Tecnológico Nacional de México

Campus Culiacán

Carrera:

Ingeniería en Sistemas Computacionales

Nombre de la materia:

Inteligencia Artificial

Tarea 1: Imágenes para la clasificación de emociones

Alumnos:

Aguilar Recio Jesús Octavio

Flores Fernandez Emily Karely

Nombre del maestro:

Zuriel Dathan Mora Felix

Grupo:

9:00 – 10:00

Introducción

Para poder hacer el detector o clasificador de emociones, primeramente, necesitamos obtener imágenes referentes a las emociones para que el programa sepa diferenciar cuales son de tristeza, alegría, angustia, etc. Es por eso por lo que en esta primera tarea se explica de donde obtuvimos dichas imágenes, como las preprocesamos y como nos van a ayudar al momento de realizar el detector ya que necesitamos asignar imágenes para entrenamiento y pruebas.

Conjunto de imágenes a utilizar:

Nosotros vamos a utilizar el conjunto de imágenes llamado FER-2013 (Facial Expression Recognition 2013) este se puede encontrar en kaggle y es uno de los conjuntos de imágenes más utilizados al momento de hacer este tipo de proyectos de detección de emociones, por ejemplo. Además, que este contiene 35, 887 imágenes en escala de grises 48x48 píxeles con 7 emociones, otra de las posibles ventajas es que ya esta estandarizado cada imagen con un tamaño e intensidad de luz en específico, este conjunto de imágenes ya esta catalogada por carpetas cada una de una emoción diferente, además que viene con imágenes para entrenamiento y pruebas.

Link del conjunto de img: https://www.kaggle.com/datasets/msambare/fer2013

Preprocesamiento:

En la cuestión del preprocesamiento las imágenes ya vienen estandarizadas con un tamaño en especifico y una intensidad de luz de la misma manera. Lo que hicimos para tener más imágenes para la hora de entrenamiento es transformar ciertas imágenes de cada emoción cambiando lo siguiente. Aumentando y disminuyendo el brillo de imágenes aleatorias de cada carpeta de emoción. También hicimos rotaciones en dos ángulos diferentes a cada imagen 15 grados y -30 grados. Estas imágenes fueron guardadas en una carpeta llamada preprocesamiento la cual contiene subcarpetas de cada emoción.

El preprocesamiento se tiene que llevar a cabo para que la detección de las emociones mostradas por cámara sea más precisa, por ejemplo, se aplica ajustes de brillo ya que al momento de mostrar la emoción por cámara no se sabe si saldrá oscuro o muy iluminado el rostro, lo mismo con la rotación ya que puede que estemos en diferentes ángulos al momento de mostrar el rostro. Entonces el preprocesamiento es obligatorio si queremos que el modelo funcione o detecte correctamente las emociones.

Otra cosa que hicimos fue hacer el archivo csv para cuando tengamos que utilizarlos más adelante en la elaboración del entrenamiento del modelo, además que aquí pues aprovechamos para etiquetar las imágenes.

Explicación de lo más relevante del código

Primero definimos las rutas de entrada de las imágenes y la ruta de salida cuando se procese las imágenes, haciendo uso de la biblioteca “os” (sirve para operaciones de sistema de archivos) llamando a la función makedirs crea la carpeta de salida si no existe.

```
# Carpeta de entrada / salida
carpeta_Entrada = "D:\\Documentos\\Octavio\\
carpeta_Salida = "D:\\Documentos\\Octavio\\T
os.makedirs(carpeta_Salida, exist_ok=True)
```

Después creamos una lista de categorías de las emociones especificadas en la tarea. Y con la misma función makedirs creamos subcarpetas para cada emoción en la carpeta de salida.

```
# Estructura de carpetas para emociones
emociones = ["angry", "distress", "happy", "sad", "surprise"]
for emocion in emociones:
    os.makedirs(os.path.join(carpeta_Salida, emocion), exist_ok=True)
```

Definimos dos funciones (ajusteBrillo y rotarImagen) para la cuestión del preprocesamiento, primero ajusteBrillo multiplica todos los valores de pixeles por el factor, también con la función np.clip asegura que los valores estén entre 0-255 y la otra función astype convierte a formato de imagen valido 8 bit. Después tenemos la otra función rotarImagen la cual obtiene dimensiones height y width y calcula el centro, creamos una matriz de rotación con la función getRotationMatrix2D de la librería “cv2” que sirve para preprocesamiento de imágenes y por último aplicamos la rotación con warpAffine usando interpolación lineal y replicando bordes.

```
# ajustar brillo y rotar imagenes
def ajusteBrillo(img, factor):
    return np.clip(img * factor, 0, 255).astype(np.uint8)

def rotarImagen(img, angulo):
    (h, w) = img.shape[:2]
    centro = (w // 2, h // 2)
    M = cv2.getRotationMatrix2D(centro, angulo, 1.0)
    giro = cv2.warpAffine(img, M, (w, h), flags=cv2.INTER_LINEAR, borderMode=cv2.BORDER_REPLICATE)
    return giro
```

Después creamos el archivo csv "etiquetas_preprocesadas" para registrar las transformaciones, con las columnas: imagen, etiqueta, conjunto y transformación. Despues hacemos la búsqueda y división de las imágenes. Con la función glob encontramos todas las imágenes jpg en subcarpetas y haciendo uso de la librería sklearn.model_selection hacemos uso de la función de entrenamiento y dividimos las imágenes en 80% entrenamiento, 15% validación y 5% de pruebas.

```
todasImagenes = glob(os.path.join(carpeta_Entrada, "**/*.jpg"), recursive=True)

# train 80 y temp 20
train_images, temp_images = train_test_split(todasImagenes, test_size=0.2, random_state=42)

# temp 15 y test 5
val_images, test_images = train_test_split(temp_images, test_size=0.25, random_state=42)
```

Despues tenemos a la función procesar_conjunto la cual lee cada imagen en escala de grises y si no puede leer la imagen, pasa a la siguiente, también aquí mismo se redimensiona a 48x48 píxeles para llevar el tamaño estándar para este tipo de modelos. Y también se encarga de extraer la emoción ósea nombre de la carpeta y el nombre del archivo.

```
def procesar_conjunto(imagenes, conjunto):
    for rutaImg in imagenes:
        img = cv2.imread(rutaImg, cv2.IMREAD_GRAYSCALE)
        if img is None:
            continue

        # tamaño para emociones 48x48
        img = cv2.resize(img, (48, 48))

        emocion = os.path.basename(os.path.dirname(rutaImg))
        img_name = os.path.basename(rutaImg)
```

Si el conjunto es train entonces creamos la lista con 5 transformaciones imagen original, ajusteBrillo oscurecida, ajusteBrillo aclarada y dos tipos de rotación, para el entrenamiento aplicamos estas transformaciones y para las validaciones solo guardamos la imagen original.

```
if conjunto == 'train':
    transformaciones = [
        (img, f"original_{img_name}", "original"),
        (ajusteBrillo(img, 0.6), f"dark_{img_name}", "brillo_reducido"),
        (ajusteBrillo(img, 1.4), f"bright_{img_name}", "brillo_aumentado"),
        (rotarImagen(img, 15), f"rot15_{img_name}", "rotacion_15"),
        (rotarImagen(img, -15), f"rotneg15_{img_name}", "rotacion_-15")
    ]
else:
    transformaciones = [(img, f"original_{img_name}", "original")]
```

Y al último guarda cada versión transformada en la carpeta correspondiente y registra los datos en el CSV.

Link drive con las carpetas imágenes y el archivo csv:

https://drive.google.com/file/d/1al0v9K729qaDkKiQAOgmo6tM1Q68npZU/view?usp=sharing