

# ANÁLISIS DE PIX2PIX COMO SEGMENTADOR SEMANTICO

Por Antonio Jesús García Nieto

CONTROL DE VERSIONES

Versión	Descripción	Fecha de última modificación	Validado
1.0	Primera versión estable	11/04/21	No

Tabla de contenid

## O

Introducción.....	4
Análisis de la arquitectura.....	4
Componentes principales de Pix2Pix.....	4
Modelo generador.....	4
Modelo discriminador.....	5
Modelo completo.....	6
Planteamiento del estudio.....	7
Resultados.....	8
Los resultados para ensamblaje con maximizador.....	8
Los resultados para ensamblaje con minimizador.....	8
Los resultado para ensamblaje con red neuronal.....	9
Conclusión del experimento.....	9
Bibliografía.....	9

## INTRODUCCIÓN

En este experimento realizaremos trataremos de usar la arquitectura Pix2Pix como segmentador semántico. Pix2Pix fue propuesta en (Isola et al., 2016), es una solución flexible que se adapta bien a numerosos problemas de traducción de imagen a imagen. Esta arquitectura se basa en la inclusión de una U-Net a una red generativas adversaria condicionadas (cGANs).

En este experimento no trabajaremos no buscamos alcanzar unos resultados, si no trabajar con la arquitectura Pix2Pix y extraer de ella conceptos importantes. La hipótesis que pretendemos comprobar es:

1. *¿Sería posible utilizar Pix2Pix como un segmentador entrenable end-2-end compuesto por varias redes?*

## ANÁLISIS DE LA ARQUITECTURA

Como comentamos anteriormente Pix2Pix es una cGAN, este tipo de arquitecturas se caracteriza por enfrentar dos modelos que realizan tareas antagónicas, mientras uno de ellos actúa de generador, el otro actúa de discriminante.

En el caso en concreto de Pix2Pix, mientras que un modelo basado en U-Net genera imágenes artificiales, el otro modelo (discriminador) intenta decidir si la imagen ha sido creada artificialmente o no.

Comenzaremos explicando los componentes de esta red y posteriormente veremos el funcionamiento de la red en mayor profundidad.

## COMPONENTES PRINCIPALES DE PIX2PIX

### Modelo generador

En concreto Pix2Pix utiliza como módulo generador una U-Net, un tipo de arquitectura que mejora el desempeño de las típicas arquitecturas FCNN (completamente convolucionales). La diferencia fundamental es que mientras las FCNN están formadas por un encoder y un decoder aislados (módulos que comprimen y descomprimen información respectivamente), las U-Net añaden saltos de conexiones entre las capas del encoder y el decoder lo que permite que el decoder tenga visión de varios mapas de características.

Lo mostramos gráficamente:

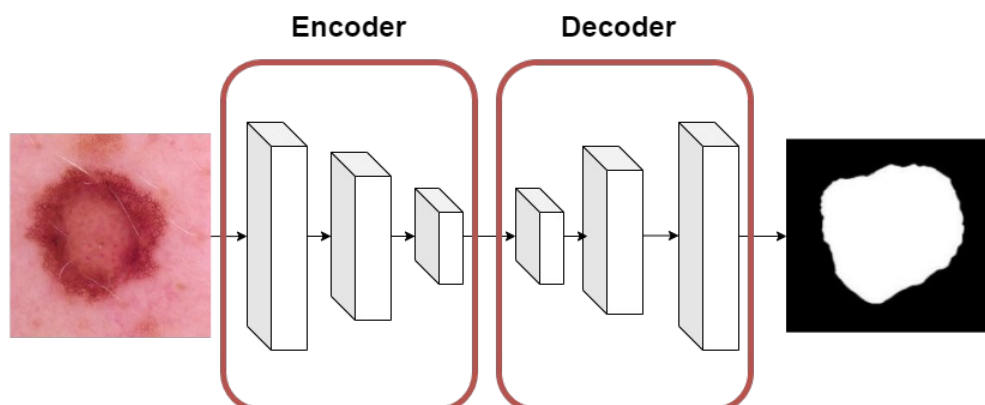


Ilustración 1. Arquitectura FCNN

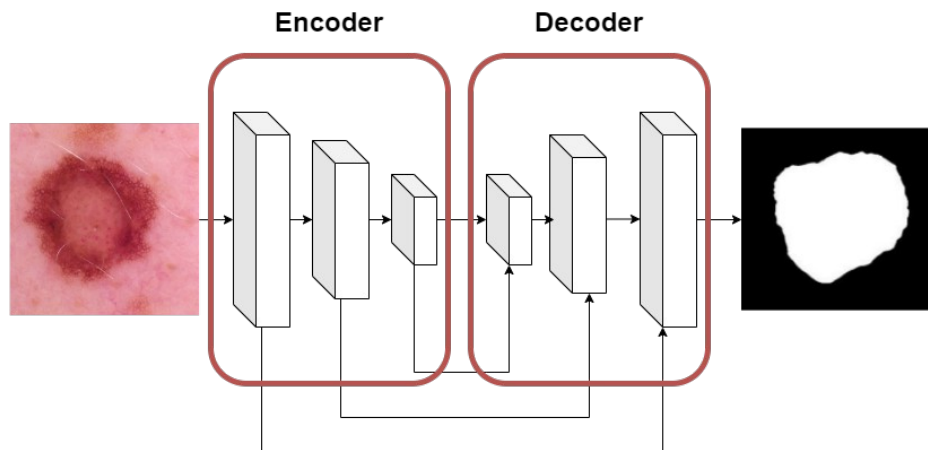


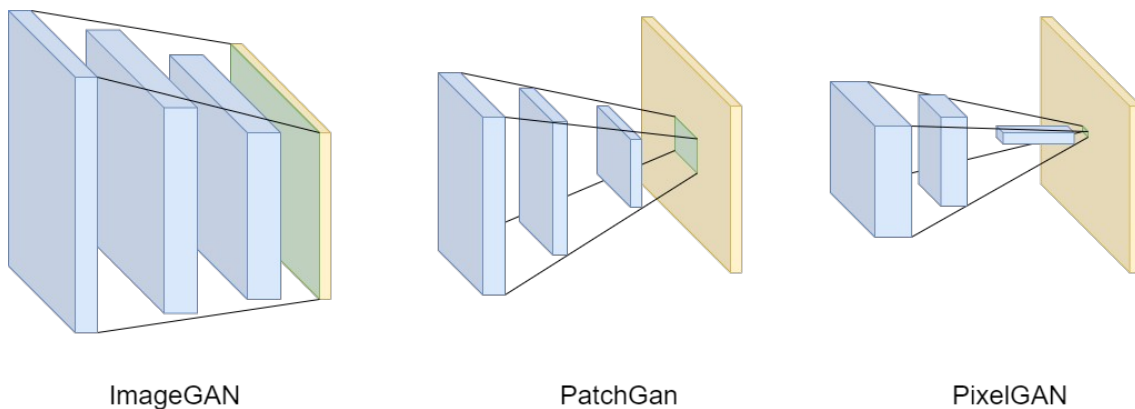
Ilustración 2. Arquitectura U-Net

## Modelo discriminador

El discriminador que propone Pix2Pix está basado en una PatchGAN. Una PatchGAN es un tipo particular de GAN que da un resultado binario sobre si un parche de  $N \times N$  píxeles de la imagen es o no generados artificialmente. Existen en total tres casos básicos de este tipo de GAN:

- **PatchGAN**. Como hemos dicho da una clasificación binaria de si un parche de  $N \times N$  píxeles es o no generado artificialmente.
- **PixelGAN**. Es un caso específico de PatchGAN en el que el  $N$  es igual a 1, es decir, consiste en la evaluación pixel a pixel.
- **ImageGAN**. Es también un caso específico de la PatchGAN, en este caso el valor de  $N$  es el del largo de la imagen (se supone una imagen cuadrada).

A modo de ilustración se presentan las siguientes imágenes:

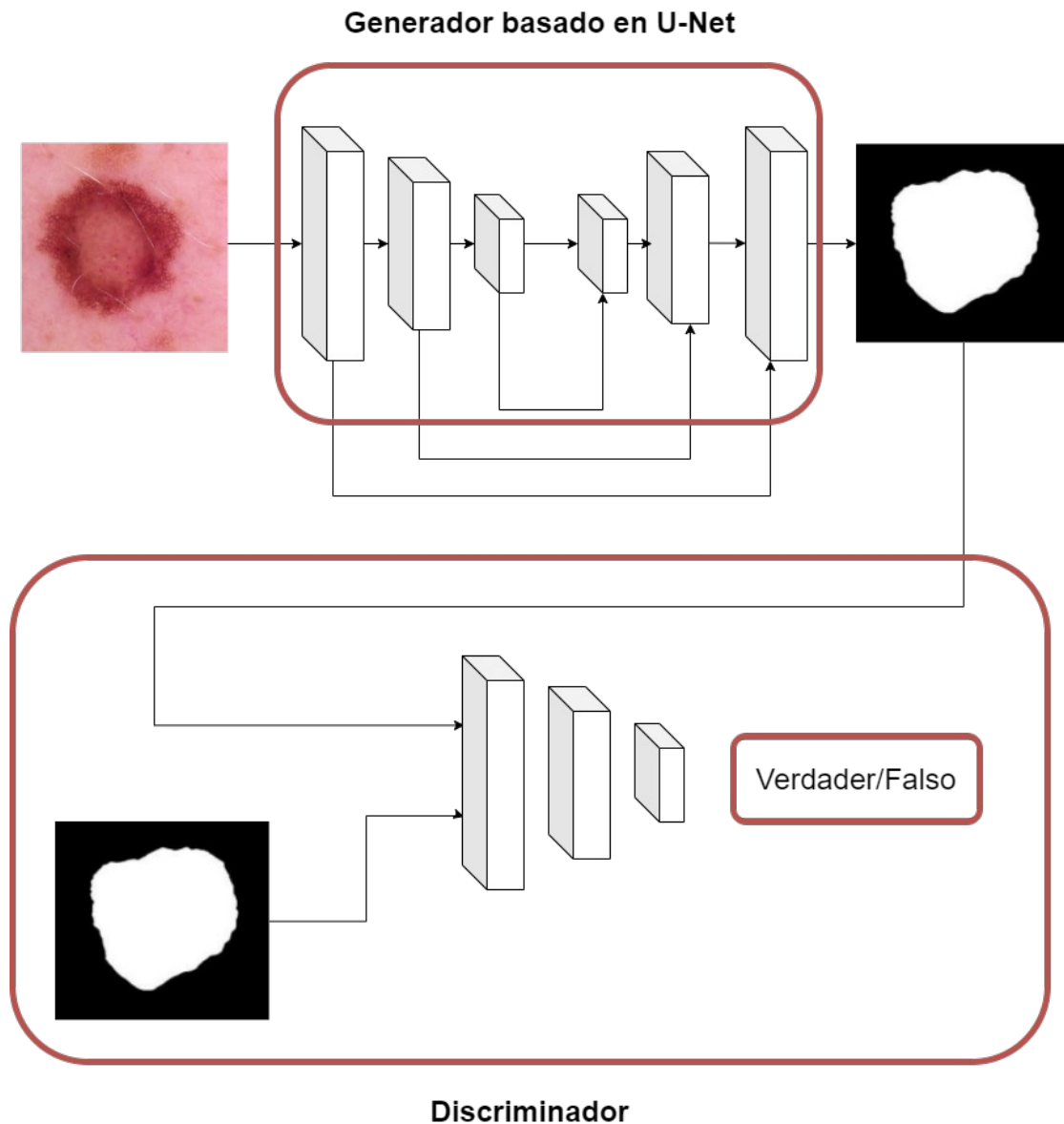


Las figuras azules serían unidades de procesamiento no lineal, la zona amarilla sería la imagen y la zona verde la superficie evaluada.

## MODELO COMPLETO

Una vez hemos introducido las dos partes de nuestra arquitectura ya podemos describir el funcionamiento general.

En primer lugar, el modelo generador (U-Net) crea la segmentación semántica del modelo. Posteriormente esta imagen entra en el modelo discriminador, la PatchGAN anteriormente explicada, este realizará una clasificación de la imagen de entrada. Lo interesante de este modelo es que se generan dos funciones de pérdida, la del generador y la del discriminador, lo que permite aprender a los dos modelos, llegando a un punto idílico en el que el generador de resultados tan buenos, que el modelo de no sea capaz de detectar la diferencia entre los datos reales y los generados.



## TRABAJO EMPÍRICO CON LA ARQUITECTURA

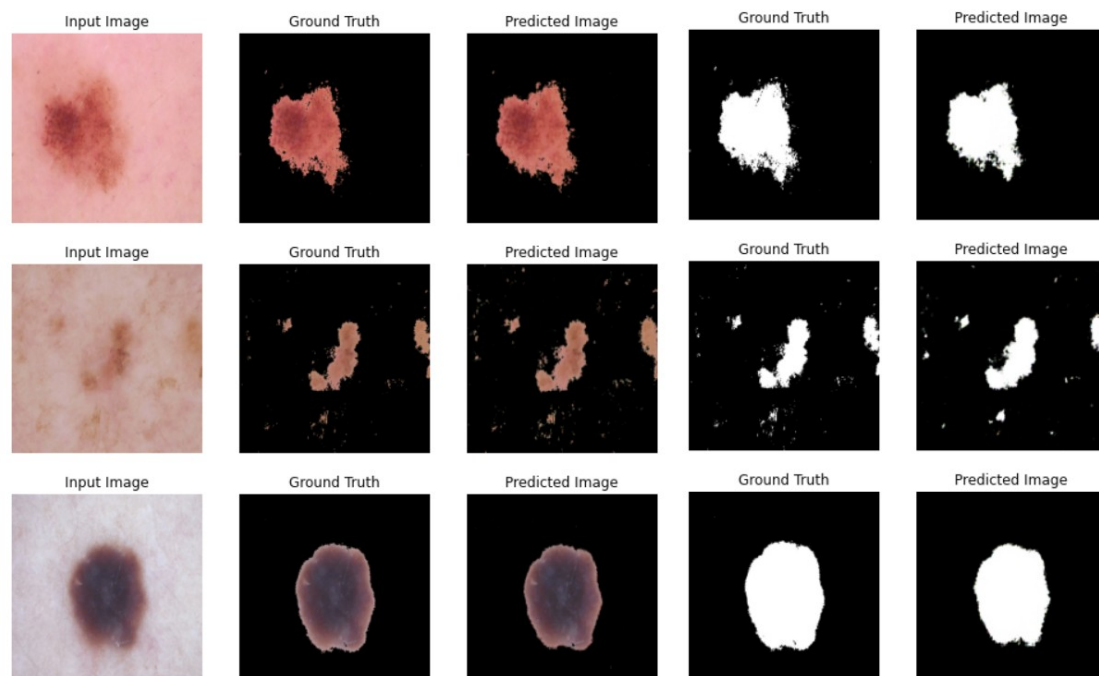
Esta arquitectura es bastante diferente a un segmentador convencional, además para trabajar con ella es necesario construirla desde 0, para ello es necesario trabajar a bajo nivel con tensorflow.

Como guía para poder construir esta red hemos seguido tanto el artículo anteriormente mencionado, en el cual hace referencia a un cuadernillo Jupyter con la construcción de la arquitectura y por otra parte se ha recurrido a un video explicativo de esta arquitectura (*Generando FLORES Realistas Con IA - Pix2Pix | IA NOTEBOOK #5 - YouTube*, n.d.).

Los detalles de la implementación se pueden encontrar en los cuadernillos referentes a este experimento en [GitHub](#).

## CONCLUSIÓN DEL EXPERIMENTO

Los resultados a priori son bastante muy buenos, como podemos observar en las siguientes imágenes, tanto para la segmentación como para la eliminación del fondo.



No obstante, se ha decidido no invertir tiempo en gestionar la evaluación de la arquitectura, pues al no ser entrenable end-to-end junto a otros modelos (debido al módulo discriminador), a su complejidad y a su alto número de parámetros entrenables (50 millones para imágenes de 256x256), se ha decidido no utilizarla en la solución propuesta.

Aún así se considera que la investigación profunda de esta red podría ser un camino interesante de seguir.

## BIBLIOGRAFÍA

*Generando FLORES realistas con IA - Pix2Pix | IA NOTEBOOK #5 - YouTube*. (n.d.). Retrieved May 17, 2021, from <https://www.youtube.com/watch?v=YsrMGcgfETY&t=120s>

Isola, P., Zhu, J.-Y., Zhou, T., & Efros, A. A. (2016). Image-to-Image Translation with Conditional Adversarial Networks. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, 2017-January*, 5967–5976. <http://arxiv.org/abs/1611.07004>

