|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | | **PRÁCTICA DE LABORATORIO** | |
|  | | | | |
| **CARRERA**:Computacion | | | | **ASIGNATURA**:VISIÓN POR COMPUTADOR |
| **NRO. PRÁCTICA**: | 3.1 | **TÍTULO PRÁCTICA**: [Boletín de Prácticas 3.1. Identificación de íconos empleando momentos de Zernike y clasificación de imágenes usando Patrones Binarios Locales (LBP).](https://avac.ups.edu.ec/presencial60/mod/assign/view.php?id=124454) | | |
| **OBJETIVO ALCANZADO:**  **Reforzar los conocimientos adquiridos en clase sobre la aplicación de las técnicas de Momentos de Zernike y los**  **Patrones Binarios Locales (LBP) para tareas de identificación y clasificación de imágenes, respectivamente.** | | | | |
| **ACTIVIDADES DESARROLLADAS** | | | | |
| **1.Parte 1: Desarrollar un programa que permita comparar los momentos de Zernike de imágenes del dataset “Fashion-Mnist”. Para ello deberá tomar en cuenta lo sigiuente (ver Ilustración 2):**  **Se escoge 4 categorias, las cuales son: Abrigo, Botin, Pantalon y Zapatillas, cada una de las categorias centa con 1000 imagenes divididas en dos subcarpetas test con 200 imagenes y train con 800 imagenes, Como se aprecia en la siguente imagen.**          **1.1.Código:**  **La realizacion de la parte se divide en 3 Stript: 1 La realizacion del calculos y comparacion de los momentos de zernike de las categorias antes mencionadas.**  **Se realiaza un metodo el cual hace el calculo de los momentos de zernique y va calculando cuanto es la diferencia entre cada imagen, tomando encuenta las dimenciones de las imagenes..**  **Se crea la matriz de comfusion en la cual se va ha almacenar los resultados por medio de listas de las comparaciones entre las categorias y entre si. Para optimixar el espacio del informe en el archivo ejercicio3-1.py se encuentra en detalle todo el codigo..**    **Terminado todas las comparaciones de las 4 categorias se guarda los resultados en un .csv.**  **Para la presentacion de resultados se crea dos script, los cuales tienen la funcionalidad de leer los datos guardados en el .csv y presentarlos de manera grafica, para un mejor entendimiento y a su ves guardar la imagen de las graficas.**  **Grafica % Prediccion:**    **Grafica Matriz de confision:**  **Resultados Obtenidos:**    **Como se puede observar en la grafica, el botin conincidio altamente con la zapatillas y esto se debe que en las imagenes tienen un alto parecido a comparacion de otras categorias.**  **En la tabla de prediccion podemos apreciar y verificar la informacion antes mencionada, la cual nos indica que el 11% en la prediccion del botin con el zapatillas, coincidieron.**  **también podemos apreciar que la predicción más limpia se encuentra en la categoría del abrigo, ya que se cuenta con el 97.50% de predicción del abrigo.**  **Para finalizar se muestra el tiempo obtenido del proceso de calculo de los momentos de Zernike y almacenamiento de la informacion con 1000 imagenes cada categoria.** | | | | |
| **2.Parte 2**  **Desarrollar un programa que permita clasificar una región de interés de una imagen, dada su**  **textura. Para ello deberá tomar en cuenta lo siguiente (ver Ilustración 2):**   1. **Programar un método que permita convertir una imagen de un espacio de color en el espacio CIELab.**   Se crea la clase imagen la cual contiene la imagen original y la imagen en su espacio de color CIELab. Además de los canales correspondientes a la misma.    Fig1. Declaración de la clase Imagen con sus atributos y métodos    Fig2. Definición de los métodos de la clase Imagen.   1. **Programar un método que dada una imagen o región de interés permita calcular el descriptor LBP. Con ello, deberá almacenar el histograma en un archivo o base de datos.**     Fig3. Llamada a los métodos desde la clase principal.cpp    Fig4. Método para guardar en un archivo los histogramas de las imágenes    Al ejecutar se obtiene los archivos ***Clase1.txt***, ***Clase2.txt*** y ***Test.txt***    Histogramas guardados en sus archivos ***.txt***   1. **Deberá calcular el descriptor LBP para al menos 2 imágenes distintas: clase 1 y clase 2.**     Fig Imágenes ***“Clase 1”***    Fig Imágenes **“*Clase 2*”**    Fig. cálculo de distancias entre clases.    Calculo de distancias entre imágenes pertenecientes a una clase.   1. **Dado un nuevo grupo de imágenes deberá calcular la precisión de su clasificador basado en el descriptor LBP para identificar a qué clase pertenece la imagen (clase 1 o clase 2).**     Fig Nuevo conjunto de datos    Resultados de test1 vs las clases. corteza1.jpg es la más cercana    Resultados de test2 vs las clases. wow.jpg es la más cercana    Resultados de clasificación | | | | |
| **RECOMENDACIONES**:  **Se recomienda realizar la practica con pocas imagenes, ya que la realizacion de los calculos consume recursos computacionales.** | | | | |
| **Conclusiones:**  **Parte 1 Zernike:**  **Parte 2 LBP:**  Se ha logrado calcular LBP de una imagen CIELab, obteniendo así su histograma por cada canal de dicha imagen para después calcular la distancia euclidia de las mismas y así poder clasificar con un conjunto de imágenes.  Los resultados de estas pruebas han sido altamente efectivos al momento de clasificar, prediciendo correctamente su tipo.  **Repositorio:**  [**https://github.com/JPizarro92/Practica3\_1LBP.git**](https://github.com/JPizarro92/Practica3_1LBP.git)  **Características:** (Win10, VS2019) | | | | |
|  | | | | |

***Nombre de estudiante*: Willan Mendieta, Jorge Pizarro**

**Firmas:**