Generic_Endoscopy_Denoising_Framework_DCNN

March 24, 2023

Endoscopy Denoising using DCNN Framework

[51]:

```
#Imports
from __future__ import print_function
import os
import numpy as np
import matplotlib.pyplot as plt
import cv2
import tensorflow as tf
from tensorflow.keras.layers import␣
 ↪Input,Conv2D,Activation,BatchNormalization,Add,Multiply,Concatenate
from tensorflow.keras.layers import GlobalAveragePooling2D
from tensorflow.keras.utils import plot_model
from tensorflow.keras.models import Model
from tensorflow.keras import metrics
import datetime
import pandas as pd
import time
import warnings
warnings.filterwarnings('ignore')
```

0.0.1 Data Source

[52]:

```
H,W,CH=[120,120,3]
BATCH_SIZE=16
NOISE_LEVELS=[15,25,50]

train_files=['Endoscopy/train/images/'+filename for filename in os.
 ↪listdir('Endoscopy/train/images/')]
test_files=['Endoscopy/test/images/'+filename for filename in os.
 ↪listdir('Endoscopy/test/images/')]
```

[53]:

```
def _parse_function(filename):
    image_string = tf.io.read_file(filename)
    image_decoded = tf.image.decode_jpeg(image_string, channels=3)
    image_decoded=tf.image.resize(image_decoded, [H,W], method='nearest')
    image = tf.cast(image_decoded, tf.float32)/255.
    noise_level=np.random.choice(NOISE_LEVELS)
    noisy_image=image+tf.random.normal(shape=(H,W,CH),mean=0,stddev=noise_level/
↪255)
    noisy_image=tf.image.resize(noisy_image, [H,W], method='nearest')
    noisy_image=tf.clip_by_value(noisy_image, clip_value_min=0.,␣
↪clip_value_max=1.)

    return noisy_image,image
```

0.0.2 Creating train dataset

[54]:

```
#Creating the Dataset
train_dataset = tf.data.Dataset.from_tensor_slices(np.array(train_files))
train_dataset = train_dataset.map(_parse_function)
train_dataset = train_dataset.batch(BATCH_SIZE)

test_dataset = tf.data.Dataset.from_tensor_slices(np.array(test_files))
test_dataset = test_dataset.map(_parse_function)
test_dataset = test_dataset.batch(BATCH_SIZE)
```

[55]:

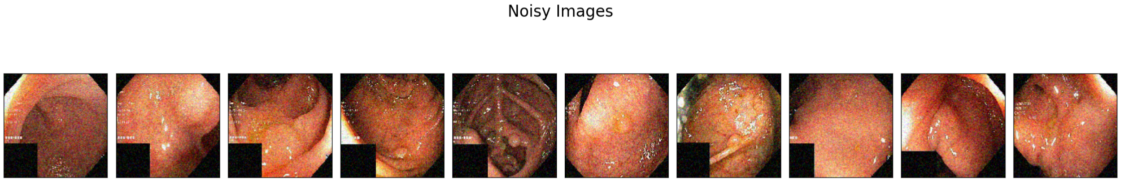
```
#data dim
iterator = iter(train_dataset)
a, b = iterator.get_next()

print('Shape of single batch of x : ',a.shape)
print('Shape of single batch of y : ',b.shape)
```

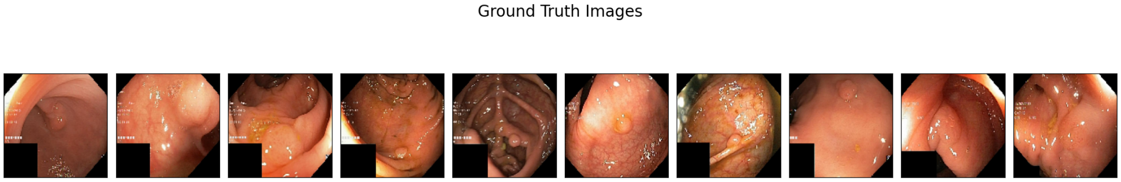
```
Shape of single batch of x :  (16, 120, 120, 3)
Shape of single batch of y :  (16, 120, 120, 3)
```

0.0.3 Exploratory data display

[56]:

```
# Display samples

fig, axs = plt.subplots(1,10,figsize=(20,4))
for i in range(10):
    axs[i].imshow(a[i])
    axs[i].get_xaxis().set_visible(False)
    axs[i].get_yaxis().set_visible(False)
    fig.suptitle('Noisy Images',fontsize=20)
fig.align_labels()
plt.tight_layout()
plt.show()

fig, axs = plt.subplots(1,10,figsize=(20,4))
for i in range(10):
    axs[i].imshow(b[i])
    axs[i].get_xaxis().set_visible(False)
    axs[i].get_yaxis().set_visible(False)
    fig.suptitle('Ground Truth Images',fontsize=20)
fig.align_labels()
plt.tight_layout()
plt.show()
```





0.0.4 Patches creation

[57]:

```
# Image patches creation

def get_patches(file_name,patch_size,crop_sizes):
    image = cv2.imread(file_name)
    image=cv2.cvtColor(image,cv2.COLOR_BGR2RGB)
    height, width , channels= image.shape
    patches = []
    for crop_size in crop_sizes: #We will crop the image to different sizes
        crop_h, crop_w = int(height*crop_size),int(width*crop_size)
        image_scaled = cv2.resize(image, (crop_w,crop_h), interpolation=cv2.
 ↪INTER_CUBIC)
        for i in range(0, crop_h-patch_size+1, patch_size):
            for j in range(0, crop_w-patch_size+1, patch_size):
                x = image_scaled[i:i+patch_size, j:j+patch_size]
                patches.append(x)
    return patches

def create_image_from_patches(patches,image_shape):
    image=np.zeros(image_shape)
    patch_size=patches.shape[1]
    p=0
    for i in range(0,image.shape[0]-patch_size+1,patch_size):
        for j in range(0,image.shape[1]-patch_size+1,patch_size):
            image[i:i+patch_size,j:j+patch_size]=patches[p]
            p+=1
    return np.array(image)

def predict_fun(model,image_path,noise_level=30):
    patches=get_patches(image_path,H,[1])
    test_image=cv2.imread(image_path)
    patches=np.array(patches)
    ground_truth=create_image_from_patches(patches,test_image.shape)

    patches = patches.astype('float32') /255.
    patches_noisy = patches+ tf.random.normal(shape=patches.
 ↪shape,mean=0,stddev=noise_level/255)
    patches_noisy = tf.clip_by_value(patches_noisy, clip_value_min=0.,␣
 ↪clip_value_max=1.)
    noisy_image=create_image_from_patches(patches_noisy,test_image.shape)
    denoised_patches=model.predict(patches_noisy)

    denoised_patches=tf.clip_by_value(denoised_patches, clip_value_min=0.,␣
 ↪clip_value_max=1.)

    denoised_image=create_image_from_patches(denoised_patches,test_image.shape)

    return patches_noisy,denoised_patches,ground_truth/255.
 ↪,noisy_image,denoised_image
```

0.0.5 Plot image patches

[58]:

```
#plot patches
def plot_patches(patches_noisy,denoised_patches):
    fig, axs = plt.subplots(2,10,figsize=(20,4))

    for i in range(10):
        axs[0,i].imshow(patches_noisy[i])
        axs[0,i].title.set_text(' Noisy')
        axs[0,i].get_xaxis().set_visible(False)
        axs[0,i].get_yaxis().set_visible(False)

        axs[1,i].imshow(denoised_patches[i])
        axs[1,i].title.set_text('Denoised')
        axs[1,i].get_xaxis().set_visible(False)
        axs[1,i].get_yaxis().set_visible(False)
    plt.show()
```

```
def plot_predictions(ground_truth,noisy_image,denoised_image):
    fig, axs = plt.subplots(1,3,figsize=(15,15))
    axs[0].imshow(ground_truth)
    axs[0].title.set_text('Ground Truth')
    axs[1].imshow(noisy_image)
    axs[1].title.set_text('Noisy Image')
    axs[2].imshow(denoised_image)
    axs[2].title.set_text('Denoised Image')
    plt.show()
```

0.0.6 PSNR measure

[59]:

```
def PSNR(gt, image, max_value=1):
    mse = np.mean((gt - image) ** 2)
    if mse == 0:
        return 100
    return 20 * np.log10(max_value / (np.sqrt(mse)))
```

0.0.7 RPDNET model architecture

[60]:

```
#Model
def CNN(input):
    x=Conv2D(64, (3,3), dilation_rate=1,padding='same',activation='relu')(input)
    x=Conv2D(64, (3,3), dilation_rate=2,padding='same',activation='relu')(x)

    y=Conv2D(64, (3,3), dilation_rate=4,padding='same',activation='relu')(input)
    y=Conv2D(64, (3,3), dilation_rate=5,padding='same',activation='relu')(y)

    z=Concatenate(axis=-1)([x,y])
    z=Conv2D(64, (3,3),padding='same',activation='relu')(z)
    add_1=Add()([z, input])

    z=Conv2D(64, (3,3),padding='same',activation='relu')(add_1)
    z=Conv2D(64, (3,3),padding='same')(z)
    add_2=Add()([z,add_1])
    add_2 = Activation('relu')(add_2)

    z=Conv2D(64, (3,3),padding='same',activation='relu')(add_2)
    z=Conv2D(64, (3,3),padding='same',activation='relu')(z)
    z=Conv2D(64, (1,1),padding='same')(z)
    add_3=Add()([z,add_2])
    add_3 = Activation('relu')(add_3)

    z = GlobalAveragePooling2D()(add_3)
    z = tf.expand_dims(z,1)
    z = tf.expand_dims(z,1)
    z=Conv2D(4, (3,3),padding='same',activation='relu')(z)
    z=Conv2D(64, (3,3),padding='same',activation='sigmoid')(z)
    mul=Multiply()([z, add_3])

    return mul
```

```
[ ]: def RPDNET():
         input = Input((H, W, CH),name='input')
         feat_extraction =Conv2D(64, (3,3),padding='same')(input)
         cnn_1=CNN(feat_extraction)
         cnn_2=CNN(cnn_1)
         cnn_3=CNN(cnn_2)
         cnn_4=CNN(cnn_3)
         x=Conv2D(3, (3,3),padding='same')(cnn_4)
         add_2=Add()([x, input])

         model=Model(input,add_2)

         return model
```

0.0.8 Model summary

```
[62]: #Model train
      tf.keras.backend.clear_session()
      tf.random.set_seed(6908)
      rpdnet = RPDNET()
      rpdnet.summary()
```

```
Model: "model"
________________________________________________________________________________
__________________
 Layer (type)                   Output Shape         Param #     Connected to
================================================================================
==================
 input (InputLayer)             [(None, 120, 120, 3  0           []
                                )]

 conv2d (Conv2D)                (None, 120, 120, 64  1792        ['input[0][0]']
                                )

 conv2d_1 (Conv2D)              (None, 120, 120, 64  36928
['conv2d[0][0]']
                                )

 conv2d_3 (Conv2D)              (None, 120, 120, 64  36928
['conv2d[0][0]']
                                )
 conv2d_2 (Conv2D)              (None, 120, 120, 64  36928
['conv2d_1[0][0]']
                                )

 conv2d_4 (Conv2D)              (None, 120, 120, 64  36928
['conv2d_3[0][0]']
                                )

 concatenate (Concatenate)      (None, 120, 120, 12  0
['conv2d_2[0][0]',
                                8)
'conv2d_4[0][0]']

 conv2d_5 (Conv2D)              (None, 120, 120, 64  73792
['concatenate[0][0]']
                                )

 add (Add)                      (None, 120, 120, 64  0
['conv2d_5[0][0]',
                                )
'conv2d[0][0]']

 conv2d_6 (Conv2D)              (None, 120, 120, 64  36928       ['add[0][0]']
                                )

 conv2d_7 (Conv2D)              (None, 120, 120, 64  36928
['conv2d_6[0][0]']
                                )

 add_1 (Add)                    (None, 120, 120, 64  0
['conv2d_7[0][0]',
                                )                                 'add[0][0]']

 activation (Activation)        (None, 120, 120, 64  0           ['add_1[0][0]']
                                )

 conv2d_8 (Conv2D)              (None, 120, 120, 64  36928
['activation[0][0]']
                                )

 conv2d_9 (Conv2D)              (None, 120, 120, 64  36928
['conv2d_8[0][0]']
                                )

 conv2d_10 (Conv2D)             (None, 120, 120, 64  4160
['conv2d_9[0][0]']
                                )
 add_2 (Add)                    (None, 120, 120, 64  0
['conv2d_10[0][0]',
                                )
'activation[0][0]']

 activation_1 (Activation)      (None, 120, 120, 64  0           ['add_2[0][0]']
                                )

 global_average_pooling2d (Glob  (None, 64)          0
['activation_1[0][0]']
 alAveragePooling2D)

 tf.expand_dims (TFOpLambda)    (None, 1, 64)        0
['global_average_pooling2d[0][0]'
                                                                 ]

 tf.expand_dims_1 (TFOpLambda)  (None, 1, 1, 64)     0
['tf.expand_dims[0][0]']

 conv2d_11 (Conv2D)             (None, 1, 1, 4)      2308
['tf.expand_dims_1[0][0]']

 conv2d_12 (Conv2D)             (None, 1, 1, 64)     2368
['conv2d_11[0][0]']

 multiply (Multiply)            (None, 120, 120, 64  0
['conv2d_12[0][0]',
                                )
'activation_1[0][0]']

 conv2d_13 (Conv2D)             (None, 120, 120, 64  36928
['multiply[0][0]']
                                )

 conv2d_15 (Conv2D)             (None, 120, 120, 64  36928
['multiply[0][0]']
                                )

 conv2d_14 (Conv2D)             (None, 120, 120, 64  36928
['conv2d_13[0][0]']
                                )

 conv2d_16 (Conv2D)             (None, 120, 120, 64  36928
['conv2d_15[0][0]']
                                )

 concatenate_1 (Concatenate)    (None, 120, 120, 12  0
['conv2d_14[0][0]',
                                8)
'conv2d_16[0][0]']

 conv2d_17 (Conv2D)             (None, 120, 120, 64  73792
['concatenate_1[0][0]']
                                )

 add_3 (Add)                    (None, 120, 120, 64  0
['conv2d_17[0][0]',
                                )
'multiply[0][0]']

 conv2d_18 (Conv2D)             (None, 120, 120, 64  36928       ['add_3[0][0]']
                                )

 conv2d_19 (Conv2D)             (None, 120, 120, 64  36928
['conv2d_18[0][0]']
                                )

 add_4 (Add)                    (None, 120, 120, 64  0
['conv2d_19[0][0]',
                                )                                 'add_3[0][0]']

 activation_2 (Activation)      (None, 120, 120, 64  0           ['add_4[0][0]']
                                )

 conv2d_20 (Conv2D)             (None, 120, 120, 64  36928
['activation_2[0][0]']
                                )

 conv2d_21 (Conv2D)             (None, 120, 120, 64  36928
['conv2d_20[0][0]']
                                )

 conv2d_22 (Conv2D)             (None, 120, 120, 64  4160
['conv2d_21[0][0]']
                                )

 add_5 (Add)                    (None, 120, 120, 64  0
['conv2d_22[0][0]',
                                )
'activation_2[0][0]']

 activation_3 (Activation)      (None, 120, 120, 64  0           ['add_5[0][0]']
                                )

 global_average_pooling2d_1 (Gl  (None, 64)          0
['activation_3[0][0]']
```

```
 obalAveragePooling2D)

 tf.expand_dims_2 (TFOpLambda)  (None, 1, 64)        0
['global_average_pooling2d_1[0][0
                                                                 ]']

 tf.expand_dims_3 (TFOpLambda)  (None, 1, 1, 64)     0
['tf.expand_dims_2[0][0]']

 conv2d_23 (Conv2D)             (None, 1, 1, 4)      2308
['tf.expand_dims_3[0][0]']

 conv2d_24 (Conv2D)             (None, 1, 1, 64)     2368
['conv2d_23[0][0]']

 multiply_1 (Multiply)          (None, 120, 120, 64  0
['conv2d_24[0][0]',
                                )
'activation_3[0][0]']

 conv2d_25 (Conv2D)             (None, 120, 120, 64  36928
['multiply_1[0][0]']
                                )

 conv2d_27 (Conv2D)             (None, 120, 120, 64  36928
['multiply_1[0][0]']
                                )

 conv2d_26 (Conv2D)             (None, 120, 120, 64  36928
['conv2d_25[0][0]']
                                )

 conv2d_28 (Conv2D)             (None, 120, 120, 64  36928
['conv2d_27[0][0]']
                                )

 concatenate_2 (Concatenate)    (None, 120, 120, 12  0
['conv2d_26[0][0]',
                                8)
'conv2d_28[0][0]']

 conv2d_29 (Conv2D)             (None, 120, 120, 64  73792
['concatenate_2[0][0]']
                                )

 add_6 (Add)                    (None, 120, 120, 64  0
['conv2d_29[0][0]',
                                )
```

```
'multiply_1[0][0]']

 conv2d_30 (Conv2D)             (None, 120, 120, 64  36928       ['add_6[0][0]']
                                )

 conv2d_31 (Conv2D)             (None, 120, 120, 64  36928
['conv2d_30[0][0]']
                                )

 add_7 (Add)                    (None, 120, 120, 64  0
['conv2d_31[0][0]',
                                )                                 'add_6[0][0]']

 activation_4 (Activation)      (None, 120, 120, 64  0           ['add_7[0][0]']
                                )

 conv2d_32 (Conv2D)             (None, 120, 120, 64  36928
['activation_4[0][0]']
                                )

 conv2d_33 (Conv2D)             (None, 120, 120, 64  36928
['conv2d_32[0][0]']
                                )

 conv2d_34 (Conv2D)             (None, 120, 120, 64  4160
['conv2d_33[0][0]']
                                )

 add_8 (Add)                    (None, 120, 120, 64  0
['conv2d_34[0][0]',
                                )
'activation_4[0][0]']

 activation_5 (Activation)      (None, 120, 120, 64  0           ['add_8[0][0]']
                                )

 global_average_pooling2d_2 (Gl  (None, 64)          0
['activation_5[0][0]']
 obalAveragePooling2D)

 tf.expand_dims_4 (TFOpLambda)  (None, 1, 64)        0
['global_average_pooling2d_2[0][0
                                                                 ]']

 tf.expand_dims_5 (TFOpLambda)  (None, 1, 1, 64)     0
['tf.expand_dims_4[0][0]']

 conv2d_35 (Conv2D)             (None, 1, 1, 4)      2308
```

