In [1]:

```
#!sudo apt install tesseract-ocr
#!pip install pytesseract

import warnings
warnings.filterwarnings('ignore')

import tensorflow as tf
import os
import matplotlib.pyplot as plt
import numpy as np
import cv2
import xml.etree.ElementTree as ET
from PIL import Image
import pandas as pd
import pytesseract
from sklearn.model_selection import train_test_split
import tensorflow as tf
import pytesseract
import csv
#from google.colab.patches import cv2_imshow
```

In [2]:

```
originalImage = "C:/Users/sesha/Untitled Folder/Untitled Folder/marmot old1/10.1.1.1.2006_3.bmp"

imageMask = "C:/Users/sesha/Untitled Folder/Untitled Folder/green/image_mask/10.1.1.1.2006_3.xml"

fileSavepath = "C:/Users/sesha/Untitled Folder/Untitled Folder/green/final_data/"

table_mask_path = "C:/Users/sesha/Untitled Folder/Untitled Folder/green/final_data/tablemask/"

col_mask_path = "C:/Users/sesha/Untitled Folder/Untitled Folder/green/final_data/colmask/"

org_image_path = "C:/Users/sesha/Untitled Folder/Untitled Folder/green/final_data/orgimage/"

dataPath = "C:/Users/sesha/Untitled Folder/Untitled Folder/marmot old1/"
```

In [3]:

```
"""
CREATE DATAFRAME OF PATHS.
dataframe
---------
image_path, xml_path

* go through every file in mamoth folder (dataPath).
* check a .bmp file, extract name, check if .xml file is present or not --> store in row
"""

image_xml_dict = {"image_path":[], "xml_path":[]}

for file in os.listdir(dataPath):
    if ".bmp" in file:
        name = file.split(".bmp")[0]
        if os.path.exists(dataPath+name+".xml"):
            image_xml_dict['image_path'].append(name+".bmp")
            image_xml_dict['xml_path'].append(name+".xml")


image_xml_df = pd.DataFrame(image_xml_dict)
```

```
image_xml_df.head(2)
```

Out[3]:

	image_path	xml_path
0	10.1.1.1.2006_3.bmp	10.1.1.1.2006_3.xml
1	10.1.1.1.2013_63.bmp	10.1.1.1.2013_63.xml

"""

793

1123

3

</size>

column

Unspecified

0

0

458

710

517

785

</bndbox>

</object>

"""

/content/drive/MyDrive/case study - II/tablenet/data/final data/

def euc_dist(point1, point2): dist = np.linalg.norm(point1 - point2) return dist

def show_image_plt(image_arr): plt.figure(figsize=(5,5)) plt.imshow(image_arr) plt.show()

def save_image(name, image_arr): im = Image.fromarray(image_arr) im.save(name)

final_dataframe_dict = {"image":[], "table_mask":[], "col_mask":[]}

for index, row in image_xml_df.iterrows():

```
    # per row --> xml path
    org_img_mask_xml = row['xml_path'] # .xml p
    ath
    image = dataPath + row['image_path'] # imag
    e .bmp path
```

image = row['image_path'] # image .bmp path

```
    # file name
```

```
name = org_img_mask_xml.split(".xml")[0]


# reading xml file
tree = ET.parse(dataPath + org_img_mask_xml
)
root = tree.getroot()


size = root.find('size')
width = int(size.find('width').text)
height = int(size.find('height').text)
depth = int(size.find('depth').text)

# creating empty mask image
col_mask_empty = np.zeros(shape=(height, wi
dth), dtype=np.uint8)
table_mask_empty = np.zeros(shape=(height,
width), dtype=np.uint8)

# finding objects
objects = tree.findall('object')
table_xmin = 0
table_ymin = 0
table_xmax = 0
table_ymax = 0
prev_dist = 0
dist = 0
forward_flag = False
backward_flag = False
newtable_flag = True

# creating empty mask image
col_mask_empty = np.zeros(shape=(height, wi
dth), dtype=np.uint8)
table_mask_empty = np.zeros(shape=(height,
width), dtype=np.uint8)

plt.figure(figsize=(5, 5))

objects = tree.findall('object')


for index, object in enumerate(objects):

    bndbox = object.find('bndbox')
    xmin = int(bndbox.find('xmin').text)
    ymin = int(bndbox.find('ymin').text)
    xmax = int(bndbox.find('xmax').text)
    ymin = int(bndbox.find('ymin').text)
    ymax = int(bndbox.find('ymax').text)

    col mask_empty[ymin:ymax, xmin:xmax] =
255

    if index == 0:

        prev_xmin = int(bndbox.find('xmin')
.text)
        prev_ymin = int(bndbox.find('ymin')
.text)
        prev_xmax = int(bndbox.find('xmax')
.text)
        prev_ymax = int(bndbox.find('ymax')
.text)


    else:

        if xmin > prev_xmin and newtable_fl
ag:


            table xmin = prev xmin
            table ymin = prev ymin
            newtable flag = False
            forward flag = True
            backward_flag = False


        if xmin < prev_xmin and newtable_fl
ag:


            table xmax = prev xmax
            table_ymax = prev_ymax


            newtable flag = False
            backward flag = True
            forward_flag = False


        if forward flag:
            dist = euc dist(np.array([xmin,
ymin]), np.array([prev_xmax, prev_ymin]))

            if prev_dist == 0:
                prev_dist = dist
            else:

                if int(np.divide(dist, prev
 dist)) > 5:
                    newtable_flag = True
                    table_mask_empty[table
ymin:prev_ymax, table_xmin:prev_xmax] = 255

                    prev_dist = 0

                if index==len(objects)-1:
                    newtable_flag = True
                    table_mask_empty[table_
ymin:ymax, table_xmin:xmax] = 255

                    prev_dist = 0


        if backward_flag:
            dist = euc_dist(np.array([xmax,
ymin]), np.array([prev_xmin, prev_ymin]))

            if prev_dist == 0:
                prev_dist = dist
            else:
                if int(np.divide(dist, prev
 dist)) > 5 or index==len(objects)-1:
                    newtable_flag = True
                    table_mask_empty[ymin:t
able_ymax, xmin:table_xmax] = 255
                    prev_dist = 0

        prev_xmin = int(bndbox.find('xmin')
.text)
        prev_ymin = int(bndbox.find('ymin')
.text)
        prev_xmax = int(bndbox.find('xmax')
.text)
        prev_ymax = int(bndbox.find('ymax')
.text)
        prev_dist = dist
```

```
    save image(table_mask_path+ name+".jpeg", t
    able mask empty)
    save image(col_mask_path + name+".jpeg", co
    l_mask_empty)

    final dataframe dict['table mask'].append(t
    able mask path+ name+".jpeg")
    final dataframe dict['col mask'].append(col
     mask path + name+".jpeg")
    final_dataframe_dict['image'].append(image)
```

