Máster SIANI

Computación Inteligente

Trabajo redes neuronales

Image Inpating

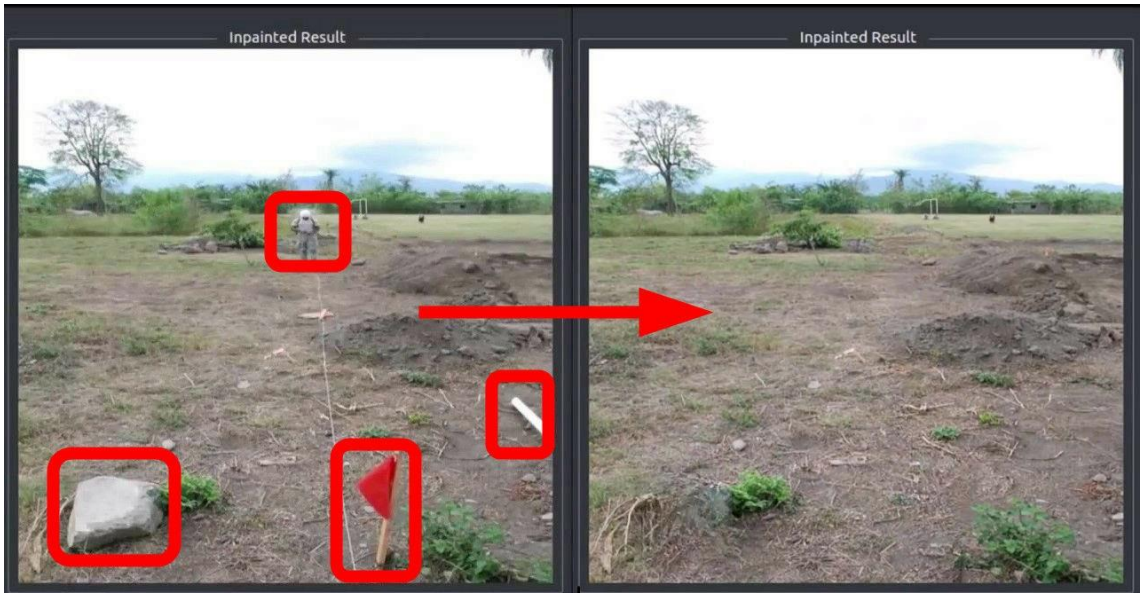
Leopoldo lopez reveron

Computación Inteligente

Universidad de las palmas de Gran Canaria

Contexto del problema:

El campo de estudio del image inpating busca conseguir que, a



partir de una imagen, a la cual se le elimina información, lograr restaurar la información de la imagen sin necesidad de ser la misma, es decir, se buscar que la información reconstruida sea imperceptible tanto al ojo humano, como en técnicas de análisis de la imagen.

Como una toma de contacto del problema se entreno un modelo con el dataset de MNIST modificado, para ver las capacidades del modelo.

Dentro de las modificaciones hechas al dataset se encuentran, pasar del formato unidimensional de las imágenes a imágenes 2D debido a que el modelo implementado se trata de un Auntoencoder convolucional con la siguiente configuración:

Encoder

nn.Conv2d(1, 8, 3, stride=2, padding=1), nn.ReLU
nn.Conv2d(8, 16, 3, stride=2, padding=1), nn.BatchNorm2d(16),
nn.ReLU
nn.Conv2d(16, 32, 3, stride=2, padding=0), nn.ReLU

Vector Latente

self.flatten = nn.Flatten(start_dim=1)
nn.Linear(3 * 3 * 32, 128), nn.ReLU
nn.Linear(128, encoded_space_dim)
nn.Linear(, 128), nn.ReLU
nn.Linear(128, 3 * 3 * 32), nn.ReLU

