TF MODEL TO TFLITE MODEL

In [1]:

```
import tensorflow as tf
from tensorflow.python.tools import freeze_graph
#from tensorflow.contrib import lite
```

In [2]:

```
TF_LITE_MODEL_FILE_NAME = "tf_lite_model.tflite"
```

model = 'C:/Users/sesha/Untitled Folder/Untitled Folder/green/model/saved_model.pb'

model=tf.keras.models.load_model('C:/Users/sesha/Untitled Folder/Untitled Folder/green/model') tf_lite_converter = tf.lite.TFLiteConverter.from_keras_model(model) tflite_model = tf_lite_converter.convert()

In [3]:

```
converter = tf.lite.TFLiteConverter.from_saved_model('C:/Users/sesha/Untitled Folder/Unti
tled Folder/green/model') # path to the SavedModel directory
tflite_model = converter.convert()
```

TABLE NET

In [4]:

```
#!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 [5]:

```
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/10.1.1.1.200
6_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/tablem
ask/"
```

```
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/orgimag
e/"

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

In [6]:

```
"""
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[6]:

	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**

In [7]:

```
# """
# <size>
#   <width>793</width>
#   <height>1123</height>
#   <depth>3</depth>
# </size>


# <object>
#   <name>column</name>
#   <pose>Unspecified</pose>
#   <truncated>0</truncated>
#   <difficult>0</difficult>
#   <bndbox>
#    <xmin>458</xmin>
#    <ymin>710</ymin>
#    <xmax>517</xmax>
#    <ymax>785</ymax>
#   </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 path
    image = dataPath + row['image_path'] # image .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, width), 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, width), 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)
        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_flag:


                table_xmin = prev_xmin
                table_ymin = prev_ymin
                newtable_flag = False
                forward_flag = True
                backward_flag = False


            if xmin < prev_xmin and newtable_flag:


                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] = 2
55


                        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:table_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", table_mask_empty)
    save_image(col_mask_path + name+".jpeg", col_mask_empty)

    final_dataframe_dict['table_mask'].append(table_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_datafr
ame_tflite.csv", index=False)
```

```
<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>
```

```
<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>
```

```
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
```

```
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
```

```
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
```

```
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
```

```
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
```

```
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
```

```
<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>
```

```
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
```

```
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
<Figure size 360x360 with 0 Axes>
```

```
<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>
```

```
<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>

<Figure size 360x360 with 0 Axes>
```

In [8]:

```
final_dataframe = pd.read_csv("C:/Users/sesha/Untitled Folder/Untitled Folder/green/final
_dataframe_tflite.csv")
final_dataframe.head(2)
```

Out[8]:

	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 [9]:

```
X_train, X_test = train_test_split(final_dataframe, test_size=0.2)
```

In [10]:

```
training_dataset = (
    tf.data.Dataset.from_tensor_slices(
        (
            tf.cast(X_train['image'].values, tf.string),
            tf.cast(X_train['table_mask'].values, tf.string),
            tf.cast(X_train['col_mask'].values, tf.string),
        )
    )
)
```

```
testing_dataset = (
    tf.data.Dataset.from_tensor_slices(
        (
            tf.cast(X_test['image'].values, tf.string),
            tf.cast(X_test['table_mask'].values, tf.string),
            tf.cast(X_test['col_mask'].values, tf.string),
        )
    )
)
```

In [11]:

```
x_test_df = pd.DataFrame(data=X_test)
print(x_test_df.head(2))
x_test_df.to_csv("x_test_csv_tflite.csv")
```

```
                                                 image  \
193  C:/Users/sesha/Untitled Folder/Untitled Folder...
322  C:/Users/sesha/Untitled Folder/Untitled Folder...

                                            table_mask  \
193  C:/Users/sesha/Untitled Folder/Untitled Folder...
322  C:/Users/sesha/Untitled Folder/Untitled Folder...

                                              col_mask
193  C:/Users/sesha/Untitled Folder/Untitled Folder...
322  C:/Users/sesha/Untitled Folder/Untitled Folder...
```

In [12]:

```
# https://www.tensorflow.org/tutorials/load_data/images

@tf.function
def load_image(image, table_mask, col_mask):

    image = tf.io.read_file(image)
    table_mask=tf.io.read_file(table_mask)
    col_mask=tf.io.read_file(col_mask)

    image=tf.io.decode_bmp(image, channels=3)
    image=tf.image.resize(image, [1024, 1024])
    image = tf.cast(image, tf.float32) / 255.0

    table_mask=tf.io.decode_jpeg(table_mask, channels=1)
    table_mask=tf.image.resize(table_mask, [1024, 1024])
    table_mask = table_mask / 255.0


    col_mask=tf.io.decode_jpeg(col_mask, channels=1)
    col_mask=tf.image.resize(col_mask, [1024, 1024])
    col_mask = col_mask / 255.0

