q4

June 9, 2024

1 Augmentation Techniques from Scratch

[53]:
```
import os
import numpy as np
import cv2
from scipy.stats import pearsonr
```

1.1 Defining the Techniques

1.1.1 Saturation

[55]:
```
def adjust_saturation(image, saturation_factor):
    hsv_image = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)
    hsv_image = hsv_image.astype(np.float32)
    h, s, v = cv2.split(hsv_image)
    s *= saturation_factor
    s = np.clip(s, 0, 255)
    hsv_image = cv2.merge([h, s, v])
    hsv_image = hsv_image.astype(np.uint8)
    adjusted_image = cv2.cvtColor(hsv_image, cv2.COLOR_HSV2BGR)

    return adjusted_image
```

1.1.2 Contrast

[56]:
```

def adjust_contrast(image, factor):
    f_image = image.astype(np.float32)
    mean = np.mean(f_image, axis=(0, 1), keepdims=True)
    adjusted_image = mean + factor * (f_image - mean)
    adjusted_image = np.clip(adjusted_image, 0, 255).astype(np.uint8)

    return adjusted_image
```

1.1.3 Brightness

[57]:
```
def adjust_brightness(image, brightness_factor):
    hsv_image = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)
    h, s, v = cv2.split(hsv_image)
    v = np.clip(v * brightness_factor, 0, 255).astype(np.uint8)
    adjusted_hsv_image = cv2.merge([h, s, v])
    adjusted_image = cv2.cvtColor(adjusted_hsv_image, cv2.COLOR_HSV2BGR)

    return adjusted_image
```

1.1.4 Flipping

[58]:
```
def flip_image(image, flip_type='horizontal'):
    if flip_type == 'horizontal':
        flipped_image = cv2.flip(image, 1)
    elif flip_type == 'vertical':
        flipped_image = cv2.flip(image, 0)
    else:
        raise ValueError("Invalid flip_type. Choose from 'horizontal' or␣
↪'vertical'.")

    return flipped_image
```

1.1.5 Rotation

[59]:
```
def rotate_image(image, angle):
    original_height, original_width = image.shape[:2]
    center = (original_width // 2, original_height // 2)
    rotation_matrix = cv2.getRotationMatrix2D(center, angle, 1.0)
    rotated_image = cv2.warpAffine(image, rotation_matrix, (original_width,␣
↪original_height))

    return rotated_image
```

1.1.6 Translation

[60]:
```
def translate_image(image, tx, ty):
    original_height, original_width = image.shape[:2]
    translation_matrix = np.float32([[1, 0, tx], [0, 1, ty]])
    translated_image = cv2.warpAffine(image, translation_matrix,␣
↪(original_width, original_height))

    return translated_image
```

1.2 Applying Transformations

[61]:
```
def save_image(image, original_name, suffix, output_dir):
    base_name, ext = os.path.splitext(original_name)
    new_name = f"{base_name}_{suffix}{ext}"
    cv2.imwrite(os.path.join(output_dir, new_name), image)

input_dir = '../images'
output_dir = 'augmented_images'

image_files = [f for f in os.listdir(input_dir) if f.endswith(('.jpeg'))]

for image_file in image_files:
    image_path = os.path.join(input_dir, image_file)
    pixels = cv2.imread(image_path)
    transformed_images = {}

    # Transformations
    transformed_images['saturated_0.5'] = adjust_saturation(pixels, 0.5)
    transformed_images['saturated_1.5'] = adjust_saturation(pixels, 1.5)
    transformed_images['flipped_horizontal'] = flip_image(pixels, 'horizontal')
    transformed_images['flipped_vertical'] = flip_image(pixels, 'vertical')
    transformed_images['contrast_0.5'] = adjust_contrast(pixels, 0.5)
    transformed_images['contrast_2.0'] = adjust_contrast(pixels, 2.0)
    transformed_images['translated_10_20'] = translate_image(pixels, 10, 20)
    transformed_images['translated_30_50'] = translate_image(pixels, 30, 50)
    transformed_images['brightened_0.5'] = adjust_brightness(pixels, 0.5)
    transformed_images['brightened_2.0'] = adjust_brightness(pixels, 2.0)
    transformed_images['rotated_90'] = rotate_image(pixels, 90)
    transformed_images['rotated_180'] = rotate_image(pixels, 180)

    for suffix, transformed_image in transformed_images.items():
        save_image(transformed_image, image_file, suffix, output_dir)
```

1.3 Comparison

To compare we compared our augmented version with the pytorch augmentations by computing the correltaions of each image.

[62]:
```
dir1 = 'augmented_images'
dir2 = '../q3/pytorch_augmented_images'

files1 = [f for f in os.listdir(dir1) if f.endswith(('.jpeg', '.png', '.jpg'))]
files2 = [f for f in os.listdir(dir2) if f.endswith(('.jpeg', '.png', '.jpg'))]

common_files = set(files1).intersection(set(files2))

def load_and_preprocess_image(image_path, size=(256, 256)):
```

```
    image = cv2.imread(image_path)
    image = cv2.resize(image, size)
    return image

correlations = {}

for fname in common_files:
    img1 = load_and_preprocess_image(os.path.join(dir1, fname))
    img2 = load_and_preprocess_image(os.path.join(dir2, fname))

    img1_flat = img1.flatten()
    img2_flat = img2.flatten()

    corr, _ = pearsonr(img1_flat, img2_flat)
    correlations[fname] = corr

with open("correlations.txt", "w") as f:
    for fname, corr in correlations.items():
        f.write(f"{fname}: {corr}\n\n")
```