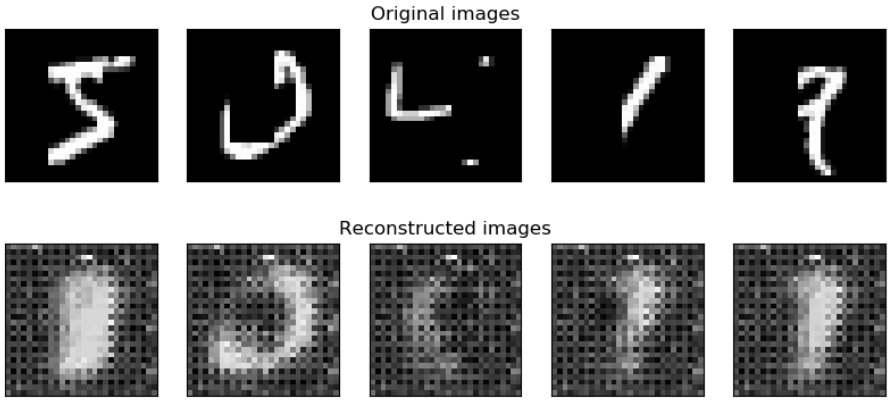
Decoder

self.unflatten = nn.Unflatten(dim=1, unflattened_size=(32, 3, 3))
nn.ConvTranspose2d(32, 16, 3, output_padding=0),
nn.BatchNorm2d(16), nn.ReLU,
nn.ConvTranspose2d(16, 8, 3output_padding=1),
nn.BatchNorm2d(8), nn.ReLU
nn.ConvTranspose2d(8, 1, 3, padding=1, output_padding=1)

La métrica de Loss utilizada es MSE y el optimizador ADAM.

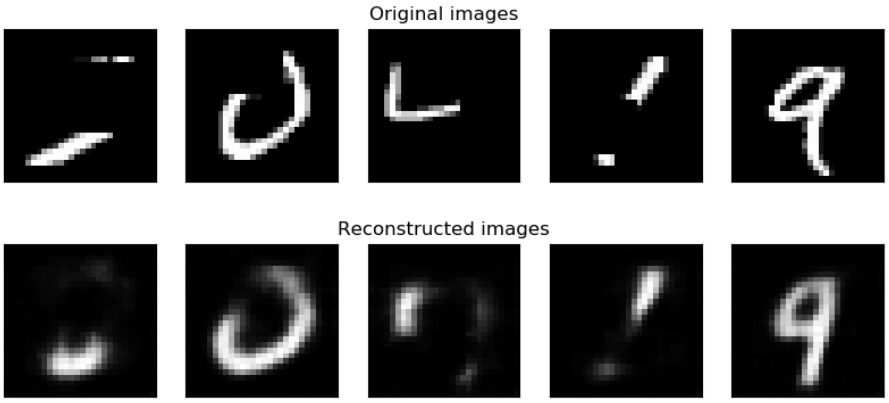
Después del entrenamiento en un GTX1070, que tardo 5 horas mientras se realizaba un uso normal del mismo, se obtuvieron los siguientes resultados:

Este es el caso de la primera época entrenada, como podemos observar el



modelo todavía no a convergido a ningún resultado y sigue arrojando ruido en las imágenes.

Este es el caso de la última época entrenada y podemos ver como el modelo ya no tiene ruido e intenta reconstruir las imágenes, aunque se ve



una perdida de calidad de la imagen y el modelo realmente no parece reconstruir la imagen.

En vista de los resultados se decidió cambiar el modelo y el dataset, pues realmente el objetivo no es reconstruir figura sino partes de imágenes. En cuanto al dataset se seleccionaron 2000 imágenes de train y 1000 de test. Para crear las imágenes a las que les falta información se optó por crearlas dinámicamente permitiendo de esta forma que los recortes de estas cambien entre épocas, evitando así sesgos de la red al ver siempre los mismos recortes en las mismas imágenes.

En cuanto al modelo tenemos la siguiente configuración adaptada para procesar imágenes en color:

Encoder

Conv2d(3, 8, 3), MaxPool2d(3), BatchNorm2d(8), ReLU
Conv2d(8, 16, 3),MaxPool2d(3), BatchNorm2d(16), ReLU
Conv2d(16, 32, 3),MaxPool2d(3,), ReLU

Vector Latente

Flatten(start_dim=1)
Linear(2048, 256), ReLU
Linear(256, 128),
Linear(128, 256), ReLU
Linear(256, 2048), ReLU
Unflatten(dim=1, unflattened_size=(32, 8, 8))

