Vision Transformers (ViT)

Métodos Generativos, curso 2025-2026

Guillermo Iglesias, guillermo.iglesias@upm.es
Jorge Dueñas Lerín, jorge.duenas.lerin@upm.es
Edgar Talavera Muñoz, e.talavera@upm.es

5 de noviembre de 2025

Escuela Técnica Superior de Ingeniería de Sistemas Informáticos | UPM

Introducción

Los Vision Transformers (ViTs) son una arquitectura presentada en el artículo *"An image is worth 16x16 words: Transformers for image recognition at scale"* del año 2020 [1] dentro del área de la visión por computador.

- Utilizan los mecanismos de atención para procesar imágenes como secuencias de píxeles
- Utilizan únicamente el encoder para generar descriptores de imágenes

Características principales

Ventajas

- Captura dependencias globales de la imagen, es decir, correlaciona elementos muy distantes
- Mejora los resultados a la hora de intepretar o generar imágenes
- Simplifican el condicionamiento de las arquitecturas

Desventajas

- Necesidad de grandes datasets
  - Muchas veces se contrarresta con un pretraining, pero no siempre es posible, y sigue siendo muy costoso
  - Hay investigación que intenta reducir el coste [2]

Metodología de los ViT

Los principales pasos en un ViT son los siguientes:

1. Tokenización de la imagen
2. Procesamiento de tokens
3. Encoding de los tokens
4. Procesamiento con Neural Network (NN)
5. Generación de la salida

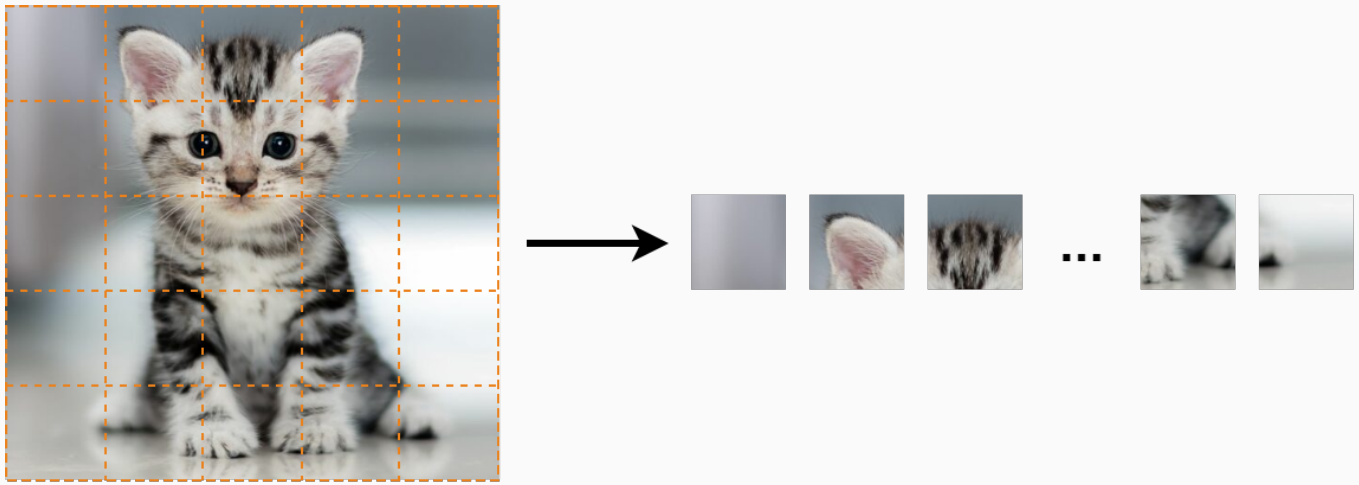
Tokenización de la imagen

De la misma manera que los Transformers en Natural Language Processing (NLP) necesitan que el texto sea procesado para ser entendible por un modelo de Artificial Intelligence (AI), las imágenes han de ser preprocesadas para ser tratadas.

→ [1.56, 0.45, -0.23, ..., 3.2]

