



Universidad Nacional de San Luis
Facultad de Ciencias Físico, Matemáticas y Naturales
Ingeniería Electrónica con Orientación en Sistemas Digitales

“Estimación automática de Condición Corporal en bovinos a partir de imágenes de contornos y *Machine Learning*”

Reta, Facundo Estanislao
Mg. Ing. Petrino, Ricardo

28 de Octubre 2020



Índice de contenidos

- 1 Introducción
- 2 BCS ¿Qué es?
- 3 Herramientas
- 4 Procesamiento de imágenes
 - Etapa I: Segmentación
 - Etapa II: Umbralización
 - Etapa III: Detección de bordes
 - Etapa IV: Rotación y traslación de contorno
 - Etapa V: Extracción de puntos anatómicos
- 5 *Machine Learning*
- 6 Conclusiones



Motivación

- Facilitar, agilizar una tarea cotidiana de clasificación manual de bovinos según su Condición Corporal en el ámbito agropecuario.
- Adquirir conocimientos, madurar y crecer en el área de la programación y el procesamiento digital de imágenes.
- Aportar soluciones a las necesidades del Proyecto Campo Demostrativo Conectado.



Objetivo

Proyecto:

- Desarrollar
- Evaluar
- Implementar

Base de datos de imágenes de bovinos

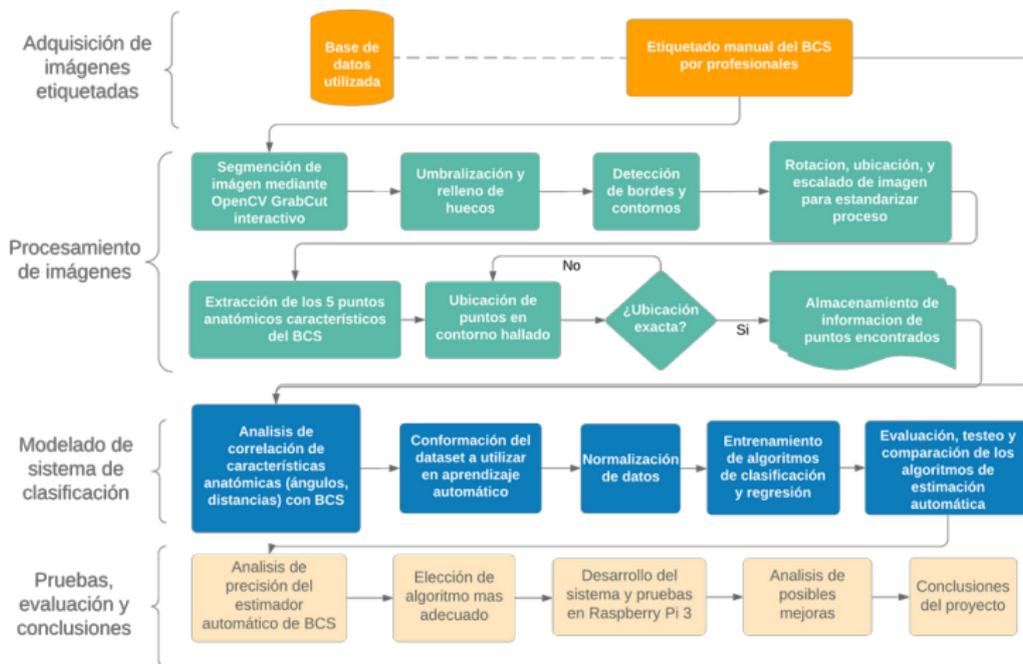


Sistema de clasificación automática del BCS

Grado de BCS



Flujo de diseño del proyecto



Condición Corporal ó BCS (*Body Condition Score*) en bovinos:

Definición

Es una medida indirecta del nivel de grasa subcutánea en el ganado bovino que se ha adoptado ampliamente para la investigación y la evaluación en el campo o para fines de gestión en granjas.

Métodos de puntuación

- Australiano (Earle, 1976)
- Británico (Mulvany, 1981)
- Neozelandés (Grainger - McGowan, 1982)
- Americano
 1. Wildman y Col. (1982)
 2. Edmonson y Col. (1989)

Procedimientos de calificación

Independientemente del sistema empleado, el evaluador estima la cantidad de grasa existente principalmente en la raíz de la cola y alrededor de la zona lumbar dando un puntaje en la escala entre 1 y 5.



Grados de BCS

A cada punto de la escala de BCS se le puede dar un significado práctico:

1 = Vaca esquelética o emaciada.

2 = Vaca flaca o delgada.

3 = Vaca normal o promedio.

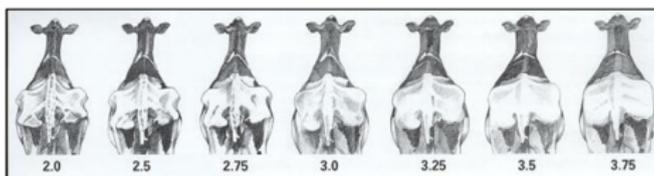
4 = Vaca gorda.

5 = Vaca obesa.



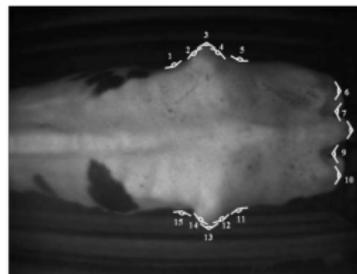
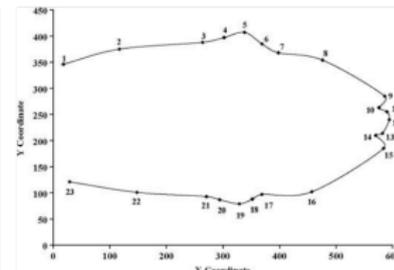
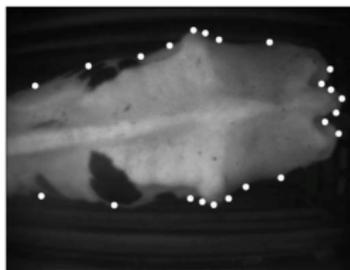
Método de Edmonson y Col. 1989

Grado de condición corporal	Vértebra en la espalda	Aspecto posterior del hueso pélvico	Aspecto lateral de la línea entre las caderas	Cavidad entre cadera y la tuberosidad isquiática	
				Aspecto posterior	Aspecto lateral
1 Subcondicionamiento severo					
2 Esqueleto obvio					
3 Buen balance de esqueleto y tejidos superficiales					
4 Esqueleto no tan obvio como tejidos superficiales					
5 Sobrecondicionamiento severo					



