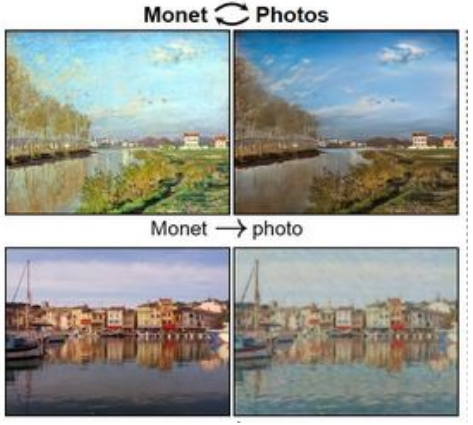
APRESENTAÇÃO 3
SCC0233



Alexandre Norcia Medeiros - 10295583
Caio Abreu de Oliveira Ribeiro - 10262839
Vinícius Torres Dutra Maia da Costa - 10262781
Gabriel Alfonso Nascimento Salgueiro - 10284368
Daniel Penna Chaves Bertazzo - 10349561

.tfrec

Conseguir carregar o dataset de modo compatível com a biblioteca Tensorflow.

Registros de dados sequenciais.

Binary file.

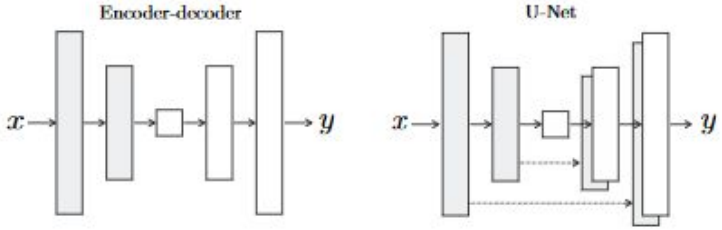
```
uint64 length
uint32 masked_crc32_of_length
byte   data[length]
uint32 masked_crc32_of_data
```

```
# Auxiliary functions to manage the tfrecords
IMAGE_SIZE = [256, 256]

# Adjusting the image dimensions and scale
def decode_image(image):
    image = tf.image.decode_jpeg(image, channels=3)
    image = (tf.cast(image, tf.float32) / 127.5) - 1
    image = tf.reshape(image, [*IMAGE_SIZE, 3])
    return image

# Parsing each element of the dataset
def read_tfrecord(example):
    tfrecord_format = {
        # "image_name": tf.io.FixedLenFeature([], tf.string),
        "image": tf.io.FixedLenFeature([], tf.string),
        # "target": tf.io.FixedLenFeature([], tf.string)
    }
    example = tf.io.parse_single_example(example, tfrecord_format)
    image = decode_image(example['image'])
    return image

# Reads the tfrecords and parses them using the read_tfrecord function
def load_dataset(filenames, labeled=True, ordered=False):
    dataset = tf.data.TFRecordDataset(filenames)
    dataset = dataset.map(read_tfrecord)#, num_parallel_calls=AUTOTUNE)
    return dataset
```