    return image, {"table_mask":table_mask, "col_mask":col_mask}



# creating dataset object
train = training_dataset.map(load_image, num_parallel_calls=tf.data.AUTOTUNE)
test = testing_dataset.map(load_image)
```

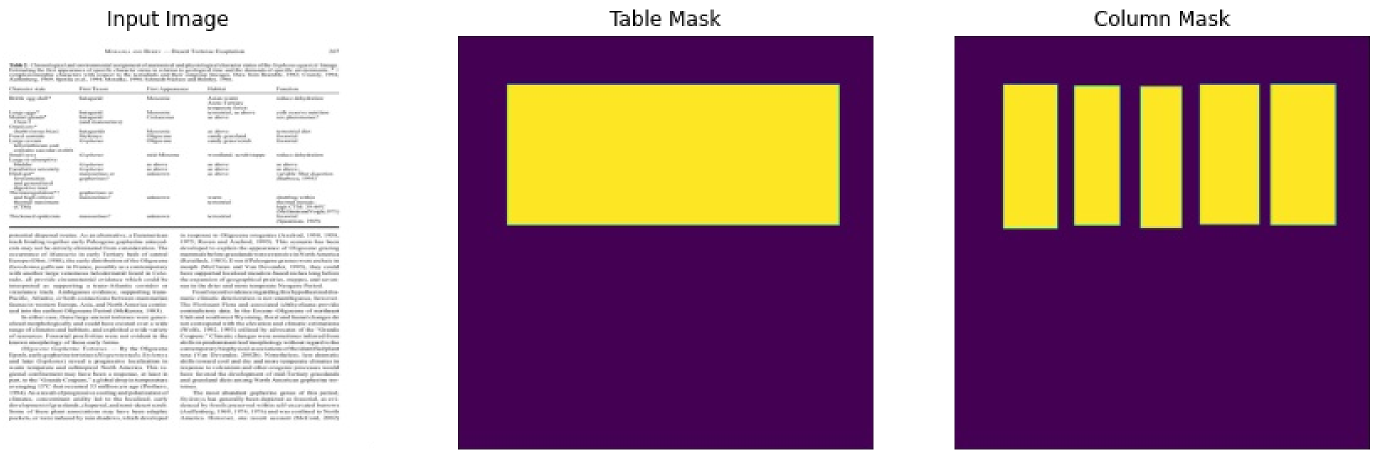
In [13]:

```
BATCH_SIZE = 1
BUFFER_SIZE = 10
train_steps = len(X_train) // BATCH_SIZE

# for feeding to training
train_dataset = train.shuffle(BUFFER_SIZE).batch(BATCH_SIZE).repeat()
train_dataset = train_dataset.prefetch(buffer_size=tf.data.experimental.AUTOTUNE)
test_dataset = test.batch(BATCH_SIZE)
```

In [14]:

```python
def display(display_list):
    plt.figure(figsize=(15, 15))
    title = ['Input Image', 'Table Mask', 'Column Mask', 'Masked image']
    for i in range(len(display_list)):
        plt.subplot(1, len(display_list), i+1)
        plt.title(title[i])

        image = display_list[i]

        plt.imshow(tf.keras.preprocessing.image.array_to_img(image))
        plt.axis('off')
    plt.show()



for image, mask in train.take(1):

    sample_image = image
    sample_table_mask = mask['table_mask']
    sample_col_mask = mask['col_mask']


    print(image.shape)
    print(mask['table_mask'].shape)
    print(mask['col_mask'].shape)
    display([image, mask['table_mask'], mask['col_mask']])
```

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



In [15]:

```python
import tensorflow as tf
from tensorflow.keras.applications import VGG19
from tensorflow.keras.layers import Input
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Conv2D
from tensorflow.keras.layers import Dropout
from tensorflow.keras.layers import UpSampling2D
from tensorflow.keras.layers import Concatenate
from tensorflow.keras.layers import Layer
from tensorflow.keras.layers import Activation
from tensorflow.keras.layers import Conv2DTranspose
from tensorflow.keras.utils import plot_model
from tensorflow.keras import backend as K
```

In [16]:

```python
"""
Table decoder
-------------
x = conv7(1x1)
```

```
x = x(upscaled) + vgg19(pool4)
x = x(upscaled) + vgg19(pool3)
x = upscaled to match input dimention (1024)


column decoder
---------------
x = conv7(1x1, relu)
x = dropout(0.8)
x = conv8(1x1)
x = upscaled() + vgg19(pool4)
x = x(upscaled) + vgg19(pool3)
x = upscaled (1024)

"""


tf.keras.backend.clear_session()


class table_mask(Layer):

    def __init__(self):
        super().__init__()
        self.conv_7 = Conv2D(kernel_size=(1,1), filters=128, kernel_regularizer=tf.keras
.regularizers.l2(0.002),activation = 'relu')
        self.upsample_pool4 = UpSampling2D(size=(2, 2), interpolation='bilinear')
        self.upsample_pool3 = UpSampling2D(size=(2, 2), interpolation='bilinear')
        self.upsample_final = Conv2DTranspose(filters=2, kernel_size=3, strides=2, paddi
ng='same', activation='softmax')

    def call(self, input, pool3, pool4):

        x = self.conv_7(input)
        x = self.upsample_pool4(x)
        x = Concatenate()([x, pool4])

        x = self.upsample_pool3(x)
        x = Concatenate()([x, pool3])

        x = UpSampling2D((2,2))(x)
        x = UpSampling2D((2,2))(x)

        x = self.upsample_final(x)

        return x

class col_mask(Layer):

    def __init__(self):
        super().__init__()
        self.conv_7 = Conv2D(kernel_size=(1,1), filters=128, kernel_regularizer=tf.keras
.regularizers.l2(0.004), kernel_initializer='he_normal',activation = 'relu')
        self.drop = Dropout(0.8)
        self.conv_8 = Conv2D(kernel_size=(1,1), filters=128, kernel_regularizer=tf.keras
.regularizers.l2(0.004), kernel_initializer='he_normal',activation = 'relu')
        self.upsample_pool4 = UpSampling2D(size=(2, 2), interpolation='bilinear')
        self.upsample_pool3 = UpSampling2D(size=(2, 2), interpolation='bilinear')
        self.upsample_final = Conv2DTranspose(filters=2, kernel_size=3, strides=2, paddi
ng='same', activation='softmax')

    def call(self, input, pool3, pool4):

        x = self.conv_7(input)
        x = self.drop(x)
        x = self.conv_8(x)

        x = self.upsample_pool4(x)
        x = Concatenate()([x, pool4])

        x = self.upsample_pool3(x)
        x = Concatenate()([x, pool3])
        x = UpSampling2D((2,2))(x)
        x = UpSampling2D((2,2))(x)