Características anatómicas

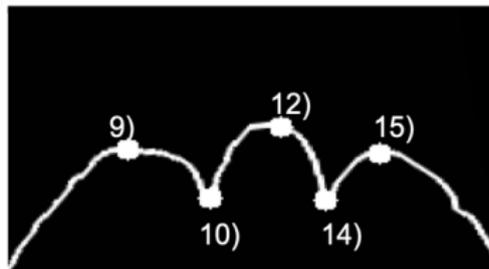
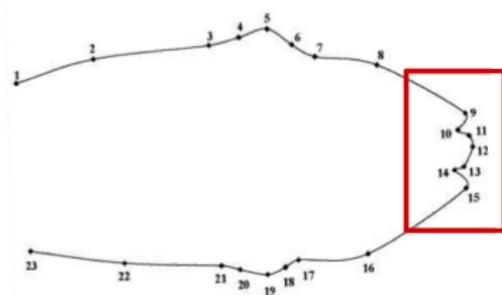
Existen 23 puntos anatómicos correspondientes a características identificables en el animal, que poseen una influencia potencial en el BCS. A través de estos puntos anatómicos se puede extraer información relevante como distancias y ángulos entre puntos los cuales poseen una correlación muy fuerte con el BCS.



Características anatómicas

En este proyecto se recurrió a ubicar sólo 5 puntos de los 23 existentes que se corresponden al cuarto trasero del animal, los cuales aportan suficiente información implícita en sus ángulos y distancias para determinar el BCS del animal.

I. Halachmi y Col. 2008

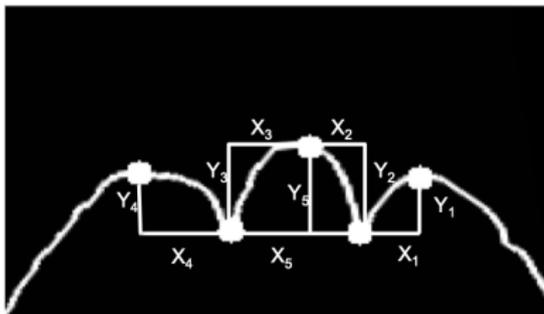


Características anatómicas

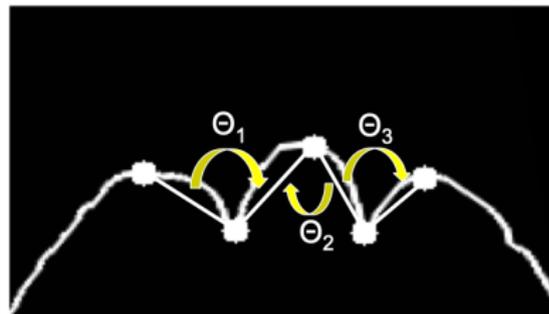
Puntos anatómicos seleccionados

- Punto 9) Pasador izquierdo
- Punto 10) Cabezal de la cola izquierda
- Punto 12) Cola
- Punto 14) Cabezal derecho de la cola
- Punto 15) Pasador derecho

Información implícita en ángulos y distancias entre puntos anatómicos:



(a)



(b)



Hardware y Software

■ Software:



■ Hardware:



Base de datos

La base de datos de BCS utilizada en este proyecto fue creada por el equipo conjunto de *Corfilac, IPLAB y Penn Veterinary Medicine*.

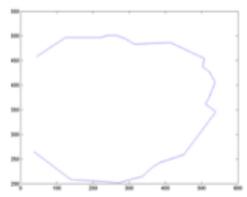
BODY CONDITION SCORE DATABASE



- Overview
- Dataset
- Publications
- Software
- People



Original Image



Shape from Anatomical Points

[Download](#) 3 3

Data BCS₁ BCS₂



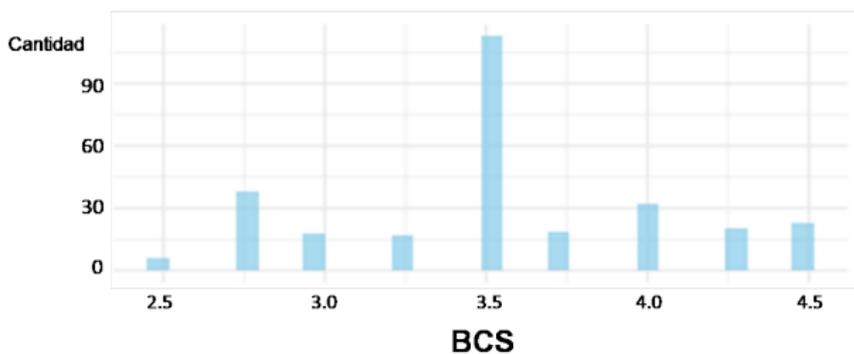
Análisis estadístico

Coordenadas x e y de cada uno de los puntos anatómicos

	P9x	P10x	P12x	P14x	P15x	P9y	P10y	P12y	P14y	P15y	BCS
1	508	502	537	510	539	454	438	406	362	346	3,00
2	542	543	571	555	582	586	557	521	498	487	3,5
3	536	535	566	546	575	565	531	506	477	457	3,5
4	560	551	602	571	604	545	503	478	453	438	4,00
5	498	504	541	525	544	485	457	425	383	359	3,00
6	575	562	594	563	593	443	402	363	351	333	4,00
...

Análisis estadístico

	BCS
Total	286
Media	3.53
SD	0.51
Min	2.5
25 %	3.25
50 %	3.5
75 %	4
Max	4.5

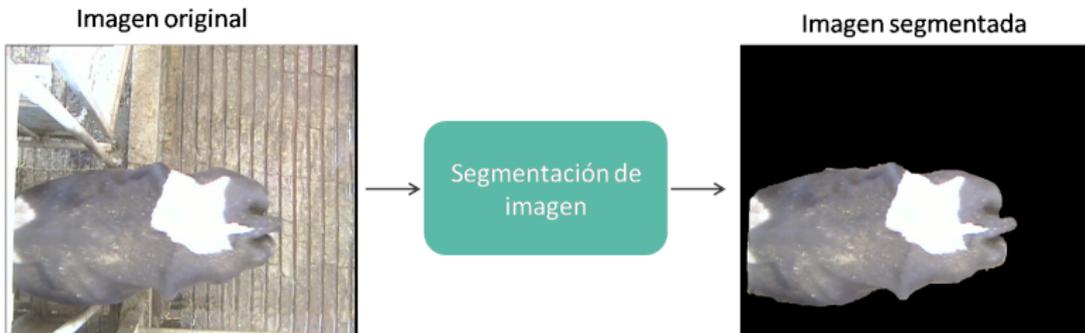


Segmentación I

Objetivo

La segmentación en el campo de la visión artificial es el proceso de dividir una imagen digital en varias partes (grupos de píxeles) u objetos.

La finalidad es el aislamiento de la zona de importancia respecto a su fondo original.



Segmentación II

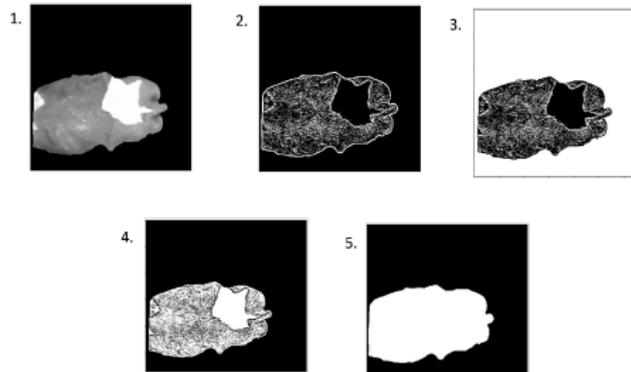
Herramientas

El algoritmo llamado *GrabCut* perteneciente a las librerías de OpenCV, es una herramienta comunmente utilizada en la segmentación de imágenes en las que automatizar este proceso se dificulta debido a características propias tales como diferencias de contrastes, brillo y posicionamiento de la zona de interés.



Umbralización de imagen

1. Imagen original en escala de grises.
cv2.adaptiveThreshold()
2. Imagen umbralizada e invertida bit a bit.
cv2.bitwise_not()
3. Imagen con huecos rellenados.
cv2.floodFill()
4. Imagen rellenada e invertida bit a bit.
cv2.bitwise_not()
5. Combinación mediante operación OR bit a bit de imágenes 4 y 2.



Algoritmo *Canny*

Canny

El método utilizado en este proyecto para la detección de bordes se basó en el algoritmo "*Canny Edge Detection*". Este proceso esta conformado por múltiples etapas:

Resumen del método

1. **Reducción de ruido:** Aplicación de filtro gaussiano 5x5.

2. **Hallar el gradiente de intensidad de la imagen:**

- *Magnitud Gradiente* $(G) = \sqrt{G_x^2 + G_y^2}$

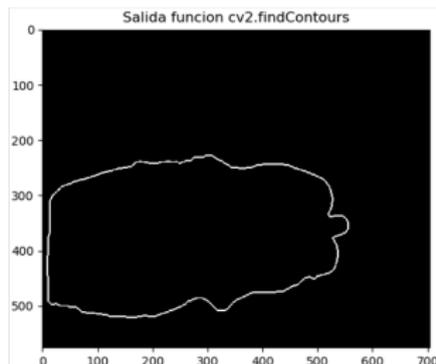
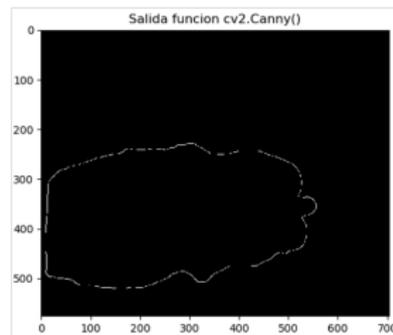
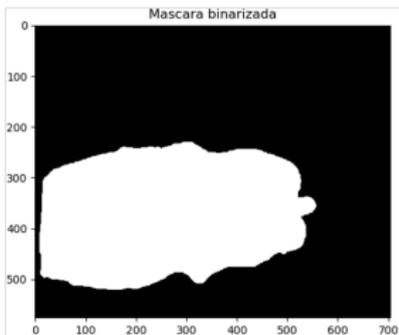
- *Dirección Gradiente* $\theta = \tan^{-1} \frac{G_x^2}{G_y^2}$

3. **Supresión no máxima:** Verificación de cada píxel en búsqueda de máximos locales en dirección del gradiente encontrado.

4. **Umbral de histéresis:** Se hizo uso de dos valores de umbral, **minVal** y **maxVal**. Cualquier borde con un gradiente de intensidad **mayor que maxVal** seguramente es borde y aquellos **debajo de minVal** seguramente no serán bordes, por lo que se descartan