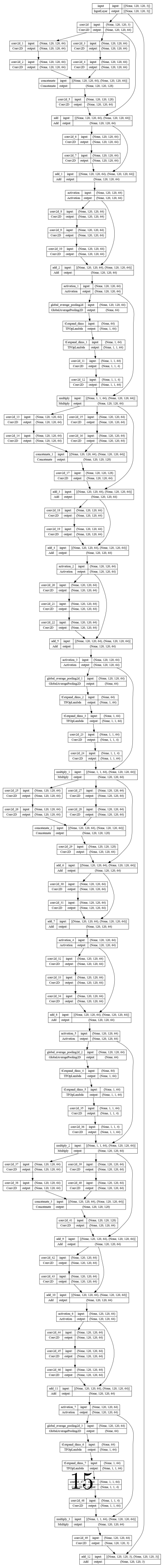
```
['tf.expand_dims_5[0][0]']

 conv2d_36 (Conv2D)             (None, 1, 1, 64)     2368
['conv2d_35[0][0]']

 multiply_2 (Multiply)          (None, 120, 120, 64  0
['conv2d_36[0][0]',
                                )
'activation_5[0][0]']

 conv2d_37 (Conv2D)             (None, 120, 120, 64  36928
['multiply_2[0][0]']
                                )

 conv2d_39 (Conv2D)             (None, 120, 120, 64  36928
['multiply_2[0][0]']
                                )

 conv2d_38 (Conv2D)             (None, 120, 120, 64  36928
['conv2d_37[0][0]']
                                )

 conv2d_40 (Conv2D)             (None, 120, 120, 64  36928
['conv2d_39[0][0]']
                                )

 concatenate_3 (Concatenate)    (None, 120, 120, 12  0
['conv2d_38[0][0]',
                                8)
'conv2d_40[0][0]']

 conv2d_41 (Conv2D)             (None, 120, 120, 64  73792
['concatenate_3[0][0]']
                                )

 add_9 (Add)                    (None, 120, 120, 64  0
['conv2d_41[0][0]',
                                )
'multiply_2[0][0]']

 conv2d_42 (Conv2D)             (None, 120, 120, 64  36928       ['add_9[0][0]']
                                )

 conv2d_43 (Conv2D)             (None, 120, 120, 64  36928
['conv2d_42[0][0]']
                                )

 add_10 (Add)                   (None, 120, 120, 64  0
['conv2d_43[0][0]',
                                )                                 'add_9[0][0]']

 activation_6 (Activation)      (None, 120, 120, 64  0
['add_10[0][0]']
                                )

 conv2d_44 (Conv2D)             (None, 120, 120, 64  36928
['activation_6[0][0]']
                                )

 conv2d_45 (Conv2D)             (None, 120, 120, 64  36928
['conv2d_44[0][0]']
                                )

 conv2d_46 (Conv2D)             (None, 120, 120, 64  4160
['conv2d_45[0][0]']
                                )

 add_11 (Add)                   (None, 120, 120, 64  0
['conv2d_46[0][0]',
                                )
'activation_6[0][0]']

 activation_7 (Activation)      (None, 120, 120, 64  0
['add_11[0][0]']
                                )

 global_average_pooling2d_3 (Gl  (None, 64)          0
['activation_7[0][0]']
 obalAveragePooling2D)

 tf.expand_dims_6 (TFOpLambda)  (None, 1, 64)        0
['global_average_pooling2d_3[0][0
                                                                 ]']

 tf.expand_dims_7 (TFOpLambda)  (None, 1, 1, 64)     0
['tf.expand_dims_6[0][0]']

 conv2d_47 (Conv2D)             (None, 1, 1, 4)      2308
['tf.expand_dims_7[0][0]']

 conv2d_48 (Conv2D)             (None, 1, 1, 64)     2368
['conv2d_47[0][0]']

 multiply_3 (Multiply)          (None, 120, 120, 64  0
['conv2d_48[0][0]',
                                )
```

```
'activation_7[0][0]']

 conv2d_49 (Conv2D)             (None, 120, 120, 3)  1731
['multiply_3[0][0]']

 add_12 (Add)                   (None, 120, 120, 3)  0
['conv2d_49[0][0]',
                                                                  'input[0][0]']

================================================================================
==================
Total params: 1,515,731
Trainable params: 1,515,731
Non-trainable params: 0
--------------------------------------------------------------------------------
------------------
```