        x = self.upsample_final(x)

        return x



input_shape = (1024, 1024, 3)
input_ = Input(shape=input_shape)

vgg19_ = VGG19(
    include_top=False,
    weights="imagenet",
    input_tensor=input_,
    input_shape=None,
    pooling=None,
    classes=1000,
    classifier_activation="softmax",
)

for layer in vgg19_.layers:
    layer.trainable = False

pool3 = vgg19_.get_layer('block3_pool').output
pool4 = vgg19_.get_layer('block4_pool').output

conv_1_1_1 = Conv2D(filters=128, kernel_size=(1, 1), activation='relu', name="block6_con
v1", kernel_regularizer=tf.keras.regularizers.l2(0.004))(vgg19_.output)
conv_1_1_1_drop = Dropout(0.8)(conv_1_1_1)

conv_1_1_2 = Conv2D(filters=128, kernel_size=(1, 1), activation='relu', name="block6_con
v2", kernel_regularizer=tf.keras.regularizers.l2(0.004))(conv_1_1_1_drop)
conv_1_1_2_drop = Dropout(0.8)(conv_1_1_2)

table_mask = table_mask()(conv_1_1_2_drop, pool3, pool4)
col_mask = col_mask()(conv_1_1_2_drop, pool3, pool4)

model_tflite = Model(input_, [table_mask, col_mask])

model_tflite.summary()
```

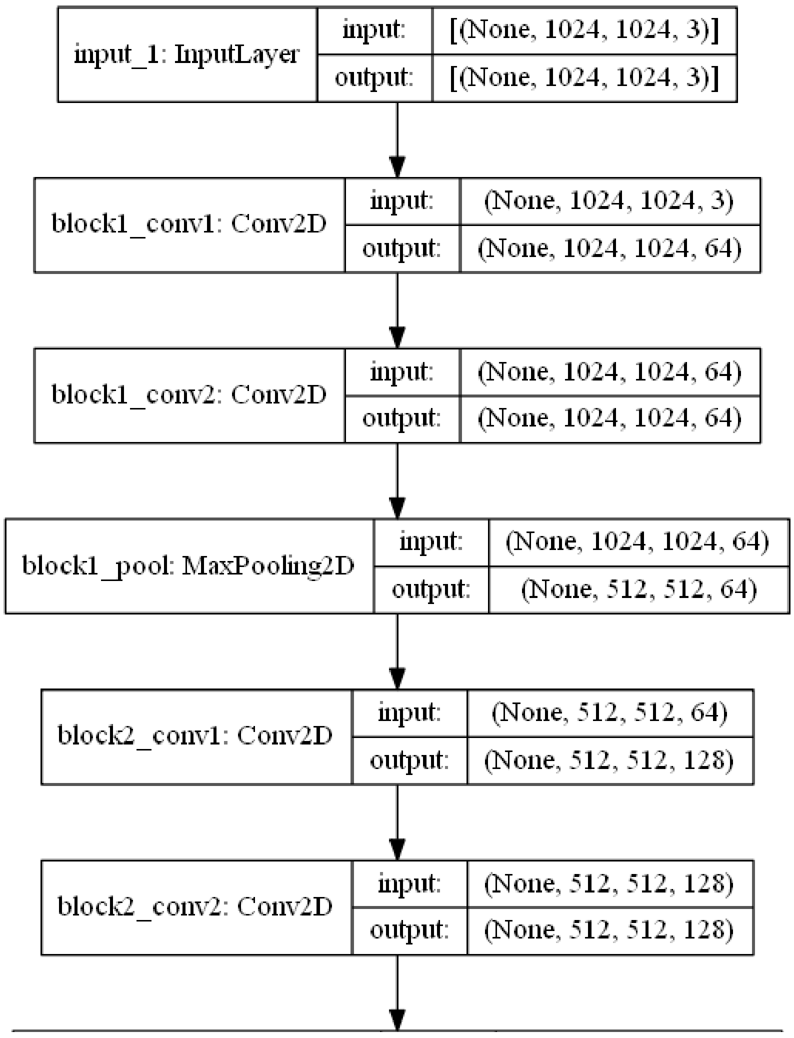
```
Model: "model"
__________________________________________________________________________________________
_________
Layer (type)                     Output Shape          Param #     Connected to

==========================================================================================
=========
input_1 (InputLayer)             [(None, 1024, 1024,  0


__________________________________________________________________________________________
_________
block1_conv1 (Conv2D)            (None, 1024, 1024, 6 1792         input_1[0][0]


__________________________________________________________________________________________
_________
block1_conv2 (Conv2D)            (None, 1024, 1024, 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, 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, 512)  0           block4_conv4[0][0]

__________________________________________________________________________________________________
block5_conv1 (Conv2D)           (None, 64, 64, 512)  2359808     block4_pool[0][0]

__________________________________________________________________________________________________
block5_conv2 (Conv2D)           (None, 64, 64, 512)  2359808     block5_conv1[0][0]

__________________________________________________________________________________________________
block5_conv3 (Conv2D)           (None, 64, 64, 512)  2359808     block5_conv2[0][0]

__________________________________________________________________________________________________
block5_conv4 (Conv2D)           (None, 64, 64, 512)  2359808     block5_conv3[0][0]

__________________________________________________________________________________________________
block5_pool (MaxPooling2D)      (None, 32, 32, 512)  0           block5_conv4[0][0]

__________________________________________________________________________________________________
block6_conv1 (Conv2D)           (None, 32, 32, 128)  65664       block5_pool[0][0]

__________________________________________________________________________________________________
dropout (Dropout)               (None, 32, 32, 128)  0           block6_conv1[0][0]
```

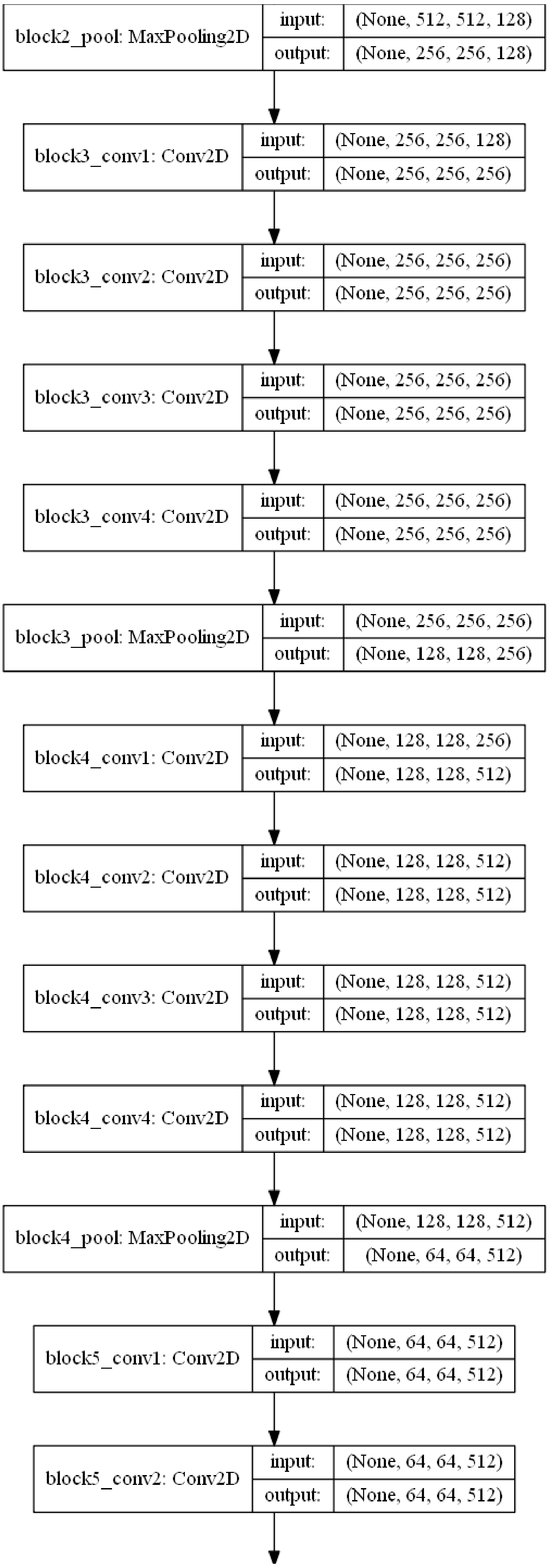
```
__________________________________________________________________________________________
_________
block6_conv2 (Conv2D)           (None, 32, 32, 128)  16512       dropout[0][0]

__________________________________________________________________________________________
_________
dropout_1 (Dropout)             (None, 32, 32, 128)  0           block6_conv2[0][0]

__________________________________________________________________________________________
_________
table_mask (table_mask)         (None, 1024, 1024, 2 32642       dropout_1[0][0]

                                                                 block3_pool[0][0]

                                                                 block4_pool[0][0]

__________________________________________________________________________________________
_________
col_mask (col_mask)             (None, 1024, 1024, 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
__________________________________________________________________________________________
_________
```

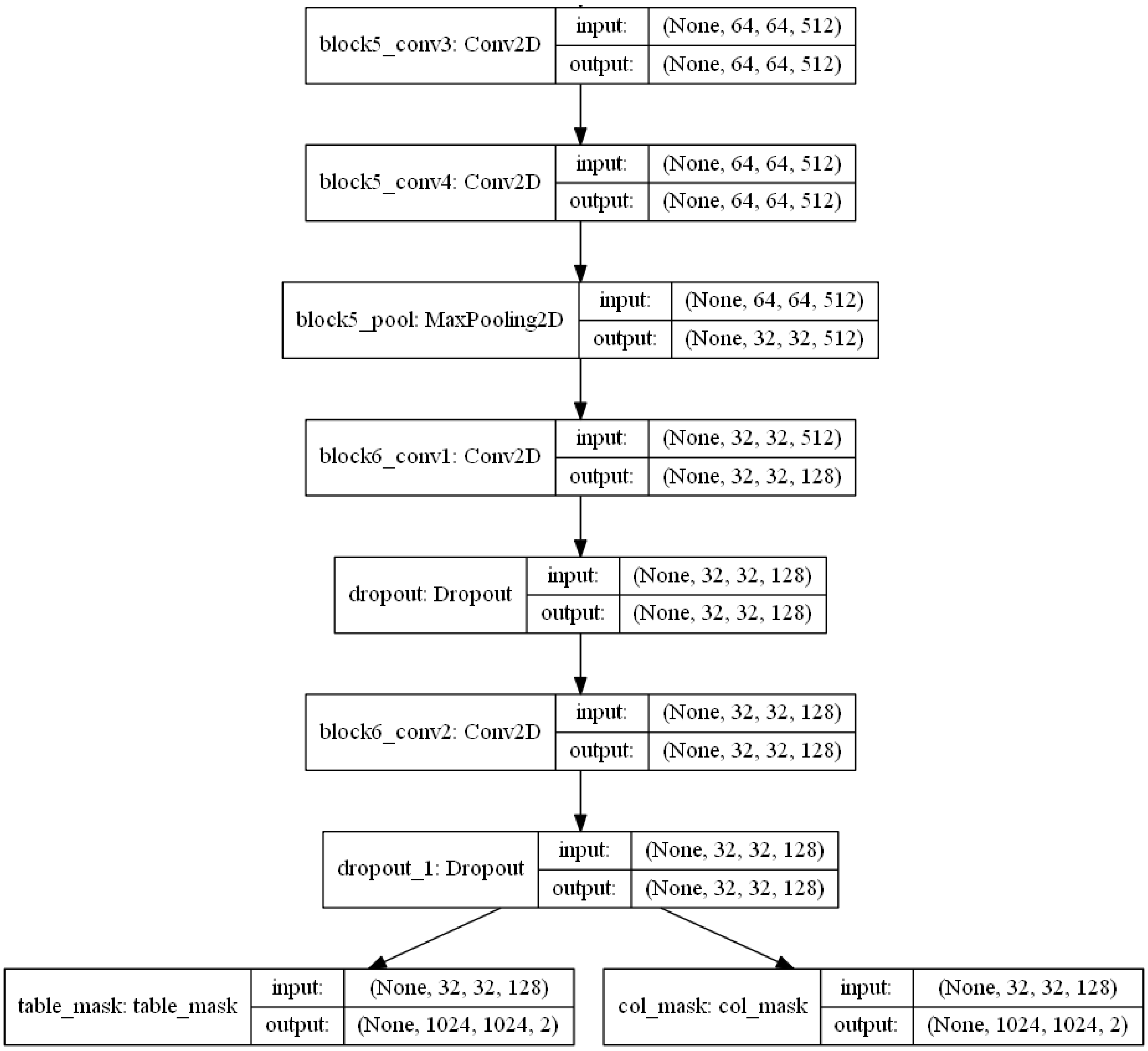
In [17]:

```
plot_model(model_tflite,show_shapes=True,show_layer_names=True)
```

Out[17]:




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)
table_mask: table_mask
input: (None, 32, 32, 128)
output: (None, 1024, 1024, 2)
col_mask: col_mask
input: (None, 32, 32, 128)
output: (None, 1024, 1024, 2)

In [18]:

```
losses = {
    "table_mask": 'sparse_categorical_crossentropy',
    "col_mask": 'sparse_categorical_crossentropy',
}

# tf.keras.losses.SparseCategoricalCrossentropy(from_logits=False)



# filepath = "/content/drive/MyDrive/case study - II/tablenet/model checkpoint/table_net.
h5"
# model_checkpoint_tflite = tf.keras.callbacks.ModelCheckpoint(filepath, monitor = "val_t
able_mask_loss", save_best_only=True, verbose = 0, mode="min")

filepath_tflite = "/content/drive/MyDrive/case study - II/tablenet/model checkpoint/model
_checkpoint_tflite/table_net.h5"
model_checkpoint_tflite = tf.keras.callbacks.ModelCheckpoint(filepath_tflite, monitor =
"val_table_mask_loss", save_best_only=True, verbose = 0, mode="min")


es = tf.keras.callbacks.EarlyStopping(monitor='val_loss', mode='min', patience=5,)


class F1_Score(tf.keras.metrics.Metric):

    def __init__(self, name='f1_score', **kwargs):
        super().__init__(name=name, **kwargs)
        self.f1 = self.add_weight(name='f1', initializer='zeros')
        self.precision_fn = tf.keras.metrics.Precision(thresholds=0.5)
        self.recall_fn = tf.keras.metrics.Recall(thresholds=0.5)

    def update_state(self, y_true, y_pred, sample_weight=None):
        p = self.precision_fn(y_true, tf.argmax(y_pred, axis=-1))
        r = self.recall_fn(y_true, tf.argmax(y_pred, axis=-1))
        # since f1 is a variable, we use assign
        self.f1.assign(2 * ((p * r) / (p + r + 1e-6)))

    def result(self):
        print("F1 Score = ",self.f1)
        return self.f1

    def reset_states(self):
        # we also need to reset the state of the precision and recall objects
        self.precision_fn.reset_states()
        self.recall_fn.reset_states()
        self.f1.assign(0)

#F1_Score(),
metrics = [F1_Score()]
#metrics=metrics,

global init_lr
init_lr = 0.0001

model_tflite.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=init_lr, epsilon=1
e-8,),
              loss=losses,
              metrics=metrics, )
```

In [19]:

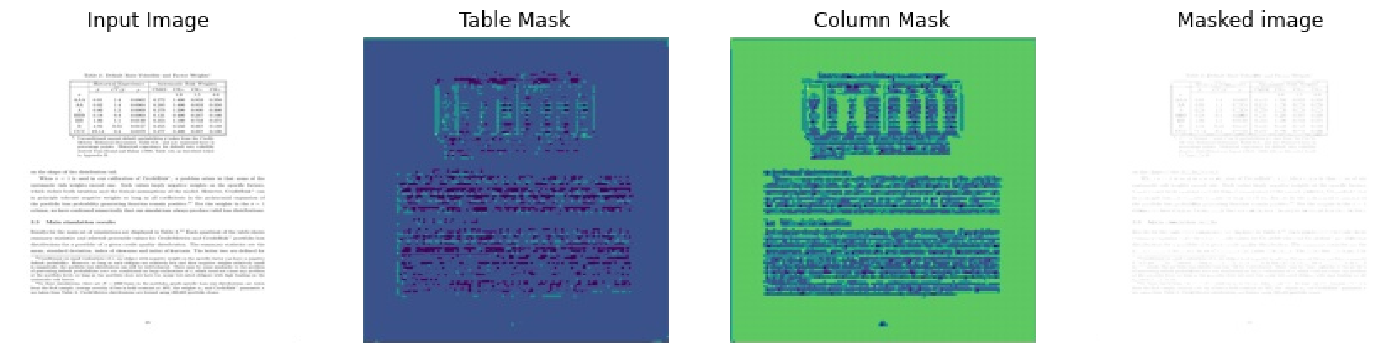
```
def show_predictions(dataset=None, num=1):

    if dataset:
        for image, mask in dataset.take(1):
            table_mask_pred, col_mask_pred = model_tflite.predict(image)

            table_mask_pred = tf.argmax(table_mask_pred, axis=-1)
            table_mask_pred = table_mask_pred[..., tf.newaxis][0]

            col_mask_pred = tf.argmax(col_mask_pred, axis=-1)
            col_mask_pred = col_mask_pred[..., tf.newaxis][0]

            im=tf.keras.preprocessing.image.array_to_img(image[0])
            im.save('image.png')

            im=tf.keras.preprocessing.image.array_to_img(table_mask_pred)
            im.save('table_mask_pred.png')

            im=tf.keras.preprocessing.image.array_to_img(col_mask_pred)
            im.save('col_mask_pred.png')



            img_org = Image.open('./image.png')
            table_mask = Image.open('./table_mask_pred.png')
            col_mask = Image.open('./col_mask_pred.png')