Tokenización de la imagen: Generación de parches

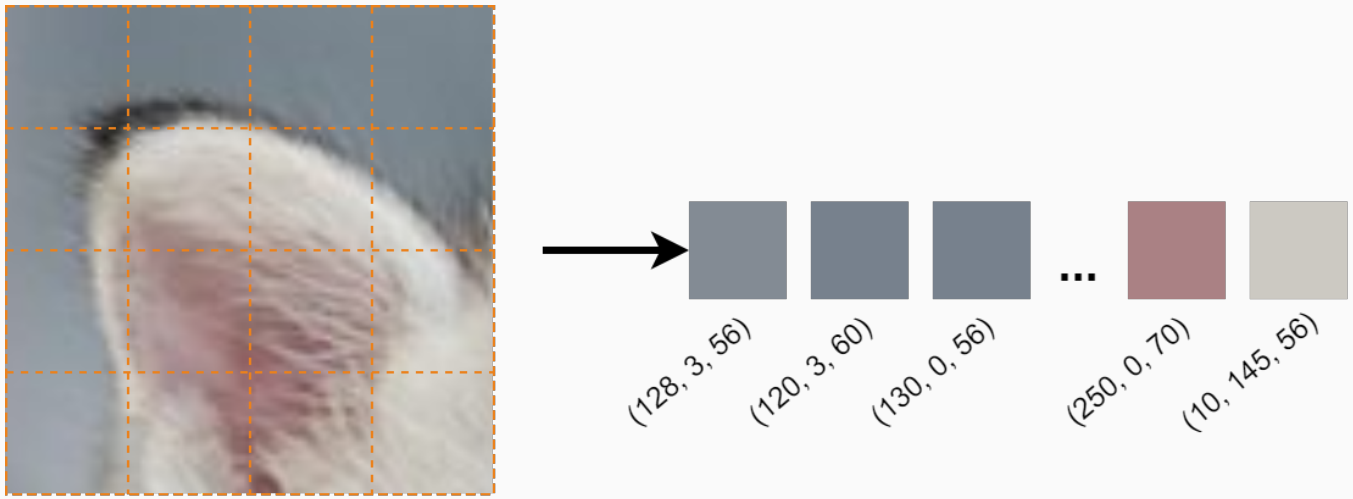
Los ViT procesan una imagen como una secuencia de tokens. Cada token es un parche de la imagen total, de tamaño $P_H x P_W x C$ (alto, ancho, canales).



*El tamaño más estándar para los parches es 16x16 píxeles

Tokenización de la imagen: Flatten de parches

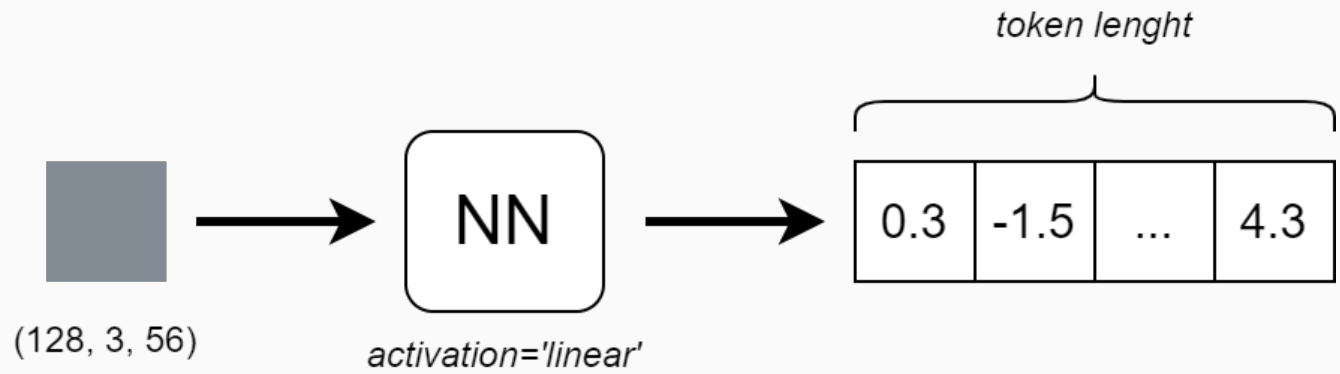
Cada parche es vectorizado aplicando un flatten.



Cada posición representa el valor de color del píxel.

