Thuật toán dùng cho Task 1:
- Xác định vị trí đặt các ảnh bằng cách khởi tạo ngẫu nhiên nhiều hình tròn cùng kích cỡ trên ảnh canvas gốc, cho đến khi số lượng hình tròn đạt đến mức phù hợp
(Với LV1 sẽ dùng canvas 500x500, 5 hình tròn có bán kính 60)
(Với LV2 sẽ dùng canvas 1000x1000, 25 hình tròn có bán kính 60)
(Với LV3 sẽ dùng canvas 2000x2000, 100 hình tròn có bán kính 60)
- Khi chọn ngẫu nhiên 1 vị trí trên canvas, chương trình kiểm tra hình tròn được đặt ở vị trí này có trùng với những hình tròn khác hay không, cho tới khi số lượng hình tròn thỏa mãn.

```
# Check if the circle overlaps with any existing circles
overlaps = False
for c in centers:
    if np.linalg.norm(c - center) < 2 * radius:
        overlaps = True
        break
```

Ở đây overlaps được phát hiện khi khoảng cách giữa 2 tâm hình tròn < 2 lần bán kính

- Khi đặt ảnh sẽ chọn ngẫu nhiên chiều quay, và chọn 1 trong 2 ảnh cat.jpg và dog.jpg để chèn vào từng hình tròn, ta xoay ảnh và dịch ảnh dùng hàm cv2.getRotationMatrix2D() và hàm cv2.warpAffine()

Thuật toán dùng cho Task 2:
-Thuật toán SSIM (Structural similarity), thuật toán gồm 5 bước :
1. Phân tách hình ảnh thành các patch:
- Các patch thường là hình vuông và được sử dụng để so sánh thông tin cấu trúc của 2 hình ảnh trong một vùng ảnh cục bộ
2. So sánh độ sáng, độ tương phản và cấu trúc mỗi patch:
- Cấu trúc là sự sắp xếp của các điểm ảnh trong patch
3. Tính toán chỉ số SSIM cho mỗi patch:
- So sánh 2 patch với vị trí tương ứng của 2 hình ảnh
4. Tính chỉ số SSIM tổng thể
5.Xác định sự tương đồng giữa hai hình ảnh

Thư viện skimage.metrics cung cấp hàm structural_similarity() trả về chỉ số SSIM giữa 2 ảnh đồng thời trả về điểm khác biệt giữa 2 ảnh

```
# Compute SSIM between the two images
(score, diff) = structural_similarity(before_gray, after_gray, full=True)
```

Sau đó chương trình xử lí khoanh vùng sự khác biệt của 2 ảnh nhờ array diff.