**Practical No. 4**

**Aim:** Program to implement image registraction/ geo referencing and supervised classification – Land use or Land cover map preparation.

**Program Code:**

import numpy as np

import cv2

from sklearn.ensemble import RandomForestClassifier

import matplotlib.pyplot as plt

from PIL import Image

image\_path = "photo.png"

img = cv2.imread(image\_path)

if img is None:

    raise FileNotFoundError(f"Image not found at {image\_path}")

img\_rgb = cv2.cvtColor(img, cv2.COLOR\_BGR2RGB)

rows, cols, ch = img\_rgb.shape

src\_points = np.float32([[0, 0], [cols-1, 0], [0, rows-1], [cols-1, rows-1]])

dst\_points = np.float32([[50, 50], [cols-50, 20], [30, rows-50], [cols-30, rows-30]])

matrix = cv2.getPerspectiveTransform(src\_points, dst\_points)

registered\_img = cv2.warpPerspective(img\_rgb, matrix, (cols, rows))

pixels = registered\_img.reshape((-1, 3))

train\_samples = np.array([

    [0, 0, 120], [10, 10, 130],

    [20, 100, 20], [30, 120, 30],

    [120, 120, 120], [150, 150, 150]

])

train\_labels = np.array([0, 0, 1, 1, 2, 2])

clf = RandomForestClassifier(n\_estimators=50, random\_state=42)

clf.fit(train\_samples, train\_labels)

pred\_labels = clf.predict(pixels)

lulc\_map = pred\_labels.reshape((rows, cols))

class\_colors = {

    0: [0, 0, 255],

    1: [0, 255, 0],

    2: [192, 192, 192]

}

colored\_map = np.zeros((rows, cols, 3), dtype=np.uint8)

for cls, color in class\_colors.items():

    colored\_map[lulc\_map == cls] = color

fig, axes = plt.subplots(1, 3, figsize=(18, 6))

axes[0].imshow(img\_rgb)

axes[0].set\_title("Original Image")

axes[0].axis("off")

axes[1].imshow(registered\_img)

axes[1].set\_title("Registered Image")

axes[1].axis("off")

axes[2].imshow(colored\_map)

axes[2].set\_title("Land Use / Land Cover Map")

axes[2].axis("off")

plt.show()

**Output:**





