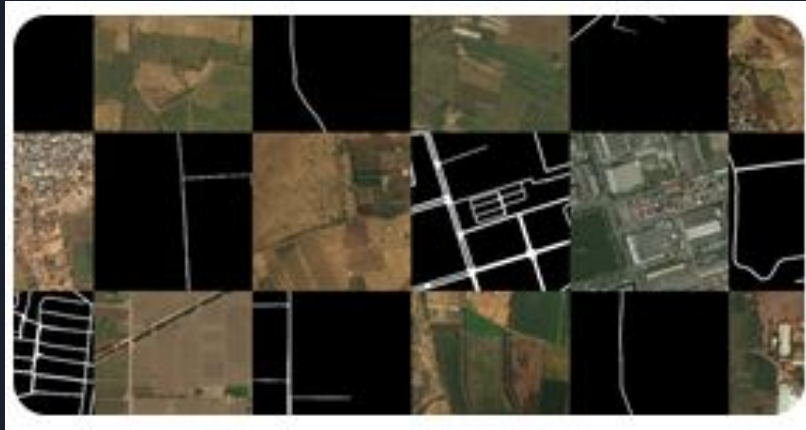


# Computer Vision 2

Carina Roldán, Eloy Zerbatto

# Road Extraction

**OBJETIVO:** Identificar carreteras y redes de calles en imágenes satelitales





# Road Extraction

## CONJUNTO DE DATOS

### Entrenamiento

- 6226 imágenes RGB
- Tamaño: 1024 x 1024
- Resolución espacial: 50 cm x pixel

### Validación

1243 imágenes

### Test

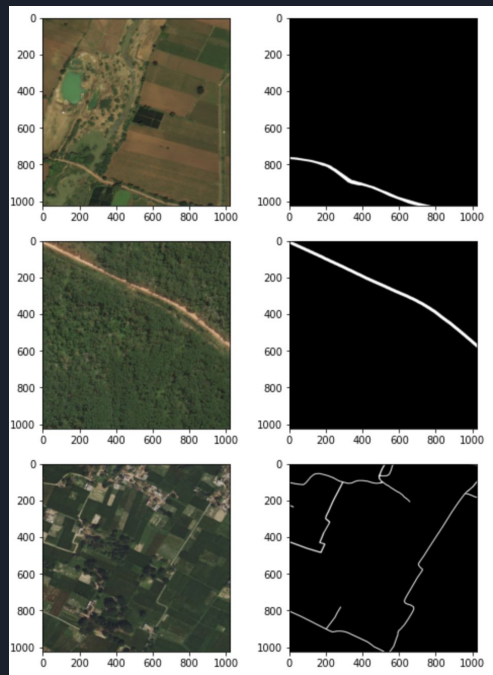
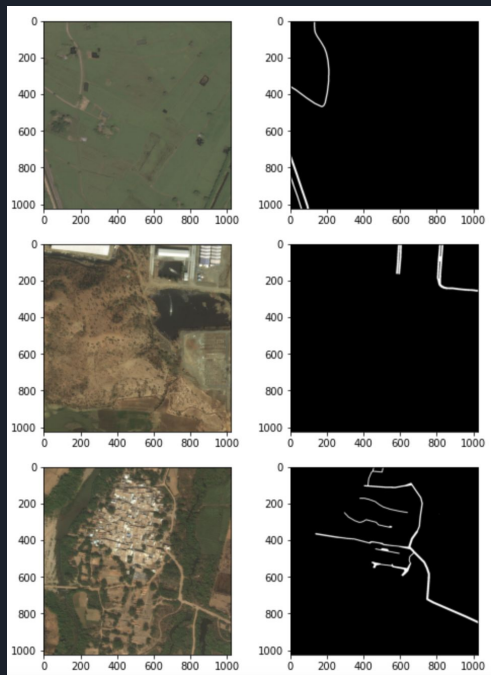
1101 imágenes  
(sin máscara)

<https://www.kaggle.com/datasets/balraj98/deepglobe-road-extraction-dataset>

Fuente: DigitalGlobe (<https://www.digitalglobe.com>)

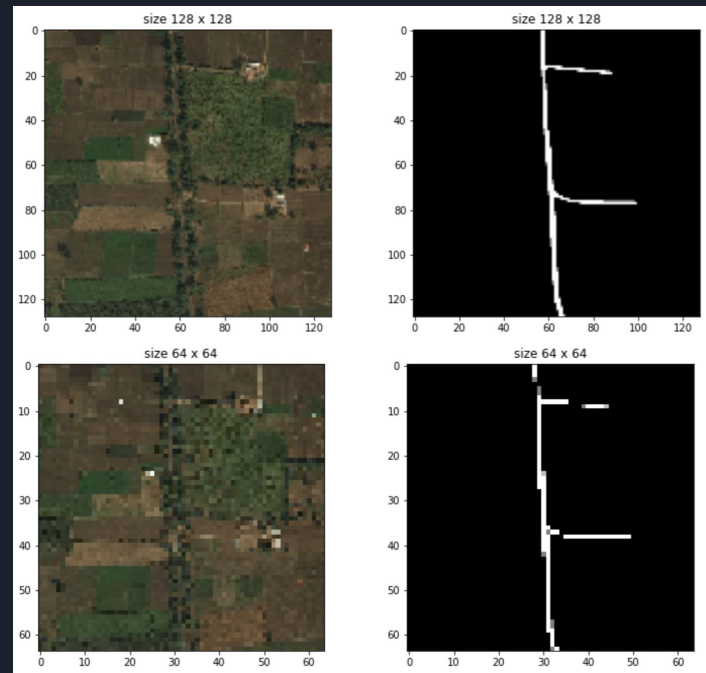
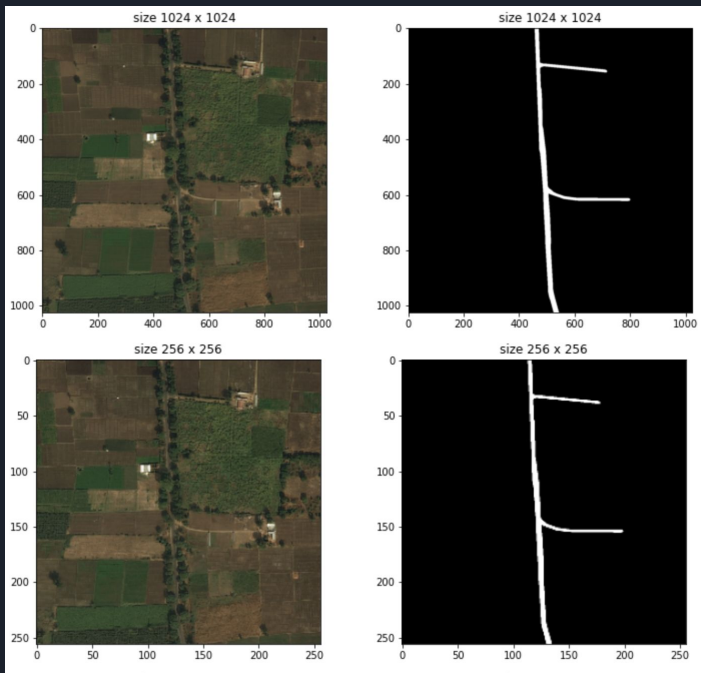
# Road Extraction