            # convert images
            img_mask = table_mask.convert('L')
            # img_mask = col_mask.convert('L')

            # grayscale
            # add alpha channel
            img_org.putalpha(img_mask)
```

```
                # save as png which keeps alpha channel
                img_org.save('output.png')


                display([image[0], table_mask_pred, col_mask_pred, img_org])

                pytesseract.pytesseract.tesseract_cmd = r'/usr/bin/tesseract'
                text = pytesseract.image_to_string(Image.open('./output.png'), lang='eng' )
# config='--psm 11'
                print(text)



class DisplayCallback(tf.keras.callbacks.Callback):

    def __init__(self):
        self.history = {'val_table_mask_loss':[]}
        self.init_lr = init_lr

    def on_epoch_end(self, epoch, logs=None):
        if epoch % 1 == 0:
            show_predictions(test_dataset, 1)


            self.history['val_table_mask_loss'].append(logs.get('val_table_mask_loss'))
            if epoch > 2:
                cur_loss = self.history['val_table_mask_loss'][epoch]
                prev_loss = self.history['val_table_mask_loss'][epoch-1]

                if cur_loss > prev_loss:
                    self.init_lr = self.init_lr * 0.93
                    K.set_value(self.model.optimizer.learning_rate, self.init_lr)
```

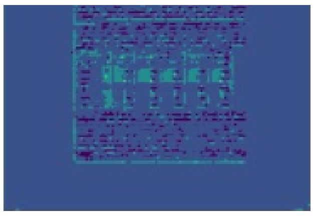
In [22]:

```
count = 0

for image, mask in test_dataset.take(10):

    print(image.shape)

    table_mask_pred, col_mask_pred = model_tflite.predict(image,batch_size = int(len(ima
ge)/2))

    table_mask_pred = tf.argmax(table_mask_pred, axis=-1)
    table_mask_pred = table_mask_pred[..., tf.newaxis][0]

    col_mask_pred = tf.argmax(col_mask_pred, axis=-1)
    col_mask_pred = col_mask_pred[..., tf.newaxis][0]

    im=tf.keras.preprocessing.image.array_to_img(image[0])
    im.save('image.png')

    im=tf.keras.preprocessing.image.array_to_img(table_mask_pred)
    im.save('table_mask_pred.png')

    im=tf.keras.preprocessing.image.array_to_img(col_mask_pred)
    im.save('col_mask_pred.png')



    img_org = Image.open('./image.png')
    table_mask = Image.open('./table_mask_pred.png')
    col_mask = Image.open('./col_mask_pred.png')


    # convert images
    img_mask = table_mask.convert('L')
    # img_mask = col_mask.convert('L')

    # grayscale
    # add alpha channel
    img_org.putalpha(img_mask)

    # save as png which keeps alpha channel
    img_org.save('output.png')


    display([image[0], table_mask_pred, col_mask_pred, img_org])

    #table_mask_iou(table_mask_pred, img_org)

    pytesseract.pytesseract.tesseract_cmd = r'C:/Users/sesha/AppData/Local/Programs/Tesse
ract-OCR/tesseract.exe'
    text = pytesseract.image_to_string(Image.open('./output.png'), lang='eng' ) # config
='--psm 11'
    print("****************************************************************************
****************")
    print(text)
    print("****************************************************************************
****************")

    table_mask_pred_flattened = tf.reshape(table_mask_pred,[-1])
    col_mask_pred_flattened = tf.reshape(col_mask_pred,[-1])
    #img_org_flattened = tf.reshape(img_org,[-1])

    m = tf.keras.metrics.MeanIoU(num_classes=2)
    m.update_state(table_mask_pred_flattened,col_mask_pred_flattened)
    print("Mean IOU = ",m.result().numpy())
    print("****************************************************************************
****************")
```

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



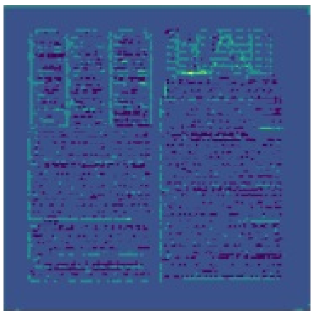
```
******************************************************************************************
******



em.







******************************************************************************************
******
Mean IOU =  0.07937212
******************************************************************************************
******
(1, 1024, 1024, 3)
```



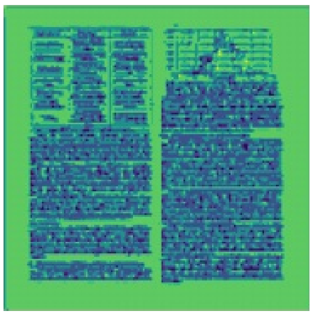
```
******************************************************************************************
******


a

******************************************************************************************
******
Mean IOU =  0.07310585
******************************************************************************************
******
(1, 1024, 1024, 3)
```

Input Image

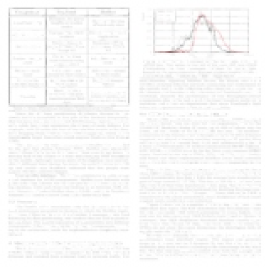
Table Mask



Column Mask



Masked image

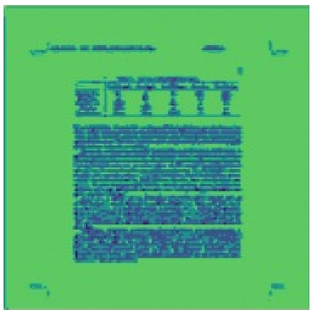


```
******************************************************************************************
******


or

******************************************************************************************
******
Mean IOU =  0.12705715
******************************************************************************************
******
(1, 1024, 1024, 3)
```

Input Image

Table Mask

Column Mask



Masked image

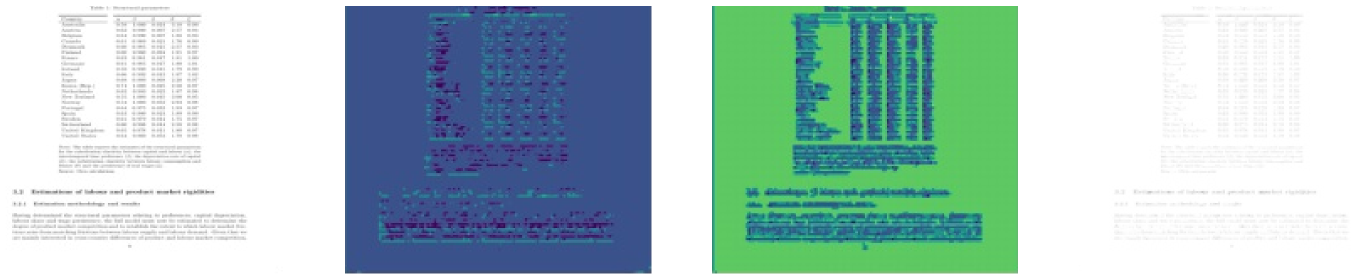
```
******************************************************************************************
******

******************************************************************************************
******
Mean IOU =  0.057753563
******************************************************************************************
******
(1, 1024, 1024, 3)
```

Input Image

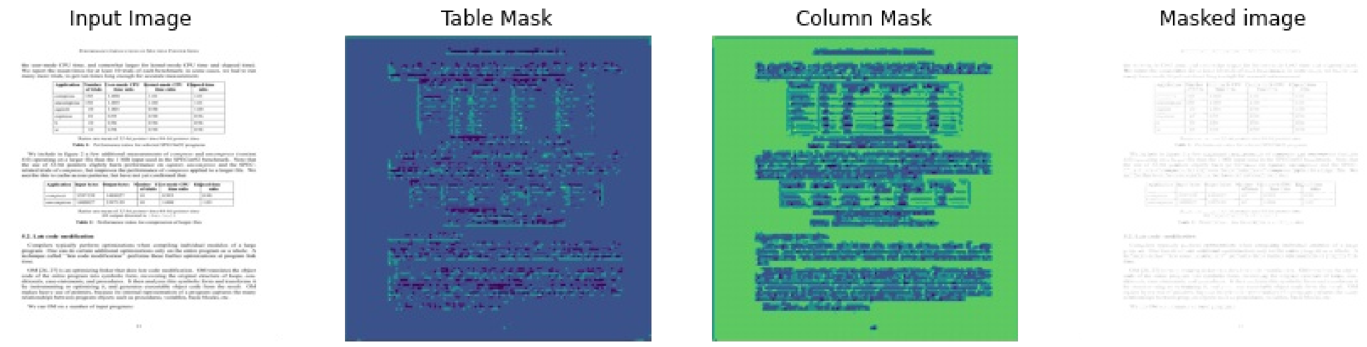
Table Mask

Column Mask

Masked image



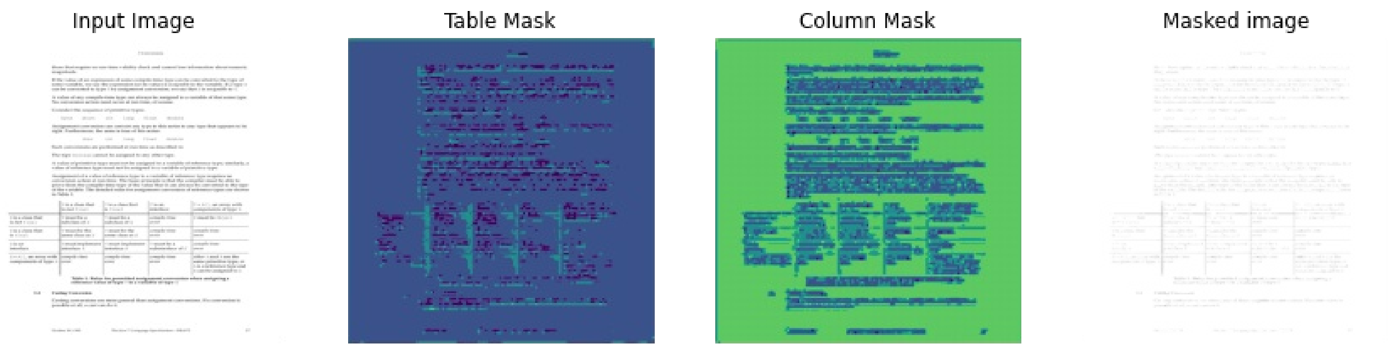
```
************************************************************************************************
******
bss

”

abeoa



************************************************************************************************
******
Mean IOU =  0.055454012
************************************************************************************************
******
(1, 1024, 1024, 3)
```