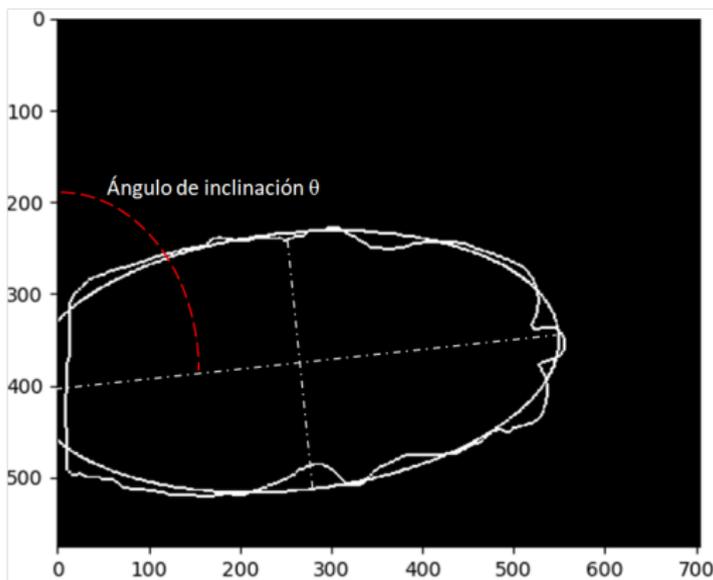


Algoritmo *Canny*: Resultado de aplicación



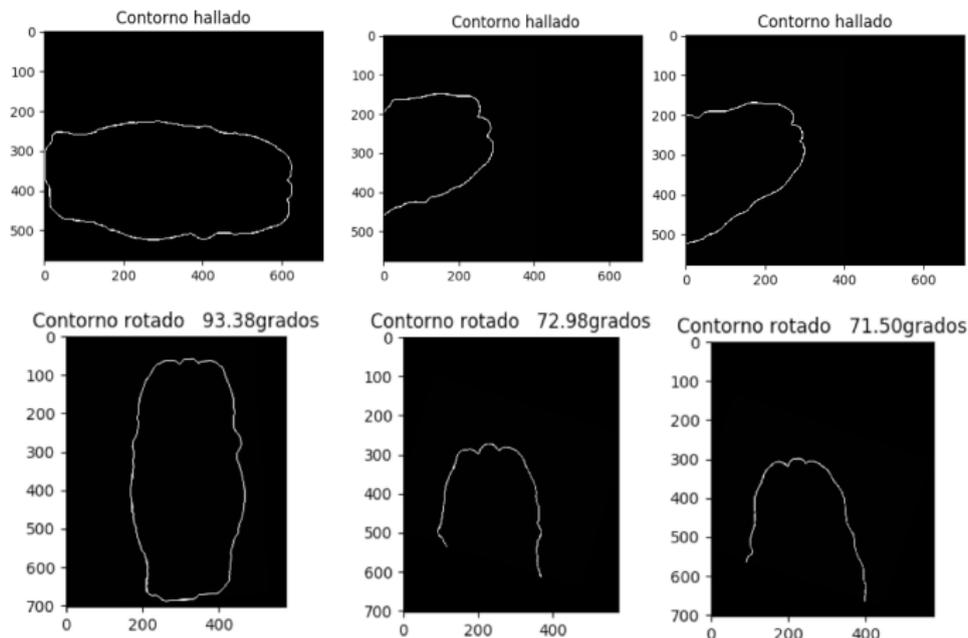
Ángulo de inclinación

Utilizando el resultado arrojado por la etapa anterior, se calcula el ángulo de inclinación θ del contorno, con el fin de estandarizar el proceso para todos los contornos a analizar:



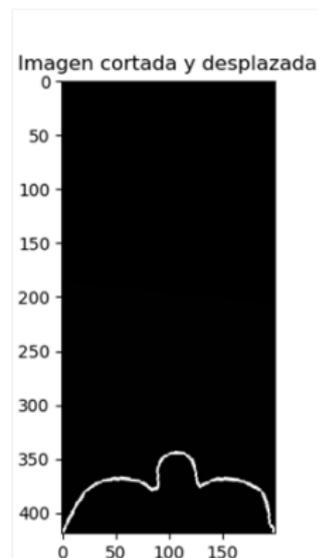
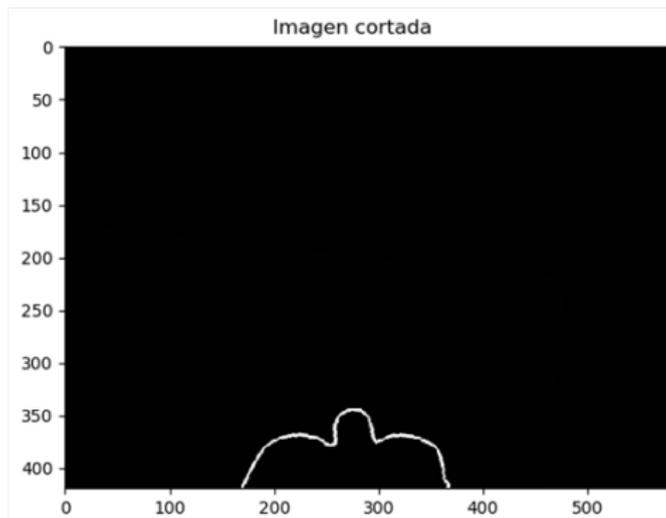
Rotación del contorno

Ya obtenido el ángulo de inclinación θ , se rota la imagen del contorno para que la misma quede totalmente vertical en el plano.



Traslado del contorno

El traslado del contorno consiste en encontrar el punto mas a la izquierda de la imagen recortada, junto al punto mas a la derecha de la misma, para luego trasladarla al origen de coordenadas con el fin de hacer mas fácil el proceso de ubicación de puntos anatómicos del animal.



Puntos anatómicos I

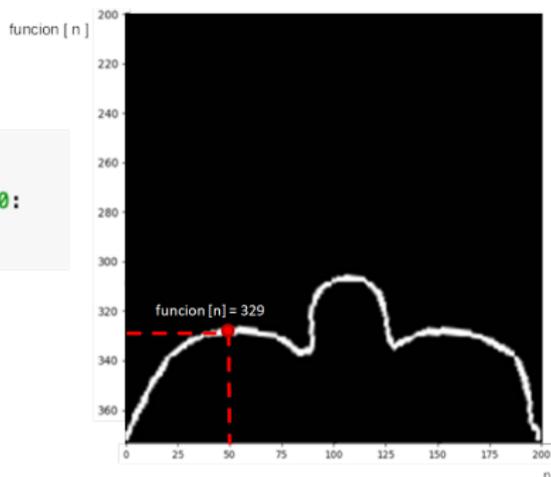
Proceso de extracción

El proceso de extracción de los 5 puntos anatómicos se basó en el análisis de valores y posiciones de un arreglo, creado a partir de la imagen de contorno obtenida en la etapa anterior.

n	0	1	2	3	4	5	6	7	8	9	10	11	12	13	~	191	192	193	194	195	196	197	198	199	200	201	202	203	204
funcion [n]	374	374	373	371	370	367	365	364	361	358	355	352	349	346	~	356	357	358	360	361	363	363	365	368	368	369	370	372	373

Código para generar el arreglo funcion []:

```
for x in range(0,w):
    for y in range(0,h):
        if img_desplazada[y,x] > 200:
            funcion[x] = y
```



Puntos anatómicos II

Análisis de vector *función*

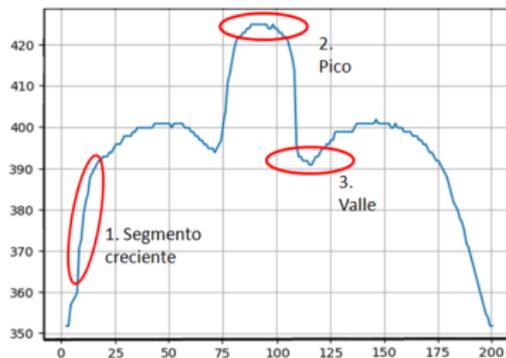
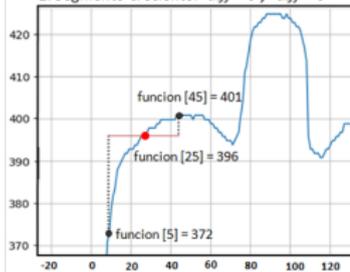
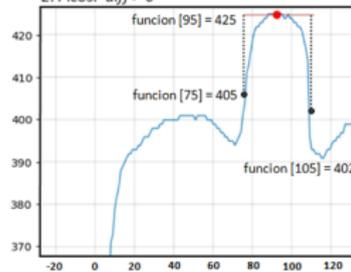
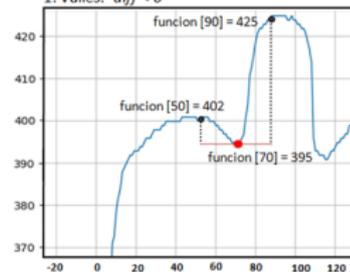
A partir de esto se analizaron cada una de las n posiciones y valores asociados del arreglo creado, con el fin de poder identificar posibles valles (**Puntos anatómicos 10 y 14**) y posibles picos (**Puntos anatómicos 9, 12, y 15**). Este análisis consistió en recorrer iterativamente el arreglo realizando comparaciones según las diferencias de cada valor de *funcion* [] con N vecinos y se asignó este resultado a una variable entera llamada *diff*:

$$diff = funcion[x] - funcion[x + i - N/2]$$

Donde x e i son las variables de iteración que permiten recorrer el arreglo y $N = 41$ elegido así mediante pruebas prácticas.



Puntos anatómicos III

1. Segmento creciente: $\text{diff} > 0$ y $\text{diff} < 0$ 2. Picos: $\text{diff} > 0$ 1. Valles: $\text{diff} < 0$ 

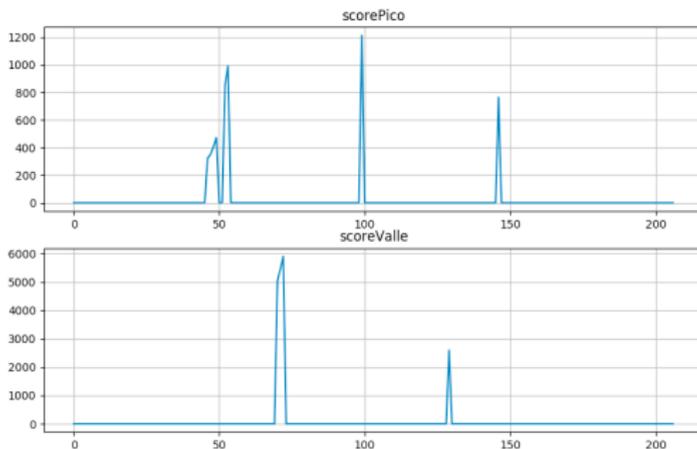
Puntos anatómicos IV

Puntuación o *score*

Se asigna una puntuación o *score* a esa posición específica del arreglo *funcion[]* la cual esta dada por el valor $diff^2$, almacenando y acumulando dicha puntuación en dos nuevos arreglos llamados **scorePico** y **scoreValle** los cuales poseen la misma dimensión que *funcion []*.

Si $diff > 0 \rightarrow$ Posible pico $\rightarrow scorePico[x] = scorePico[x] + diff^2$

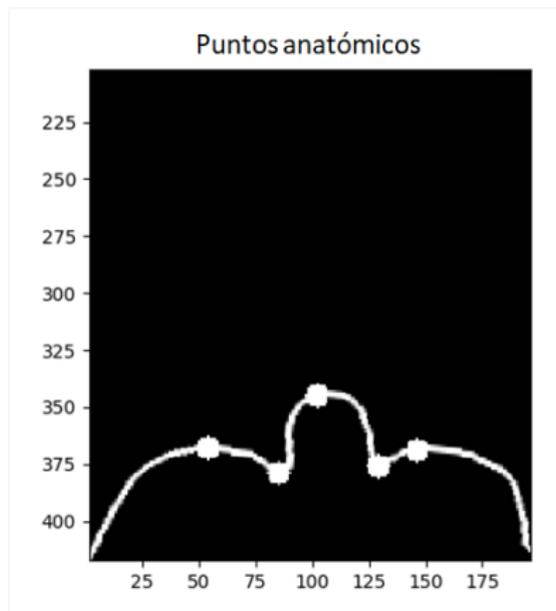
Si $diff < 0 \rightarrow$ Posible valle $\rightarrow scoreValle[x] = scoreValle[x] + diff^2$



Puntos anatómicos V

Ubicación automática de puntos anatómicos

Se ubican los puntos hallados mediante detección de máximos locales en vectores *scorePico* y *scoreValle*:



Machine Learning: Introducción

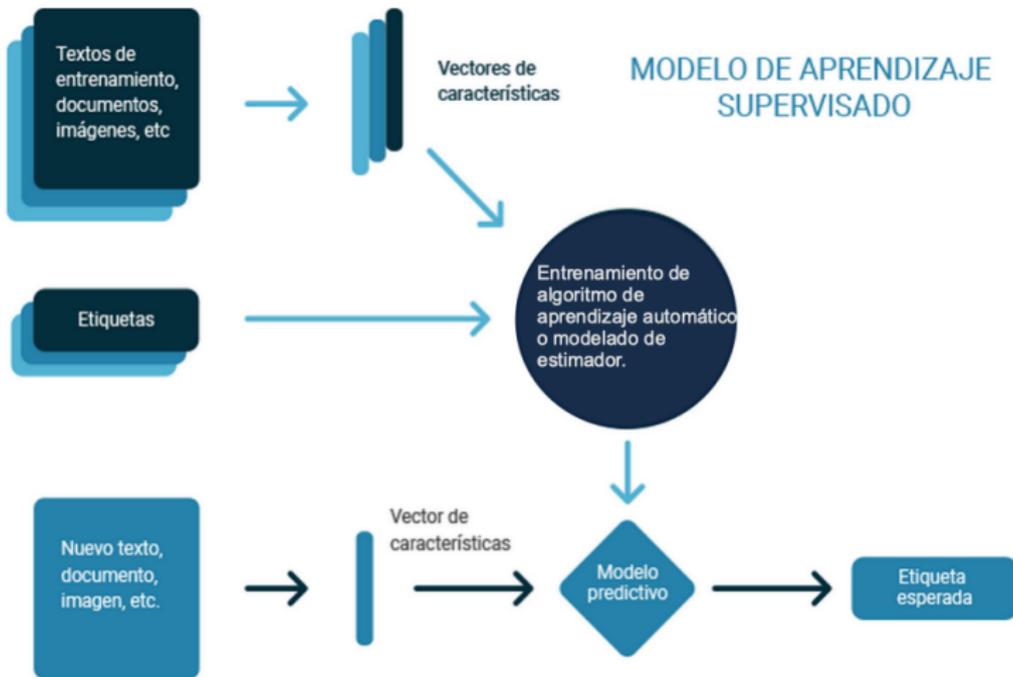
Definición

“Se dice que un programa de computadora aprende de la experiencia E con respecto a alguna clase de tareas T y medida de desempeño P , si su desempeño en tareas en T , medido por P , mejora con la experiencia E .”