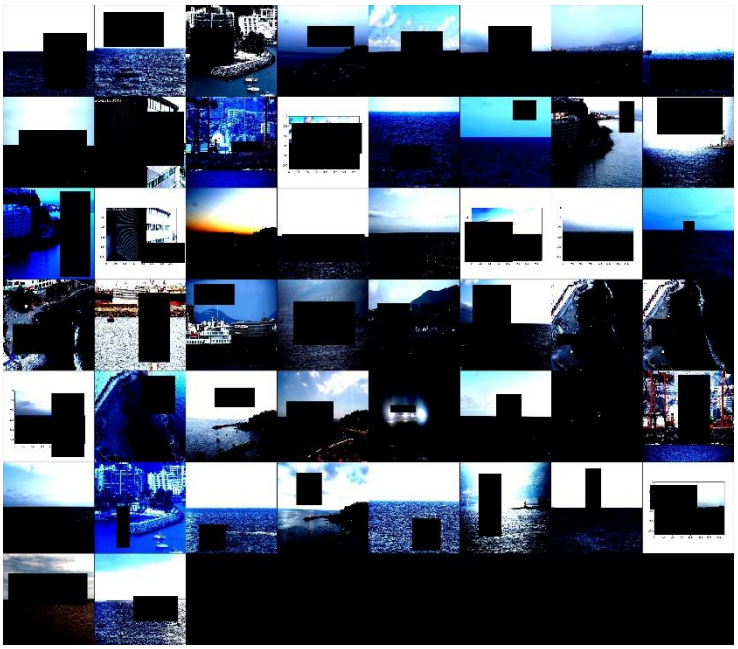
Decoder

Upsample(), ConvTranspose2d(32, 16, 8), BatchNorm2d(16), ReLU
Upsample(), ConvTranspose2d(16, 8, 3), BatchNorm2d(8), ReLU
Upsample(), ConvTranspose2d(8, 3, 19)

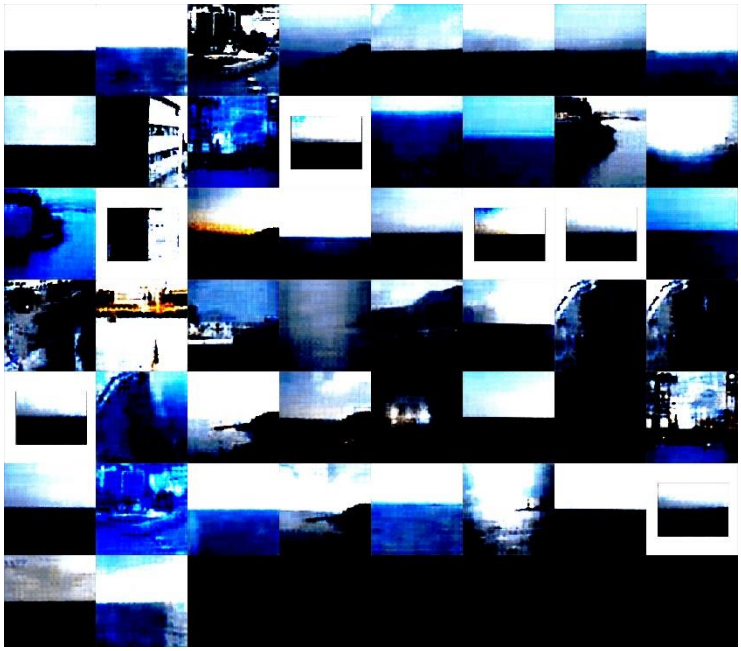
Loss: MSE
Optimizador: Adam(lr = 1e-3, weight_decay = 1e-8)

Después del entrenamiento obtuvimos los siguientes resultados:

