```
import pandas as pd
import numpy as np
import os
from PIL import Image


from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
def image_to_numpy(image_path):
    with Image.open(image_path) as img:
        return np.array(img)


input_folder = "/content/drive/MyDrive/Learner_1"
y_test_folder = "/content/drive/MyDrive/ODsamples"
y_test_image_list = sorted(os.listdir(y_test_folder))
y_test_arrays = {}
y_pred_image_list = sorted(os.listdir(input_folder))
y_pred_arrays = {}


new_size = (768, 512)
def resize_image(image_array, new_size):
    img = Image.fromarray(image_array)
    resized_img = img.resize(new_size, Image.ANTIALIAS)
    return np.array(resized_img)
for image_name in y_test_image_list:
    image_path = os.path.join(y_test_folder, image_name)
    image_array = image_to_numpy(image_path)
    resized_image_array = resize_image(image_array, new_size)
    y_test_arrays[image_name] = resized_image_array
```

```
<ipython-input-5-1b5c4a6e4276>:4: DeprecationWarning: ANTIALIAS is deprecated and will be removed in Pillow 10 (2023-07-01). Us
  resized_img = img.resize(new_size, Image.ANTIALIAS)
```

```
for image_name in y_pred_image_list:
    if image_name.startswith("Prediction_") and image_name.endswith(".png"):
        image_path = os.path.join(input_folder, image_name)
        image_array = image_to_numpy(image_path)
        y_pred_arrays[image_name] = image_array


wrongly_predicted_coords = []
for image_name in y_test_image_list:
    if image_name in y_pred_arrays:
        y_test_array = y_test_arrays[image_name]
        y_pred_array = y_pred_arrays[image_name]
        diff_indices = np.where(y_test_array != y_pred_array)
        wrongly_predicted_coords.extend(list(zip(diff_indices[0], diff_indices[1])))


excel_file = "/content/drive/MyDrive/ambiguties_tested.csv"
df = pd.read_csv(excel_file)
modified_y_pred_arrays = y_pred_arrays.copy()


for index, i in df.iterrows():
    excel_filename = i['File_Name']
    variable_name = excel_filename.replace("amb_", "").replace(".png", "")

    if variable_name in modified_y_pred_arrays:
        image_array = modified_y_pred_arrays[variable_name]
        row_coord = i['Row_Coordinate']
        col_coord = i['Column_Coordinate']
        center_pixel_value = i['Center_Pixel_Value']

        image_array[row_coord, col_coord] = center_pixel_value
        modified_y_pred_arrays[variable_name] = image_array
```

```
output_image_directory = '/content/drive/MyDrive/CORRECTED_images'
os.makedirs(output_image_directory, exist_ok=True)

for image_name, image_array in modified_y_pred_arrays.items():
    modified_image = Image.fromarray(image_array)
    modified_image_path = os.path.join(output_image_directory, image_name)
    modified_image.save(modified_image_path)


!zip -r /content/drive/MyDrive/CORRECTED_images.zip /content/drive/MyDrive/CORRECTED_images
```

```
  adding: content/drive/MyDrive/CORRECTED_images/ (stored 0%)
  adding: content/drive/MyDrive/CORRECTED_images/Prediction_drishtiGS_091.png (deflated 25%)
  adding: content/drive/MyDrive/CORRECTED_images/Prediction_drishtiGS_092.png (deflated 25%)
  adding: content/drive/MyDrive/CORRECTED_images/Prediction_drishtiGS_093.png (deflated 23%)
  adding: content/drive/MyDrive/CORRECTED_images/Prediction_drishtiGS_094.png (deflated 26%)
  adding: content/drive/MyDrive/CORRECTED_images/Prediction_drishtiGS_095.png (deflated 25%)
  adding: content/drive/MyDrive/CORRECTED_images/Prediction_drishtiGS_096.png (deflated 24%)
  adding: content/drive/MyDrive/CORRECTED_images/Prediction_drishtiGS_097.png (deflated 21%)
  adding: content/drive/MyDrive/CORRECTED_images/Prediction_drishtiGS_098.png (deflated 21%)
  adding: content/drive/MyDrive/CORRECTED_images/Prediction_drishtiGS_099.png (deflated 22%)
  adding: content/drive/MyDrive/CORRECTED_images/Prediction_drishtiGS_100.png (deflated 22%)
  adding: content/drive/MyDrive/CORRECTED_images/Prediction_drishtiGS_101.png (deflated 21%)
```

```
import pandas as pd
import numpy as np
import cv2
import os
from PIL import Image


def get_image_size(image_path):
    with Image.open(image_path) as img:
        return img.size


folder1_path = "/content/sample_data/y_pred"
folder2_path = "/content/sample_data/y_test"
image1_list = os.listdir(folder1_path)
image2_list = os.listdir(folder2_path)


image1_path = os.path.join(folder1_path, image1_list[0])
size1 = get_image_size(image1_path)
print(f"images size: {size1}")

image2_path = os.path.join(folder2_path, image2_list[0])
size2 = get_image_size(image2_path)
print(f"labels size: {size2} bytes")

     images size: (768, 512)
     labels size: (2049, 1749) bytes


def image_to_numpy(image_path):
    with Image.open(image_path) as img:
        img_array = np.array(img)
    return img_array
```

```
input_folder = "/content/sample_data/y_pred"
image_list = os.listdir(input_folder)
image_arrays = {}


for image_name in image_list:
  image_path = os.path.join(input_folder, image_name)
  image_array = image_to_numpy(image_path)
  image_arrays[f'amb_{image_name}'] = image_array


image_names = sorted(os.listdir("/content/drive/MyDrive/img"))
y_test = sorted(os.listdir("/content/sample_data/y_test"))
y_pred = sorted(os.listdir("/content/sample_data/y_pred"))


data = {'image_names': image_names,'y_test': y_test,'y_pred': y_pred}
df = pd.DataFrame(data)
df.to_excel("all_file_names.xlsx", index=False)


def image_to_numpy(image_path):
    img = Image.open(image_path)
    return np.array(img)


image_directory = "/content/sample_data/y_pred"


image_files = [f for f in os.listdir(image_directory) if f.endswith('.png')]


for image_file in image_files:
    image_name = image_file[:-4]
    image_path = os.path.join(image_directory, image_file)
    globals()[image_name] = image_to_numpy(image_path)
```

```
excel_file = "/content/ambiguties_test (1).xlsx"
df = pd.read_excel(excel_file)


for index, row in df.iterrows():
    excel_filename = row['File_Name']
    row_coord = row['Row_Coordinate']
    col_coord = row['Column_Coordinate']
    center_pixel_value = row['Center_Pixel_Value']

    variable_name = excel_filename.replace("amb_", "").replace(".png", "")

    if variable_name in globals():
        image_array = globals()[variable_name]

        image_array[row_coord, col_coord] = center_pixel_value
        globals()[variable_name] = image_array


output_image_directory = '/content/sample_data/numpy_images'
for index, row in df.iterrows():
    excel_filename = row['File_Name']
    variable_name = excel_filename.replace("amb_", "").replace(".png", "")

    if variable_name in globals():
        image_array = globals()[variable_name]
        modified_image = Image.fromarray(image_array)
        modified_image_path = os.path.join(output_image_directory, f"{excel_filename}")
        modified_image.save(modified_image_path)


!zip -r /content/numpy_images.zip /content/sample_data/images

       adding: content/sample_data/images/ (stored 0%)
       adding: content/sample_data/images/Prediction_drishtiGS_098.png (deflated 21%)
       adding: content/sample_data/images/Prediction_drishtiGS_092.png (deflated 25%)
       adding: content/sample_data/images/Prediction_drishtiGS_096.png (deflated 24%)
       adding: content/sample_data/images/Prediction_drishtiGS_094.png (deflated 26%)
       adding: content/sample_data/images/Prediction_drishtiGS_099.png (deflated 22%)
```

```
  adding: content/sample_data/images/Prediction_drishtiGS_101.png (deflated 21%)
  adding: content/sample_data/images/Prediction_drishtiGS_093.png (deflated 23%)
  adding: content/sample_data/images/Prediction_drishtiGS_097.png (deflated 21%)
  adding: content/sample_data/images/Prediction_drishtiGS_095.png (deflated 25%)
  adding: content/sample_data/images/Prediction_drishtiGS_091.png (deflated 25%)
  adding: content/sample_data/images/Prediction_drishtiGS_100.png (deflated 22%)
```

✓ 0s completed at 3:43 PM