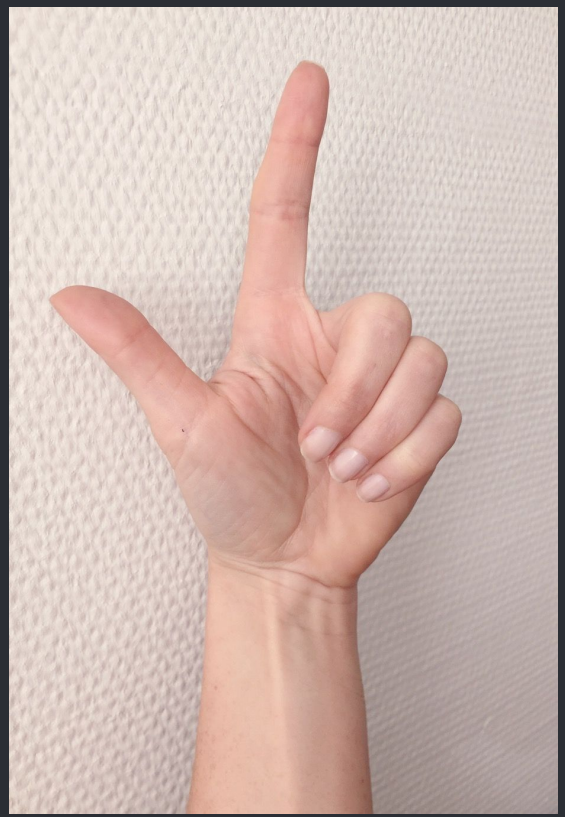
Detección de Digitos en las Manos

Proyecto de Machine Learning

Carlos Cortés

Overview

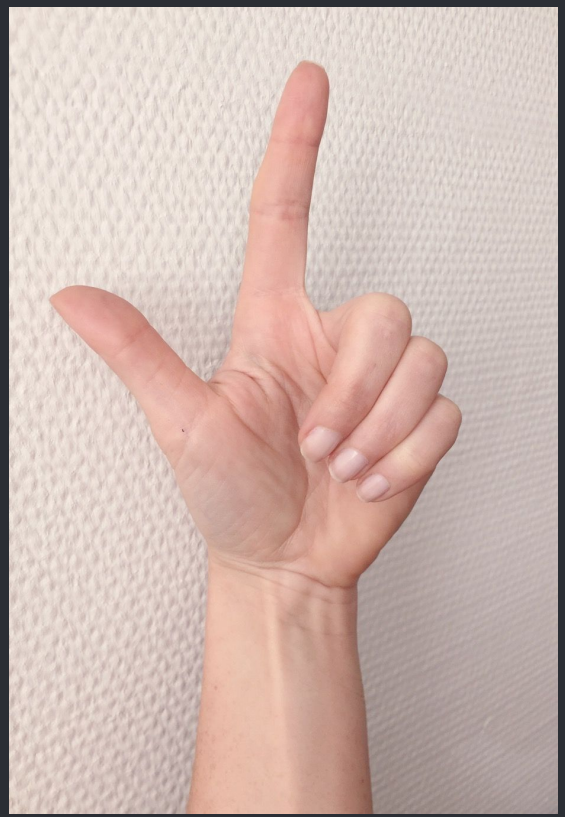
Objetivo: Clasificar imágenes de manos haciendo dígitos de lenguaje de señas.



Overview

Objetivo: Clasificar imágenes de manos haciendo dígitos de lenguaje de señas.

La idea surgió de querer reconocer el lenguaje de señas.
Así que se intentó comenzar por reconocer únicamente los diez dígitos del lenguaje de señas.



Método

Lo primordial es recordar que las imágenes son matrices.
Podemos desdoblar la matriz en un gran vector para realizar machine learning con cada imagen como instancia.
Se probó con dos métodos:

Método

Lo primordial es recordar que las imágenes son matrices.
Podemos desdoblar la matriz en un gran vector para realizar machine learning con cada imagen como instancia.
Se probó con dos métodos:

K-NN

Es muy sencillo de realizar y da buenos resultados.

Método

Lo primordial es recordar que las imágenes son matrices.
Podemos desdoblar la matriz en un gran vector para realizar machine learning con cada imagen como instancia.
Se probó con dos métodos:

K-NN

Es muy sencillo de realizar y da buenos resultados.

Regresión Logística.

Supuestamente da mejores resultados.

Preview

La base de datos de imágenes fue tomada del github de la referencia.
Consta de 2062 imágenes etiquetadas del 0-9.
Cada imagen es de 100x100 y en formato jpg (por lo que fue necesario convertirlas a escala de grises) [1].

Preview

				
0	1	2	3	4
				
5	6	7	8	9

Dependencias

- *Mlpack*.
  Biblioteca principal para métodos de Machine Learning.

Dependencias

- *Mlpack*.
  Biblioteca principal para métodos de Machine Learning.
- *Armadillo*.
  Biblioteca para matrices que Mlpack utiliza.

Dependencias

- *Mlpack*.
  Biblioteca principal para métodos de Machine Learning.
- *Armadillo*.
  Biblioteca para matrices que Mlpack utiliza.
- *OpenCV*.
  Biblioteca que utilizo para manipular imágenes y convertirlas en vectores.

Código

Visita Rápida

Kdevelop.jpg

Mejor Puntaje

Partiendo el Dataset en 1856 de entramiento y 206 de prueba.

Con k = 10 vecinos más cercanos.

K-NN

154 casos correctos.

Porcentaje = 74.75%.

Entre (68.85% , 80.65%) al 95% de confianza.

Mejor Puntaje

Partiendo el Dataset en 1856 de entramiento y 206 de prueba.

Con k = 10 vecinos más cercanos.

K-NN

154 casos correctos.

Porcentaje = 74.75%.

Entre (68.85% , 80.65%) al 95% de confianza.

Regresión Logística.

25 casos correctos.

Porcentaje = 12.13%.

Entre (7.69% , 16.57%) al 95% de confianza.

Better, Stronger

Lo siguiente a hacer para mejorar/perfeccionar el proyecto:

Better, Stronger

Lo siguiente a hacer para mejorar/perfeccionar el proyecto:

- Incluir una cámara para realizar clasificación en tiempo real.

Better, Stronger

Lo siguiente a hacer para mejorar/perfeccionar el proyecto:

- Incluir una cámara para realizar clasificación en tiempo real.
- Intentar aplicar PCA o alguna otra técnica que pudiera ayudar a aumentar el poder predictivo.

Better, Stronger

Lo siguiente a hacer para mejorar/perfeccionar el proyecto:

- Incluir una cámara para realizar clasificación en tiempo real.
- Intentar aplicar PCA o alguna otra técnica que pudiera ayudar a aumentar el poder predictivo.
- Seguir intento variantes de regresión logística (básicamente intentar redes neuronales).

Bibliography

[1] ardamavi. Sign-Language-Digits-Dataset. 2018. Recuperado de https://github.com/ardamavi/Sign-Language-Digits-Dataset