

Antonio Muñoz Barrientos

26 de febrero de 2024

Machine learning y Deep learning

Examen Parcial 1: Machine\_Learning Aplicado al Reconocimiento de Firmas.

**Etapa 1: Cheques firmados a mano**

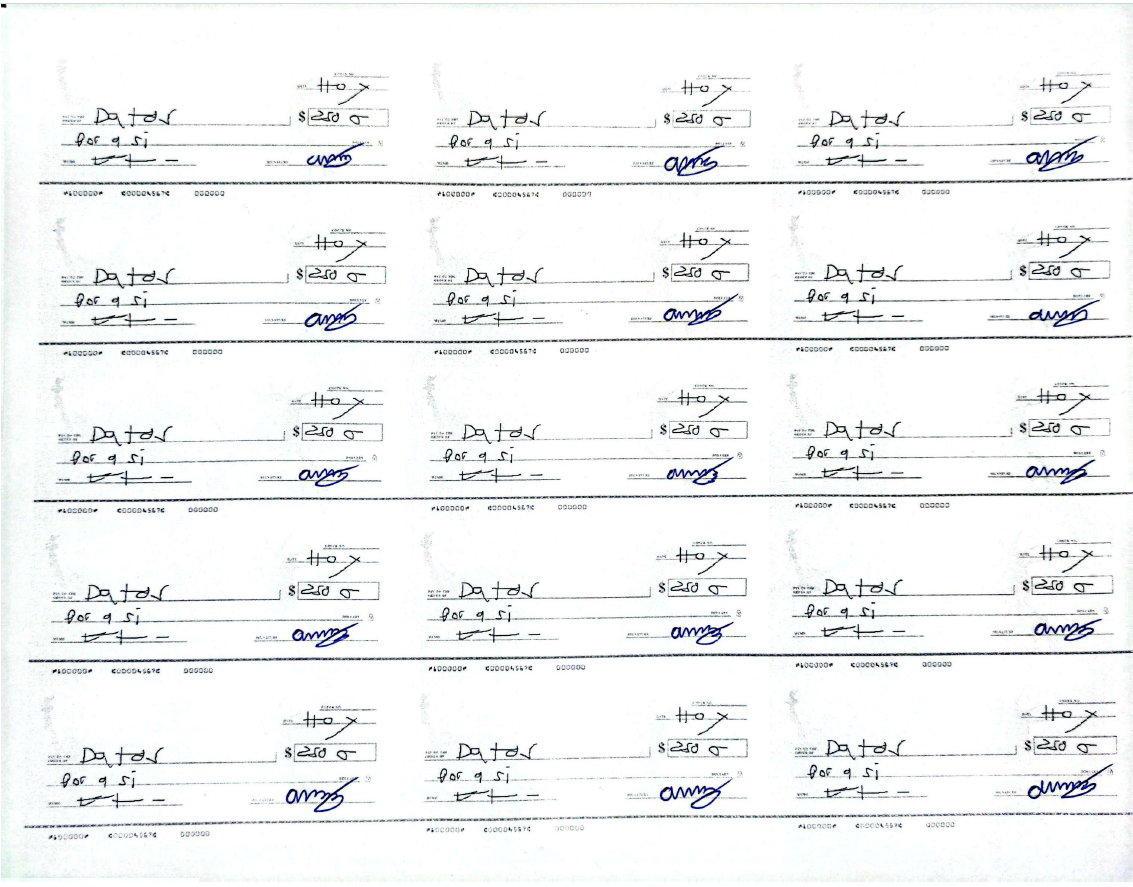
****

Ilustración Cheques firmados a mano

**Etapa 2: Algoritmo de Segmentación**

1. Leer la imagen y convertirla a un arreglo de valores
2. Se divide en 4 para extraer solo el cuadrante donde se encuentre la firma
3. Binarizamos la imagen
4. Extraemos todos los contornos en la imagen binaria
5. Ordenamos los contornos de mayor a menor por cantidad de contenido
6. Asignar media\_anterior = False
7. Asignar median\_index = 7
8. Por cada contorno en contornos:
   1. Obtenemos el cuadrado del contorno (x,y,w,h)
   2. Si x <= 3 o y <= 3 pasar al siguiente contorno
   3. Si media\_anterior es False:
      1. Calcular la media de los valores (x,y,w,h)
      2. Actualizar media\_anterior = media
      3. Actualizar valores del rectángulo x,y,w,h
      4. Dibujar los contornos obtenidos
   4. Si media\_anterior es True:
      1. Calcular la media de los valores (x,y,w,h)
      2. Si media – media\_anterior <= median\_index:
         1. xmin = x\_anterior
         2. ymin = y\_anterior
         3. xmax = x
         4. ymax = y
         5. h\_actual = h
         6. w\_actual = w
         7. Dibujar los contornos obtenidos
         8. Media\_anterior = False
      3. Si no
         1. Obtener las coordenadas del rectángulo de la firma (ymin, y\_max + h\_actual), (xmin, x\_max + w\_actual)
         2. Fin recorrer contornos
9. Recortar la firma con las coordenadas del cuadrado obtenido
10. Obtener la imagen con los contornos obtenidos.

**Base de datos de firmas extraídas**

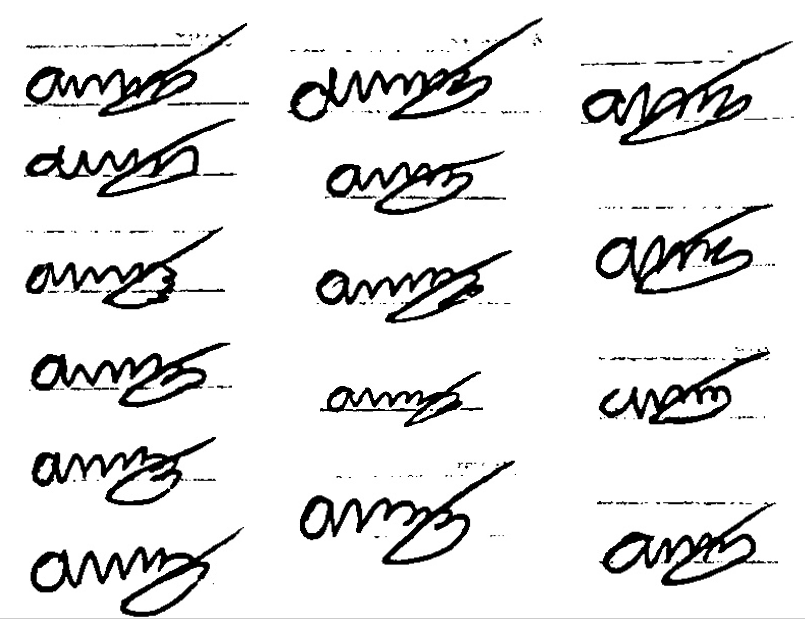
****

Ilustración : Firmas extraídas

**Etapa 3: Creación de la base de datos**

Primero debemos escalar las firmas al mismo tamaño para obtener mejores resultados, en este caso las transformaremos a imágenes de 300 x 300.

Con el uso de morfología matemática logramos obtener variables que nos permitan describir la firma en términos de datos. Primero sacamos el área y el perímetro de la firma utilizando un proceso de erosión y dilatación (opening), para posteriormente obtener el contorno. Con la suma de los valores del contorno tenemos el área y con la longitud del arreglo del contorno tenemos el perímetro.

Diagrama

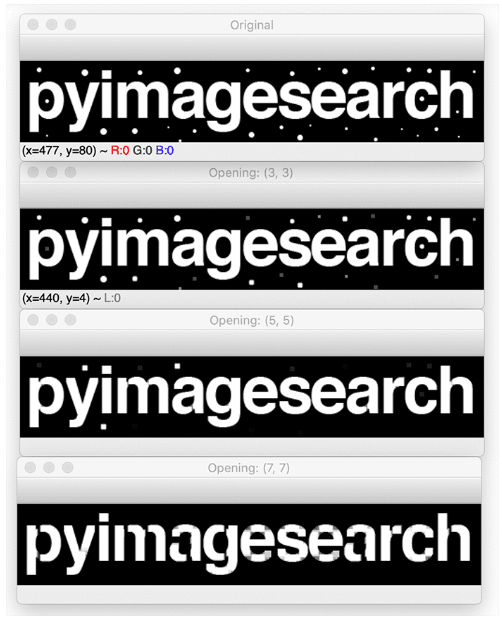
Descripción generada automáticamente con confianza media

Ilustración : Firma con opening

Ilustración : Imagen con opening

Luego a cada imagen le aplicamos una máscara (kernel) de 3 diferentes estructuras, un círculo, un rectángulo y una línea diagonal. Todos los pixeles que tenga los mismos bits activados entre la máscara y la imagen se suman. En total tenemos 13 estructuras diferentes. Esto nos genera el siguiente dataset:

Texto, Calendario

Descripción generada automáticamente con confianza media

Ilustración : Data set de firmas

Para crear más muestras correctas de nuestra firma (datos sintéticos positivos) calculamos la media de cada columna. Con un valor aleatorio determinamos si al valor de la media le sumamos, restamos o no hacemos nada con la desviación estándar. Gracias a esta forma aleatoria podemos generar diferentes muestras pero que sigan en el mismo rango de valores. También le asignamos el valor de válido 1 para saber que si es nuestra firma. Generamos 45 datos sintéticos para obtener 60 muestras de nuestra firma.

Texto

Descripción generada automáticamente

Ilustración : Datos sintéticos

Ahora generamos datos sintéticos negativos con valores aleatorios que no se crucen con nuestros valores correctos, esto para evitar que haya confusiones. Asignamos el valor de válido como 0 para identificar que no es nuestra firma. Generamos 60 para obtener 60 muestras no correctas de nuestra firma.

Tabla

Descripción generada automáticamente con confianza baja

Ilustración : Datos sintéticos negativos

Texto

Descripción generada automáticamenteJuntamos ambas muestras y obtenemos nuestro data set terminado.

Ilustración : Dataset completo

**Etapa 4: Entrenamiento de la Red neuronal**

Primero es necesario escalar los datos utilizados para el modelo. La red neuronal procesa mucho mejor los datos normalizados. Utilizamos la clase StandarScale() para normalizar los datos.

Pantalla de computadora con letras

Descripción generada automáticamente con confianza media

Ilustración : Datos normalizados

Para entrenar la red neuronal dividimos el conjunto de entrenamiento 70% y de prueba 30%. Instanciamos la red neuronal con 101 capas ocultas y el modelo de activación de Relu. Al final obtenemos la matriz de confusión, la precisión y el error globales. La variable objetivo es la columna Valido (1 si es mi firma, 0 no es mi firma).

Texto

Descripción generada automáticamente

Ilustración : Precisión del modelo

**Conclusiones**

Gracias a técnicas de procesamiento de imágenes como la binarización y la búsqueda de contornos podemos extraer de cualquier documento una firma. Es importante que debemos tratar de obtener todo el contenido de la firma incluso si esta es de varias partes, gracias a la función construida podemos obtener varias partes de las firmas.

Con el uso de estructuras morfológicas podemos obtener un conjunto de estructuras y características para transformar la firma en datos que el modelo pudiera leer. Luego dividimos el conjunto de datos obtenido para el entrenamiento y prueba del modelo. Se obtuvo una precisión del 100%.

**Referencias**

* CV2: Guía maestra OpenCV para desarrolladores de Python: <https://konfuzio.com/es/cv2/>
* OpenCV-Python Tutorials: <https://docs.opencv.org/4.x/d6/d00/tutorial_py_root.html>

- OpenCV Resize image using cv2.resize(): https://www.tutorialkart.com/opencv/python/opencv-python-resize-image/#gsc.tab=0

- Morphological Transformations: https://docs.opencv.org/4.x/d9/d61/tutorial\_py\_morphological\_ops.html

- OpenCV Morphological Operations: https://pyimagesearch.com/2021/04/28/opencv-morphological-operations/

- sklearn.neural\_network.MLPClassifier: https://scikit-learn.org/stable/modules/generated/sklearn.neural\_network.MLPClassifier.html