Francis Jay L. Villas ICE 415 BSIT – BTM 4B

ACTIVITY

Think and Write

Research on at least three digital image processing systems or applications and identify the algorithms used. Describe the system/application and write the algorithm/s.

DIP System/Application	*Algorithms used*
OpenCV (Open Source Computer Vision)	Image Filtering- Involves applying a filter or a kernel to an input image to obtain an output image. The purpose of filtering is to enhance certain features, remove noise, or extract useful information from the image.
Snapchat (Snapchat Filters)	Facial Landmark Detection - involves identifying and locating key points on a face, such as the eyes, nose, mouth, and other facial features.
Document Scanning Apps (CamScanner)	Edge Detection- involves identifying boundaries within an image where intensity changes significantly

ANALYSIS

Do some Analysis

With the algorithms you have listed in the activity, list at least three possible applications that can make use of such algorithms. Algorithms may be combined or modified.

Image Filtering:

Art Restoration Application - used in art restoration to remove noise, scratches, or imperfections from historical paintings or photographs, preserving and restoring the visual quality.

Facial Landmark Detection:

Virtual Try-On Platforms - virtual try-on platforms use facial landmarks to accurately map facial features and enable users to virtually try on cosmetics, accessories, or glasses before making a purchase.

Edge Detection:

Security Surveillance Systems - applied in video surveillance systems to detect suspicious activities or intrusions

Code for Problem Set :

```
#Francis Jay L. Villas
#Problem Set 8- M3L2
#Library
library(magick)

image <- image_read(file.choose())
print(image)
#Reference Morphs
morphology_types()
kernel_types()

image1 <- image_morphology(image,
                           method = 'HitAndMiss',
                           kernel = 'Binomial',
                           iterations = 4)
layout(t(1:2))
plot(image)
plot(image1)

image2 <- image_morphology(image,
                           method = 'Erode',
                           kernel = 'Diamond',
                           iterations = 4)
layout(t(1:2))
plot(image)
plot(image2)
```

```
image3 <- image_morphology(image,
                           method = 'Dilate',
                           kernel = 'Square',
                           iterations = 3)
layout(t(1:2))
plot(image)
plot(image3)
image4 <- image_morphology(image,
                           method = 'Open',
                           kernel = 'Disk',
                           iterations = 2)
layout(t(1:2))
plot(image)
plot(image4)
image5 <- image_morphology(image,
                           method = 'Close',
                           kernel = 'Square',
                           iterations = 5)
layout(t(1:2))
plot(image)
plot(image5)

#Segmentation
#Method
image_use <- image_scale(image,300)
image_connect(image_use, connectivity = 8)
image_split(image_use, keep_color = TRUE)
image_fuzzycmeans(image_use, min_pixels = 1, smoothing = 1.5)
```

```
image_segment <- image_resize(image_use, "400x400")
image_segment <- image_quantize(image_segment, 8)
layers <- image_split(image_segment)
image_flatten(layers)
image_lat(image_segment, geometry = "10x10+5%")
```

Screen Capture: