapíxeles. El trabajo anterior [16, 1] que aplicó Transformers a la generación de imágenes demostró resultados prometedores para imágenes de hasta un tamaño de 64 × 64 píxeles pero, debido al aumento cuadrático del costo en la longitud de la secuencia, no se puede simplemente escalar a resoluciones más altas. La síntesis de imágenes de alta resolución requiere un modelo que comprenda la composición global de las imágenes, lo que le permite generar patrones localmente realistas y globalmente consistentes. Por lo tanto, en lugar de representar una imagen con píxeles, la representamos como una composición de componentes de imagen perceptivamente ricos de un libro de códigos.

Aprendiendo un código efectivo, como se describe en la sección 3.1, podemos reducir significativamente la longitud de descripción de las composiciones, lo que nos permite modelar de manera eficiente sus interrelaciones globales dentro de imágenes con una arquitectura de Transformer como se describe en la sección 3.2. Este enfoque, resumido en la Figura 4, es capaz de generar imágenes de alta resolución realistas y consistentes tanto en un entorno incondicional como condicional.

3.1 Libro de códigos en los Transformers

Para utilizar la arquitectura Transformer altamente expresiva para la síntesis de imágenes, necesitamos expresar los componentes de una imagen en forma de secuencia. En lugar de basarse en píxeles individuales, la complejidad requiere un enfoque que utilice un libro de códigos discreto de representaciones aprendidas, de modo que cualquier imagen $x \in \mathbb{R}^{H\times W\times 3}$ se puede representar mediante una colección espacial de entradas del libro de códigos $z_q \in \mathbb{R}^{h\times w\times n_z}$, donde n_z es la dimensionalidad de los códigos. Una representación equivalente es una secuencia de índices hw que especifican las entradas respectivas en el libro de códigos aprendido. Para aprender eficazmente un libro de códigos espacial tan discreto, proponemos incorporar directamente los sesgos inductivos de las CNN e incorporar ideas del aprendizaje de la representación neural discreta [14]. Primero, aprendemos un modelo convolucional que consta de un encoder E y un decoder G, de modo que, tomados en conjunto, aprenden a representar imágenes con códigos de un libro de códigos discreto aprendido $Z = \{z_k\}_{k=1}^{K} \subset \mathbb{R}^{n_z}$ (consulte la Figura.4 para obtener una descripción general). Más precisamente, aproximamos una imagen dada x por $\hat{x} = G(z_q)$. Nosotros obtenemos z_q usando la codificación $\hat{z} = E(x) \in \mathbb{R}^{hxwxn_z}$ y una cuantificación posterior por elementos $q(.)$ de cada código espacial $\hat{z_{ij}} \in \mathbb{R}^{n_z}$ en su entrada de libro de códigos más cercana z_k :

$$z_q = q(\hat{z}) := (argmin\|\hat{\mathbf{z}}_{\mathbf{ij}} - \mathbf{z}_{\mathbf{k}}\|) \in R^{h \times w \times n_z} \tag{2}$$

La reconstrucción $\hat{x} \approx x$ esta dado por:

$$\hat{x} = G(z_q) = G(q(E(x))). \tag{3}$$

El backpropagation a través de la operación de cuantificación no diferenciable en la ecuación (3) se logra mediante un estimador de gradiente directo (construido en Pytorch), que simplemente copia los gradientes del de encoder al encoder [17], de modo que el modelo y el libro de códigos se pueden entrenar de un extremo a otro mediante la función de pérdida:

$$\pounds_{VQ}(E, G, Z) = \|x - \hat{x}\|^2 + \|sg[E(x)] - z_q\|_2^2 + \|sg[z_q] - E(x)\|_2^2. \tag{4}$$

Aquí $\zeta_{rec} = \|x - \hat{x}\|^2$ es una perdida de reconstrucción, $sg[.]$ denota la operación *stop gradient*, y $\|sg[z_q] - E(x)\|_2^2$ es la llamada *commitment loss* presentada en [14].