Tom M. Mitchell - Machine Learning 1997



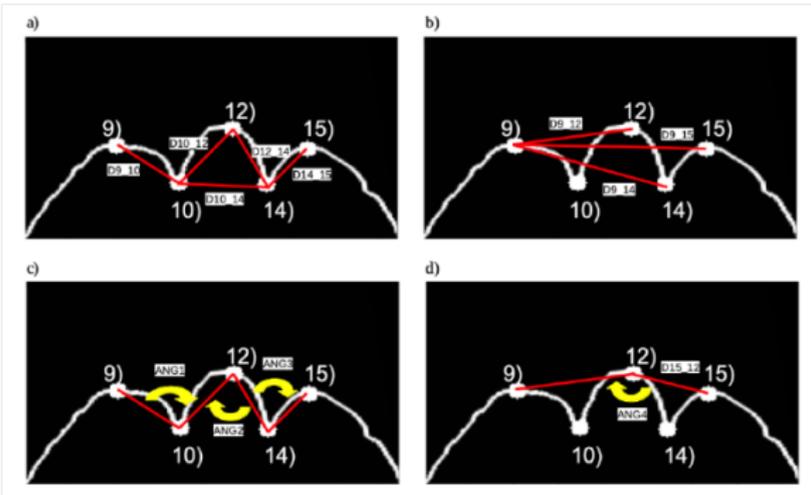
Machine Learning: Modelo básico



Etapa 1. ML

Adquisición de datos

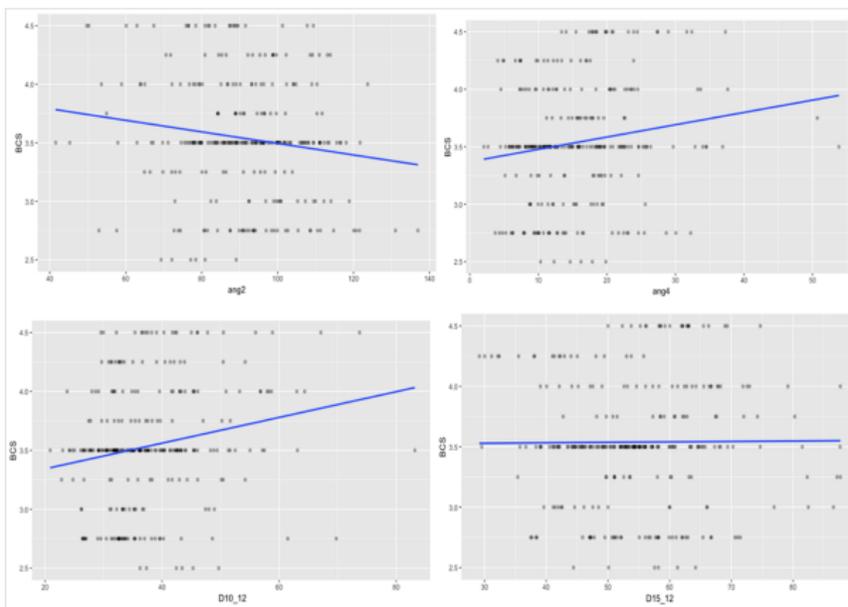
Se extraen los datos de todos los animales del dataset, como también de animales “desconocidos” para el sistema:



Etapa 2. ML

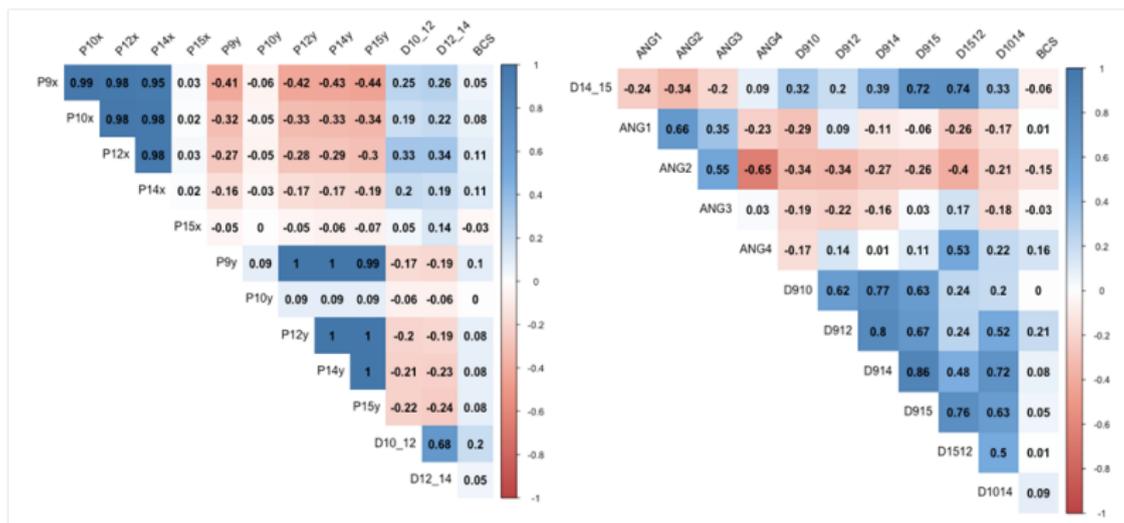
Análisis de datos: Cualitativos

Se analiza la correlación de datos frente a etiquetas BCS. Esto se mide haciendo uso del coeficiente de *Pearson* ya que define una medida de la correspondencia o relación lineal entre dos variables cuantitativas aleatorias.



Etapa 2. ML

Análisis de datos: Cuantitativos



Etapa 2. ML

Estandarización de datos

Muchas características poseen escalas muy diversas y valores atípicos muy grandes por lo que provocan dificultades para visualizar los datos y, lo que es más importante, pueden degradar el rendimiento predictivo de muchos algoritmos de aprendizaje automático. El método elegido, dentro de todas las variantes y opciones disponibles, fue el escalador ***preprocessing.MaxAbsScaler()***.

Proceso de estandarización				
P9x	P10x	P12x	...	D1014
a_1	b_1	c_1	...	n_1
a_2	b_1	c_2	...	n_2
...
$a_i = \frac{a_i}{\maxabs_{P9x}}$	$b_i = \frac{b_i}{\maxabs_{P10x}}$	$c_i = \frac{c_i}{\maxabs_{P12x}}$...	$n_i = \frac{n_i}{\maxabs_{D1014}}$



Etapa 2. ML

Conformación del dataset

Se conforma el dataset para utilizar en etapa de entrenamiento y testeo de modelos predictivos, contando con 286 filas y 22 columnas de datos.

[8 rows x 23 columns]

Dataset normalizado:

	P9x	P10x	P12x	P14x	P15x	P9y	P10y \
0	0.734104	0.747024	0.770445	0.761194	0.783430	0.662774	0.659639
1	0.783237	0.808036	0.819225	0.828358	0.845930	0.855474	0.838855
2	0.774566	0.796131	0.812052	0.814925	0.835756	0.824818	0.799699
3	0.809249	0.819940	0.863702	0.852239	0.877907	0.795620	0.757530
4	0.719653	0.750000	0.776184	0.783582	0.790698	0.708029	0.688253

	P12y	P14y	P15y	...	ANG1	ANG2	ANG3	ANG4 \
0	0.623656	0.583871	0.575707	...	0.639278	0.736765	0.567982	0.280832
1	0.800307	0.803226	0.810316	...	0.823382	0.783498	0.502894	0.037890
2	0.777266	0.769355	0.760399	...	0.726816	0.688499	0.585214	0.131872
3	0.734255	0.730645	0.728785	...	0.594366	0.474626	0.411795	0.180721
4	0.652842	0.617742	0.597338	...	0.816810	0.803213	0.785357	0.289190

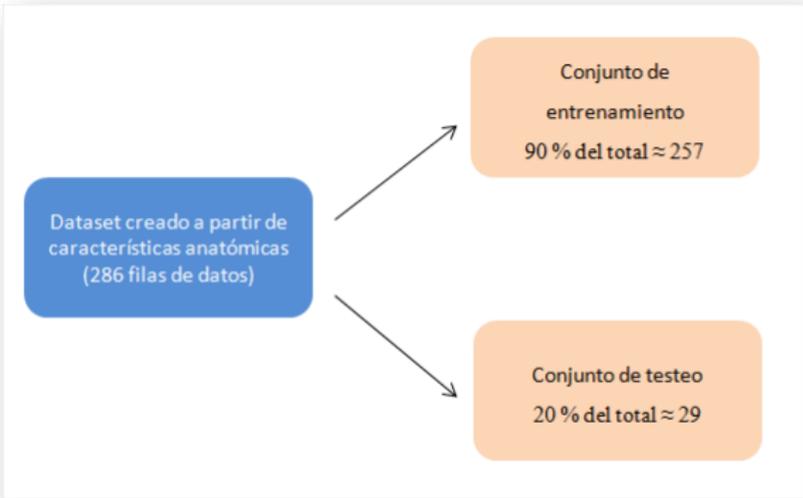
	D912	D914	D915	D1512	D1014	BCS
0	0.601459	0.827518	0.704540	0.684805	0.993500	3.0
1	0.763406	0.800000	0.669551	0.407712	0.782761	3.5
2	0.709888	0.796493	0.720028	0.568332	0.716459	3.5
3	0.848134	0.833273	0.725420	0.456879	0.700078	4.0
4	0.791720	0.948831	0.841046	0.753708	1.000000	3.0



Etapa 3. ML

Conjuntos de entrenamiento y testeo

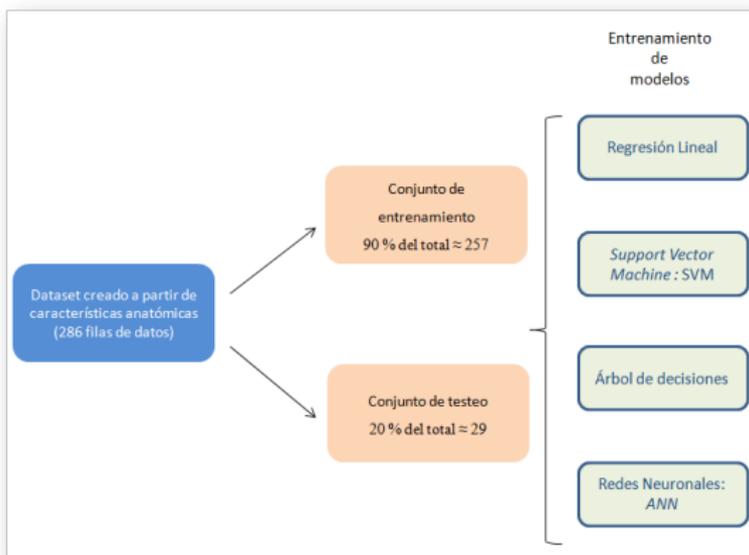
Se conforma el dataset para utilizar en etapa de entrenamiento y testeo de modelos predictivos, contando con 286 filas y 22 columnas de datos.



Etapa 3. ML

Entrenamiento

Se entrenaron 4 algoritmos de *Machine Learning* diferentes para luego comparar resultados:



Etapa 4. ML

Testeo de resultados

A partir del entrenamiento, se prueba cada uno de los algoritmos utilizando el conjunto de testeo conformado anteriormente.

Con el fin de comparar resultados con el material de investigación consultado durante todo el proyecto, la métrica elegida fue: **Error Medio Cuadrático** o *mean_squared_error* (*MSE*), la cual consiste en la sumatoria de las diferencias cuadráticas entre el valor real y el valor estimado por el algoritmo de cada muestra.

$$MSE(y, \hat{y}) = \frac{1}{n_{samples}} \sum_{i=1}^{n_{samples}} (y_i - \hat{y}_i)^2$$

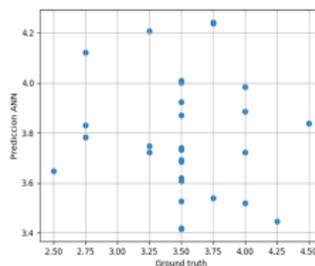
donde y_i es el valor real de cada i -ésima muestra de BCS, \hat{y}_i es el i -ésimo valor estimado por el algoritmo, y $n_{samples}$ son cada una de las muestras reales de BCS del conjunto de testeo.



Etapa 4. ML

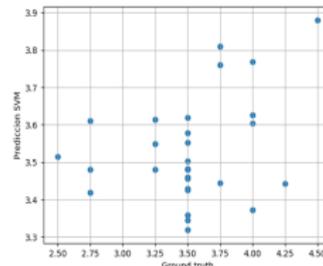
Análisis de resultados

a)



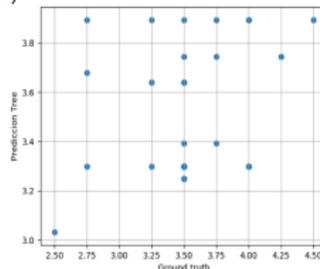
ANN Precision: 46.990000%
Error cuadrático medio: 0.28

b)



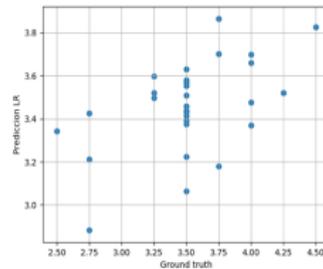
SVM Precision: 58.320000%
Error cuadrático medio: 0.17

c)



ARBOL Precision: 55.540000%
Error cuadrático medio: 0.20

d)

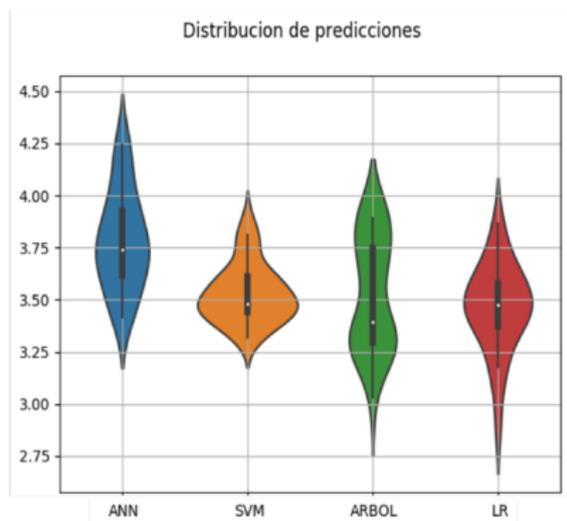
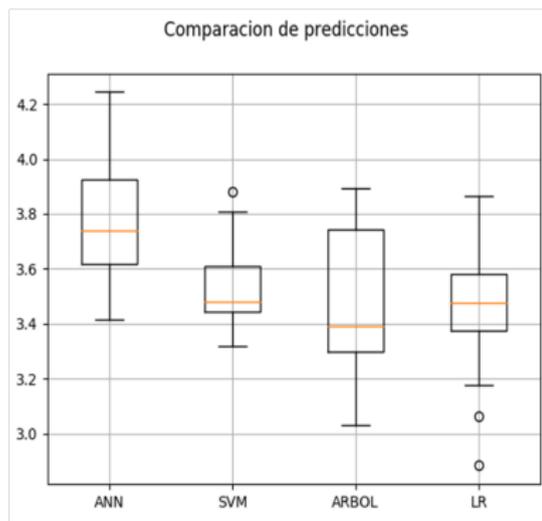


LR Precision: 61.940000%
Error cuadrático medio: 0.14



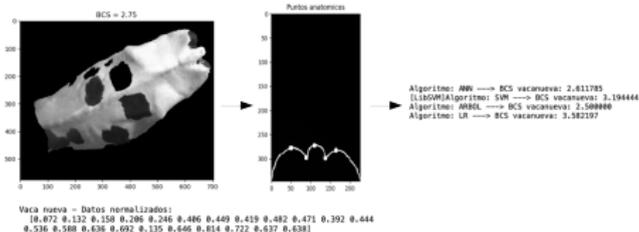
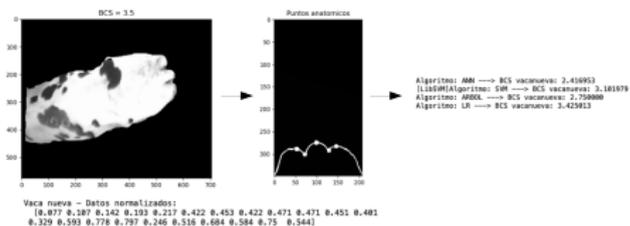
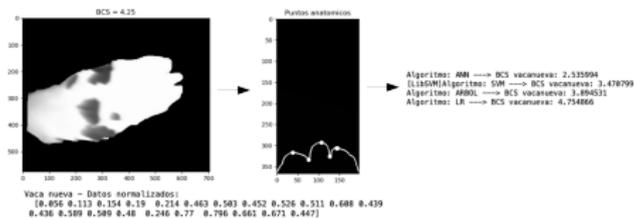
Etapa 4. ML

Análisis de resultados



Etapa 5. ML

Pruebas y mejoras

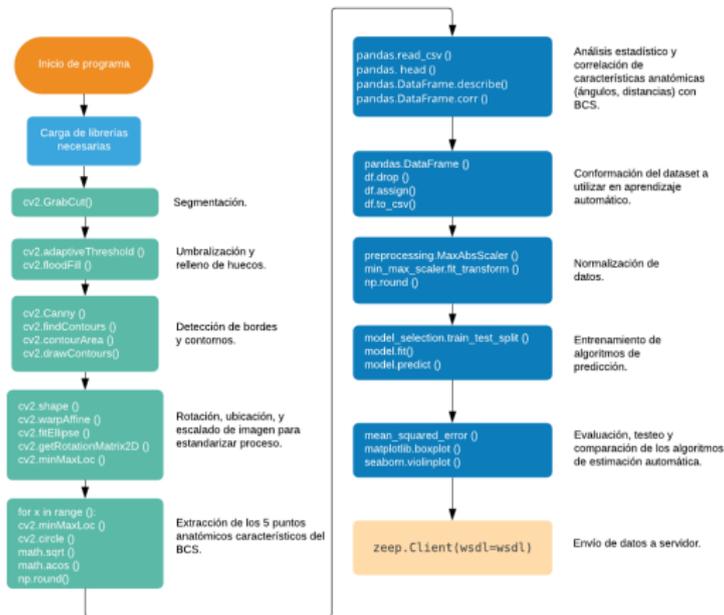


Código de programa

Resumen de ejecución

Flujo de programa

Facundo E. Reta | Octubre 2020



Conclusiones

- Se logró evaluar distintas técnicas y algoritmos de *Machine Learning*; se midieron todos los resultados devueltos haciendo uso de herramientas adecuadas, hallando distintas precisiones en cada uno de ellos.
- Se logró desarrollar un sistema de análisis de imágenes y clasificación objetiva de vacas según su BCS, logrando una precisión aproximada del **62 %**, utilizando lenguaje de programación Python, librerías OpenCV de visión artificial de libre uso y distribución e implementando sobre una placa de desarrollo Raspberry Pi 3B.
- Se obtuvieron respuestas de procesamiento muy rápidas, en el orden de los 5 segundos para el procesamiento completo de imagen en cada animal, y su respectiva estimación de BCS.



Trabajo a futuro

Como trabajo a futuro puede enumerarse lo siguiente:

- Adquisición de imágenes *in situ* mediante ambientes controlados, extendiendo el número de muestras ya obtenidas de *Corfilac, IPLAB and Penn Veterinary Medicine joint team*, con el fin de mejorar la precisión de estimación.
- Adquisición de video y procesamiento de imágenes en tiempo real. Clasificación de animales según su BCS en tiempo real.
- Implementación de técnicas automáticas de segmentación, haciendo uso de aprendizaje automático no supervisado o *deep learning*.
- Aceleración de los algoritmos desde perspectiva de Software y Hardware.
- Desarrollo de entorno GUI con el fin de hacer mas amigable el sistema a potenciales usuarios.
- Medición del impacto y precisión a partir del uso de este sistema en campo.



Preguntas



Agradecimientos

¡MUCHAS GRACIAS!