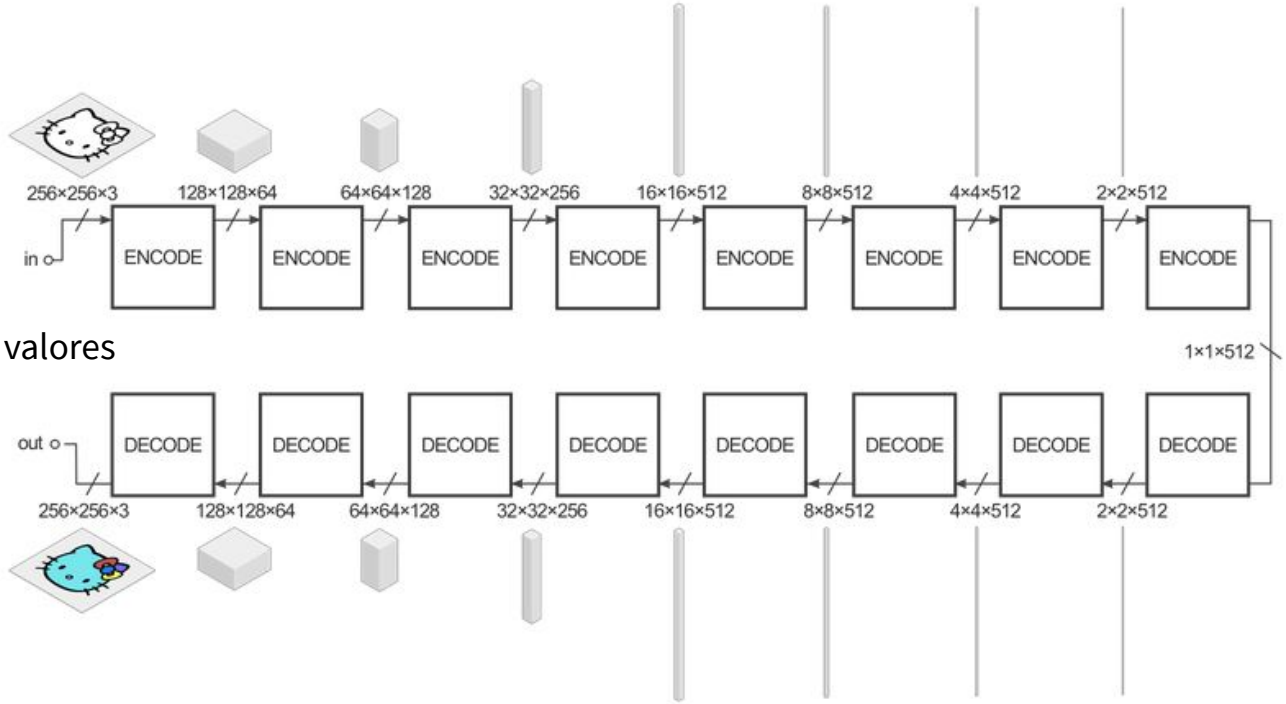
```
monet_ds = load_dataset(MONET_FILENAMES, labeled=True).batch(1)
photo_ds = load_dataset(PHOTO_FILENAMES, labeled=True).batch(1)
```

```
example_monet = next(iter(monet_ds))
example_photo = next(iter(photo_ds))
```