3.1.1 Libro de códigos rico en percepción

El uso de Transformers para representar imágenes como una distribución sobre los componentes de la imagen latente requiere que superemos los límites de la compresión y aprendamos un rico libro de códigos. Para ello, se propone VQGAN, una variante del VQVAE original, y utilizamos un discriminador y una pérdida de percepción [18, 19] para mantener una buena calidad de percepción a una mayor tasa de compresión. Tenga en cuenta que esto contrasta con trabajos anteriores que aplicaron modelos autorregresivos basados en píxeles [20, 15] y basados en Transformers [3] sobre un modelo de cuantificación superficial. Más específicamente, reemplazamos la pérdida L2 utilizada en [14] por una pérdida perceptiva y se presenta un procedimiento de entrenamiento adversario con un discriminador D [21] basado en parche que tiene como objetivo diferenciar entre imágenes reales y reconstruidas:

$$\pounds_{GAN}(\{E, G, Z\}, D) = [\log D(x) + log(1 - D(\hat{x}))] \tag{5}$$

El objetivo completo para encontrar el modelo de compresión óptimo $Q_* = E_*, G_*, Z_*$. Para agregar contexto de todas partes, aplicamos una sola capa de atención en la resolución más baja. Este procedimiento de entrenamiento reduce significativamente la longitud de la secuencia al desenrollar el código latente y, por lo tanto, permite la aplicación de potentes modelos Transformers.

3.2 Aprendiendo la composición de imágenes con Transformers

3.2.1 Transformers Latentes

Con E y G disponibles, ahora podemos representar imágenes en términos de índices de libro de códigos de sus codificaciones. Más precisamente, la codificación cuantificada de una imagen x viene dada por $z_q = q(E(x)) \in \mathbb{R}^{h \times w \times n_z}$ y es equivalente a una secuencia $s \in \{0, ..., |z| - 1\}^{h \times w}$ de índices del libro de códigos, que se obtiene reemplazando cada código por su índice en el libro de códigos Z.

Al mapear los índices de una secuencia s de nuevo a sus correspondientes entradas del libro de códigos, $z_q = (z_{q_{ij}})$ se recupera y decodifica fácilmente en una imagen $\hat{x} = G(z_q)$.

Por lo tanto, después de elegir algún orden de los índices en s, la generación de imágenes se puede formular como predicción autoregresiva del índice siguiente: Dados los índices $s_{<i}$, el Transformer aprende a predecir la distribución de los índices siguientes posibles, es decir, $p(s_i|s_{<i})$ para calcular la probabilidad de la representación completa como $p(s) = \prod_i p(s_i|s_{<i})$. Esto nos permite maximizar directamente la probabilidad logarítmica de las representaciones de datos.

3.2.2 Síntesis Condicionada

En muchas tareas de síntesis de imágenes, un usuario exige control sobre el proceso de generación proporcionando información adicional a partir de la cual se sintetizará un ejemplo. Esta información,

que llamaremos c, podría ser una sola etiqueta que describa la clase de imagen general o incluso otra imagen en sí. La tarea es entonces aprender la probabilidad de la secuencia dada esta información c.

$$p(s|c) = \prod_i p(s_i|s_{<i}, c) \tag{6}$$

Si la información de condicionamiento c tiene extensión espacial, primero aprendemos otro VQGAN para obtener nuevamente una representación basada en índices $r \in \{0, ..., |Z_c| - 1\}^{h_c \times w_c}$con el libro de códigos recién obtenido Z_c Debido a la estructura autorregresiva del Transformer , entonces podemos simplemente anteponer r a s y restringir el cálculo de la probabilidad logarítmica negativa a las entradas $p(s_i|s_{<i}, r)$. Esta estrategia de "solo deencoder" también se ha utilizado con éxito para tareas de resumen de texto [22].

3.2.3 Generando Imágenes en Alta Resolución

El mecanismo de atención del Transformer pone límites a la secuencia de longitud $h \cdot w$ de sus entradas s. Si bien podemos adaptar el número de bloques de submuestreo m del presente VQGAN para reducir imágenes de tamaño $H \times W \; para \; h = (H/2^m) \times w = W/2^m$, observamos una degradación de la calidad de reconstrucción más allá de un valor crítico. valor de m, que depende del conjunto de datos considerado. Para generar imágenes en el régimen de megapíxeles, tenemos que trabajar en forma de parches y recortar imágenes para restringir la longitud de sa un tamaño máximo posible durante el entrenamiento. Para muestrear imágenes, usamos el Transformer en forma de ventana deslizante como se ilustra en la Figura **??**.