Tokenización de la imagen: Proyección linear de los tokens

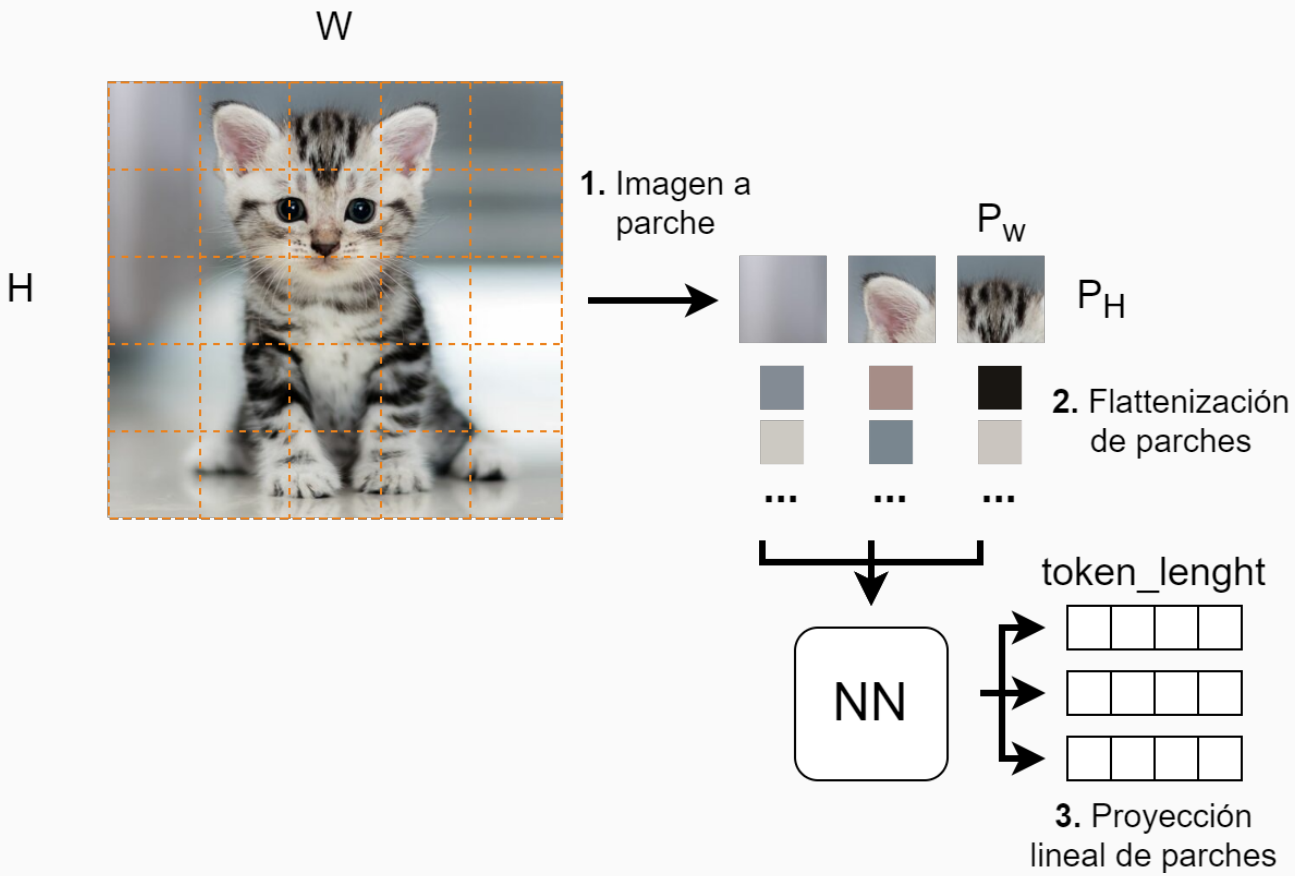
Una práctica común es proyectar los tokens a un espacio latente de un tamaño arbitrario. Se utiliza para mejorar la representación de los parches, lo que es equivalente a los embeddings de palabras en NLP.



*En [1] el *token lenght* se define a 768.

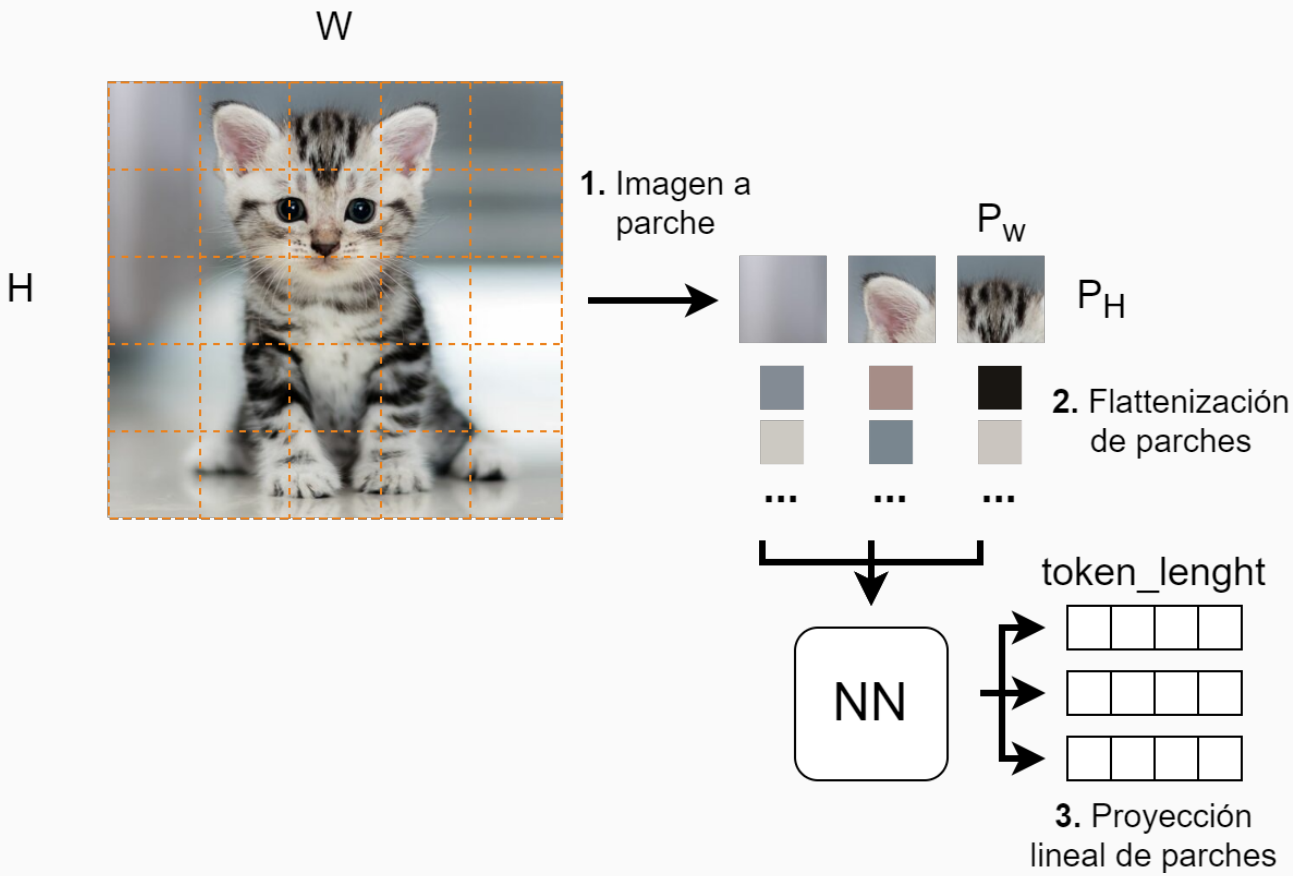
Tokenización de la imagen: Resumen

Con todo el proceso definido anteriormente, es posible generar una representación latente de una imagen, dividida en parches, que además pueden adaptarse al proceso de aprendizaje.



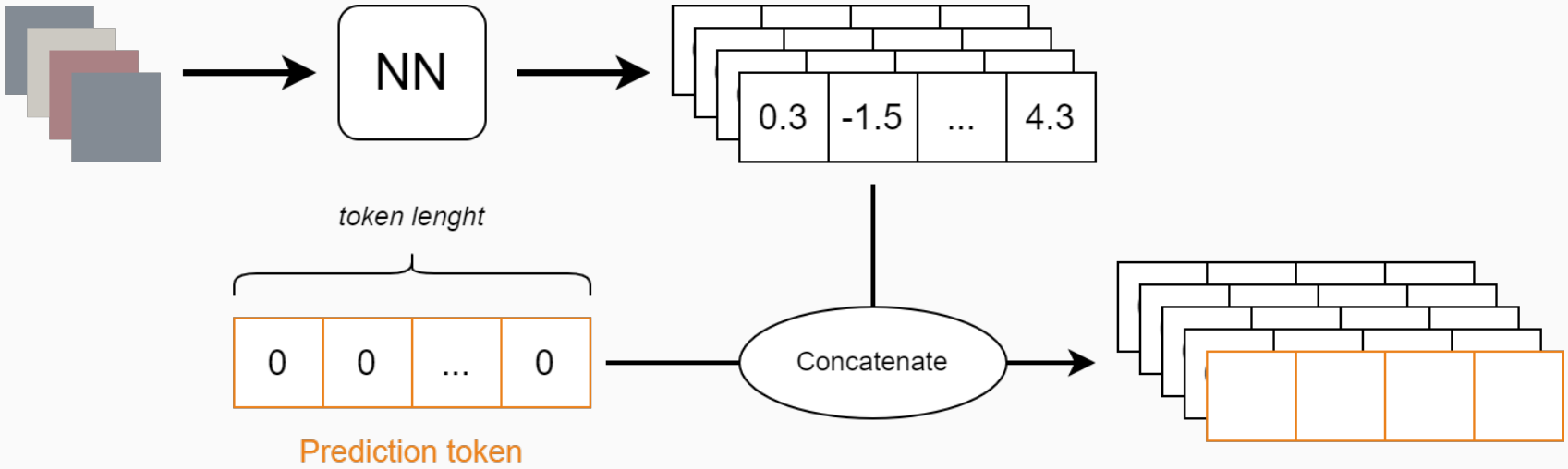
Procesamiento de tokens

El siguiente paso consiste en procesar los tokens para añadir información extra necesaria para mantener las relaciones espaciales y generar una salida.