```
************************************************************************************************
******
Ww

ra

ox



¥
```

```
fal ach. aren De PP Mi eeompe
A " WET Ban fin
* opmpe © :
= ea
oan seimses sion.
za ny wileal we
ebel a Ht wr
```

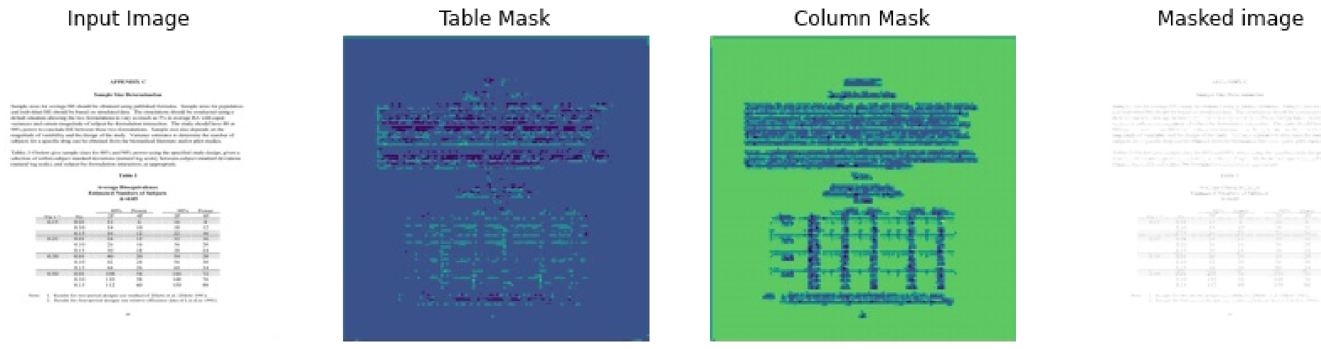
```
**********************************************************************************************
Mean IOU =  0.0918481
**********************************************************************************************
(1, 1024, 1024, 3)
```



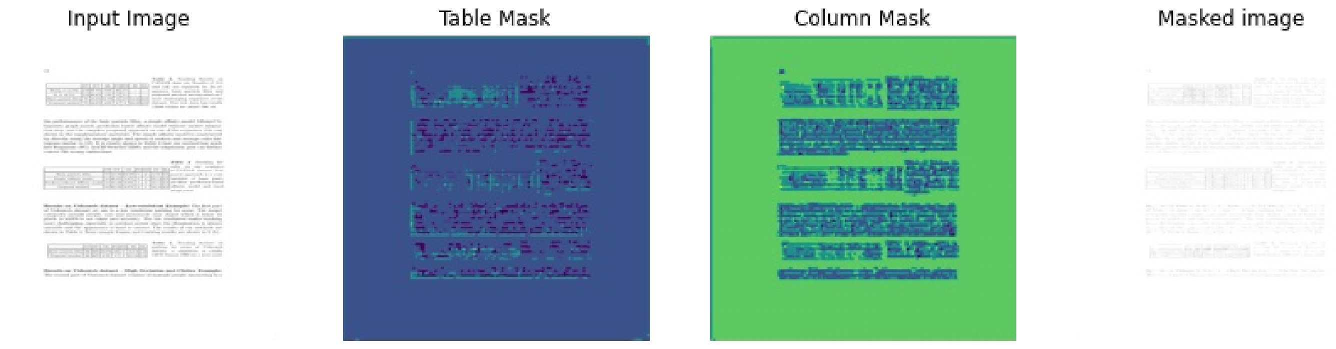
```
**********************************************************************************************
```

```
esting Wither wants
```

```
a


************************************************************************************************
Mean IOU =  0.07874289
************************************************************************************************
(1, 1024, 1024, 3)
```

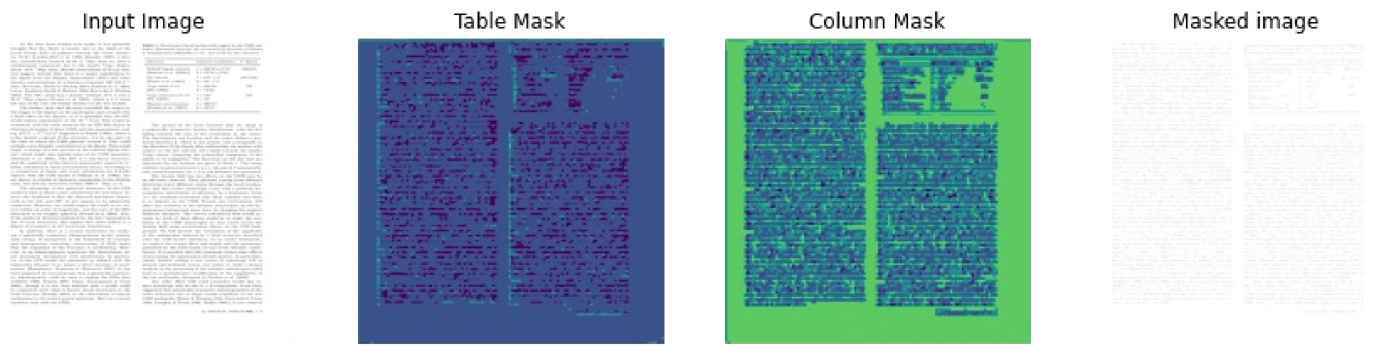


```
************************************************************************************************









a on
a wate rain! a fe 20m oe sop les
* site . 2 - a . .
: rene 2 cee
nna. atal sshainilit fags af ee ae eit lean ilevte. otakrmr dhe omnis
? J w 2 hey
isthe ` Head "
liao # =
aes - a

************************************************************************************************
Mean IOU =  0.057444554
************************************************************************************************
(1, 1024, 1024, 3)
```



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

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

```
******
Mean IOU =  0.055701926
****************************************************************************************
******
(1, 1024, 1024, 3)
```



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

****************************************************************************************
******
Mean IOU =  0.13336223
****************************************************************************************
******
```

```
In [ ]:
```