[63]:

```
dot_img_file = 'rpdnet.png'
tf.keras.utils.plot_model(rpdnet, to_file=dot_img_file, show_shapes=True)
```

[63]:


input | input: | [(None, 120, 120, 3)]
InputLayer | output: | [(None, 120, 120, 3)]
conv2d | input: | (None, 120, 120, 3)
Conv2D | output: | (None, 120, 120, 64)
conv2d_1 | input: | (None, 120, 120, 64)
Conv2D | output: | (None, 120, 120, 64)
conv2d_3 | input: | (None, 120, 120, 64)
Conv2D | output: | (None, 120, 120, 64)
conv2d_2 | input: | (None, 120, 120, 64)
Conv2D | output: | (None, 120, 120, 64)
conv2d_4 | input: | (None, 120, 120, 64)
Conv2D | output: | (None, 120, 120, 64)
concatenate | input: | [(None, 120, 120, 64), (None, 120, 120, 64)]
Concatenate | output: | (None, 120, 120, 128)
conv2d_5 | input: | (None, 120, 120, 128)
Conv2D | output: | (None, 120, 120, 64)
add | input: | [(None, 120, 120, 64), (None, 120, 120, 64)]
Add | output: | (None, 120, 120, 64)
conv2d_6 | input: | (None, 120, 120, 64)
Conv2D | output: | (None, 120, 120, 64)
conv2d_7 | input: | (None, 120, 120, 64)
Conv2D | output: | (None, 120, 120, 64)
add_1 | input: | [(None, 120, 120, 64), (None, 120, 120, 64)]
Add | output: | (None, 120, 120, 64)
activation | input: | (None, 120, 120, 64)
Activation | output: | (None, 120, 120, 64)
conv2d_8 | input: | (None, 120, 120, 64)
Conv2D | output: | (None, 120, 120, 64)
conv2d_9 | input: | (None, 120, 120, 64)
Conv2D | output: | (None, 120, 120, 64)
conv2d_10 | input: | (None, 120, 120, 64)
Conv2D | output: | (None, 120, 120, 64)
add_2 | input: | [(None, 120, 120, 64), (None, 120, 120, 64)]
Add | output: | (None, 120, 120, 64)
activation_1 | input: | (None, 120, 120, 64)
Activation | output: | (None, 120, 120, 64)
global_average_pooling2d | input: | (None, 120, 120, 64)
GlobalAveragePooling2D | output: | (None, 64)
tf.expand_dims | input: | (None, 64)
TFOpLambda | output: | (None, 1, 64)
tf.expand_dims_1 | input: | (None, 1, 64)
TFOpLambda | output: | (None, 1, 1, 64)
conv2d_11 | input: | (None, 1, 1, 64)
Conv2D | output: | (None, 1, 1, 4)
conv2d_12 | input: | (None, 1, 1, 4)
Conv2D | output: | (None, 1, 1, 64)
multiply | input: | [(None, 1, 1, 64), (None, 120, 120, 64)]
Multiply | output: | (None, 120, 120, 64)
conv2d_13 | input: | (None, 120, 120, 64)
Conv2D | output: | (None, 120, 120, 64)
conv2d_15 | input: | (None, 120, 120, 64)
Conv2D | output: | (None, 120, 120, 64)
conv2d_14 | input: | (None, 120, 120, 64)
Conv2D | output: | (None, 120, 120, 64)
conv2d_16 | input: | (None, 120, 120, 64)
Conv2D | output: | (None, 120, 120, 64)
concatenate_1 | input: | [(None, 120, 120, 64), (None, 120, 120, 64)]
Concatenate | output: | (None, 120, 120, 128)
conv2d_17 | input: | (None, 120, 120, 128)
Conv2D | output: | (None, 120, 120, 64)
add_3 | input: | [(None, 120, 120, 64), (None, 120, 120, 64)]
Add | output: | (None, 120, 120, 64)
conv2d_18 | input: | (None, 120, 120, 64)
Conv2D | output: | (None, 120, 120, 64)
conv2d_19 | input: | (None, 120, 120, 64)
Conv2D | output: | (None, 120, 120, 64)
add_4 | input: | [(None, 120, 120, 64), (None, 120, 120, 64)]
Add | output: | (None, 120, 120, 64)
activation_2 | input: | (None, 120, 120, 64)
Activation | output: | (None, 120, 120, 64)
conv2d_20 | input: | (None, 120, 120, 64)
Conv2D | output: | (None, 120, 120, 64)
conv2d_21 | input: | (None, 120, 120, 64)
Conv2D | output: | (None, 120, 120, 64)
conv2d_22 | input: | (None, 120, 120, 64)
Conv2D | output: | (None, 120, 120, 64)
add_5 | input: | [(None, 120, 120, 64), (None, 120, 120, 64)]
Add | output: | (None, 120, 120, 64)
activation_3 | input: | (None, 120, 120, 64)
Activation | output: | (None, 120, 120, 64)
global_average_pooling2d_1 | input: | (None, 120, 120, 64)
GlobalAveragePooling2D | output: | (None, 64)
tf.expand_dims_2 | input: | (None, 64)
TFOpLambda | output: | (None, 1, 64)
tf.expand_dims_3 | input: | (None, 1, 64)
TFOpLambda | output: | (None, 1, 1, 64)
conv2d_23 | input: | (None, 1, 1, 64)
Conv2D | output: | (None, 1, 1, 4)
conv2d_24 | input: | (None, 1, 1, 4)
Conv2D | output: | (None, 1, 1, 64)
multiply_1 | input: | [(None, 1, 1, 64), (None, 120, 120, 64)]
Multiply | output: | (None, 120, 120, 64)
conv2d_25 | input: | (None, 120, 120, 64)
Conv2D | output: | (None, 120, 120, 64)
conv2d_27 | input: | (None, 120, 120, 64)
Conv2D | output: | (None, 120, 120, 64)
conv2d_26 | input: | (None, 120, 120, 64)
Conv2D | output: | (None, 120, 120, 64)
conv2d_28 | input: | (None, 120, 120, 64)
Conv2D | output: | (None, 120, 120, 64)
concatenate_2 | input: | [(None, 120, 120, 64), (None, 120, 120, 64)]
Concatenate | output: | (None, 120, 120, 128)
conv2d_29 | input: | (None, 120, 120, 128)
Conv2D | output: | (None, 120, 120, 64)
add_6 | input: | [(None, 120, 120, 64), (None, 120, 120, 64)]
Add | output: | (None, 120, 120, 64)
conv2d_30 | input: | (None, 120, 120, 64)
Conv2D | output: | (None, 120, 120, 64)
conv2d_31 | input: | (None, 120, 120, 64)
Conv2D | output: | (None, 120, 120, 64)
add_7 | input: | [(None, 120, 120, 64), (None, 120, 120, 64)]
Add | output: | (None, 120, 120, 64)
activation_4 | input: | (None, 120, 120, 64)
Activation | output: | (None, 120, 120, 64)
conv2d_32 | input: | (None, 120, 120, 64)
Conv2D | output: | (None, 120, 120, 64)
conv2d_33 | input: | (None, 120, 120, 64)
Conv2D | output: | (None, 120, 120, 64)
conv2d_34 | input: | (None, 120, 120, 64)
Conv2D | output: | (None, 120, 120, 64)
add_8 | input: | [(None, 120, 120, 64), (None, 120, 120, 64)]
Add | output: | (None, 120, 120, 64)
activation_5 | input: | (None, 120, 120, 64)
Activation | output: | (None, 120, 120, 64)
global_average_pooling2d_2 | input: | (None, 120, 120, 64)
GlobalAveragePooling2D | output: | (None, 64)
tf.expand_dims_4 | input: | (None, 64)
TFOpLambda | output: | (None, 1, 64)
tf.expand_dims_5 | input: | (None, 1, 64)
TFOpLambda | output: | (None, 1, 1, 64)
conv2d_35 | input: | (None, 1, 1, 64)
Conv2D | output: | (None, 1, 1, 4)
conv2d_36 | input: | (None, 1, 1, 4)
Conv2D | output: | (None, 1, 1, 64)
multiply_2 | input: | [(None, 1, 1, 64), (None, 120, 120, 64)]
Multiply | output: | (None, 120, 120, 64)
conv2d_37 | input: | (None, 120, 120, 64)
Conv2D | output: | (None, 120, 120, 64)
conv2d_39 | input: | (None, 120, 120, 64)
Conv2D | output: | (None, 120, 120, 64)
conv2d_38 | input: | (None, 120, 120, 64)
Conv2D | output: | (None, 120, 120, 64)
conv2d_40 | input: | (None, 120, 120, 64)
Conv2D | output: | (None, 120, 120, 64)
concatenate_3 | input: | [(None, 120, 120, 64), (None, 120, 120, 64)]
Concatenate | output: | (None, 120, 120, 128)
conv2d_41 | input: | (None, 120, 120, 128)
Conv2D | output: | (None, 120, 120, 64)
add_9 | input: | [(None, 120, 120, 64), (None, 120, 120, 64)]
Add | output: | (None, 120, 120, 64)
conv2d_42 | input: | (None, 120, 120, 64)
Conv2D | output: | (None, 120, 120, 64)
conv2d_43 | input: | (None, 120, 120, 64)
Conv2D | output: | (None, 120, 120, 64)
add_10 | input: | [(None, 120, 120, 64), (None, 120, 120, 64)]
Add | output: | (None, 120, 120, 64)
activation_6 | input: | (None, 120, 120, 64)
Activation | output: | (None, 120, 120, 64)
conv2d_44 | input: | (None, 120, 120, 64)
Conv2D | output: | (None, 120, 120, 64)
conv2d_45 | input: | (None, 120, 120, 64)
Conv2D | output: | (None, 120, 120, 64)
conv2d_46 | input: | (None, 120, 120, 64)
Conv2D | output: | (None, 120, 120, 64)
add_11 | input: | [(None, 120, 120, 64), (None, 120, 120, 64)]
Add | output: | (None, 120, 120, 64)
activation_7 | input: | (None, 120, 120, 64)
Activation | output: | (None, 120, 120, 64)
global_average_pooling2d_3 | input: | (None, 120, 120, 64)
GlobalAveragePooling2D | output: | (None, 64)
tf.expand_dims_6 | input: | (None, 64)
TFOpLambda | output: | (None, 1, 64)
tf.expand_dims_7 | input: | (None, 1, 64)
TFOpLambda | output: | (None, 1, 1, 64)
conv2d_47 | input: | (None, 1, 1, 64)
Conv2D | output: | (None, 1, 1, 4)
conv2d_48 | input: | (None, 1, 1, 4)
Conv2D | output: | (None, 1, 1, 64)
multiply_3 | input: | [(None, 1, 1, 64), (None, 120, 120, 64)]
Multiply | output: | (None, 120, 120, 64)
conv2d_49 | input: | (None, 120, 120, 64)
Conv2D | output: | (None, 120, 120, 3)
add_12 | input: | [(None, 120, 120, 3), (None, 120, 120, 3)]
Add | output: | (None, 120, 120, 3)