GT

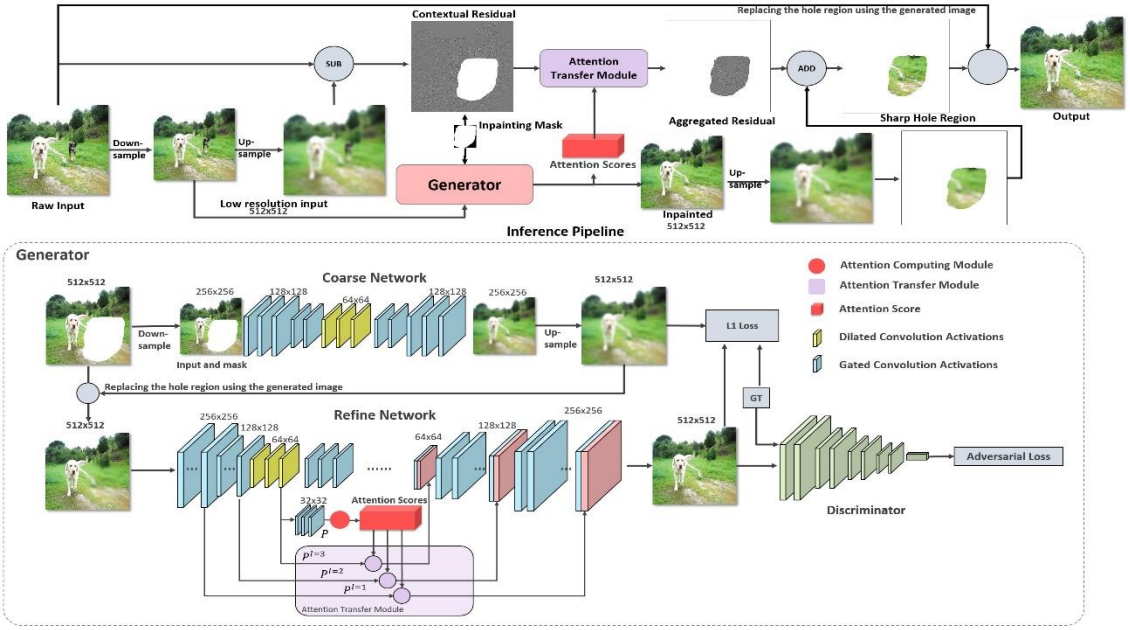


Inference



Como se puede observar el modelo logra reconstruir las imágenes, aunque se observa la perdida de resolución del caso anterior. El modelo por tiempo, se entrenó 400 épocas de las 10000 que se suelen utilizar en el estado del arte. Por otro lado, se tardó en entrenar en una GTX1070 3 días.

A la vista de estos resultados se revisó el estado del arte, en busca de soluciones que ya se hayan propuesto para este problema, en este caso se

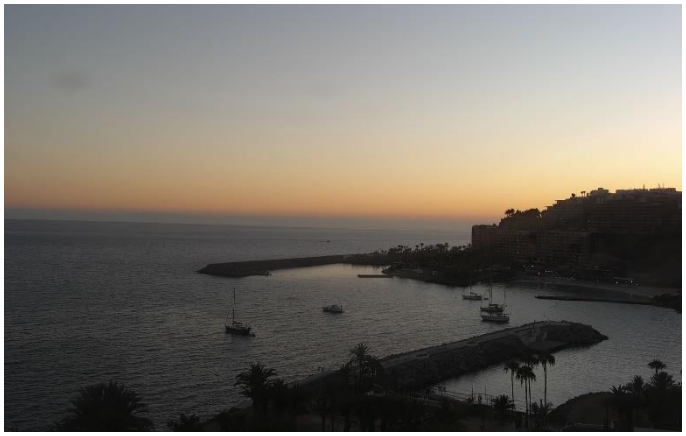


encontró que las soluciones son bastante más complejas que un simple Autoencoder, como podemos ver en el siguiente diagrama:

Lo que más destaca de estes soluciones es la función de error que utilizan, que se trata de una reconversión de la transformada de Fourier, investigando más a fondo, esta decisión se debe a las características de la transformada, que buscan minimizar el error que se produce en la métrica PSNR (Proporción Máxima de Señal a Ruido), mejorando de esta forma significativamente el resultado, Además como viene siendo costumbre en las soluciones del estado del arte, se hace uso de los Transformers en pos de utilizar los mecanismos de atención que tienes estos modelos, lo cual viene en consonancia con el modelo al tener un área especifica la cual se quiere reconstruir.

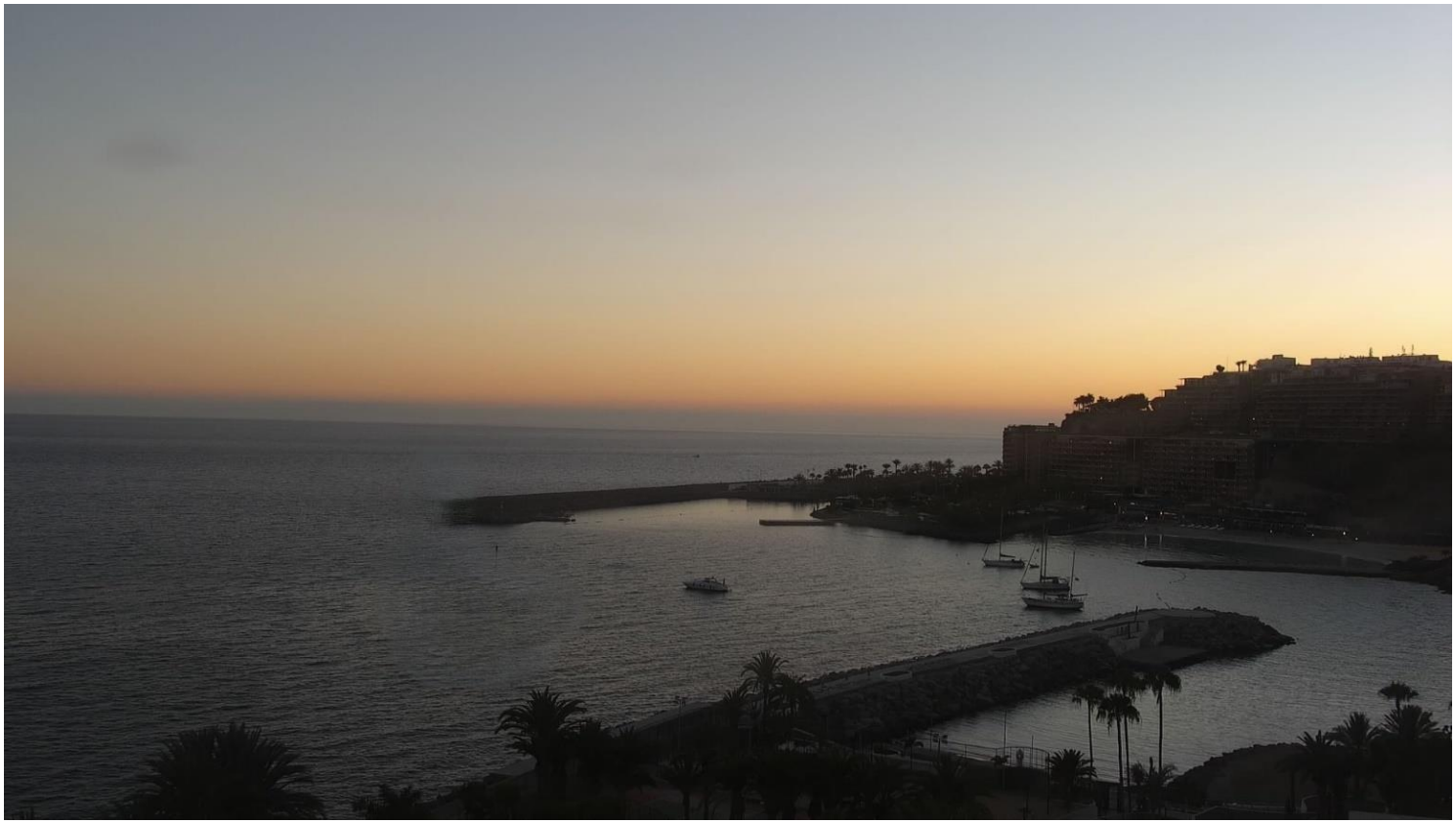
Después del estudio se probo el modelo obteniendo estos resultados:

GT



Mascara

Inferencia



Como podemos ver el barco en el mar a desaparecido casi por completo, además por la densidad en el negro de la máscara podemos decidir cómo se realiza la transparencia al reconstruir la imagen, dejando el halo que se observa en la inferencia.

El repositorio de código donde se encuentra todo el desarrollo es:

https://github.com/qwerteleven/image-impating