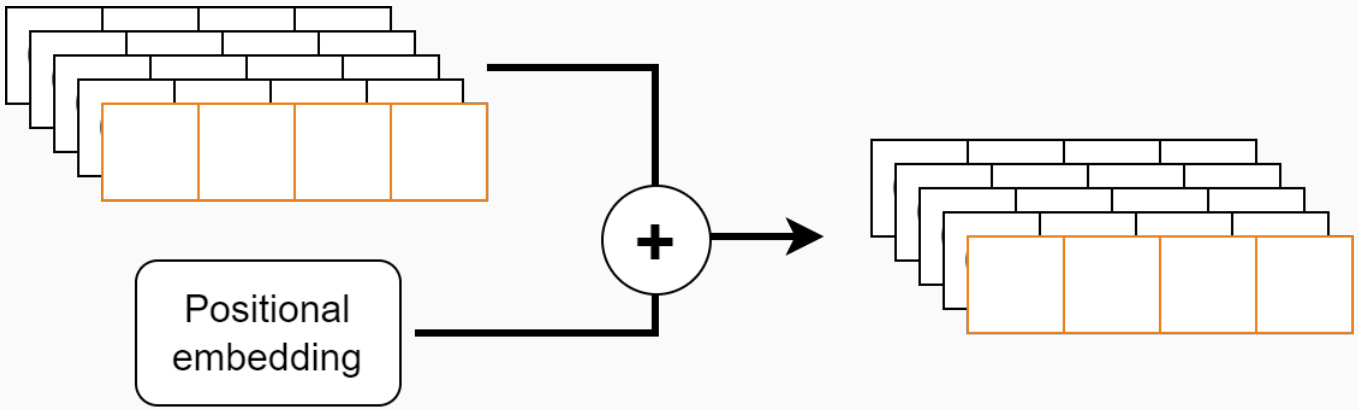
Procesamiento de tokens: Concatenar el *prediction token*

El *prediction token* es un vector que siempre se inicializa a 0, usado para generar la salida de predicción tras en encoding.



Procesamiento de tokens: Position embedding

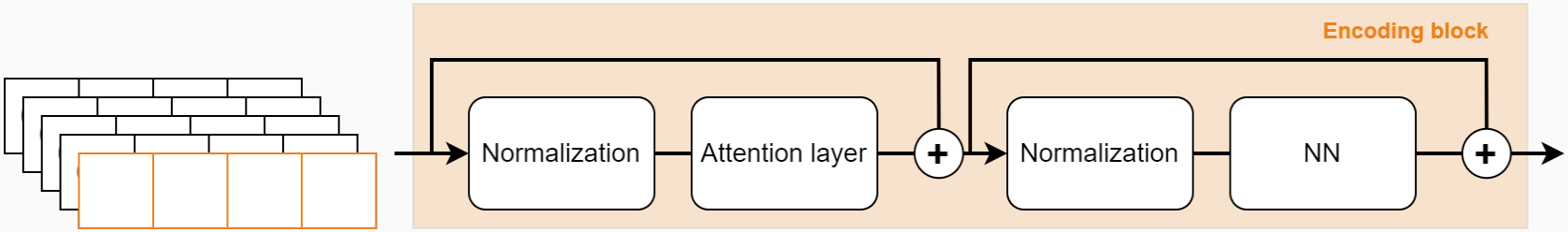
Para mantener las relaciones espaciales de los parches de una imagen se añade un positional embedding que ordena los embeddings. Esto ya se hacía en los Transformers para NLP.



*Es importante recalcar que el positional embedding es sumado a los anteriores. Normalmente se una una función sinusoidal.