creating dataframe --> (original_image, table_mask, col_mask)

final_dataframe = pd.DataFrame(final_dataframe_dict)
final_dataframe.head(2)
final_dataframe.to_csv("C:/Users/sesha/Untitled Folder/Untitled Folder/green/final_dataframe.csv", index=False)

```
In [4]:
```

```
Out[4]:
```

	image	table_mask	col_mask
0	C:/Users/sesha/Untitled Folder/Untitled Folder...	C:/Users/sesha/Untitled Folder/Untitled Folder...	C:/Users/sesha/Untitled Folder/Untitled Folder...
1	C:/Users/sesha/Untitled Folder/Untitled Folder...	C:/Users/sesha/Untitled Folder/Untitled Folder...	C:/Users/sesha/Untitled Folder/Untitled Folder...

```
In [5]:
```

```
In [6]:
```

```
In [7]:
```

```
image  \
37  C:/Users/sesha/Untitled Folder/Untitled Fold
er...
32  C:/Users/sesha/Untitled Folder/Untitled Fold
```

```
er...

                                           tabl
e_mask  \
37  C:/Users/sesha/Untitled Folder/Untitled Fold
er...
32  C:/Users/sesha/Untitled Folder/Untitled Fold
er...

                                             co
l_mask
37  C:/Users/sesha/Untitled Folder/Untitled Fold
er...
32  C:/Users/sesha/Untitled Folder/Untitled Fold
er...
```

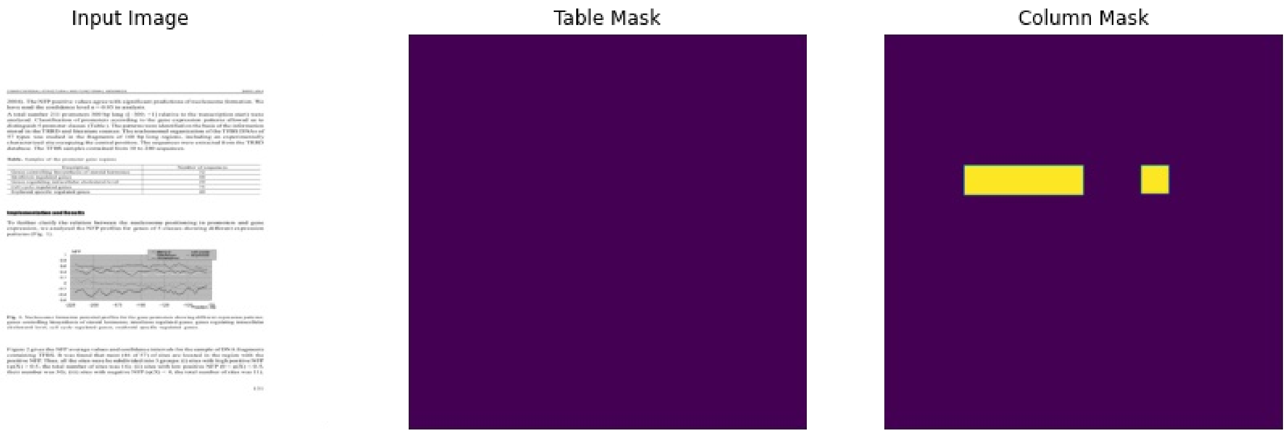
In [ ]:

In [8]:

In [9]:

In [10]:

```
(1024, 1024, 3)
(1024, 1024, 1)
(1024, 1024, 1)
```



In [11]:

In [12]:

```
Model: "model"
________________________________________________
________________________________________________
__
Layer (type)                    Output Shape
Param #     Connected to
================================================
================================================
==
input_1 (InputLayer)            [(None, 1024, 10
24,  0


________________________________________________
________________________________________________
__
block1_conv1 (Conv2D)           (None, 1024, 102
4, 6 1792         input_1[0][0]


________________________________________________
________________________________________________
__
block1_conv2 (Conv2D)           (None, 1024, 102
4, 6 36928        block1_conv1[0][0]


________________________________________________
________________________________________________
__
block1_pool (MaxPooling2D)      (None, 512, 512,
64) 0           block1_conv2[0][0]


________________________________________________
________________________________________________
__
block2_conv1 (Conv2D)           (None, 512, 512,
128 73856       block1_pool[0][0]
```

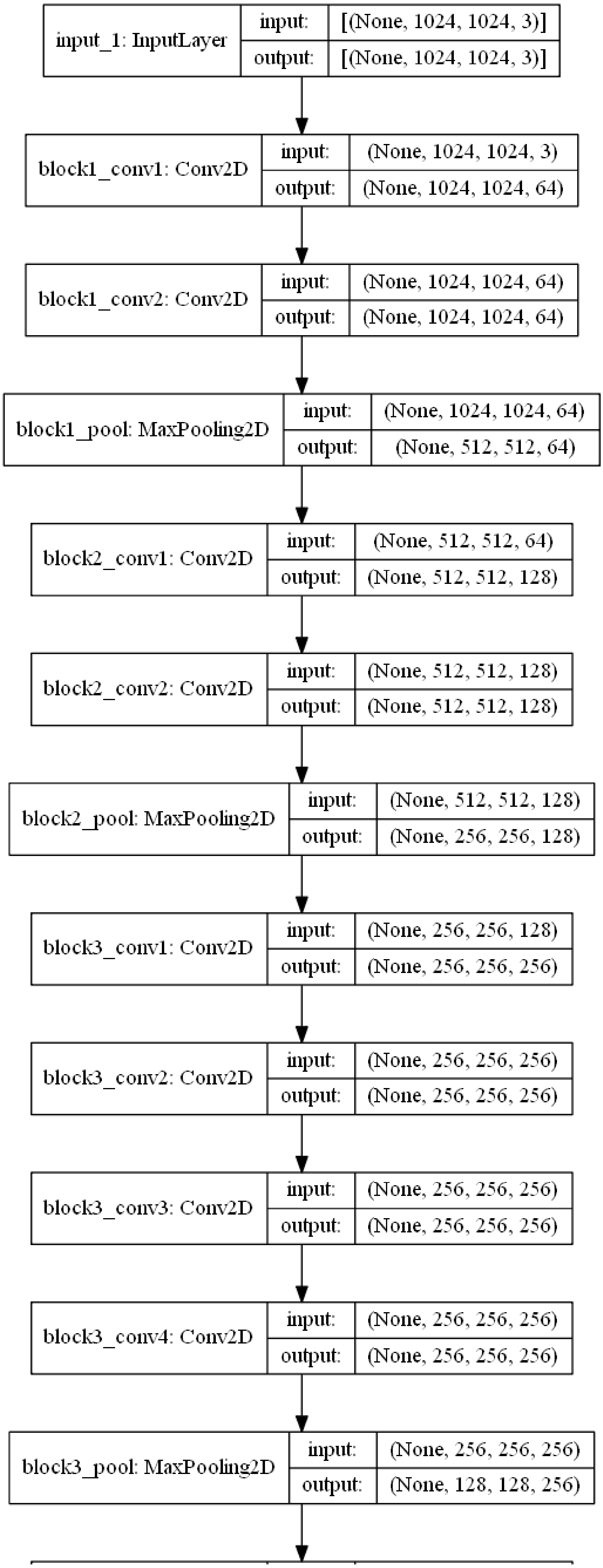
```
________________________________________________
________________________________________________
__
block2_conv2 (Conv2D)           (None, 512, 512,
128 147584      block2_conv1[0][0]

________________________________________________
________________________________________________
__
block2_pool (MaxPooling2D)      (None, 256, 256,
128 0           block2_conv2[0][0]

________________________________________________
________________________________________________
__
block3_conv1 (Conv2D)           (None, 256, 256,
256 295168      block2_pool[0][0]

________________________________________________
________________________________________________
__
block3_conv2 (Conv2D)           (None, 256, 256,
256 590080      block3_conv1[0][0]

________________________________________________
________________________________________________
__
block3_conv3 (Conv2D)           (None, 256, 256,
256 590080      block3_conv2[0][0]

________________________________________________
________________________________________________
__
block3_conv4 (Conv2D)           (None, 256, 256,
256 590080      block3_conv3[0][0]

________________________________________________
________________________________________________
__
block3_pool (MaxPooling2D)      (None, 128, 128,
block3_pool (MaxPooling2D)       (None, 128, 128,
256 0           block3_conv4[0][0]

________________________________________________
________________________________________________
__
block4_conv1 (Conv2D)            (None, 128, 128,
512 1180160     block3_pool[0][0]

________________________________________________
________________________________________________
__
block4_conv2 (Conv2D)            (None, 128, 128,
512 2359808     block4_conv1[0][0]

________________________________________________
________________________________________________
__
block4_conv3 (Conv2D)            (None, 128, 128,
512 2359808     block4_conv2[0][0]

________________________________________________
________________________________________________
__
block4_conv4 (Conv2D)            (None, 128, 128,
512 2359808     block4_conv3[0][0]

________________________________________________
________________________________________________
__
block4_pool (MaxPooling2D)       (None, 64, 64, 5
12)  0           block4_conv4[0][0]

________________________________________________
________________________________________________
__
block5_conv1 (Conv2D)            (None, 64, 64, 5
12)  2359808     block4_pool[0][0]
```

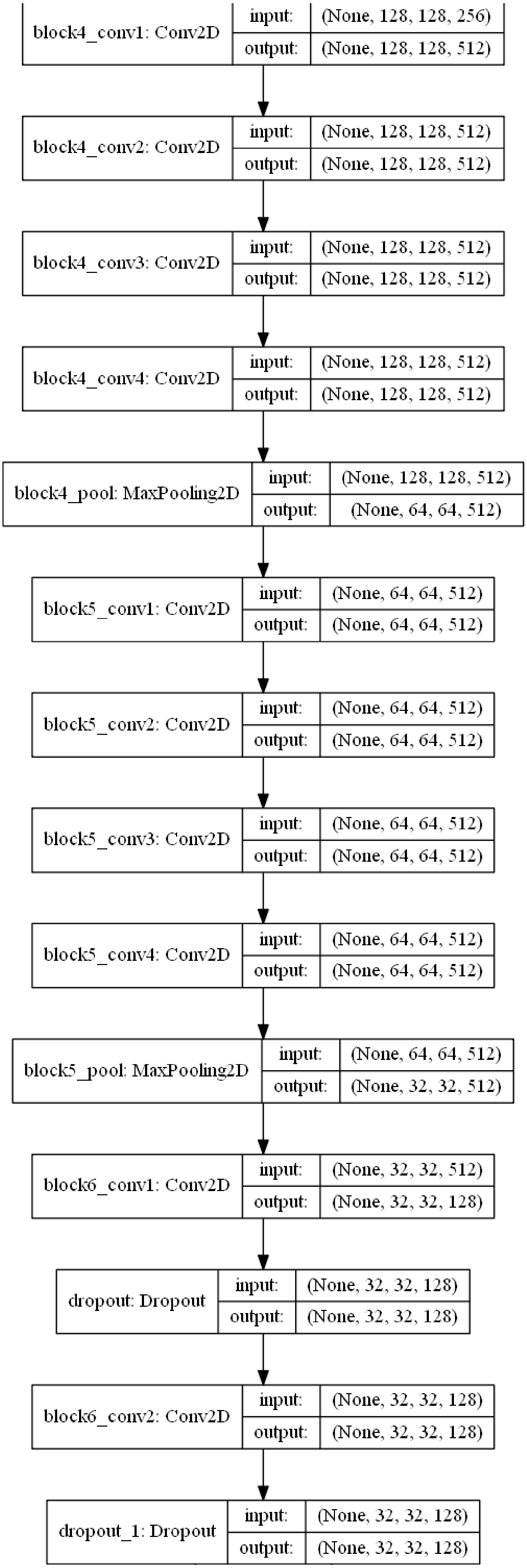
```
________________________________________________
________________________________________________
__
block5_conv2 (Conv2D)           (None, 64, 64, 5
12)  2359808     block5_conv1[0][0]

________________________________________________
________________________________________________
__
block5_conv3 (Conv2D)           (None, 64, 64, 5
12)  2359808     block5_conv2[0][0]

________________________________________________
________________________________________________
__
block5_conv4 (Conv2D)           (None, 64, 64, 5
12)  2359808     block5_conv3[0][0]

________________________________________________
________________________________________________
__
block5_pool (MaxPooling2D)      (None, 32, 32, 5
12)  0           block5_conv4[0][0]

________________________________________________
________________________________________________
__
block6_conv1 (Conv2D)           (None, 32, 32, 1
28)  65664       block5_pool[0][0]

________________________________________________
________________________________________________
__
dropout (Dropout)               (None, 32, 32, 1
28)  0           block6_conv1[0][0]

________________________________________________
________________________________________________
```

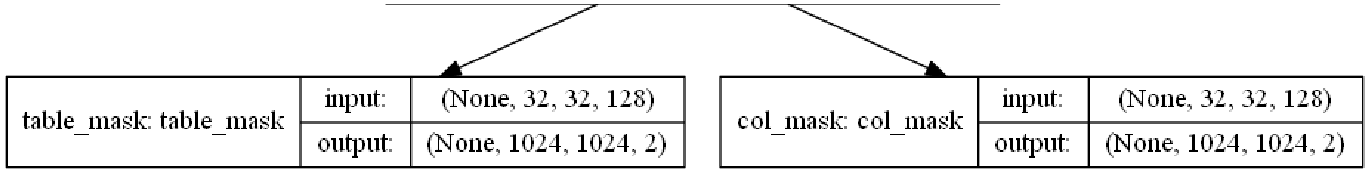
```
__
block6_conv2 (Conv2D)           (None, 32, 32, 1
28)  16512       dropout[0][0]


________________________________________________
________________________________________________
__
dropout_1 (Dropout)             (None, 32, 32, 1
28)  0           block6_conv2[0][0]


________________________________________________
________________________________________________
__
table_mask (table_mask)         (None, 1024, 102
4, 2 32642       dropout_1[0][0]


block3_pool[0][0]

block4_pool[0][0]
________________________________________________
________________________________________________
__
col_mask (col_mask)             (None, 1024, 102
4, 2 49154       dropout_1[0][0]


block3_pool[0][0]

block4_pool[0][0]
================================================
================================================
==
Total params: 20,188,356
Trainable params: 163,972
Non-trainable params: 20,024,384

________________________________________________
________________________________________________
__
```

In [13]:

Out[13]:


input_1: InputLayer | input: [(None, 1024, 1024, 3)] | output: [(None, 1024, 1024, 3)]
block1_conv1: Conv2D | input: (None, 1024, 1024, 3) | output: (None, 1024, 1024, 64)
block1_conv2: Conv2D | input: (None, 1024, 1024, 64) | output: (None, 1024, 1024, 64)
block1_pool: MaxPooling2D | input: (None, 1024, 1024, 64) | output: (None, 512, 512, 64)
block2_conv1: Conv2D | input: (None, 512, 512, 64) | output: (None, 512, 512, 128)
block2_conv2: Conv2D | input: (None, 512, 512, 128) | output: (None, 512, 512, 128)
block2_pool: MaxPooling2D | input: (None, 512, 512, 128) | output: (None, 256, 256, 128)
block3_conv1: Conv2D | input: (None, 256, 256, 128) | output: (None, 256, 256, 256)
block3_conv2: Conv2D | input: (None, 256, 256, 256) | output: (None, 256, 256, 256)
block3_conv3: Conv2D | input: (None, 256, 256, 256) | output: (None, 256, 256, 256)
block3_conv4: Conv2D | input: (None, 256, 256, 256) | output: (None, 256, 256, 256)
block3_pool: MaxPooling2D | input: (None, 256, 256, 256) | output: (None, 128, 128, 256)


block4_conv1: Conv2D
input: (None, 128, 128, 256)
output: (None, 128, 128, 512)
block4_conv2: Conv2D
input: (None, 128, 128, 512)
output: (None, 128, 128, 512)
block4_conv3: Conv2D
input: (None, 128, 128, 512)
output: (None, 128, 128, 512)
block4_conv4: Conv2D
input: (None, 128, 128, 512)
output: (None, 128, 128, 512)
block4_pool: MaxPooling2D
input: (None, 128, 128, 512)
output: (None, 64, 64, 512)
block5_conv1: Conv2D
input: (None, 64, 64, 512)
output: (None, 64, 64, 512)
block5_conv2: Conv2D
input: (None, 64, 64, 512)
output: (None, 64, 64, 512)
block5_conv3: Conv2D
input: (None, 64, 64, 512)
output: (None, 64, 64, 512)
block5_conv4: Conv2D
input: (None, 64, 64, 512)
output: (None, 64, 64, 512)
block5_pool: MaxPooling2D
input: (None, 64, 64, 512)
output: (None, 32, 32, 512)
block6_conv1: Conv2D
input: (None, 32, 32, 512)
output: (None, 32, 32, 128)
dropout: Dropout
input: (None, 32, 32, 128)
output: (None, 32, 32, 128)
block6_conv2: Conv2D
input: (None, 32, 32, 128)
output: (None, 32, 32, 128)
dropout_1: Dropout
input: (None, 32, 32, 128)
output: (None, 32, 32, 128)



In [14]:

In [15]:

**https://stackoverflow.com/questions/31653576/how-to-calculate-the-mean-iu-score-in-image-segmentation/31775111 from sklearn.metrics import confusion_matrix
import numpy as np**

def table_mask_iou(y_pred,y_true):

```
    # ytrue, ypred is a flatten vector
    y pred = y pred.flatten()
    y true = y true.flatten()
    y pred = y pred.flatten()
    y true = y true.flatten()
    current = confusion_matrix(y_true, y_pred
    , labels=[0, 1])
    # compute mean iou
    intersection = np.diag(current)
    ground truth set = current.sum(axis=1)
    predicted set = current.sum(axis=0)
    union = ground_truth_set + predicted_set
    - intersection
    IoU = intersection / union.astype(np.floa
    t32)
    return np.mean(IoU)
```

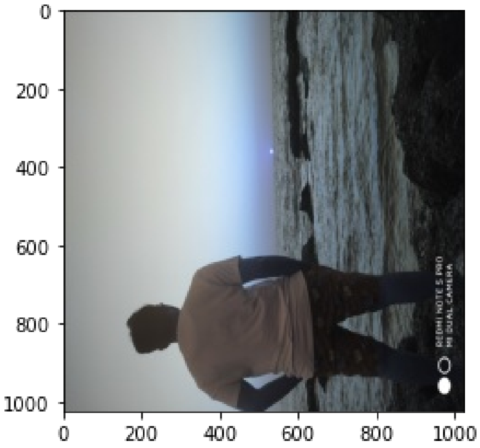
EPOCHS = 50 VAL_SUBSPLITS = 30 VALIDATION_STEPS = len(X_test)//BATCH_SIZE//VAL_SUBSPLITS

history = model.fit(train_dataset, epochs=EPOCHS, steps_per_epoch=train_steps, validation_data=test_dataset, validation_steps=VALIDATION_STEPS, callbacks=[model_checkpoint, es, DisplayCallback()])

In [16]:

In [17]:

```
(1, 1024, 1024, 3)
```



In [18]:

In [19]:

In [20]:

```
IOU_df = pd.DataFrame(IOU)

print(IOU_df)

IOU_df.to_csv("IOU_latest.csv")
```

In [21]:

In [22]:

```
   Unnamed: 0                Images  IOU_rati
os
0           0     10.1.1.7.2164_21.bmp    0.6868
87
1           1    10.1.1.160.615_15.bmp    0.5725
52
2           2    10.1.1.193.1803_3.bmp    0.6419
45
3           3   10.1.1.185.1566_14.bmp    0.7335
50
```

```
4             4     10.1.1.13.2943_5.bmp     0.5946
02
..          ...                      ...
...
94           94     10.1.1.8.2198_13.bmp     0.6675
52
95           95     10.1.1.160.701_6.bmp     0.6841
00
96           96   10.1.1.185.1552_1.bmp     0.4139
66
97           97   10.1.1.172.1007_3.bmp     0.5634
26
98           98   10.1.1.120.1527_3.bmp     0.5680
21

[99 rows x 3 columns]
```

In [25]:

In [26]:

```
    Unnamed: 0                    Images  IOU_ratio
s
96          96  10.1.1.185.1552_1.bmp    0.41396
6
54          54  10.1.1.100.302_10.bmp    0.53028
0
48          48  10.1.1.190.1808_4.bmp    0.55917
7
44          44  10.1.1.192.1811_6.bmp    0.56202
4
97          97  10.1.1.172.1007_3.bmp    0.56342
6
..         ...                    ...         ..
.
43          43   10.1.1.180.557_4.bmp    0.74850
7
27          27  10.1.1.160.546_36.bmp    0.75043
2
```

```
74            74   10.1.1.40.3122_3.bmp    0.75520
5
60            60    10.1.1.1.2019_2.bmp    0.78148
5
70            70    10.1.1.1.2057_5.bmp    0.78822
5

[99 rows x 3 columns]
```

In [27]:

best = [] medium = [] worst = []

a = pd.read_csv("IOU_latest.csv", usecols=['0'])

print(a)

for i in a.itertuples(): print(i) if i=='0': continue if float(i)>=0.70: best.append(i) if float(i)>=0.58 and i<=0.65: medium.append(i) if float(i)<=0.55: worst.append(i) print("") print("Best:",best) print("**") print("Medium:",medium) print("**") print("Worst:",worst) print("**")**

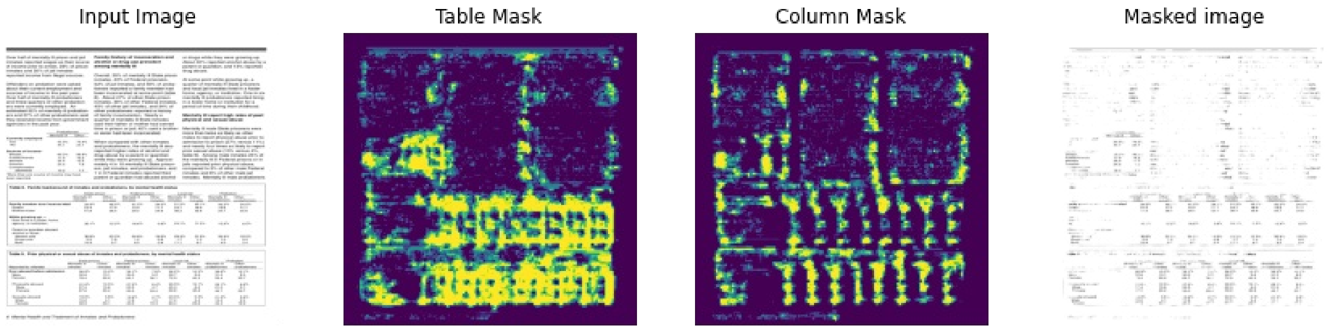
Best: [0.73354995, 0.7504325, 0.7273352, 0.7384416, 0.71315396, 0.7485073, 0.71735144, 0.7186574, 0.7814851, 0.7387027, 0.78822494, 0.7552048, 0.7066281, 0.7025455, 0.7294334]

Medium: [0.6419449, 0.594602, 0.60271955, 0.64220315, 0.62896365, 0.6481284, 0.61418533, 0.6200449, 0.63652116, 0.61110693, 0.5968022, 0.63183904, 0.6234341, 0.58831954, 0.63048756, 0.64278275, 0.6146496, 0.6349157, 0.5942844, 0.63654244, 0.59677905, 0.62716645, 0.62189436, 0.6100571, 0.6353363, 0.63046706, 0.64714867, 0.6194117, 0.62059885, 0.62035084, 0.61281496, 0.61763203, 0.6420317, 0.6460918, 0.629436, 0.6434793]

Worst: [0.53027964, 0.41396585]

In [24]:

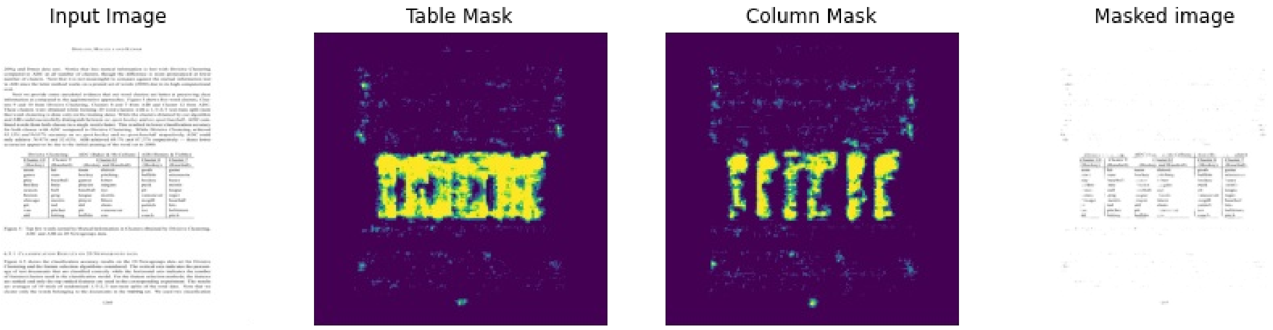
```
(1, 1024, 1024, 3)
```



```
************************************************
***********************************************
```

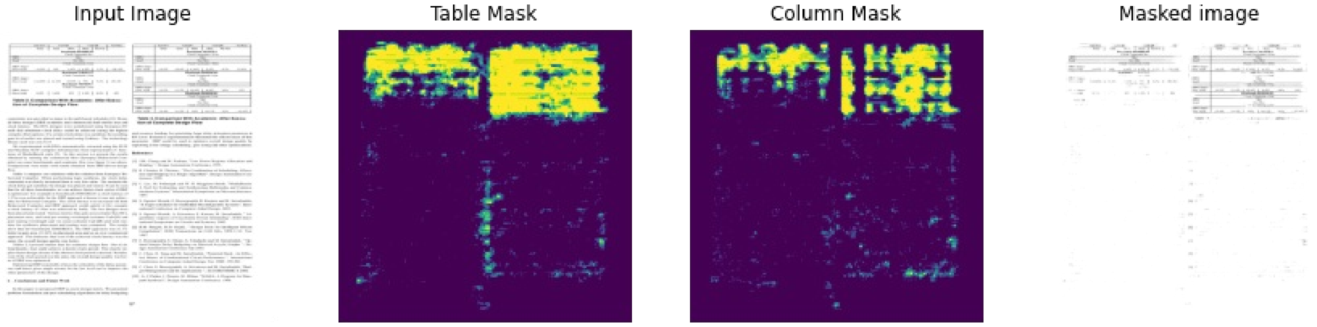
all Ao : aand or fe cee
ne 5 1 Boo an iy
! f
ie 4
"
ote 4 ae vere
: ion Dos uster
poutine ou : he . cr in sh
wT
med
ther .
ae = ss , rt
: 750% W es fen
1 ay also
: ree nd a
Pee tana on an o
come w rox " ri
once “a ce
peneaton a ano > ‘eiterat
soa amen em a Hi; fied ir inves
ohol rund
‘on : i
_ —— Eisien, lacnied "inmates, “amma initio,
coc ag Negi ataw bags Sts wea aoa gags
3 es bs we gens mo oaks
BS ee ee sex Ment asx
Bese ow wy ; 33
Bom an 5 2%
bie 8 a Taw : Tron” tau
ai mt Ieee weal tna, __tseataa__ neo

```
tena Bimer” teed ai Blnes cian E
ieee nena ana
: geo gah ongme Pgte geghe
roateaty atime! mo. 126 ag ae 0.0%
eo 28 Pt) Hy ta

************************************************
***********************************************
Mean IOU =  0.53027964
************************************************
***********************************************
(1, 1024, 1024, 3)
```



```
************************************************
***********************************************
23 2 atl
```

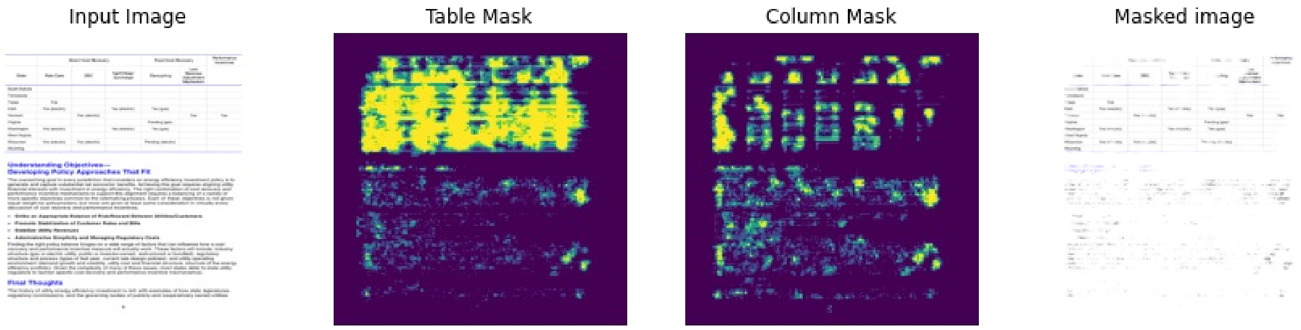
```
Hivisee i tag | ADE (Bax 3: MeCallum) | Ais (Sk
ashy)
Shuster 10°) "Cluster 9 ‘Chusteri> ‘Cluster 8” |
Cluster 7
(ockey) | (Baseball) (Hockey and Baseball) | (Ho
ckey) | (Baseball)
eam [hit team detroit ‘goals game
way yaseball nes bitter nockey >ases

wkey vase ayers angers puck moms
aso ball sseball nyt i league
stor: areg cague testis vancouver | roger
icage norris, yer blues negill basebaii
it ted abl shots patric hits

an pitcher pit neu wer ice ba
tht hitting: buffalo sus | coach pitch

‘ F x




**************************************************
*************************************************
Mean IOU =  0.6918371
**************************************************
*************************************************
(1, 1024, 1024, 3)
```



```
**************************************************
*************************************************
```

ena fren aa Bard ere ence =B
oe ead a ie
Star an a na
i
aac 1 perl
SET) ea ves Less oases |
“usted 3, Be. AF ane

```
;
ia 5
isl
rea he
ig
" iy < a

***************************************************
**************************************************
Mean IOU =  0.69578433
***************************************************
**************************************************
(1, 1024, 1024, 3)
```



```
***************************************************
**************************************************
rect ort an enme hemes were










aes

= ° nue

slate Aula ase sBc Hing a. iaastement
```

Mectz:nism
seu Tlakota
tennessee
Fexas Yes
Utah Yes (electric) Yes (e'2tric) Yes (gas)
Yes (2 nittie} Yes Yes

virginia Pending (gas)
Washington Yes (eiatiic) ‘Yes teivetric) Yes (gas)
West Virginia
Wisconsin Yes (esti) Yes (¢stric) Per-ding (eizstde)
Wyoming _ “| _ _

wl 2

ye : —
pone a . sigs oe ee »
wiry -.

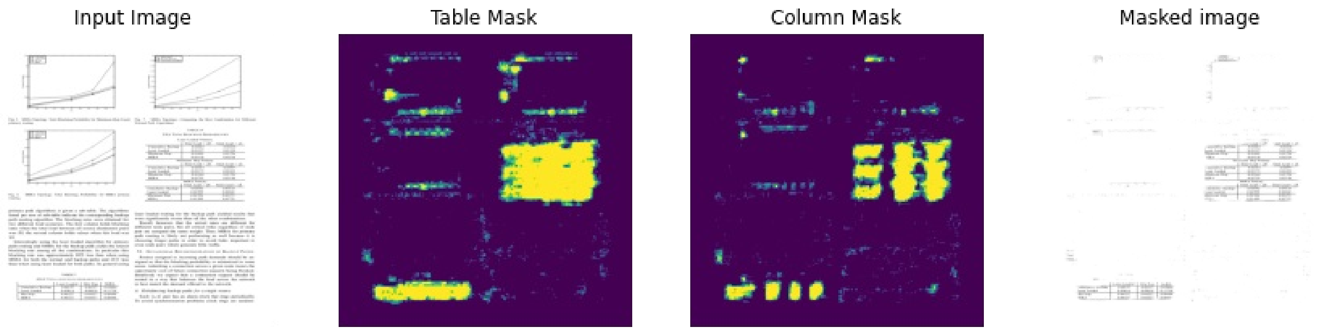
au i : ed
wor ae He cn =
win " ® '

a at '

ia #4 ye

ne ’
“pre ; :
q au Py
i 1 Me ony
4 : see 00 ing
re argu

```
************************************************
***********************************************
Mean IOU =  0.5210244
************************************************
***********************************************
(1, 1024, 1024, 3)
```



```
************************************************
***********************************************
```

was,
Lost Loatod | tin Hp 1 eda,

wtiilanive sit ap | OGeSST | UIT | Dow T

Least Losded oos6i6 | o0s616 | 0.11728

in Hop ‘Oae527 —| 06537

NUR, 0.06527 | 0.06527 | ones”

1 ina Bac

fatal Lead = 44
O03

Least Loaded 0.05330
Mfininran Hop 0.03756
wes 0.01338 0.02526

Wiliin Hop Primary

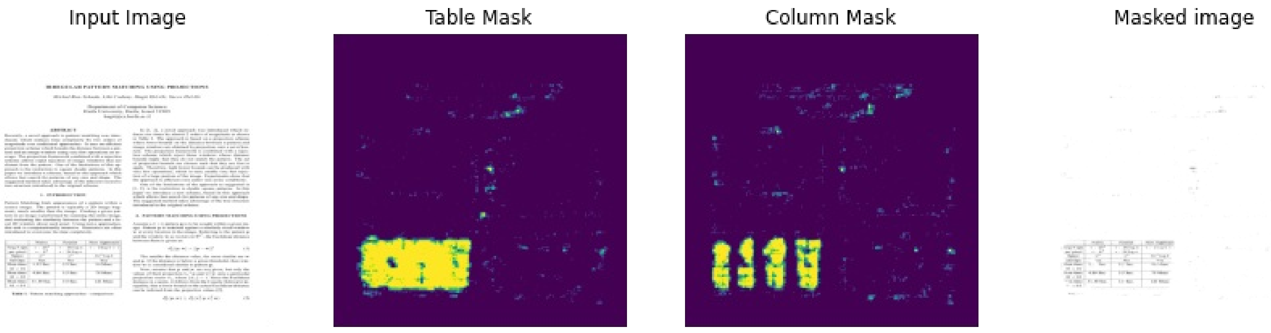
“Total Load = ut

otal Loa = 24

```
east Loaded 0.03173 0.05035
inwitint Hop ‘Ooze 0.03 756
ages 0.01791 0.03196

MIRA Primary
‘Total Loat = 20

Toul Load = 24
```

```
cmalane aia C0208 TOSS
east Lowes 0.02999 o.oasa2
‘linia Hop C0526 ‘OaszTT

0.01490 0.02728
```

```
************************************************
***********************************************
Mean IOU =  0.7175415
************************************************
***********************************************
(1, 1024, 1024, 3)
```



```
************************************************
***********************************************
```

| Aug Fops
per pixel

‘Space

Int Ops

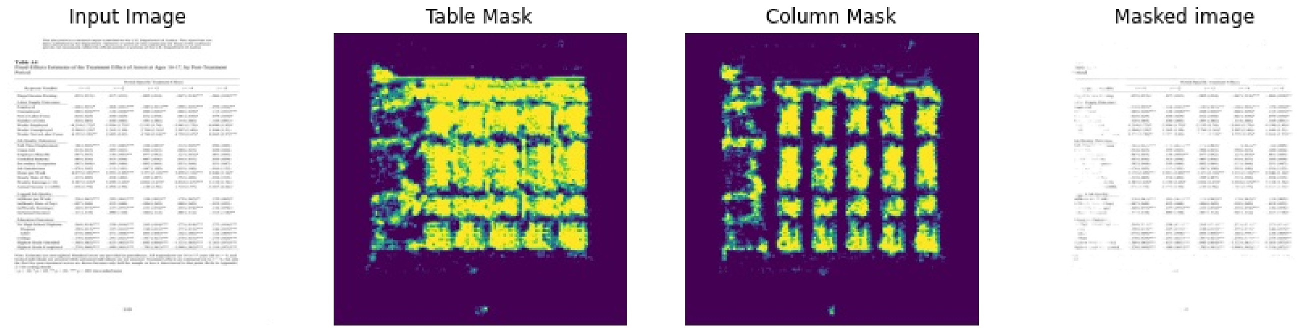
Run time
16 = 16

Hun time
32 = 32

la time
«64

We

```
9G.


************************************************
***********************************************
Mean IOU =  0.7138675
************************************************
***********************************************
(1, 1024, 1024, 3)
```



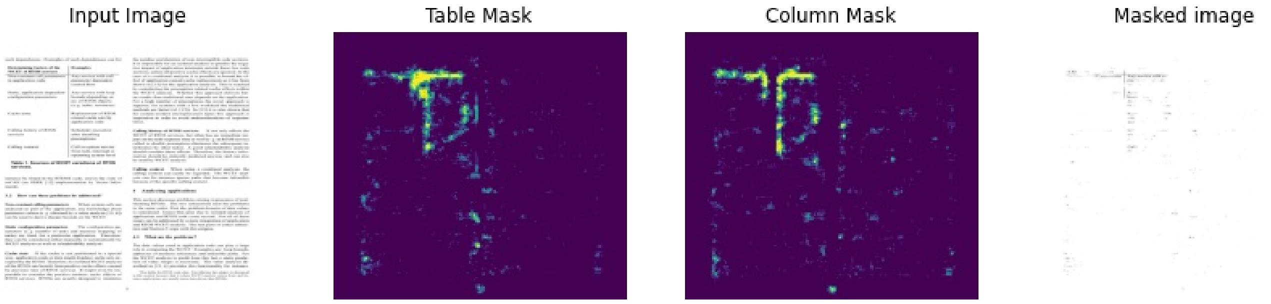
```
************************************************
***********************************************


Period-Specific Treatment Effects
```

a ras2 ms t+ ras

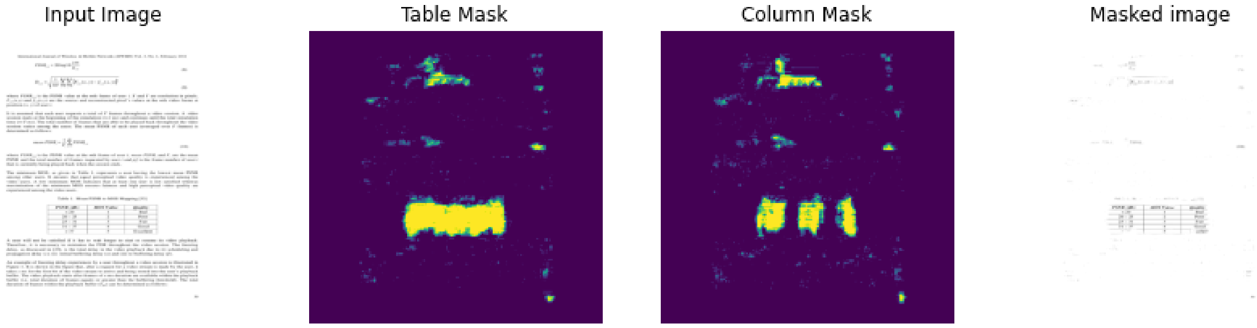
```
1025 (.013)+ 017 (013) =.005 (.014) 047 (OEE) 06
6 (.018)*
<8 (.021)* oes (021 089 (OZT)=A" 90 L022)" 370 (
.024)*~
O88 (.028)"** 128 (,028)*" 080 (.028)*! 068 (.02
9)* 115 (.033)*"
024 (.029) 030 (.029) 012 (.030) 061 (.030)* 079
(.034)"
838 (085) ~.030 (085) =.091 (086) =.114 (086) ~.
169 (.091)>
vat 4214(1.72)" —-3.956(1.73)" —-2.193(01.74) 3.
051 (1.75) 8 90,(1.85)"
2 904 (1.29)" 1.265 (1.30) 2.760.136)" 2.507 (1.
40)+ 8441.51)
* $3731.90)" Z.8n7 12.03) Sez aayen 4.759 2.47)"
R.E69 (2 97)"
18S CORSE EE LOn spe ime L083)" 045)" 156 (045)
914 (023) 085 (023) 206 (023) 984 (.023) 920 (02
4)
067 (.053) 186(.053)"* 077 (.052) 127 (.053)* 06
1 (.055)
983 (.036) 13 (036) 207 (036) 916 (037) 935 (.03
9)
982 (044) O49 (044) 002 (.044 S83 (084) 31 (.047
)
876 (102) =113 (101) 47 (.100) 19 (.100) 16 (.12
2)
S75 (1.09) 3.993 (1.09)""" —3.575(1.10)°* —-E.8B
Y(T.1O)"°* 2.846 (1.16)
15 (480) O34 (482) 389 (487) 87 (488) 934 (.515)
3.587 (1.65)* 3891.65)" 4.622 (1.67) 6.034 (1.67
)"*" —3.110(4.76)+
ete) 148 (£96) SLOT) Bor aye
sy, A Job Quatity:
fa( Hess por eek) 226 (.062)"** = 202. (.965)""*
de (.062)"* 178 C062)" 4:29 (.065)"
fra Fe FPay) 007 (.048) 035 (.048) 004 (.049) 08
0 (.049) 019 (.052)
chi ses) 260(.073)***  237(.073)""*  « 233(.074
```

```
obi ses) 260(.073)^^^ —-237(.073)""^ —«.233(.074
)"* ——-255(.074)"** 136.078)"
Anes Linamuiged VEE (110) 090.110) 044 4.112) GE
(M2) A313 118)
268 (48a 258 (05) 2as eae" 275 Ley 273 Lersy
190 (083) 187 (013)" 180 (043) 171 (02° 146 (.01
5)"
071 (.008)* 085 (.008)*. 102 008)" 128 (.009)**
291 (.021)""" 307 (021)9** = 278 (UTI 275 (.024)
*
623 (.0633°"" 885 (064)/9** 1.125 (.065)°"" 1.26
3 (.073)**
90 (061\9" 785 LOGIY**™ —1.040.(.062)""" 1.316 C
071)"
```

```
****************************************************
*****************************************************
Mean IOU =  0.6588544
****************************************************
*****************************************************
(1, 1024, 1024, 3)
```



```
************************************************************
**********************************************************




"eu =
TE TH palsies _| Any service wath &
rar ee









weet
awe
4
os
aa
an 4

************************************************************
**********************************************************
Mean IOU =  0.6147557
************************************************************
**********************************************************
(1, 1024, 1024, 3)
```



```
************************************************************
```

****

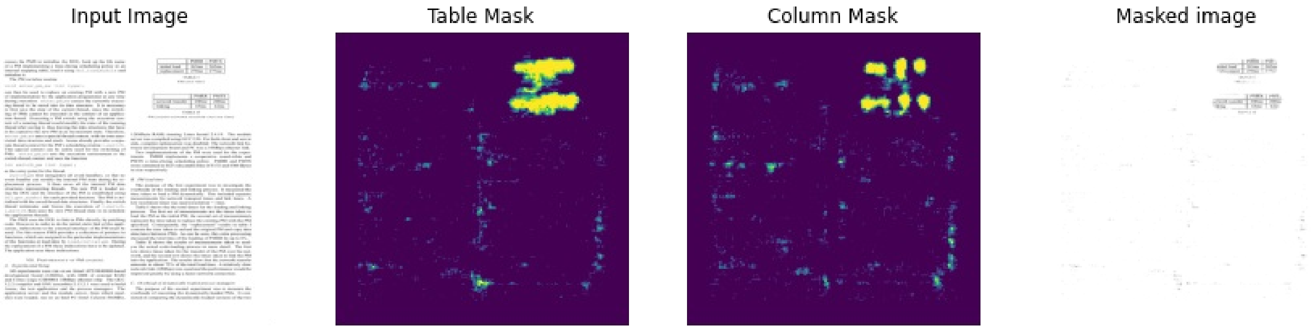
coy

al D s x

PSNR (dB) MOS Vaiue Quatity
<20 1 Bad
20-25 2 Poor
25— 31 3 Fair
37 a Good

as 5 7 cellent

```
so

*************************************************
************************************************
Mean IOU =  0.71671945
*************************************************
************************************************
(1, 1024, 1024, 3)
```



```
*************************************************
************************************************
```

```
[LPMiRR | PT:
nial load [| 263ms | 265m:
Bite

fetwork

transfer
```

“aking

Plecement || 279ms

TABLE’
Mba

TAPLE

```
************************************************
***********************************************
Mean IOU =  0.6606277
************************************************
***********************************************
```

```
from prettytable import PrettyTable
x = PrettyTable() x.field_names = ["Image
Number","MeanIOU Score"] x.add_row(['1', '0.6466862'])
```

```
Number','MeanIOU Score'] x.add_row(['1','0.5[illegible]'])
x.add_row(['2','0.625956']) x.add_row(['3','0.7644789'])
x.add_row(['4','0.61763203']) x.add_row(['5','0.56443346'])
x.add_row(['6','0.58016765']) x.add_row(['7','0.7130594'])
x.add_row(['8','0.61871165']) x.add_row(['9','0.63872015'])
x.add_row(['10','0.5968022']) print(x)
```

```
In [ ]:
```