## EDA - Imágenes de muestra y sus máscaras



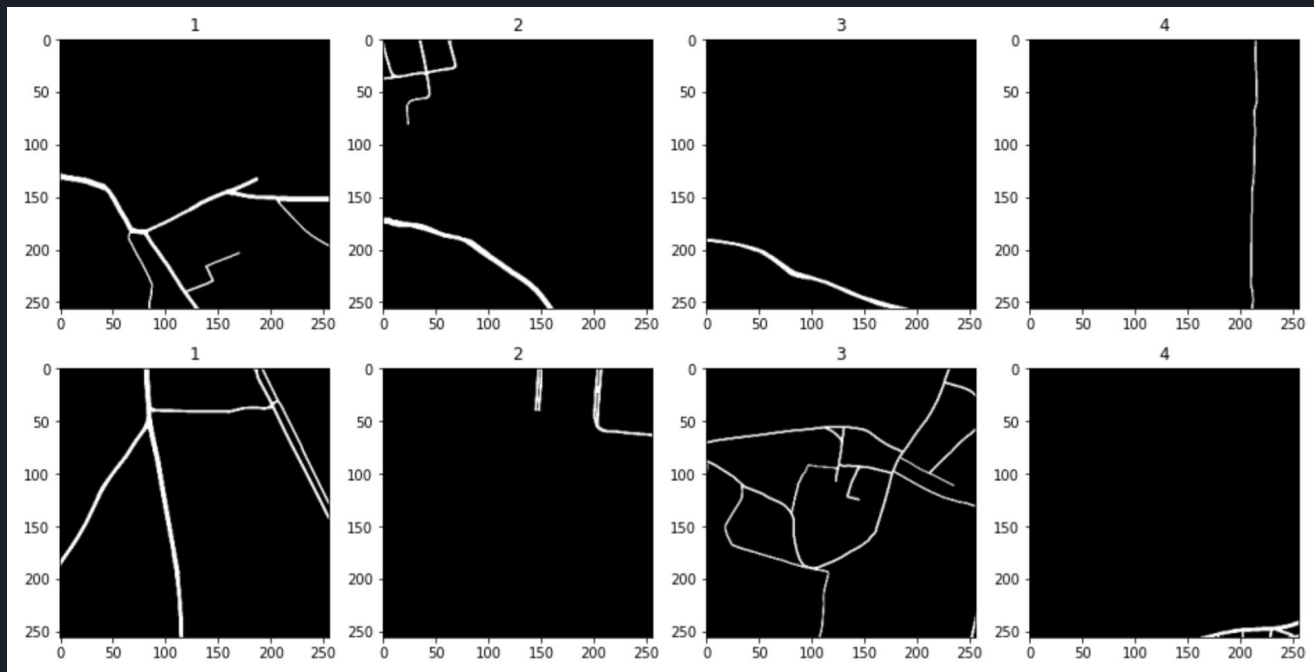
# Road Extraction

## EDA - Explorando la misma imagen a distintas escalas



# Road Extraction

## EDA - Explorando máscaras *resizeadas* a 256x256





# Road Extraction

## SOLUCIÓN PROPUESTA

Realizar una segmentación  
para identificar dos clases:  
“carretera” y “fondo”

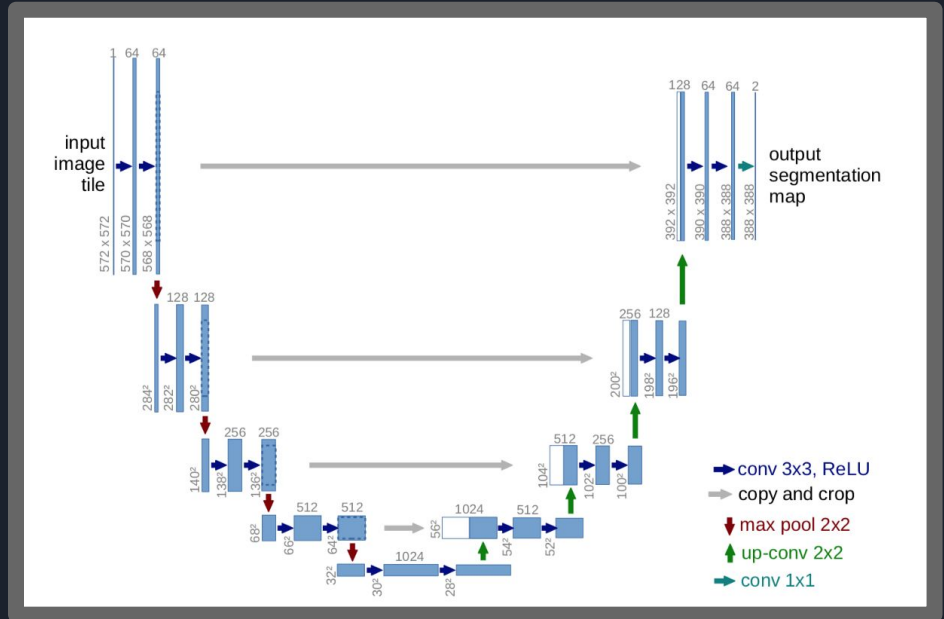
<https://www.kaggle.com/datasets/balraj98/deepglobe-road-extraction-dataset>

Fuente: DigitalGlobe (<https://www.digitalglobe.com>)

# MODELO A ENTRENAR

# MODELO A ENTRENAR

- Encoder/decoder con skip connections
- Inspirado en **U-NET**

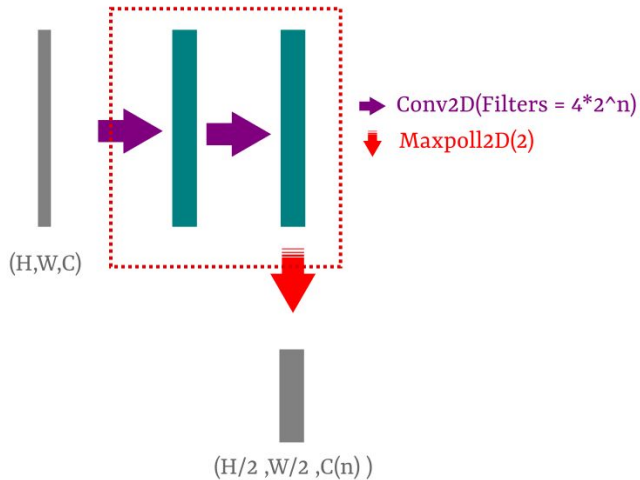




# Road Extraction

## MODELO A ENTRENAR

## Encoder



bloques de dos conv2D  
(Kernel 3x3) + LeakyRelu

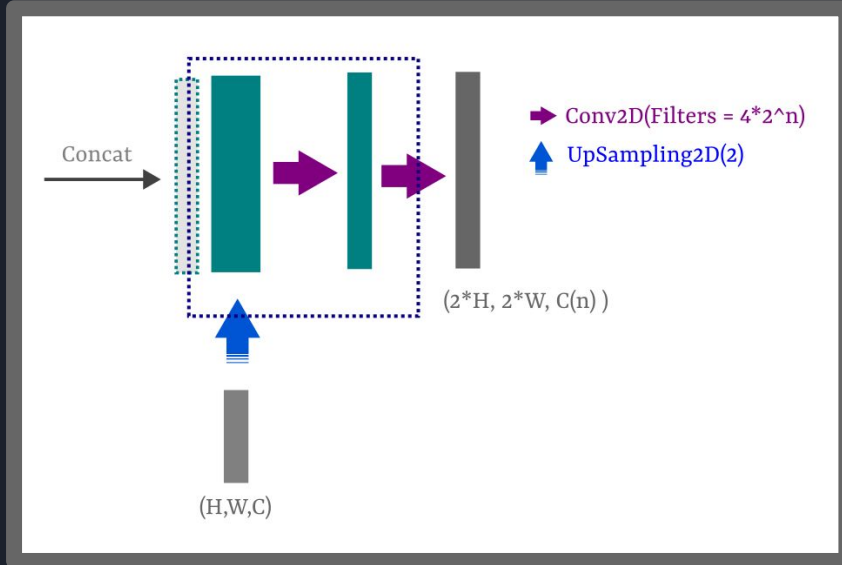
skip connection

maxpool2D (poolsize = 2)

padding Same

# Road Extraction

## MODELO A ENTRENAR



## Decoder

`Upsampling(Size = 2)`

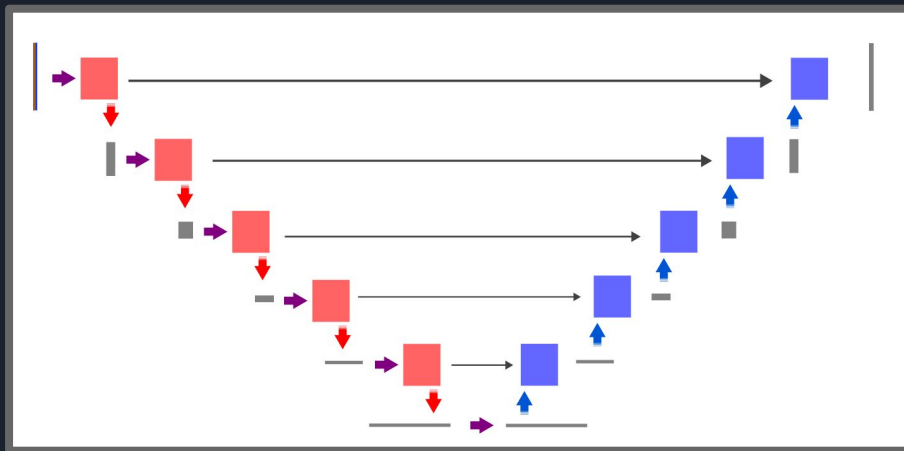
concatenación

bloques de dos conv2D  
(Kernel 3x3) + LeakyRelu

padding Same

# Road Extraction

## MODELO A ENTRENAR



Total params: 492,081

Trainable params: 492,081

Non-trainable params: 0

input = (256,256,3)

output = (256,256,1)



# Road Extraction

## ESTRATEGIA DE DISEÑO

- Se tomó como inspiración **U Net**
- Se reemplazaron las **ConvTraspuestas** por **Upsampling+Conv2D** (arquitectura más fácil de modificar para las pruebas)

### Búsqueda de hiperparámetros

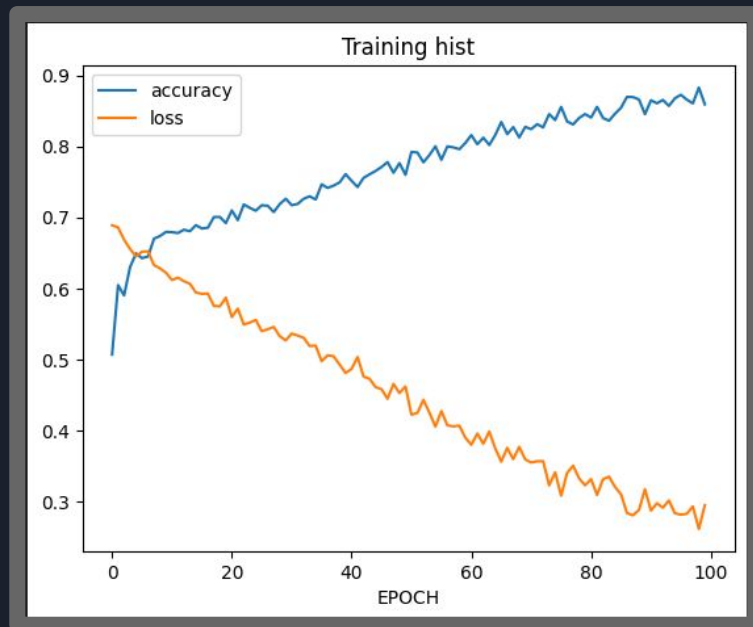
Combinación de *cantidad de filtros*, *learning rate*, *tamaño de la imagen de entrada* en iteraciones sobre unas *pocas EPOCHS* y sobre un *set de datos reducido*.

# Road Extraction

## RESULTADOS PRELIMINARES

Predicciones luego de **100 epochs** (rápidas) del pre entrenamiento:

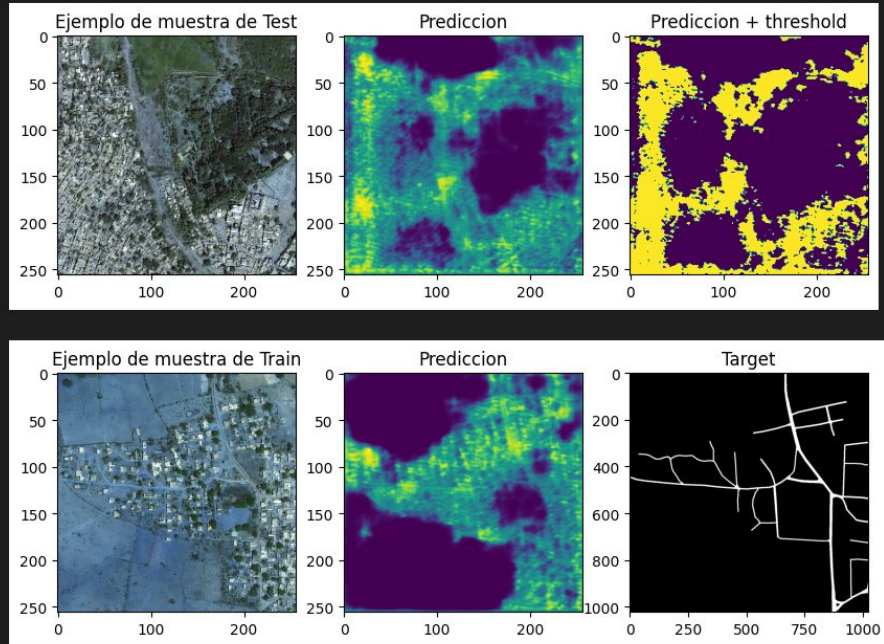
- En el hist se ve que la pérdida (*loss*) aún puede seguir bajando.



# Road Extraction

## RESULTADOS PRELIMINARES

- Se ve que no se resuelve el problema pero ya detecta zonas urbanas.





# Road Extraction

## ESTRATEGIA DE DISEÑO (cont.)

Una vez que se decidió el tamaño de la red y de las entradas, se corrió un entrenamiento con un *learning rate* de **0.01** por **100** epoch con un set de datos pequeño.

### Objetivo

- verificar que el modelo tendía a aprender
- buscar un conjunto de pesos iniciales (como un pre entrenamiento)



# Road Extraction

## ESTRATEGIA DE DISEÑO (cont.)

- Se realizó un entrenamiento tomando el estado anterior de los pesos, con un learning rate variable (haciendo uso de un scheduler).
- Se inició en **0.001** durante unas pocas epochs y luego con **0.0001**.

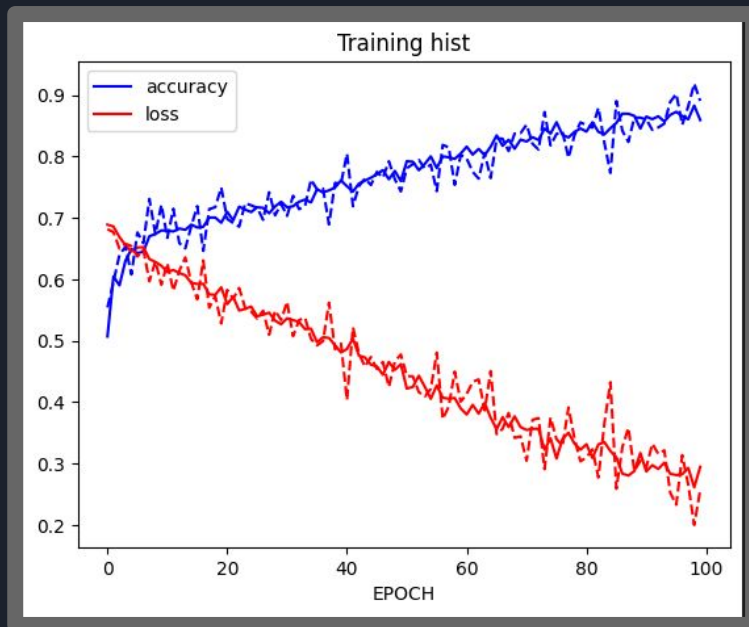


# Road Extraction

## RESULTADOS DEL ENTRENAMIENTO FINAL

Predicciones luego de **100 epochs** del entrenamiento:

- La tendencia en la gráfica nos indica que si continuamos entrenando el modelo, el mismo podría seguir mejorando.





# Road Extraction

## RESULTADOS DE LOS EXPERIMENTOS

### ### Experimento A

- \* Total params: 119,329
- \* Trainable params: 119,329
- \* Non-trainable params: 0
- \* Input = (256,256,3)
- \* Output = (235,235,1)

#### *\*Entrenamiento\**

- \* 20 EPOCH
- \* Arquitectura tipo encoder/decoder con skip connections. (Conv2D + Conv2DTranspose)
- \* Learning rate variable con decaimiento exponencial. (lr inicial: 0.001)
- \* Accuracy en test: 74%



# Road Extraction

## RESULTADOS DE LOS EXPERIMENTOS

### ### Experimento B

- \* Total params: 1,902,913
- \* Trainable params: 1,902,913
- \* Non-trainable params: 0
- \* Input = (256,256,3)
- \* Output = (256,256,1)

#### *\*Entrenamiento\**

- \* 20 EPOCH
- \* Arquitectura tipo encoder/decoder con skip connections. (Conv2D + Conv2DTranspose)
- \* Learning rate variable con decaimiento exponencial. (lr inicial: 0.001)
- \* Accuracy en test: 74%



# Road Extraction

## RESULTADOS DE LOS EXPERIMENTOS

### ### Experimento C

- \* Total params: 492,081
- \* Trainable params: 492,081
- \* Non-trainable params: 0
- \* Input = (256,256,3)
- \* Output = (256,256,1)

#### *\*Entrenamiento\**

- \* 20 EPOCH
- \* Arquitectura tipo encoder/decoder con skip connections. (Conv2D + Conv2DTranspose)
- \* Learning rate variable con decaimiento exponencial. (lr inicial: 0.001)
- \* Accuracy en test: 76%



# Road Extraction

## RESULTADOS DE LOS EXPERIMENTOS

### ### Experimento D

- \* Total params: 476,257
- \* Trainable params: 476,257
- \* Non-trainable params: 0
- \* Input = (256,256,3)
- \* Output = (235,235,1)

#### *\*Entrenamiento\**

- \* 50 EPOCH
- \* Arquitectura tipo encoder/decoder con skip connections. (Conv2D + Conv2DTranspose)
- \* Learning rate variable con decaimiento exponencial. (lr inicial: 0.001)
- \* Accuracy en test: 79%



# RESULTADOS DE LOS EXPERIMENTOS

## ### Experimento FINAL

- \* Total params: 476,257
- \* Trainable params: 476,257
- \* Non-trainable params: 0
- \* Input = (256,256,3)
- \* Output = (256,256,1)
- \* Arquitectura tipo encoder/decoder con skip connections. (Conv2D + Upsampling)

### *\*Entrenamiento\**

#### *\_Primera parte del entrenamiento:\_*

- \* 100 muestras
- \* 5 EPOCH , lr = 0.001
- \* 95 EPOCH , lr = 0.0001

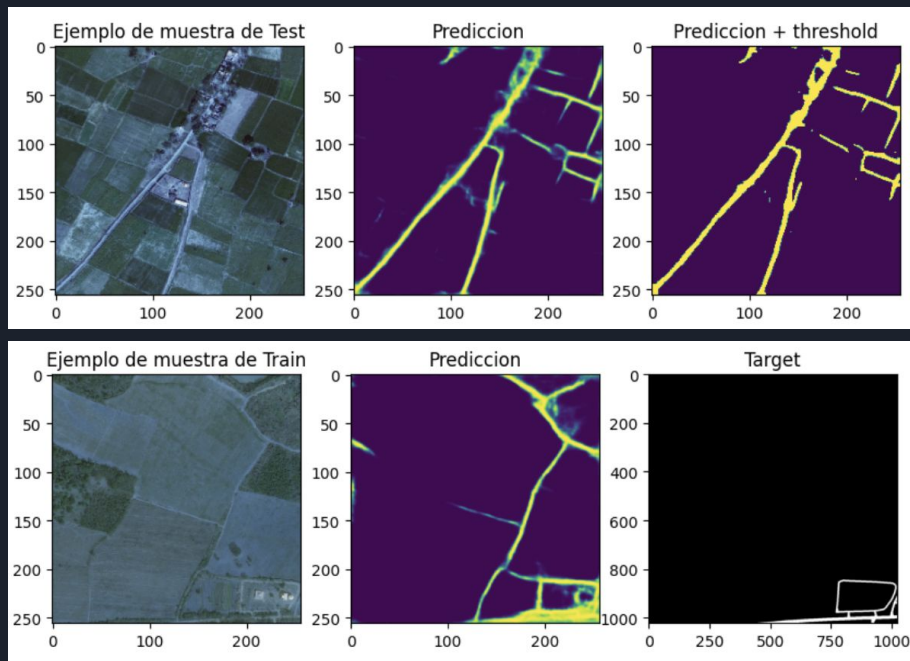
#### *\_Segunda parte del entrenamiento:\_*

- \* Pesos de la primera parte
- \* +4200 muestras
- \* 5 EPOCH , lr = 0.001
- \* 95 EPOCH , lr = 0.0001

- \* Accuracy en test: 92% en 100 epoch

# Road Extraction

## RESULTADOS





# Road Extraction

## CONCLUSIONES

- Se buscó diseñar una arquitectura *custom* que sea capaz de resolver la problemática de segmentación semántica de rutas.
- Para los datos de entrada se buscó un *shape* mínimo que no afecte el proceso de entrenamiento pero que a su vez permita implementar un modelo reducido.





# Road Extraction

## CONCLUSIONES (CONT.)

- Se realizaron experimentos con distintas funciones de activación, estrategias de *learning rate* y variantes de la arquitectura inicial.
- Inicialmente, el resultado del entrenamiento no fue satisfactorio debido al fuerte desbalance de clases en los datos de entrada. Para salvar este problema y obtener mejores resultados se utilizaron clases con pesos.



# Road Extraction

## CONCLUSIONES (CONT.)

Para la arquitectura elegida, la pérdida durante el entrenamiento muestra una tendencia a bajar.

Esto indicaría que es posible proseguir con el entrenamiento y así lograr una *accuracy* en test superior al **92%**, valor alcanzado hasta el momento.



¡Gracias!