El VQGAN asegura que el contexto disponible aún sea suficiente para modelar fielmente las imágenes, siempre que las estadísticas del conjunto de datos sean aproximadamente invariantes espacialmente o información de acondicionamiento espacial está disponible. En la práctica, este no es un requisito restrictivo, porque cuando se viola, es decir, síntesis de imagen incondicional en datos alineados, podemos simplemente condicionar en coordenadas de imagen, similar a [23].

Figure 5: Ventana de atención deslizante.

3.2.4 Resumen de metodología

En base a los objetivos planteados y una descripción detallada del enfoque a utilizar, se detallarán las actividades a realizar:

1. Las redes convolucionales codificarán una imagen de baja resolución, reduciendo sus dimensiones hasta un espacio latente, donde los componentes principales de la imagen serán almacenados en un libro de códigos que le permitirá mantener sus características y de esa manera hacerlos más eficiente al momento de generar las imágenes.
2. Una vez codificados los componentes, se usarán los Transformers para expresar los componentes de la imagen que estarán disponibles en el libro de códigos e intentara generar una nueva imagen parecida al original pero con una mayor resolución y realismo.
3. Cuando los Transformers hayan terminado de modelar las componentes, las redes convolucionales decodificaran la imagen en su forma de píxeles.
4. Y por último pasara por un discriminador (error del modelo) donde se analizará si la imagen generada es igual o parecido a la imagen solicitada, si es verdadera entonces mostrara la imagen generada, si es falsa entonces repetirá el proceso hasta que se asemeje a la imagen original.

4 Experimentos

En esta capitulo describiremos los resultados que obtuvimos en nuestros experimentos de acuerdo a los objetivos propuestos.

4.1 Rendimiento de los Transformers

4.1.1 Introducción

En esta sección se desarrollara los resultados esperados de nuestro objetivo especifico 1. Consiste en evaluar la capacidad de nuestro modelo VQGAN y analizar si se siguen manteniendo las ventajas de los Transformers comparándolas con el modelo PixelSNAIL que sigue un enfoque convolucional.

4.1.2 Atención en el espacio latente

Los Transformers muestran resultados de vanguardia en una amplia variedad de tareas, incluido el modelado de imágenes autorregresivas. Sin embargo, las evaluaciones de trabajos anteriores se limitaron a Transformers que trabajaban directamente en píxeles (de baja resolución) [16, 6, 24]. Esto plantea la pregunta de si el enfoque propuesto conserva las ventajas de los Transformers sobre los enfoques convolucionales. Para responder a esta pregunta, utilizamos una variedad de tareas condicionales e incondicionales y comparamos el desempeño entre el enfoque propuesto basado en Transformers y un enfoque convolucional. Para cada tarea, entrenamos un VQGAN con $m = 4$ bloques de submuestreo y, si es necesario, otro para la información de condicionamiento, y luego entrenamos tanto un Transformer como un modelo PixelSNAIL [25] en las mismas representaciones, como se ha realizado en enfoques anteriores de dos etapas [15].

Para una comparación completa, variamos las capacidades del modelo entre los parámetros 85M y 310M y ajustamos el número de capas en cada modelo para que coincidan entre sí. Observamos que PixelSNAIL se entrena aproximadamente dos veces más rápido que el Transformer y, por lo tanto, para una comparación justa, se muestra la probabilidad logarítmica negativa (negative log-likelihood) tanto para la misma cantidad de tiempo de entrenamiento (P-SNAIL time) como para la misma cantidad de pasos de entrenamiento (P-SNAIL steps).

Data # params	Transformer P-SNAIL steps	Transformer P-SNAIL time	PixelSNAIL fixed time
RIN / 85M	4.78	4.84	4.96
LSUN-CT / 310M	4.63	4.69	4.89
IN / 310M	4.78	4.83	4.96
D-RIN / 180 M	4.70	4.78	4.88
S-FLCKR / 310 M	4.49	4.57	4.64

Table 2: Comparación de las arquitecturas Transformer y PixelSNAIL en diferentes conjuntos de datos y tamaños de modelos. Para todas las configuraciones, los Transformers superan al modelo de última generación de la familia PixelCNN, PixelSNAIL en términos de NLL. Esto es válido tanto al comparar NLL en momentos fijos (PixelSNAIL entrena aproximadamente 2 veces más rápido) como cuando se entrena para un número fijo de pasos. Ver sección 4.1 para las abreviaturas.

4.2 Rendimiento del VQGAN

4.2.1 Introducción

En esta sección se desarrollara los resultados esperados de nuestro objetivo especifico 2. Consiste en realizar pruebas con nuestro modelo VQGAN y analizar su rendimiento generando imágenes de alta resolución.

Primero usaremos el modelo clip para representar la imagen solicitada de una cadena de texto a una imagen, luego de eso nuestro modelo VQGAN tendrá como entrada la imagen generada por clip y como salida la misma imagen pero con una resolución mas elevada y realista.

4.2.2 Síntesis en alta resolución

El enfoque de ventana deslizante permite la síntesis de imágenes más allá de una resolución de 256×256 píxeles. Evaluamos este enfoque en la generación de imágenes de forma condicional e incondicional en DRIN, COCO-Stuff y S-FLCKR (figura 6, 7, 8 y 9). Tenga en cuenta que, en principio, este enfoque se puede utilizar para generar imágenes de proporción y tamaño arbitrarios, dado que las estadísticas de imagen del conjunto de datos de interés son aproximadamente invariantes en el espacio o hay información espacial disponible.

Figure 6: Aplicando el enfoque de la ventana deslizante (Fig. 5) a varias tareas de síntesis de imágenes condicionales. Arriba: profundidad de imange en RIN, segunda fila: super-resolución estocástica sobre IN, tercera y cuarta fila: síntesis semántica en S-FLCKR, abajo: síntesis guiada por bordes sobre IN. Las imágenes resultantes varían entre 368×496 y 1024×576, por lo que se ven mejor con el zoom.

Figure 7: Un Monstruo en el Bosque.

Figure 8: Noche Oscura.

Figure 9: Noche Oscura.

4.2.3 Qué tan buena es la VQGAN?

Los FID de reconstrucción obtenidos a través del libro de códigos proporcionan una estimación del FID alcanzable del modelo generativo entrenado en él. Para obtener ganancias de rendimiento de calidad de nuestro VQGAN sobre VAE discretos entrenados sin pérdidas perceptivas y adversarias. Evaluamos esta métrica en ImageNet y se muestran los resultados los resultados en la tabla 3. Nuestro VQGAN supera a los modelos no adversarios al tiempo que proporciona una comprensión significativamente mayor. Como se esperaba, las versiones más grandes de VQGAN (ya sea en

términos de tamaños de libro de códigos más grandes o longitudes de códigos aumentadas) mejoran aún más el rendimiento. Una comparación cualitativa se observa en la figura 10

Modelo	Tamaño del libro	dim Z	FID/val	FID/train
VQVAE-2	64×64 & 32×32	512	n/a	~ 10
DALL-E	32×32	8192	32.01	33.88
VQGAN	16×16	1024	7.94	10.54
VQGAN	16×16	16384	4.98	7.41
VQGAN*	32×32	8192	1.49	3.24
VQGAN*	64×64 & 32×32	512	1.45	2.78

Table 3: FID en ImageNet entre la división de validación reconstruida y la división de validación original (FID/val) y de entrenamiento (FID/train).

Figure 10: Comparación de las capacidades de reconstrucción entre VQVAEs y VQGAN. Los números entre paréntesis indican el factor de compresión y el tamaño del libro de códigos. Con el mismo factor de compresión y tamaño de libro de códigos, los VQGAN producen reconstrucciones más realistas en comparación con los reconstrucciones borrosas de los VQVAE. Esto permite mayores tasas de compresión para VQGAN al tiempo que conserva reconstrucciones realistas.