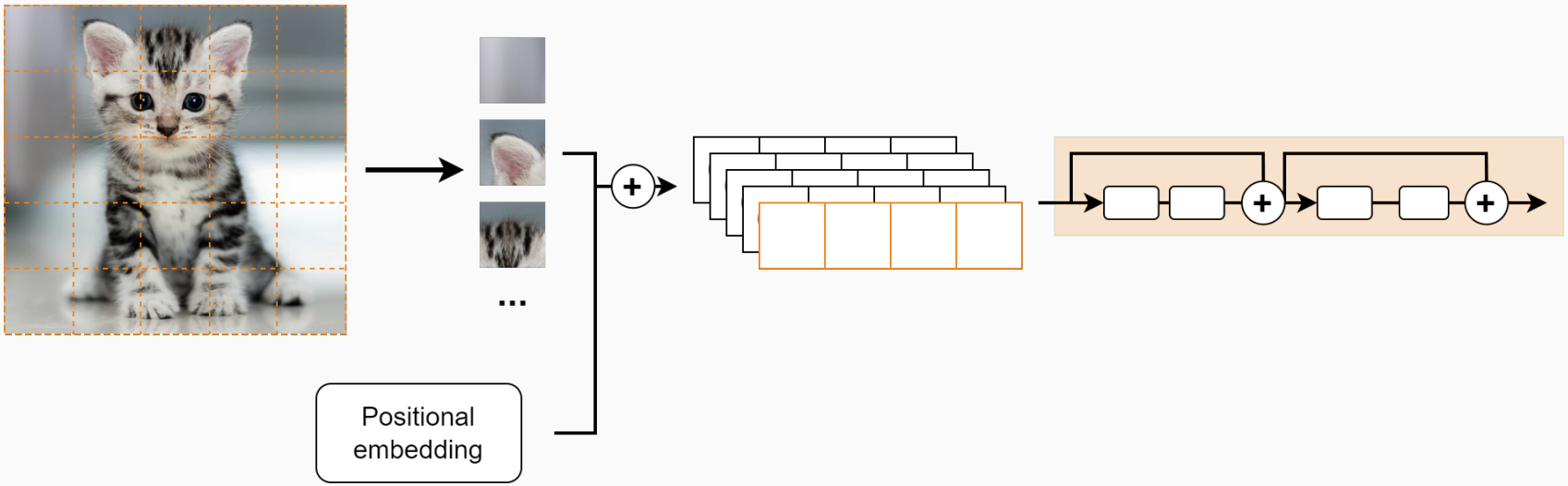
Encoding

La información generada previamente se procesa con bloques de encoding usando capas de atención. Se sigue el mismo procedimiento que en los Transformers vanilla.



Encoding: Intuición

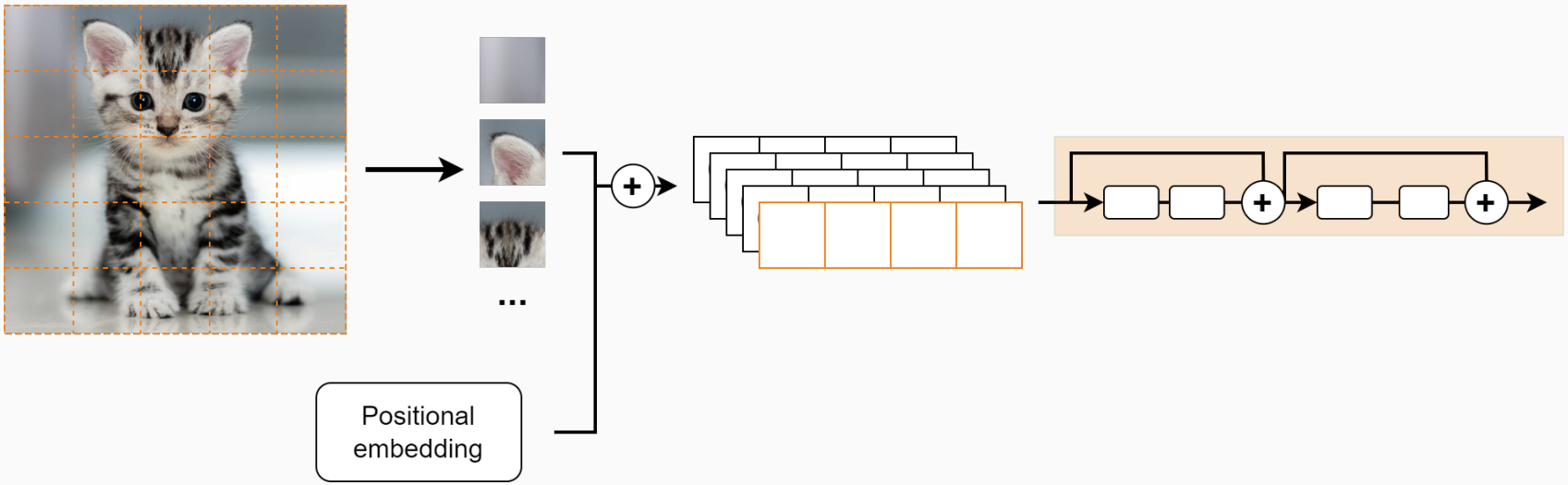
En este punto cabe recordar que el encoder está procesando cada parche de la imagen, proyectado a un espacio latente que guarda las relaciones entre parches. De esta forma es posible extraer características de las imágenes.