Arquitetura do Gerador



Encoder-Decorder / UNET

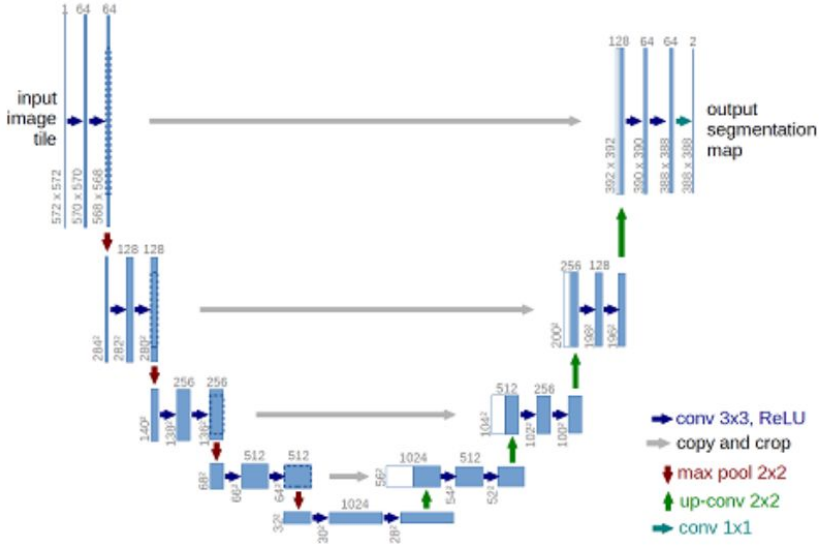


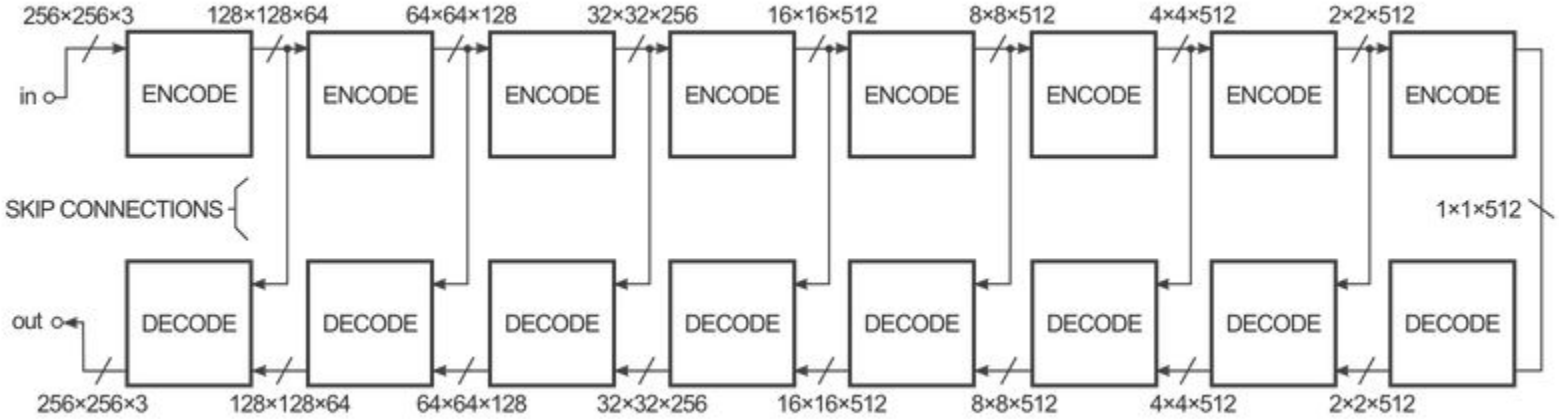
196,608 valores

512 valores

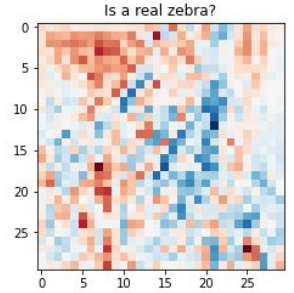
Arquitetura do gerador

UNET





Discriminator Loss Function



zebra real

cavalo fake

cavalo real

zebra fake

cavalo real

cavalo fake

Discriminador de cavalo

Discriminador de cavalo

Resultado Ideal

1 1 1 ... 1 1 1
1 1 1
1 1 1
..
1 1 1 ... 1 1 1

0 0 0 ... 0 0 0
0 0 0
0 0 0
...
0 0 0 ... 0 0 0

Discriminator Loss Function

DISCRIMINATOR LOSS

Loss_t = [loss(m_real, 1) + loss(m_fake, 0)] / 2

GENERATOR LOSS

Loss = loss(m_fake, 1)

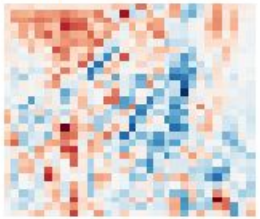
cavalo real

Discriminador de cavalo

Resultado Durante Treinamento

0.324 .08 ... 0.00
0.88 ...
... m_real
0.123 ... 0.02

Representação



Resultado Ideal

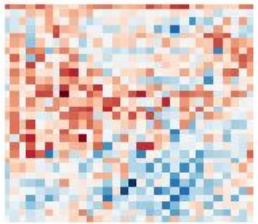
1 1 1 ... 1 1 1
1 1 1
1 1 1
..
1 1 1 ... 1 1 1

cavalo fake

Discriminador de cavalo

Resultado Durante Treinamento

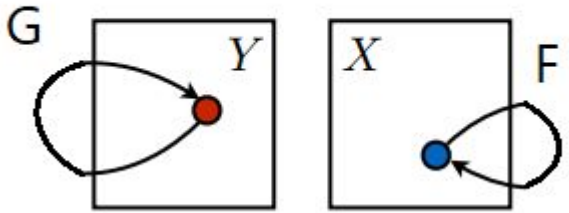
0.324 .08 ... 0.00
0.88 ...
... m_fake
0.123 ... 0.02



0 0 0 ... 0 0 0
0 0 0
0 0 0
...
0 0 0 ... 0 0 0

Generator Loss Functions

$$Identity\ loss = |G(Y) - Y| + |F(X) - X|$$



- Generator loss
- Cycle loss function
- Identity loss function