

Corso di Laurea triennale in Ingegneria e Scienze Informatiche

## Image Denoising Technique with Neural Network

*Relatore:*

**Prof. Lazzaro Damiana**

*Co/Contro Relatore*

**Dott. Ezio Greggio**

*Candidato:*

**Matteo Vanni**

*Matricola: 0000935584*



# Contents



# 1 Articolanzio

## 1.1 Panoramica sul problema

L'immagine denoising è il processo di rimozione di rumore da un'immagine.

Il rumore, che è causato da svariate fonti, quali foto fatte in condizioni di scarsa illuminazione o problemi che corrompono i file, causa perdita d'informazione sull'immagine.

**Cos'è il rumore?** Un'aggiunta casuale di pixel che non appartengono all'immagine originale e ce ne sono di varie tipologie:

Impulse Noise(IN) dove i pixel sono completamente diversi da quelli attorno. Esistono due categorie di IN: Salt and Pepper Noise(SPN) e Random Valued Impulse Noise(RVIN).

Additive White Gaussian Noise(AWGN) cambia ogni pixel dall'originale di una piccola quantità.

## 1.2 Utilizzo di modelli di Deep Learning

È essenziale rimuovere il rumore e ristabilire l'immagine originale dove riottenere l'immagine originale è importante per prestazioni robuste o ricostruire le informazioni mancanti è molto utile, come immagini astronomiche di oggetti molto lontani.

Le reti neurali convoluzionali lavorano bene con le immagini e ne utilizzeremo N, menzionate in alcuni paper di ricerca e compareremo i risultati di ogni modello.

## 1.3 Dataset utilizzati

Il primo dataset usato per addestrare i modelli sarà Oxford-IIIT Pet da Tensorflow, in modo poi da testare i modelli con immagini che non conoscono da altri dataset(colonscopie)

```
1 def load_dataset(split='train', img_size=(256,256), batch_size=16):
2     #Carica il dataset Oxford-IIIT Pet da tfds e applica il preprocessing
3
4     #Ritorna un dataset in batch.
5
6     # as_supervised=False per mantenere dict
7     dataset = tfds.load('oxford_iiit_pet', split=split, as_supervised=False)
8     # Preprocessing
9     dataset = dataset.map(lambda sample: preprocess(sample, img_size))
10    # Ottimizza il caricamento
11    dataset = dataset.shuffle(1024).batch(batch_size).prefetch(tf.data.AUTOTUNE)
```

12

```

return dataset

import numpy as np

def incmatrix(genl1,genl2):
    m = len(genl1)
    n = len(genl2)
    M = None #to become the incidence matrix
    VT = np.zeros((n*m,1), int) #dummy variable

    #compute the bitwise xor matrix
    M1 = bitxormatrix(genl1)
    M2 = np.triu(bitxormatrix(genl2),1)

    for i in range(m-1):
        for j in range(i+1, m):
            [r,c] = np.where(M2 == M1[i,j])
            for k in range(len(r)):
                VT[(i)*n + r[k]] = 1;
                VT[(i)*n + c[k]] = 1;
                VT[(j)*n + r[k]] = 1;
                VT[(j)*n + c[k]] = 1;

            if M is None:
                M = np.copy(VT)
            else:
                M = np.concatenate((M, VT), 1)

    VT = np.zeros((n*m,1), int)

    return M

```

Kvasir-seg: 1000(MILLEH) immagini di polipi,  
 gli animali di arvard, le colonscopie

## 2 Decsrizione delle reti

## 2.1 Rumore

E FAI RUMORE **QU** Prima funzione di rumore -j Random noise con fattore 0.3 massimo Rumore gaussiano Rumore a più layer

## 2.2 Autoencoder

Il primo approccio è stato quello di usare l'autencoder dell'articolo da cui sto copiando para para tutto.

Spiegazione del numero di layer usati e del tipo di mse loss fun, dam e learning rate di 1e-3

## 2.3 RIDNet

stessa cosa dell'autencoder (vedere se aggiungere tutto il codice gigante del setup)





## 3 Analisi ed ottimizzazione

### 3.1 Prestazioni

PSNR è il metodo più comune per misurare la qualità delle immagini.

Il PSNR è definito come il rapporto tra il massimo valore possibile di un segnale e il valore del rumore che disturba la qualità della sua rappresentazione.

Normalmente misurato in una scala logaritmica in decibel(dB).

Data l'immagine originale( $g$ ) e l'immagine rumorosa( $f$ ), il PSNR è definito come:

$$PSNR = 20 \log_{10} \left( \frac{MAX_f}{\sqrt{MSE}} \right)$$

dove  $MAX_f$  è il valore massimo del pixel dell'immagine e si calcola come

$$MSE = \frac{1}{mn} \sum_0^{m-1} \sum_0^{n-1} \|f(i, j) - g(i, j)\|^2$$

mentre  $MSE$  (*Mean Square Error*) è l'errore quadratico medio tra l'immagine originale e quella rumorosa.

### 3.2 Prima run

#### 3.2.1 Autoencoder

#### 3.2.2 RIDNet

### 3.3 Quantizzazione dei modelli

### 3.4 Analisi dei modelli

#### 3.4.1 Performance su dataset sconosciuti



## 4 Bibliografia

- jnome<sub>ℓ</sub>