Salida del modelo

Una vez procesada la información, se genera la salida de la red de neuronas. Hasta este punto, la arquitectura del ViT es general y puede usarse para múltiples problemas. Pero en este ejemplo se explicará cómo hacer una clasificación.

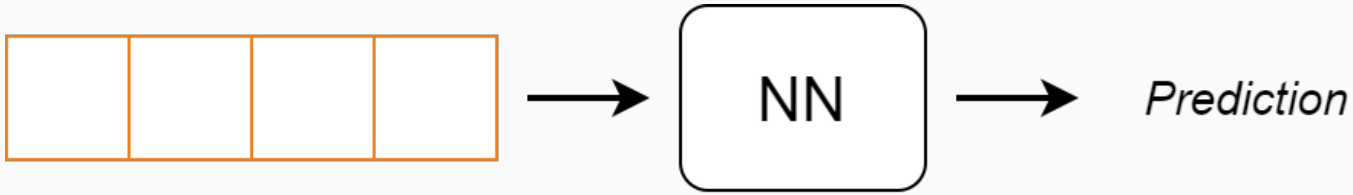
Para ello se usa el prediction token anteriormente definido.



Salida del modelo

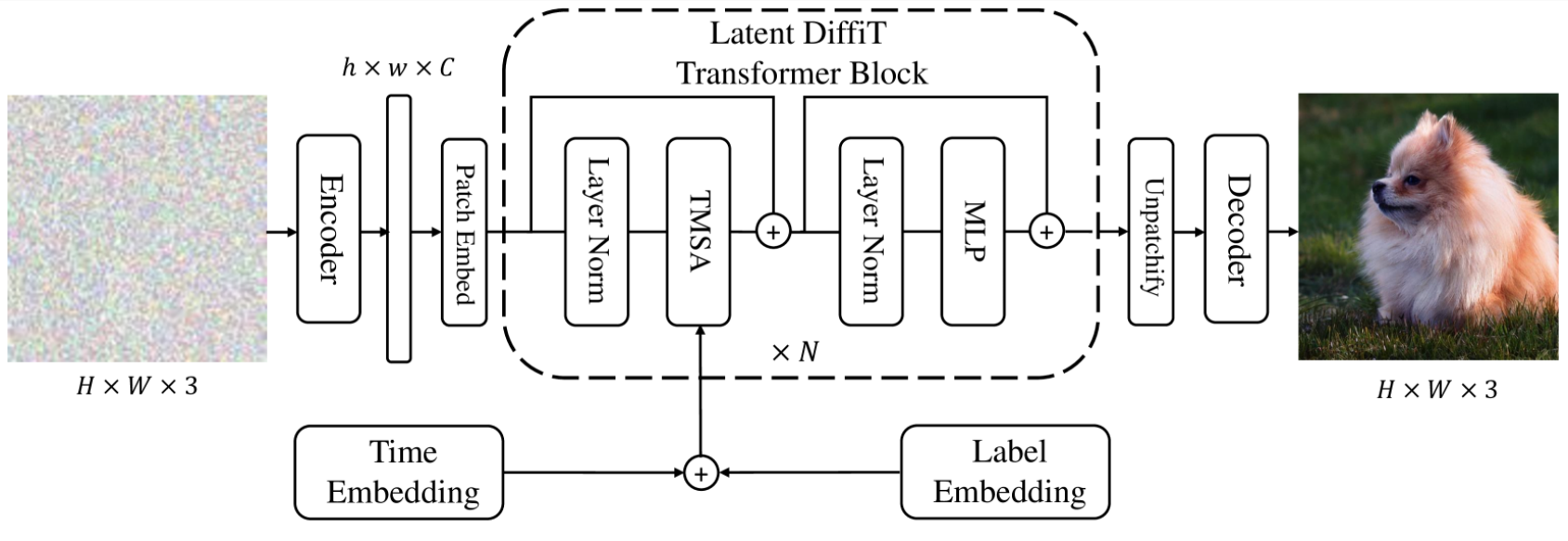
El prediction token es finalmente procesado por una cabeza de clasificación compuesta por un perceptrón.

Normalmente, este perceptrón no es más que una capa con activación lineal.



Generación de contenido: DiffiT

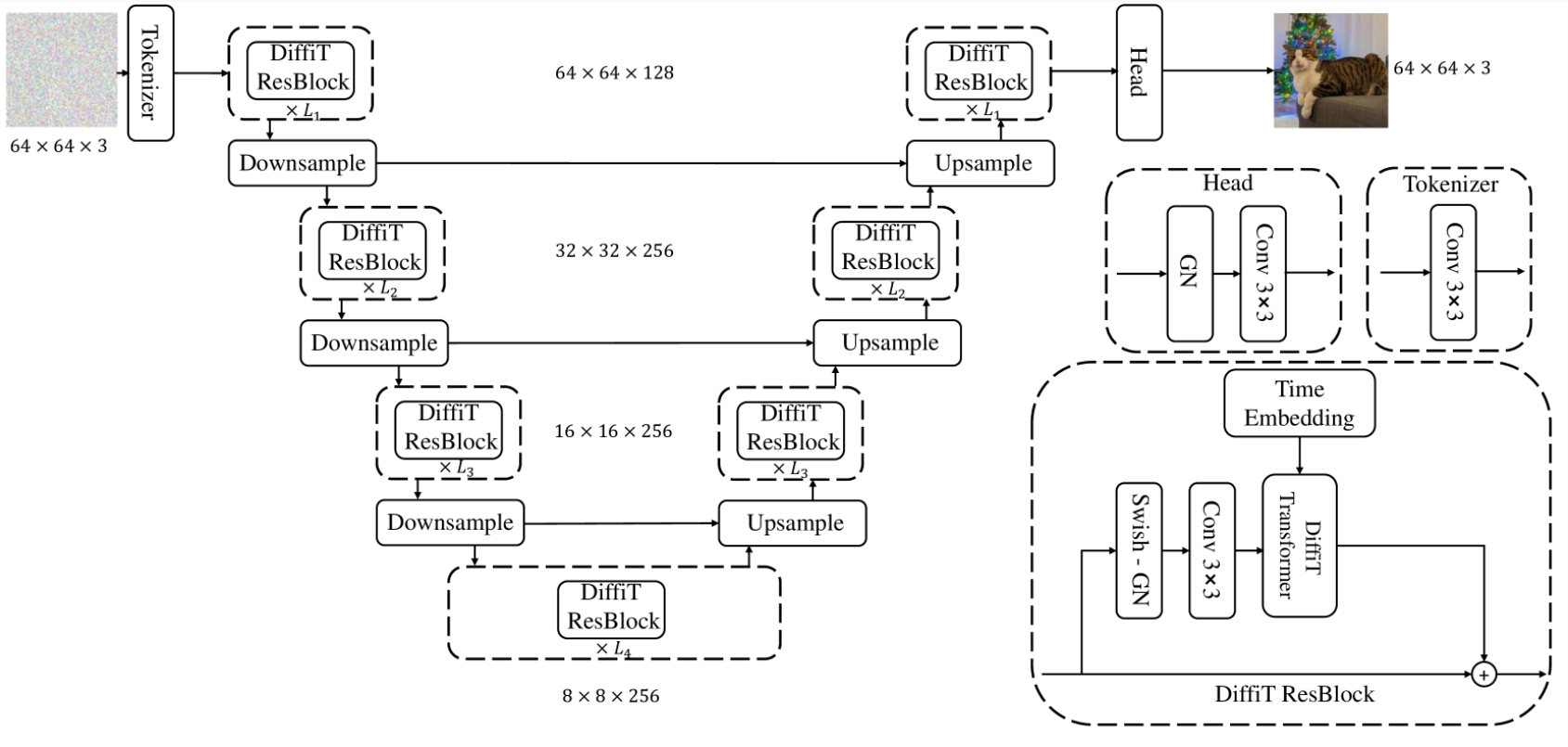
Existen arquitecturas como Diffusion Vision Transformer (DiffiT) [3] que combinan la potencia de los ViT con la capacidad de generación de los Diffusion models para generar contenido.



[3]

Generación de contenido: DiffiT

Combinando una arquitectura de Autoencoder (AE) e introduciendo bloques de ViT con un embedding temporal consiguen generar los mejores resultados del estado del arte.



[3]

Referencias i

[1] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al.
An image is worth 16x16 words: Transformers for image recognition at scale.
arXiv preprint arXiv:2010.11929, 2020.

[2] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zi-Hang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan.
Tokens-to-token vit: Training vision transformers from scratch on imagenet.
In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 558–567, 2021.

Referencias ii

[3] Ali Hatamizadeh, Jiaming Song, Guilin Liu, Jan Kautz, and Arash Vahdat.
Diffit: Diffusion vision transformers for image generation.
In *European Conference on Computer Vision*, pages 37–55. Springer, 2024.

Contribuciones de las diapositivas

- **Autor original de las diapositivas:** Guillermo Iglesias Hernández