1.3.1 Results

Below are the correlation values for the 'dl_book_colour' image:

buddha_colour_saturated_0.5.jpeg: 0.9922868238804489

buddha_colour_saturated_1.5.jpeg: 0.9977504243945872

buddha_colour_brightened_0.5.jpeg: 0.9907473076256899

buddha_colour_brightened_2.0.jpeg: 0.9511899341734003

buddha_colour_translated_10_20.jpeg: 0.9969093588960934

buddha_colour_translated_30_50.jpeg: 0.9980011965825497

buddha_colour_contrast_0.5.jpeg: 0.9364953558394352

buddha_colour_contrast_2.0.jpeg: 0.9483491425528454

buddha_colour_flipped_horizontal.jpeg: 0.999781662014373

buddha_colour_flipped_vertical.jpeg: 0.9997867869564403

buddha_colour_rotated_90.jpeg: 0.9983660468219826

buddha_colour_rotated_180.jpeg: 0.9784485376802984

We noted that ur brightening and contrast functions were just slightly off from the pytorch standard.

The rest of the results can be viewed below:

buddha_colour_saturated_1.5.jpeg: 0.9977504243945872

buddha_colour_brightened_2.0.jpeg: 0.9511899341734003

tajin__colour__flipped__vertical.jpeg: 0.9999094151098544

buddha__colour__brightened__0.5.jpeg: 0.9907473076256899

macbook__colour__saturated__0.5.jpeg: 0.9933166261379309

dl__book__colour__brightened__0.5.jpeg: 0.9920402746110856

lp__world__colour__contrast__2.0.jpeg: 0.9554330398169275

lp__world__colour__contrast__0.5.jpeg: 0.9378049487216751

dl__book__colour__translated__30__50.jpeg: 0.9978899324388919

dl__book__colour__rotated__180.jpeg: 0.9719812290410986

tajin__colour__saturated__1.5.jpeg: 0.9960967710313156

lp__world__colour__saturated__0.5.jpeg: 0.9856090957261219

dl__book__colour__translated__10__20.jpeg: 0.9973112609268913

macbook__colour__flipped__horizontal.jpeg: 0.9998551347813078

tajin__colour__brightened__0.5.jpeg: 0.9950185136669101

dl__book__colour__saturated__0.5.jpeg: 0.988989217010966

dl__book__colour__saturated__1.5.jpeg: 0.9950324001742756

buddha__colour__translated__10__20.jpeg: 0.9969093588960934

tajin__colour__rotated__180.jpeg: 0.9738250354949266

dl__book__colour__brightened__2.0.jpeg: 0.9928663316876847

dl__book__colour__contrast__0.5.jpeg: 0.9364434126765929

lp__world__colour__flipped__vertical.jpeg: 0.9998935633304774

tajin__colour__brightened__2.0.jpeg: 0.9942277152099921

macbook__colour__translated__10__20.jpeg: 0.9977512812022342

macbook__colour__rotated__90.jpeg: 0.9989192049920855

buddha__colour__saturated__0.5.jpeg: 0.9922868238804489

buddha__colour__contrast__0.5.jpeg: 0.9364953558394352

macbook__colour__brightened__0.5.jpeg: 0.9931289235920808

dl__book__colour__flipped__horizontal.jpeg: 0.9998589743245696

lp__world__colour__translated__30__50.jpeg: 0.99856915002376

lp__world__colour__brightened__2.0.jpeg: 0.9862050512624521

macbook__colour__saturated__1.5.jpeg: 0.9975407402527472

macbook__colour__translated__30__50.jpeg: 0.9984287964812518

lp__world__colour__flipped__horizontal.jpeg: 0.9998822279293001

buddha_colour_flipped_horizontal.jpeg: 0.9997816620143739

macbook_colour_rotated_180.jpeg: 0.9930033977523309

dl_book_colour_flipped_vertical.jpeg: 0.999872043316193

lp_world_colour_saturated_1.5.jpeg: 0.9936735919889206

tajin_colour_saturated_0.5.jpeg: 0.9920134069861877

tajin_colour_translated_30_50.jpeg: 0.9986058869769754

tajin_colour_flipped_horizontal.jpeg: 0.99990253174572

tajin_colour_contrast_0.5.jpeg: 0.9428165103496648

buddha_colour_contrast_2.0.jpeg: 0.9483491425528454

dl_book_colour_contrast_2.0.jpeg: 0.9537274104273332

lp_world_colour_rotated_180.jpeg: 0.9876592266562696

lp_world_colour_rotated_90.jpeg: 0.9963340101613245

macbook_colour_flipped_vertical.jpeg: 0.9998792867284101

tajin_colour_translated_10_20.jpeg: 0.998267272422891

dl_book_colour_rotated_90.jpeg: 0.9906991010261856

lp_world_colour_translated_10_20.jpeg: 0.9981167151546495

tajin_colour_rotated_90.jpeg: 0.992473227857493

macbook_colour_brightened_2.0.jpeg: 0.999410850776757

buddha_colour_rotated_90.jpeg: 0.9983660468219826

lp_world_colour_brightened_0.5.jpeg: 0.9945058161129994

buddha_colour_rotated_180.jpeg: 0.9784485376802984

tajin_colour_contrast_2.0.jpeg: 0.9709173749393627

macbook_colour_contrast_0.5.jpeg: 0.9477140754317523

buddha_colour_flipped_vertical.jpeg: 0.9997867869564403

macbook_colour_contrast_2.0.jpeg: 0.9342006169375852

buddha_colour_translated_30_50.jpeg: 0.9980011965825497