
RECONOCIMIENTO DE OPERACIONES MATEMATICAS

BRAYAN STEVEN MARTINEZ VANEGAS

¹Información de los autores: Se deben incluir el nombre completo de cada autor, su título académico más alto, cargo, compañía o universidad, y correo electrónico del autor de correspondencia. Si los autores pertenecen a la misma institución se puede utilizar sólo una información institucional que abarque todos los autores.

RESUMEN

El objetivo es desarrollar un programa con la capacidad de extraer números y operadores matemáticos de una imagen o video en tiempo real. Además, determinar si hay una expresión matemática para darle solución y visualizarla. Para conseguir imágenes apropiadas de cada símbolo, se usan algoritmos de procesamiento de imagen digital, también, se usa una base de datos con 1269 muestras de dígitos y símbolos para entrenar un modelo de machine learning que permite clasificar una nueva entrada, siendo esta una imagen que contiene un carácter desconocido. Como resultado, el programa retorna en imágenes individuales cada carácter presente en una imagen ingresada, también, el resultado de una expresión matemática si esta existe en dicha imagen. Adicionalmente, es posible entrenar de forma manual por medio de una interfaz gráfica, un nuevo modelo lineal a partir de un conjunto de imágenes, pudiendo ser utilizado posteriormente. Se concluye que la dificultad para reducir el ruido en imágenes es notoria debido a las infinitas condiciones que se pueden presentar.

Palabras clave: reconocimiento

RECOGNITION OF MATHEMATICAL OPERATIONS

ABSTRACT

The objective is to develop a program with the ability to extract numbers and mathematical operators from an image or video in real time. Also, determine if there is a mathematical expression to solve it and visualize it. To obtain appropriate images of each symbol, digital image processing algorithms are used, also, a database with 1269 samples of digits and symbols is used to train a machine learning model that allows to classify a new entry, this being an image that contains an unknown character. As a result, the program returns in individual images each character present in an entered image, also, the result of a mathematical expression if it exists in said image. Additionally, it is possible to manually train, through a graphical interface, a new linear model from a set of images, which can then be used. It is concluded that the difficulty to reduce the noise in images is notorious due to the infinite conditions that may occur.

Keywords: recognition.

1. INTRODUCCIÓN

Cuando se refiere a OCR (Reconocimiento Óptico de Caracteres), se habla de hallar con la ayuda de la computación, unos objetos específicos: números, letras, o símbolos en general en una imagen. La manipulación de estos símbolos o caracteres depende de la aplicación: detección de placas vehiculares, digitalización de textos o resolución de operaciones matemáticas como es el caso.

Se acude a un campo de la 'inteligencia artificial', la 'visión artificial', con el fin de obtener información digital de una imagen haciendo uso de los métodos correctos, el sistema puede aprovechar esta información para cumplir diversas funciones automatizando procesos.

Las etapas para el tratamiento de imágenes son:

1. Captura: El medio digital de la realidad tomado con una cámara fotográfica, de video o con escáner
2. Preprocesamiento: A una imagen digital se le aplican diversas técnicas para mejorar atributos de la imagen, su contraste, brillo o tamaño entre otros. Se busca eliminar información sobrante.
3. Segmentación: Dividir la imagen corregida en regiones de interés usando puntos de referencia como cambios abruptos en el color.
4. Extracción de características: Los resultados de la segmentación son analizados, y se obtiene información particular de un conjunto de elementos agrupados, pudiéndose diferenciar de otros conjuntos. Esto responde al entrenamiento de un modelo que permite en base a las características, determinar a que conjunto corresponde una nueva entrada.
5. Identificación de objetos: El sistema obtiene un resultado de una nueva imagen utilizando las características extraídas de las imágenes anteriores.

Objetivo específico:

Las imágenes contienen caracteres escritos a mano, esto quiere decir que el programa debe de tener la capacidad de reconocer tales caracteres nuevos.

Hay varias formas de segmentación: binarizado, detección de límites verticales y umbralización. Se hace énfasis en el binarizado

Para reconocer los caracteres, es necesario contar con un modelo entrenado a base de imágenes previamente clasificadas, este le permite al sistema 'aproximar' el valor de una nueva imagen.

No se encontraron bases de datos con imágenes de números y símbolos matemáticos clasificados.

2. MATERIALES Y METODO

Lenguaje de programación: Python, soporta programación orientada a objetos y es multiplataforma.

Spyder IDE Python (Entorno de desarrollo integrado): Usado para compilar el código en Python y visualizar los resultados.

Librerías: Una librería es un conjunto de funciones compiladas que permiten agilizar trabajo siendo fáciles de implementar en cualquier el programa.

OpenCV (Open Source Computer Vision Library): Es una librería de software libre multiplataforma de alta eficiencia que puede tomar provecho del procesamiento en varios núcleos, enfocada en aplicaciones de tiempo real y desarrollada por Intel, basada en C y C++. Permite editar imágenes y videos (convertir a escala de grises, modificar tamaño entre muchas más funciones.)

Scikit-learn: Es una librería de software libre para 'machine learning' (aprendizaje automático) que contiene una gran variedad de algoritmos para regresión y clasificación entre otros. Puede trabajar en conjunto con otras librerías científicas de Python como Numpy.

Numpy: Librería incluida en Python para la informática científica que contiene entre muchas otras, funciones de álgebra lineal.

Tkinter: Es una librería incluida en Python para el desarrollo de GUI (Interfaz gráfica de usuario).

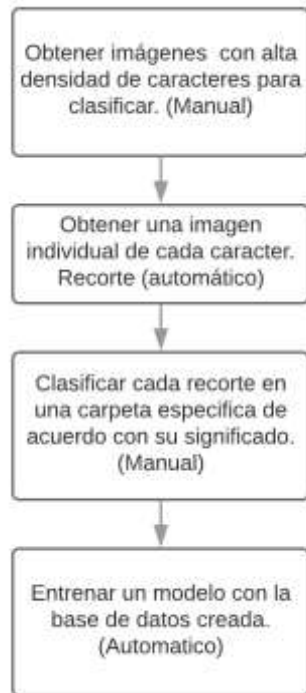
Entrenar el Sistema

Es obligatorio entrenar con un conjunto de datos previamente clasificados, gracias a ellos se puede determinar el valor de un dato nuevo desconocido.

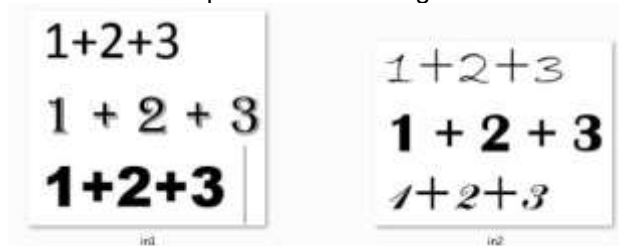
El resultado de este entrenamiento es un archivo que se puede usar en cualquier ocasión, es decir, solo se debe de entrenar una vez.

Puesto que ni los datos, ni el resultado de un entrenamiento previo fueron encontrados, se hace necesario crearlos. Esta tarea se realizará con el paquete de OpenCV y Scikit-learn.

Proceso que seguir para entrenar un sistema.



Obtener imágenes con caracteres para clasificar: Consta de recopilar un grupo de imágenes de las cuales se puedan obtener muchos caracteres diferentes con mismas características. De forma normalizada, se usan varias fuentes disponibles en los editores de texto y se guardan todas en una carpeta llamada 'imágenes'.

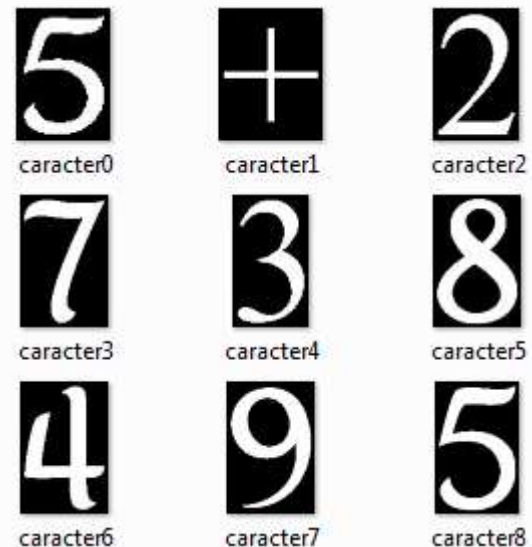


Es necesario que sean diferentes para tener una variedad, los datos nuevos no siempre son iguales a los de entrenamiento.

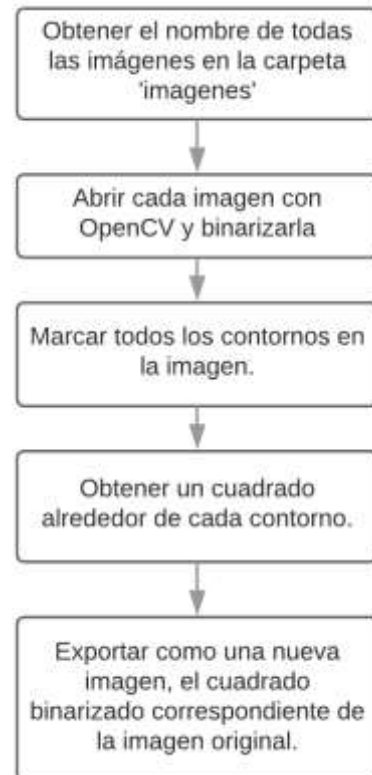
Obtener una imagen individual de cada carácter.

(Recorte): Se le llama recortes porque son segmentos de las imágenes anteriores.

El resultado final es este.



El proceso sigue:



Marcar todos los contornos en la imagen:



Obtener un cuadrado alrededor de cada contorno



Exportar en una nueva imagen, el cuadrado correspondiente de la imagen original:

El cuadrado verde donde se indica el número **DOS** será una imagen independiente.



Y el mismo proceso se aplica para todos los cuadrados hasta llegar al resultado final mostrado anteriormente.

Clasificar cada recorte en una carpeta específica de acuerdo con su significado: Este proceso, al comienzo se hace de forma manual, puesto que no se dispone de un modelo los clasifique de forma automática en su respectiva carpeta.

Se hace un grupo de carpetas que corresponden a las categorías de entrenamiento, es decir, se hace una carpeta para almacenar los numero **uno**, otra carpeta para almacenar los numero **dos**, y así para el resto de las clasificaciones que se tenga.

El nombre de las carpetas se da en códigos **ASCII**, esto facilita el reconocimiento de una carpeta por parte de Python y la reducción de condicionales. Además, hay símbolos que no se pueden asignar de nombre a un carpeta.

ASCII	Símbolo	ASCII	Símbolo
48	0	57	9
49	1	40	(
50	2	41)
51	3	42	*
52	4	43	+
53	5	45	-
54	6	47	/
55	7	120	x
56	8	121	y



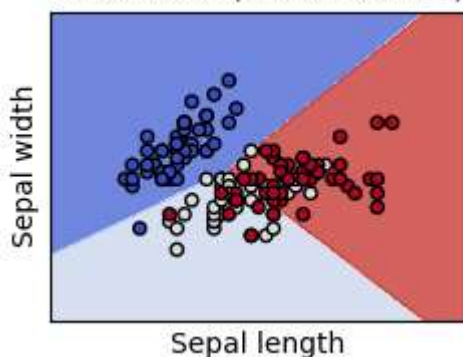
Entrenar un modelo con la base de datos creada: Es el ultimo paso de esta sección. Se usa la librería de Scikit-Learn para obtener un archivo del modelo entrenado con las imágenes anteriormente recortadas y clasificadas.

Antes de entrenar el modelo, se debe de asegurar que todas las imágenes tengan la misma dimensión, se sigue:



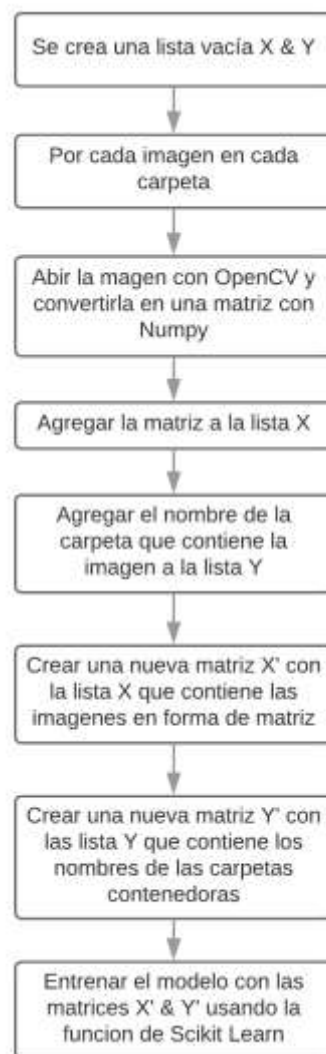
Luego de que todas las imágenes están iguales en tamaño, se puede entrenar el modelo usando SVC (Support Vector Classification). Hay varias opciones, entre ellas, una lineal.

LinearSVC (linear kernel)



Dadas las pruebas, fue esta forma de clasificación la que brindo resultados aceptables.

Usando SVC con Scikit-learn se tiene:



La función 'fit' de Scikit-learn debe de recibir algo similar a esto:

X	Y – número ASCII
Imagen de un '2' binarizada y en forma de matriz.	'50'
Imagen de un '7' binarizada y en forma de matriz.	'55'
Imagen de un '+' binarizada y en forma de matriz.	'43'
Imagen de un '2' binarizada y en forma de matriz.	'50'
Imagen de un '!' binarizada y en forma de matriz.	'40'

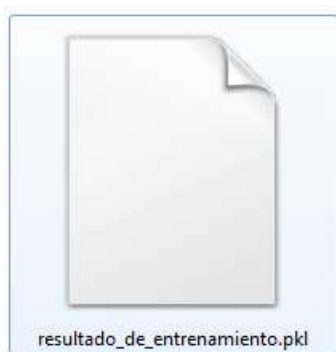
La imagen, y su respectiva clasificación, con estos datos, el modelo se puede entrenar. Es un proceso

que tarda en función del número de datos que se le entregan y la capacidad de procesamiento del computador. Con 1000 muestras tarda alrededor de 30 segundos o incluso menos.

Cuando termina el proceso, el modelo estará en un objeto de Python, para usarlo posteriormente no hace falta volver a entrenar, sino, guardar el objeto como un archivo que puede ser cargado luego. Se realiza con esta función:

```
joblib.dump(modelo, "nombre_del_archivo.pkl")
```

Retornará un archivo como el siguiente:



Que se puede cargar desde Python en forma de objeto:

```
clf = joblib.load("resultado.pkl")
```

Con esto termina la sección de entrenamiento del modelo, dejando como resultado una base de datos que se puede seguir ampliando.

Predecir

Cargado el 'modelo' como objeto, solo es necesario cargar una imagen y usar los métodos anteriores para extraer los recortes, cada uno de estos es analizado y el resultado es la suma en cadena de texto con la predicción de cada carácter.

Todos los servicios están disponibles de forma ordenada en un GUI (Grafical User Interface), esta fue desarrollada con apoyo de la librería Tkinter disponible en Python.



3. RESULTADOS

- Una base de datos con símbolos clasificados en 21 carpetas: (paréntesis, llaves, corchetes, asterisco, suma, resta, letra x, letra 'y', números desde el 0 hasta el 9).
- Un modelo entrenado con Scikit-learn a base de 1269 muestras y que puede ser usado en el futuro.
- Una GUI con manejo de errores y 3 opciones de uso: Reconocer la operación matemática de una imagen estática, entrenar un nuevo modelo de forma manual a base de imágenes nuevas y hacer el uso de una webcam en tiempo real.

4. DISCUSIÓN (O ANÁLISIS DE RESULTADOS)

La discusión debe considerar los resultados en relación con las hipótesis formuladas en la introducción y el lugar del estudio en el contexto de otros trabajos. Las secciones de Resultados y Discusión (o análisis de resultados) pueden ser combinadas.

5. CONCLUSION

Las bases de datos con números no son fáciles de encontrar en la red, hay poco material de apoyo en la misma.

Hay librerías que contienen una base datos, pero esta no puede ser actualizada fácilmente.

El uso de binarización puede quedar corto ante los otros métodos, pero sin duda, un uso inteligente de todos juntos es lo mejor.

El programa tiene debilidad con imágenes que presentan mucha suciedad o defectos de iluminación.

Para una investigación futura, se puede ampliar el programa para que tome videos e imágenes de internet y entrene constantemente con tal contenido.

6. AGRADECIMIENTOS

Al semillero de visión artificial de la Universidad EIA.

7. REFERENCIAS BIBLIOGRÁFICAS

Referencias de internet:

OpenCV. (3 de Marzo de 2018). Obtenido de Contours : Getting Started: https://docs.opencv.org/3.3.1/d4/d73/tutorial_py_contours_begin.html

OpenCV. (3 de Marzo de 2018). Obtenido de Smoothing Images: https://docs.opencv.org/3.3.0/d4/d13/tutorial_py_filtering.html

ANEXO:

Video de su funcionamiento: <https://youtu.be/drQtyEAPDIY>