4.2.4 Comparación con Image-GPT

Para evaluar más a fondo la eficacia de nuestro enfoque, lo comparamos el estado del arte del modelo Transformer generativo sobre imágenes conocido como ImageGPT [3]. Mediante el uso de inmensas cantidades de cálculo, los autores demostraron que los modelos Transformer se pueden aplicar a la representación de píxeles de imágenes y, por lo tanto, lograron resultados impresionantes tanto en el aprendizaje de la representación como en la síntesis de imágenes. Sin embargo, como su enfoque se limita al espacio de píxeles, no se escala más allá de una resolución de 192×192. Como nuestro enfoque aprovecha un método de compresión fuerte para obtener representaciones ricas en contexto de imágenes y luego aprende un modelo Transformer, podemos sintetizar imágenes con mucho mayor resolución. Comparamos ambos enfoques en la figura 11. Ambas imágenes muestran que nuestro enfoque es capaz de sintetizar terminaciones consistentes de una fidelidad dramáticamente incrementada. Los resultados de [3] se obtienen de https://openai.com/blog/image-gpt/

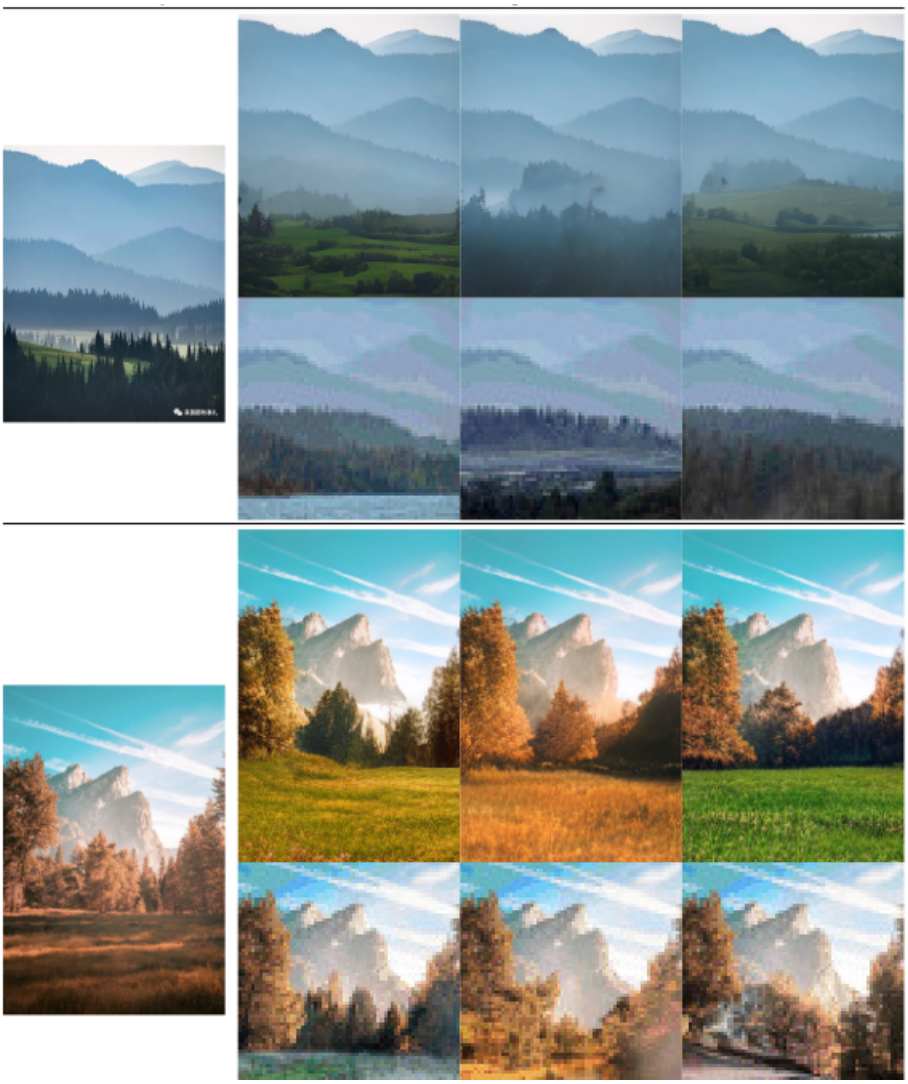

Figure 11: Comparando nuestro enfoque con el enfoque basado en píxeles de [3].

5 Conclusión

Este trabajo abordó los desafíos fundamentales que anteriormente limitaban a los Transformers a imágenes de baja resolución. Se propuso un enfoque que representa las imágenes como una composición de componentes de imagen perceptivamente ricos y, por lo tanto, supera la complejidad cuadrática inviable al modelar imágenes directamente en el espacio de píxeles. Modelar componentes con una arquitectura CNN y sus composiciones con una arquitectura Transformer aprovecha todo el potencial de sus fortalezas complementarias y, por lo tanto, permite representar los primeros resultados en la síntesis de imágenes de alta resolución.

Bibliografía

[1] Ashish Vaswani et al. "Attention Is All You Need". In: *CoRR* abs/1706.03762 (2017). arXiv: 1706.03762. URL: http://arxiv.org/abs/1706.03762.

[2] Naihan Li et al. "Neural speech synthesis with transformer network". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 01. 2019, pp. 6706–6713.

[3] Mark Chen et al. "Generative pretraining from pixels". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 1691–1703.

[4] Niki Parmar et al. "Image transformer". In: *International Conference on Machine Learning*. PMLR. 2018, pp. 4055–4064.

[5] Dirk Weissenborn, Oscar Täckström, and Jakob Uszkoreit. "Scaling autoregressive video models". In: *arXiv preprint arXiv:1906.02634* (2019).

[6] Rewon Child et al. "Generating long sequences with sparse transformers". In: *arXiv preprint arXiv:1904.10509* (2019).

[7] Taesung Park et al. "Semantic image synthesis with spatially-adaptive normalization". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 2337–2346.

[8] Aaron Van Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. "Pixel recurrent neural networks". In: *International Conference on Machine Learning*. PMLR. 2016, pp. 1747–1756.

[9] Bin Dai and David P. Wipf. "Diagnosing and Enhancing VAE Models". In: *CoRR* abs/1903.05789 (2019). arXiv: 1903.05789. URL: http://arxiv.org/abs/1903.05789.

[10] Diederik P Kingma and Max Welling. "Stochastic gradient VB and the variational auto-encoder". In: *Second International Conference on Learning Representations, ICLR*. Vol. 19. 2014, p. 121.

[11] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. "Stochastic backpropagation and approximate inference in deep generative models". In: *International conference on machine learning*. PMLR. 2014, pp. 1278–1286.

[12] Ian Goodfellow et al. "Generative adversarial networks". In: *Communications of the ACM* 63.11 (2020), pp. 139–144.

[13] Jinlin Liu, Yuan Yao, and Jianqiang Ren. "An acceleration framework for high resolution image synthesis". In: *arXiv preprint arXiv:1909.03611* (2019).

[14] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. "Neural discrete representation learning". In: *arXiv preprint arXiv:1711.00937* (2017).

[15] Ali Razavi, Aaron van den Oord, and Oriol Vinyals. "Generating diverse high-fidelity images with vq-vae-2". In: *Advances in neural information processing systems*. 2019, pp. 14866–14876.

[16] Niki Parmar et al. "Image Transformer". In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by Jennifer Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR, Oct. 2018, pp. 4055–4064. URL: https://proceedings.mlr.press/v80/parmar18a.html.

[17] Y Bengio. "N. L, , eonard, and A. Courville. Estimating or, propagating gradients through stochastic neurons for conditional, computation". In: *arXiv preprint arXiv:1308.3432* 2.3 (2013).

[18] Alexey Dosovitskiy and Thomas Brox. "Generating images with perceptual similarity metrics based on deep networks". In: *Advances in neural information processing systems* 29 (2016), pp. 658–666.

[19] Fabian Mentzer et al. "High-fidelity generative image compression". In: *arXiv preprint arXiv:2006.09965* (2020).

[20] Aaron van den Oord et al. "Conditional image generation with pixelcnn decoders". In: *arXiv preprint arXiv:1606.05328* (2016).

[21] Phillip Isola et al. "Image-to-image translation with conditional adversarial networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 1125–1134.

[22] Peter J Liu et al. "Generating wikipedia by summarizing long sequences". In: *arXiv preprint arXiv:1801.10198* (2018).

[23] Chieh Hubert Lin et al. “Coco-gan: Generation by parts via conditional coordinating”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 4512–4521.

[24] Jonathan Ho et al. “Axial Attention in Multidimensional Transformers”. In: *CoRR* abs/1912.12180 (2019). arXiv: 1912.12180. URL: http://arxiv.org/abs/1912.12180.

[25] Xi Chen et al. “Pixelsnail: An improved autoregressive generative model”. In: *International Conference on Machine Learning*. PMLR. 2018, pp. 864–872.