[64]:

```
#Compile model
rpdnet.compile(optimizer=tf.keras.optimizers.Adam(1e-03), loss=tf.keras.losses.
 ↪MeanAbsoluteError(),
               metrics=['mean_absolute_error'])
```

0.0.9 Set callback

[65]:

```
#Callback
def scheduler(epoch,lr):
    return lr*0.9

checkpoint_path = "rpdnet.h5"
checkpoint_dir = os.path.dirname(checkpoint_path)
cp_callback = tf.keras.callbacks.ModelCheckpoint(filepath=checkpoint_path,
                                                                ␣
 ↪save_weights_only=False,verbose=0,save_best_only=False)

lrScheduler = tf.keras.callbacks.LearningRateScheduler(scheduler)
callbacks = [cp_callback,lrScheduler]
```

0.0.10 Train model

[ ]:

```
#Train model
nepochs=25
history=rpdnet.fit( train_dataset,shuffle=True,epochs=nepochs,
                   validation_data= test_dataset,callbacks=callbacks,
                   max_queue_size=10,workers=1,verbose=1,
                   use_multiprocessing=True
                  )
```

```
Epoch 1/25
13/13 [==============================] - 519s 40s/step - loss: 0.0858 -
mean_absolute_error: 0.0858 - val_loss: 0.1317 - val_mean_absolute_error: 0.1317
- lr: 9.0000e-04
Epoch 2/25
13/13 [==============================] - 531s 41s/step - loss: 0.0692 -
mean_absolute_error: 0.0692 - val_loss: 0.1294 - val_mean_absolute_error: 0.1294
- lr: 8.1000e-04
Epoch 3/25
13/13 [==============================] - 492s 38s/step - loss: 0.0673 -
mean_absolute_error: 0.0673 - val_loss: 0.1225 - val_mean_absolute_error: 0.1225
- lr: 7.2900e-04
Epoch 4/25
11/13 [=======================>…] - ETA: 1:14 - loss: 0.0629 -
```

```
mean_absolute_error: 0.0629
```

[ ]:

```
#history info.
print(history.history.keys())
```

0.0.11 Plot training performance

[ ]:

```
def plot_history(model_history,keys):
    m,val_m = keys
    plt.plot(model_history.history[m])
    plt.plot(model_history.history[val_m])
    plt.ylabel(m)
    plt.xlabel('epoch')
    plt.legend(['train', 'validation'], loc='upper left')
    plt.show()

plot_history(history,['loss','val_loss'])
```

0.0.12 Model evaluation

[ ]:

```
path='Endoscopy/data/e2.jpg'
patches_noisy,denoised_patches,ground_truth,noisy_image,denoised_image=predict_fun(rpdnet,path
print('PSNR of Noisy Image : ',PSNR(ground_truth,noisy_image))
print('PSNR of Denoised Image : ',PSNR(ground_truth,denoised_image))
plot_patches(patches_noisy,denoised_patches)

#result
plot_predictions(ground_truth,noisy_image,denoised_image)
```

[ ]:

```
path='Endoscopy/data/e3.jpg'
patches_noisy,denoised_patches,ground_truth,noisy_image,denoised_image=predict_fun(rpdnet,path
print('PSNR of Noisy Image : ',PSNR(ground_truth,noisy_image))
print('PSNR of Denoised Image : ',PSNR(ground_truth,denoised_image))
plot_patches(patches_noisy,denoised_patches)

#result
plot_predictions(ground_truth,noisy_image,denoised_image)
```

[ ]:

```
path='Endoscopy/data/e4.jpg'
patches_noisy,denoised_patches,ground_truth,noisy_image,denoised_image=predict_fun(rpdnet,path
print('PSNR of Noisy Image : ',PSNR(ground_truth,noisy_image))
print('PSNR of Denoised Image : ',PSNR(ground_truth,denoised_image))
plot_patches(patches_noisy,denoised_patches)

#result
plot_predictions(ground_truth,noisy_image,denoised_image)
```

0.0.13 Residual learning

[ ]:

```
# Residual Learning
model_res=Model(rpdnet.input,rpdnet.get_layer('conv2d_49').output)
path='Endoscopy/data/e4.jpg'
patches_noisy,denoised_patches,ground_truth,noisy_image,residual_image=predict_fun(model_res,p
```

0.0.14 Residual data visualisation

[ ]:

```
#Display predicted result
fig, axs = plt.subplots(1,3,figsize=(15,15))
axs[0].imshow(ground_truth)
axs[0].title.set_text('Ground Truth')
axs[1].imshow(noisy_image)
axs[1].title.set_text('Noisy Image')
axs[2].imshow(residual_image)
axs[2].title.set_text('Residual Image')

plt.show()
```

[ ]:

```
import keract
from tensorflow.keras.applications.mobilenet import decode_predictions,␣
 ↪preprocess_input
from tensorflow.keras.preprocessing.image import  load_img, img_to_array
```

0.0.15 Activation silency map

[ ]:

```
#Activation silency map
image = load_img(path, target_size= (H, W))
image = img_to_array(image)
image = image.reshape((1, image.shape[0], image.shape[1], image.shape[2]))
image = preprocess_input(image)
y_hat = rpdnet.predict(image)
```

[ ]:

```
#layers
layers=['conv2d_1','conv2d_3','conv2d_2','conv2d_4']

activations= keract.get_activations(rpdnet, image, layer_names= layers,
                                    nodes_to_evaluate= None, output_format=␣
 ↪'simple',  auto_compile= True)
keract.display_activations(activations, cmap='viridis', save= False, directory=␣
 ↪'activations')
```

[ ]: