Procesamiento de imágenes (pre TP1)

Víctor A. Bettachini
Datamining en ciencia y tecnología 2023
Especialización en Explotación de Datos y Descubrimiento del Conocimiento
bettachini@gmail.com

Resumen

Cuca.

1. Introducción

Cuca

2. Materiales y métodos

Datos 210 imágenes de flores acompañados de un listado de las correspondientes especies dentro de una variedad de 10. Las imágenes en formato png tienen una dimensión de 128 x 128 píxeles con tres canales de color. El conjunto se descargó de una fuente pública [1].

Recurso informático Un cuaderno (notebook) Jupyter provisto por los docentes en el sitio web denominado "Campus" [2] es la plantilla donde se escribió código en lenguaje Python. Este explotó funciones de las bibliotecas OpenCV (`cv2`) y Clustimage para el trabajo con imágenes.

3. Resultados

3.1. Preprocesamiento de los datos

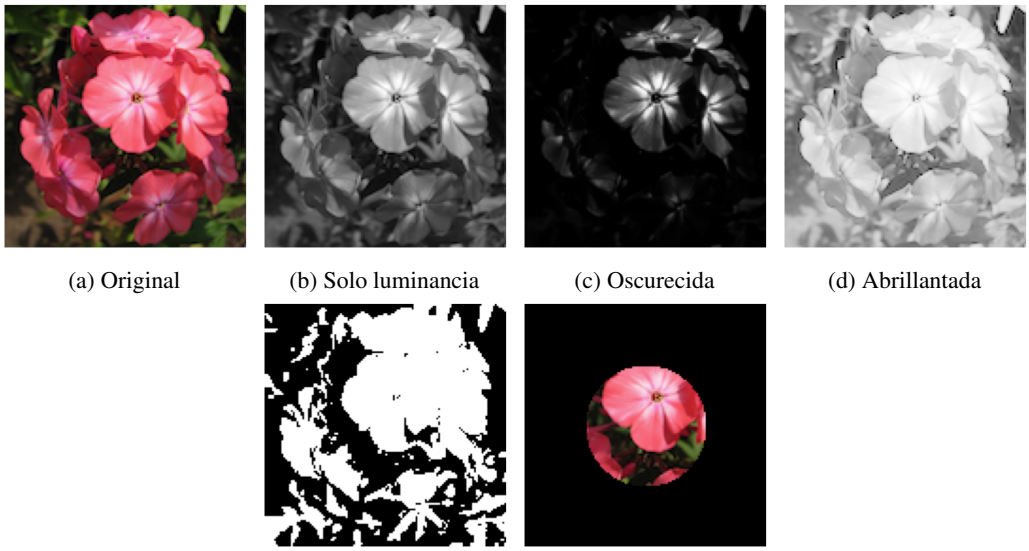
Carga del conjunto de datos La función `cv2.imread` carga los canales es azúl, verde, rojo (BGR: blue, green, red). Invertiendo la carga en un array (arreglo de la biblioteca numpy) se los ordena como RGB con la sentencia `cv2.imread(path[0])[...,::-1]`, en este caso para la primer imagen en el directorio al que apunta `path`. La figura 1a muestra en colores naturales la imagen generada a partir del array por la función `imshow` de la biblioteca matplotlib.

3.2. Manipulación de datos

Escala de grises Se utilizó la combinación lineal que preserva la luminancia perceptual de la codificación de color sRGB de la Commission Internationale de l'éclairage en 1931 según el consorcio W3 [3] $Y_{\mathrm{lineal}} = 0{,}2126 R_{\mathrm{lineal}} + 0{,}7152 G_{\mathrm{lineal}} + 0{,}0722 B_{\mathrm{lineal}}$. La figura 1b muestra la luminancia obtenida en una escala lineal de grises.

Brillo En la documentación de OpenCV se indica que el ajuste de contraste y brillo se realiza con una función lineal [4] $Y_{\mathrm{linealfinal}} = \alpha Y_{\mathrm{linealinicial}} + \beta$, donde α la ganancia controla el contraste y β el sesgo controla el brillo. Pero como lo que se muestra con imshow siempre está normalizado entre máximo y mínimo, resultando en una imagen indistingiblede la original.

Para observar una variación del brillo utilizaré una función exponencial aplicada al valor normalizado de la luminancia. Puesto que los niveles en cada color está estratificado en 8 bits al normalizar por



(a) Original (b) Solo luminancia (c) Oscurecida (d) Abrillantada

(e) Binarizada (f) Viñetado

Figura 1: Procesamientos de la primer imagen en el conjunto de datos

255 la luminancia está acotada al intervalo $[0, 1]$. Un exponente mayor que la unidad redundará en reducir la luminancia, como ejemplifica la figura 1c para un exponente de $3{,}5$. El efecto contrario se obtiene con el mismo proceder pero con un exponente como $0{,}4$ cuyo redundó en la figura 1d.

Imágen binarizada Se creó una matríz de ceros de la misma dimensión que la de luminancia. A los píxeles con valores por sobre la mediana (`statistics.median`) se les asigno el valor unidad en la nueva matriz que se muestra en la figura 1e

recorte círculo Nuevamente partiendo de la matriz de ceros, se copian píxeles de la imagen original cuando se cumple una condición con un valor de radio r del cuatro del número de columnas. La condición define un círculo centrado, pues $(i - i_0)^2 + (j - j_0)^2 < r^2$ con (i, j) e (i_0, j_0) filas y columnas del arreglo y sus valores centrales respectivamente. Con esto se obtiene la figura 1f.

3.3. Búsqueda de *features*

Análisis de componentes principales Una centena de componentes principales por imagen se obtuvieron con el método `exctract_feat` [5].

4. Discusión

Referencias

[1] Olga Belitskaya. *Flower Color Images*. 2020. URL: `https://www.kaggle.com/datasets/olgabelitskaya/flower-color-images`.

[2] Juan A. Kamienkowski. *Curso: Data Mining en Ciencia y Tecnología*. 2023. URL: `https://datamining.dc.uba.ar/campus/course/view.php?id=37`.

[3] *A Standard Default Color Space for the Internet - sRGB*. World Wide Web Consortium. 5 de nov. de 1996. URL: `https://www.w3.org/Graphics/Color/sRGB`.

[4] *OpenCV: Changing the contrast and brightness of an image!* URL: `https://docs.opencv.org/3.4/d3/dc1/tutorial_basic_linear_transform.html`.

[5] E. Taskesen. *PCA — clustimage clustimage documentation*. 2020. URL: `https://erdogant.github.io/clustimage/pages/html/Feature%20Extraction.html`.

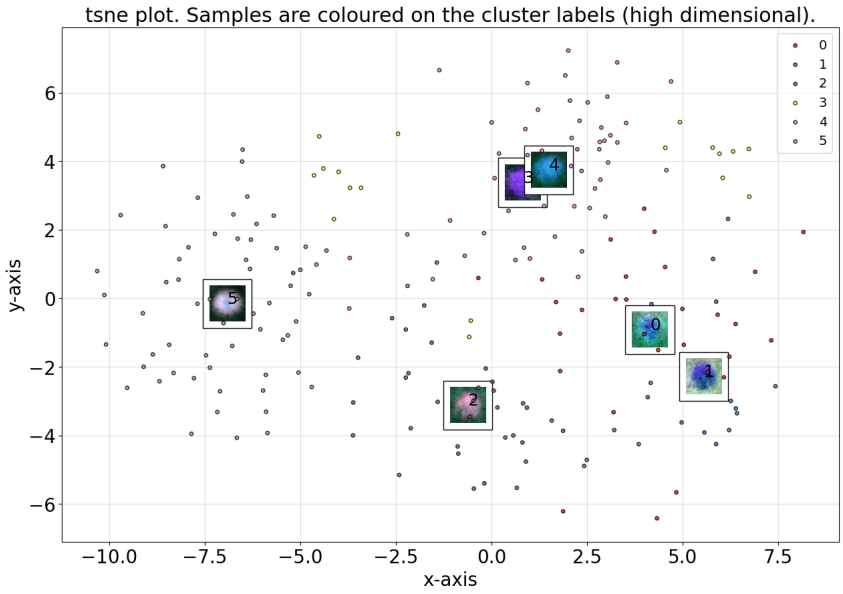


Figura 2: Ubicación de cada imágen en componentes principales(tsne plot)