Clasificador Bayesiano

Flores-Silva P.

Abstract—Se entrenó un clasificador Bayesiano ingenuo con datos de color de cinco imágenes de prueba para cada píxel de cada imagen. Los datos se obtuvieron gracias a la segmentación de tres zonas de interés mediante la app color thresholder de matlab, la normalización de los tamaños del personaje de interés, así como el tamaño de las imágenes de entrenamiento . Para la clasificación se consideró que cada color de cada píxel de las imágenes de prueba provienen de una distribución gaussiana. El resultado de la clasificación fue satisfactorio por lo que se concluye que el clasificador tiene equivocaciones muy pequeñas con respecto a las clases de interés. Es claro que el clasificador se equivocará en regiones que no corresponden a las clases. Para resolver esto se proponen, en las conclusiones, dos métodos que podrían mejorar la clasificación en dichas regiones. NOTA: Se sugiere revisar las libretas de los scripts desarrollados en el github: https://github.com/Pedri0/ClasificadorBayesianoconImagenes.
ya que ahi se actualizará la documentacion del codigo.

Index Terms—Visión computacional, Imágenes, Aprendizaje de Maquina, Clasificador Bayesiano Ingenuo, Naive Bayes.

1 Introducción

CON la creciente industria social electrónica, tales como facebook, TikTok, foros en internet entre otros, en años recientes, la cantidad de datos generados en esta industria crece de manera acelerada cada día. El creciente poder computacional y el avance de las ciencias computacionales permite manejar todo tipo de datos en tiempos aceptables para una gran cantidad de los mismos. La modernización de la industria tiene como objetivo la automatización adquisición de datos de forma electrónica, esto debido a que las nuevas tecnologías proveen métodos que permiten el fácil manejo y análisis de los datos.

La importancia de los datos puede resumirse en que gracias a ellos es posible extraer conocimiento o la deducción de ciertos eventos. Existen muchos métodos para realizar dicha tarea, entre ellos se encuentra la estadística inferencial.

El teorema de Bayes es uno de los grandes pilares de la estadística inferencial, es también, importante en una gran variedad de modelos de aprendizaje de maquina. El razonamiento de Bayesiano es un enfoque lógico para actualizar la probabilidad de una hipótesis o evento a la luz de nueva evidencia (como los datos) y, por lo tanto, juega un papel fundamental en la ciencia [1].

El análisis Bayesiano nos permite responder preguntas para las cuales no se desarrollaron enfoques estadísticos frecuentistas. De hecho, la idea de asignar una probabilidad a una hipótesis no es parte de este paradigma [2].

1.1 Estadística Básica y el teorema de Bayes

Un experimento estadístico puede definirse en términos generales como un proceso que da lugar a uno y sólo uno de varios resultados posibles. La colección de todos los posibles resultados se llama el espacio muestral, denotado como Ω. La probabilidad de un evento E la denotamos como $P(E)$ y se calcula, de acuerdo al concepto clásico de probabilidad como el cociente entre el numero de eventos favorables E y el numero total de eventos, es decir, $P(E) = \frac{|E|}{|\Omega|}$, donde $|x|$ es la cardinalidad del conjunto x.

El complemento de E es definido como el evento para el cual E no ocurre y se denota como E^c, con $P(E) = 1 - P(E)$. Además, $P(A|B)$ denota la probabilidad condicional de A dado B, interpretada como la probabilidad de que ocurra A dado que ocurrió B. finalmente, decimos que dos eventos son mutuamente excluyentes si la ocurrencia de uno de estos eventos descarta la posibilidad de que ocurra el otro evento.

Sean A y B dos eventos de un espacio muestral Ω, que puede ser finito o infinito, sea $P : \Omega \to [0, 1]$ una distribución de probabilidad en Ω tal que $0 < P(A) < 1$, $0 < P(B) < 1$ y $P(\Omega) = 1$. Definimos la intersección de los eventos A y B, denotado por $A \cap B$, como el evento tal que A y B ocurren. La unión de eventos, denotado por $A \cup B$, es el evento para el cual A, B o ambos ocurren.

La probabilidad condicional es una relación fundamental dada por:

$$P(A|B) = \frac{P(B \cap A)}{P(B)} \tag{1}$$

El teorema de bayes prueba que la probabilidad condicional es:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{2}$$

Si Ω está dividido en n eventos $A_1, ..., A_n$ mutuamente excluyentes y si B es un evento que es un subconjunto de la unión de todos los A_i y con probabilidad mayor que cero, entonces, para cada A_i la formula generalizada de Bayes es:

$$P(A_i|B) = \frac{P(B|A_i)P(A_i)}{\sum_{j=i}^{n} P(B|A_j)P(A_j)} \tag{3}$$

Que puede reescribirse como:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B|A)P(A) + P(B|A^c)P(A^c)} \tag{4}$$

• *P. Flores-Silva estudiante del PCIC en la UNAM.*
E-mail: flosipan@ciecias.unam.mx

El teorema de Bayes puede ser usado para obtener la probabilidad a posteriori de una hipótesis dados los datos observados:

$$P(hipotesis|datos) = \frac{P(datos|hipotesis)P(hipotesis)}{P(datos)} \quad (5)$$

donde $P(datos|hipotesis)$ es la función de verosimilitud (probabilidad) de los datos dada la hipótesis. $P(hipotesis)$ es la probabilidad a priori de la hipótesis y $P(datos)$ es la probabilidad de observar los datos independientemente de la hipótesis. La probabilidad a posteriori también se conoce como el grado (inicial) de creencia en la hipótesis. Es decir, lo anterior cuantifica la plausibilidad a priori de la hipótesis [2].

1.2 Clasificador de Bayesiano Ingenuo (Naive Bayes)

Se asume que el conjunto de datos contiene n muestras ($\vec{x_i}$, $i = 1, ..., n$) que consisten en p atributos, es decir $\vec{x_i} = (x_{i1}, x_{i2}, ..., x_{ip})$. Se supone que cada muestra pertenece a una y solo una clase $y \in \{y_1, y_2, ..., y_m\}$. La mayoría de los modelos predictivos en el aprendizaje de maquina generan una escala numérica s para cada instancia de $\vec{x_i}$. Esta escala cuantifica el grado de pertenencia a la clase y_j.
El aprendizaje ingenuo de Bayes se refiere a la construcción de un modelo probabilístico Bayesiano que asigna una probabilidad de clase a posteriori a una instancia $P(Y = y_j|X = \vec{x_i})$. El clasificador ingenuo de Bayes usa dichas probabilidades para asignar una muestra a una clase. Del teorema de Bayes tenemos que:

$$P(y_j|\vec{x_i}) = \frac{P(\vec{x_i}|y_j)P(y_j)}{P(\vec{x_i})} \quad (6)$$

Donde el numerador puede reescribirse como (omitiendo el índice i):

$$P(\vec{x}|y_j)P(y_j) = P(\vec{x}, y_j) = P(x_1, ..., x_p, y_j) = ... \quad (7)$$
$$P(x_1|x2, ..., x_p, y_j)P(x_2|x3, ..., x_p, y_j)...P(x_p|y_j)P(y_j) \quad (8)$$

Al suponer que cada x_i son independientes entre si, entonces $P(x_k|x2, ..., x_p, y_j) = P(x_k|y_j)$ para cualquier $k = 1, ..., p$. De esta manera podemos reescribir el numerador como:

$$P(\vec{x}|y_j)P(y_j) = \prod_{k=1}^{p} P(x_k|y_j)P(y_j) \quad (9)$$

Así, escribimos (6) como:

$$P(y_j|\vec{x}) = \frac{\prod_{k=1}^{p} P(x_k|y_j)P(y_j)}{P(\vec{x})} \quad (10)$$

Notese que el denominador no depende de las clase y es una constante, por lo que se considera un factor de escala y dado que se está interesado en asignar una muestra a exactamente una clase, entonces es posible omitir el calculo de dicha constante y calcular unicamente el el valor del numerador para cada clase y seleccionar aquella clase cuyo valor de $P(y_j|\vec{x})$ sea máximo, ya que representa mayor probabilidad de que dicha muestra pertenezca a y_j. La clase seleccionado se conoce como la clase máximo a posteriori (MAP) y es calculada como $\hat{y}$ para la muestra $\vec{x}$ como sigue:

$$\hat{y} = \prod_{k=1}^{p} P(x_k|y_j)P(y_j) \quad (11)$$

Un modelo que implementa (11) se le llama clasificador Bayesiano ingenuo.

2 Procedimiento experimental

Se va a implementar un clasificador Bayesiano Ingenuo para clasificar tres regiones de interés para una imagen dada de entrada, el resultado deberá ser la misma imagen con cada una de las tres regiones coloreada con su respectivo color promedio, es decir la imagen de salida tendrá tres colores basado en la clasificación. Para realizar esta tarea se deben obtener los datos por separado y definir las regiones de interés para calcular su estadística para que con esto sea posible entrenar el clasificador Bayesiano. Las imágenes de entrenamiento se muestran en la figura 1, mientras que las imágenes de validación se muestran en la figura 2.

Fig. 1. Imágenes de entrenamiento. Las imágenes originales no contienen el texto *Imagen X*

Fig. 2. Imágenes de validación. Las imágenes originales no contienen el texto *Imagen X*

Las regiones de interés o clases serán los ojos, las orejas y el cuerpo, ya que los colores de cada una de estas clases tienen valores en el espacio de color RGB distantes por lo que, de esta manera es fácil identificarlos. Notemos que las tres clases son contiguas.
Para las imágenes de entrenamiento se procuró que el tamaño del pokemon (en píxeles) sea aproximadamente igual para cada imagen, esto se fue posible al escalar únicamente el pokemon a dimensiones determinadas con el programa GIMP. Para el caso de los datos de entrenamiento es importante que el paso anterior ya que la probabilidad a priori de cada clase depende de la cantidad de píxeles de la clase y la cantidad de píxeles en la imagen. teniendo esto en cuenta también se ajustó el tamaño de la imagen total a (512,512,3) píxeles antes de realizar el escalado del pokemon. Si no se realizase el escalamiento tanto de la imagen y del pokemon en sí, las probabilidades de cada región contendrían una varianza grande y por ende sería difícil poder clasificarlas correctamente, es decir los dos pasos mencionados con anterioridad aseguran una uniformidad consistente de las probabilidades a priori de cada clase.
En el caso de las imágenes de validación no se realizó ningún tratamiento previo. De esta forma, cada una de las imágenes mantiene su tamaño original de acuerdo a como fueron encontradas en internet.

3 Obtención de datos

Como ya se mencionó hay tres clases de interés que vienen o serán caracterizadas por su color. Dado que se trabaja con las imágenes a color cada clase tendrá tres atributos o vectores $\vec{x_i}$ característicos. El espacio de color seleccionado fue el BGR ya que la idea de este sistema es mucho más intuitiva a los demás. De esta manera un vector $\vec{x_i}$ para una clase dada i tendrá la forma $\vec{x_i} = (B, G, R)$, donde los valores de B, G, R varían entre $(0, 255)$. Nótese que los valores B, G, R serán los promedios de los promedios de los valores de cada región en cada imagen, por ejemplo, para la clase cuerpo y una imagen dada se analiza cada píxel de la imagen talque dicho píxel corresponda a la clase, se realiza la suma de cada valor de B, G, R por separado y cada valor se divide entre la cantidad de píxeles correspondientes a la clase. Este proceso se realiza para cada clase en cada imagen de entrenamiento para así obtener los promedios de los promedios de cada imagen. La obtención de las probabilidades a priori es un proceso similar, en lugar de dividir la suma de cada valor B, G, R por la cantidad de píxeles que corresponden a la clase, se divide por la cantidad total de píxeles en la imagen.

Para poder realizar lo anterior es necesario conocer las posiciones de los píxeles de cada clase. Sin embargo, este método es muy complicado de realizar con las imágenes originales. En una forma más intuitiva de realizar esto es mediante la obtención de las mascaras de cada clase y realizar el proceso anteriormente mencionado teniendo cuidado de no promediar píxeles cuyo valor sea (0,0,0) (negro).
Las mascaras obtenidas para cada región y cada imagen se muestran en las figuras 3, 4, 5. Fue posible obtener las mascaras de cada imagen gracias a la herramienta o aplicación de matlab llamada ***Color Thresholder***. Cabe mencionar que este proceso es fácilmente programable en python (ver [3]). Sin embargo, por razones de tiempo no fue posible desarrollar este programa.
Finalmente en base a las probabilidades obtenidas de cada clase para cada imagen se obtiene la varianza utilizando el mismo proceso al anterior pero considerando las medias previamente calculadas.

Fig. 3. Mascaras del cuerpo. Las imágenes no contienen el texto *Imagen X*

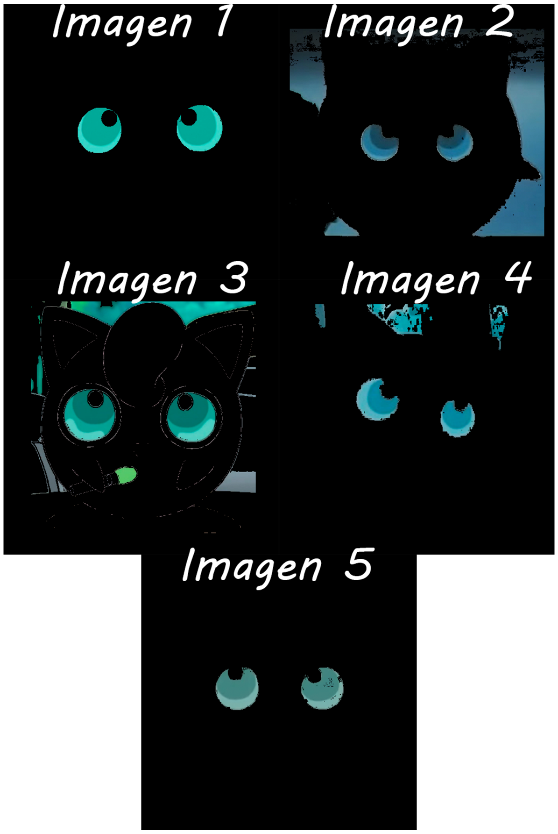

Fig. 4. Mascaras de los ojos. Las imágenes no contienen el texto *Imagen X*

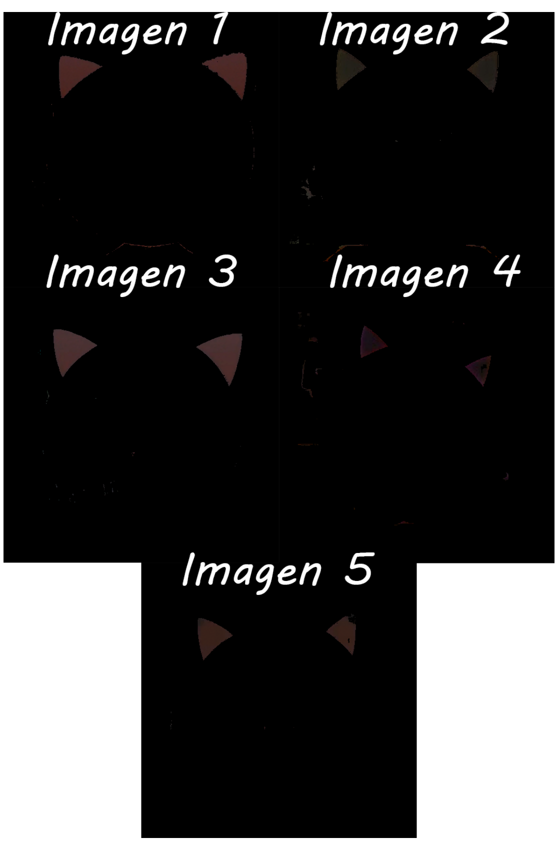

Fig. 5. Mascaras de las orejas. Las imágenes no contienen el texto *Imagen X*

El código para obtener el promedio y la varianza de los atributos así como las probabilidades a priori de cada clase de cada imagen considerando las mascaras se encuentra en la libreta de jupyter llamada **BayesDatosPokemon.ipynb**. Dicha libreta genera nueve archivos .csv. Tres de ellos contienen la probabilidad de cada clase para cada imagen, otros tres contienen la varianza de cada clase para cada imagen y los otros tres contienen las probabilidades a priori de cada clase para cada imagen. Un ejemplo de la estructura de un archivo de medias o promedios y varianzas se muestra a continuación.

i	Media Azul Orejas	Media Verde Orejas	Media Rojo Orejas
0	valor	valor	valor
1	valor	valor	valor
2	valor	valor	valor
3	valor	valor	valor
4	valor	valor	valor

TABLE 1
Ejemplo de un archivo csv para la media o varianza de una clase dada. i denota el índice que se relaciona con la imagen, es decir la imagen 1 corresponde al índice 0

Un ejemplo del archivo csv de las probabilidades a priori se muestra a continuación.

i	Probabilidad Orejas
0	valor
1	valor
2	valor
3	valor
4	valor

TABLE 2
Ejemplo de un archivo csv para la probabilidad a priori de una clase dada. i denota el índice que se relaciona con la imagen, es decir la imagen 1 corresponde al índice 0

4 Clasificación

Una vez que ya se tienen los datos, se hace la suposición de que cada atributo sigue una distribución (probabilidad condicional de los atributos dada una clase) gaussiana con valores de μ y σ correspondientes al promedio de promedios de cada atributo previamente calculado. Por ejemplo dada una clase, digamos los ojos, calculamos el promedio del promedio de cada atributo, es decir el promedio del archivo mediaOjos para cada columna (atributos). el valor de σ se calcula con el promedio del promedio de los valores de la varianza para cada atributo. Para hacer esto:

```
#Vamos a calcular la media de: las medias de cada color (BGR), las varianzas de cada color. Tambien,
#vamos a calculas la varianza de: las medias de cada color (BGR), las varianzas de cada color
#Para ello hacemos:
medCuerpo = mediaCuerpo.mean(axis=0)
medOjos = mediaOjos.mean(axis=0)
medOrejas = mediaOrejas.mean(axis=0)

#calculamos la varianza de cada columna para cada dataframe
varCuerpo = varianzaCuerpo.mean(axis=0)
varOjos = varianzaOjos.mean(axis=0)
varOrejas = varianzaOrejas.mean(axis=0)

#Convertimos el dataframe prob(Clase) a un array numpy
```

```
pCuerpo=probCuerpo.to_numpy()
pOjos = probOjos.to_numpy()
pOrejas = probOrejas.to_numpy()
```

Listing 1. μ y σ de la probabilidad condicional de los atributos dada una clase .

Aplicando el logaritmo en cada lado de la ec (11) y realizando operaciones sencillas vemos que (11) es equivalente a:

$$\hat{y} = argmax_y \left(ln(P(y)) + \sum_{i=1}^{n} ln(P(x_i|y)) \right) \tag{12}$$

donde $ln(P(x_i|y))$ es:

$$ln(P(x_i|y)) = \left(\frac{-(\mu_y - x_y)^2}{2\sigma^2} + ln\left(\frac{1}{\sigma}\right) - \frac{ln(2\pi)}{2} \right) \tag{13}$$

Antes de proceder a la clasificación se define la función logaritma auxiliar de la siguiente forma:

```
import math as m
#definimos la funcion log que come los parametros
    varianza, media y x
def log(x,med,var):
    ln = ((-1/(2*pow(var,2)))*(pow((med-x),2)))+(m.
    log((1/var)))-(m.log(2*m.pi)/2)
    return ln
```

Listing 2. Función auxiliar logaritmica note que corresponde a $ln(P(x_i|y))$.

Para clasificar leemos la imagen y creamos dos arrays de las mismas dimensiones de la imagen:

```
img = cv2.imread('5.jpg')
Imagen = np.zeros(img.shape,dtype=float)
Imagen2 = np.zeros(img.shape,dtype=float)
```

Listing 3. Leer imagenes y crear dos arrays

Finalmente, para clasificar debemos calcula $\hat{y}$ para cada píxel, es decir recorrer la imagen de entrada por cada píxel en ella y para cada píxel calcular $\hat{y}$ usando las tres clases y los tres atributos, escoger el que tenga mayor valor y pintar (asignar los valores del píxel) el píxel del array que creamos según el color promedio (B, G, R) de la clase cuyo valor fue máximo. Se espera que la imagen resultante el color del cuerpo sea uniforme y color rosa, el color de los ojos sea uniforme y color azul verdoso y las orejas sea uniforme y color café oscuro. Finalmente para las zonas aledañas o fondo no se espera un color en particular ya que este varia y el programa tomara el color de la clase ganadora. Se consideraron dos arrays ya que uno de ellos tendrá los valores de la clasificación sin considerar las probabilidades a priori, este array sera el de Imagen, y el array Imagen2 si considerará las probababilidades a priori esto para poder observar los resultados.
Dado que el código de esta parte es mas ancho que las columnas del documento, se sugiere revisar la libreta de jupyter **ClasificadorBayesianoIngenuoPokemon.ipynb** (es fácil encontrarlo ya que se documento el código).

5 Resultados y Discusión

Las imágenes resultantes de la clasificación se muestran en las imágenes 6 y 7.

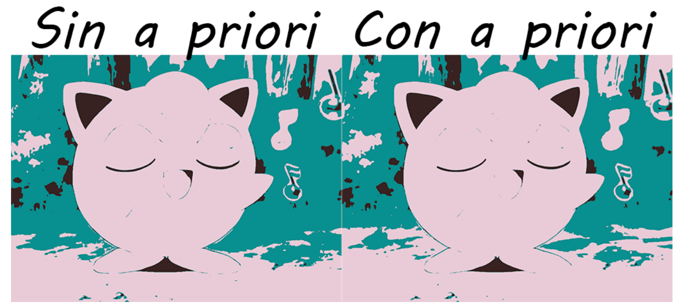

Fig. 6. Resultados de la clasificación

Los resultados para la primera imagen son tal y como los esperados con respecto a las zonas de interés. Mientras que el fondo observamos en su mayoria las clases cuerpo y ojos, este resultado es predecible ya que los colores del fondo son muy similares a cada clase dominante. Veamos que visualmente el suelo o tierra es mas semejante al rosa que al azul verdoso y el follaje es mucho más semejante al azul verdoso que al rosa, por lo que es obvio esperar este tipo de resultados.
Los resultados de considerar la probabilidad a priori aparentan ser sumamente similares, salvo por regiones donde existen curvas en la imagen original, es decir los bordes se observan un poco mejor definidos cuando no se considera la probabilidad a priori. La perdida mínima visual de los bordes al considerar las probabilidades a priori se explica fácilmente al ver que la probabilidad a priori del cuerpo aproximadamente dos y tres veces más grande que las clases ojos y orejas respectivamente, por lo que para un píxel entre el negro y el rosa el clasificador tendrá su $haty$ como la clase cuerpo, ya que la probabilidad a priori de esta es mayor.

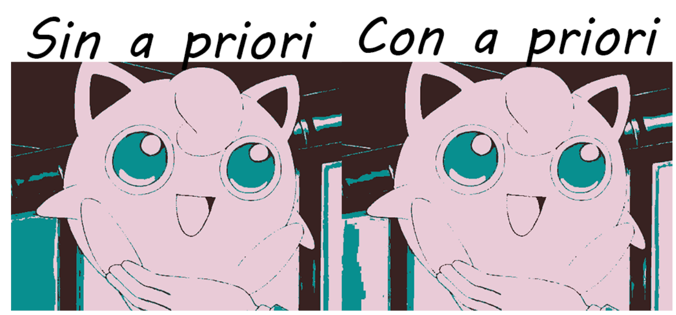

Fig. 7. Resultados de la clasificación

Notemos que los resultados para la segunda imagen de prueba son bastante acertados incluso sin considerar la probabilidad a priori. Notemos que las únicas diferencias apreciables entre las dos imágenes son quizá el fondo, esto es claro ya que los colores del fondo no tienen ninguna relación con las tres clases y para estos casos las probabilidades a priori juegan un papel importante. Por otro lado notemos que la parte de los ojos que era blanca (esta zona no se considera en el color de los ojos) el clasificador los clasifica como si fueran una parte del cuerpo, este hecho también es obvio ya que visualmente el color blanco se asemeja más al rosa que a alguna otra de las clases disponibles. Finalmente en ambas imágenes se aprecia en la zona de la boca que el área que se detecto como oreja difiere ligeramente entre imágenes, nuevamente esto se debe a las probabilidades a priori.

6 Conclusiones

Se desarrolló y entrenó un clasificador Bayesiano ingenuo a partir de cinco imágenes similares en cuanto al color de las clases de interés. Con éste fue posible clasificar dos imágenes, que no eran parte de los datos de entrenamiento, con resultados visuales aceptables. De esta manera consideramos que el clasificador tiene muy pocas equivocaciones, ya que si fuera el caso las imágenes resultantes no serían similares a las imágenes de entrada con respecto a las clases. Claramente el clasificador tendrá demasiados errores para las regiones desconocidas como el fondo.

Es posible mejorar esto último al considerar una cuarta clase como el fondo y realizar una regularización del mismo. También esto puede arreglarse al medir el grado de incerteza de la clasificación con respecto a las tres clases y si la incerteza es mayor a un cierto umbral pintar el array que contiene los resultados en las posiciones de dichos píxeles otro color como por ejemplo amarillo.

Con esto en mente consideramos que nuestro clasificador es bueno para clasificar los colores rosa, café y azul verdoso. Sería interesante observar su desempeño en imágenes que no contengan a nuestro personaje de Pokemon si no que algún otro personaje o incluso fotos con personas.

Finalmente cabe mencionar que, para este caso, las probabilidades a priori de las clases aparentan no juegan un papel fundamental en la clasificación ya que las variaciones de los resultados considerando las probabilidades a priori y sin considerarlas son bastante similares. Sin embargo, dado que el conjunto de entrenamiento es muy reducido, no podemos asegurar que las probabilidades realmente no importan.

References

[1] D. Berry, *Statics: A Bayesian Perspective*, 1st ed. Duxbury Press, 1996.

[2] D. Berrar, *(Bayes Theorem and Naive Bayes Classifier.) Encyclopedia of Bioinformatics and Computational Biology*, 1st vol. Elsevier, pp 403-412, 2018.

[3] A. Rosebrock, *OpenCV and Python Color Detection*, Guia para pyimagesearch www.pyimagesearch.com/2014/08/04/opencv-python-color-detection/. Consultado el 02.04.20