Algoritmo de segmentación

Robótica y percepción computacional

Alejandro Cobo Cabornero, 150333
Facundo Navarro Olivera, 140213
Diego Sánchez Lizuain, 150072

Contenido

Solución propuesta

Tras comparar el rendimiento de varios clasificadores, como el basado en la distancia euclídea, el *K-Nearest Neighbors*, el *Gaussian Naive Bayes* o el *Quadratic Discriminant Analysis (QDA)*, hemos determinado que los dos clasificadores más apropiados son el euclídeo y el QDA, puesto que son los que menor tiempo de segmentación presentan.

Tras comparar ambos, hemos determinado que el QDA ofrece un menor error y es el que finalmente hemos decidido utilizar.

Se ha utilizado la implementación incluida en la librería *sklearn.qda*.

Entrenamiento

Para el entrenamiento, se utilizan las imágenes "linea.png" y "lineaMarcada.png", adjuntas en la carpeta src/imgs.

Se seleccionan los píxeles correspondientes de la imagen de entrenamiento y se normalizan sus valores RGB. Estos datos se usan para llamar a la función *fit* del clasificador QDA.

El clasificador almacena las medias y las matrices de covarianza de las clases.

Algoritmo de segmentación

Para segmentar una imagen, se normalizan sus valores RGB y se descarta un canal. Esta imagen se pasa al método *predict* del clasificador QDA.

El clasificador tiene una frontera de decisión cuadrática, y supone una distribución gaussiana de los datos. Ajusta los datos de entrada con las densidades de probabilidad condicionada de las clases, usando el teorema de Bayes.

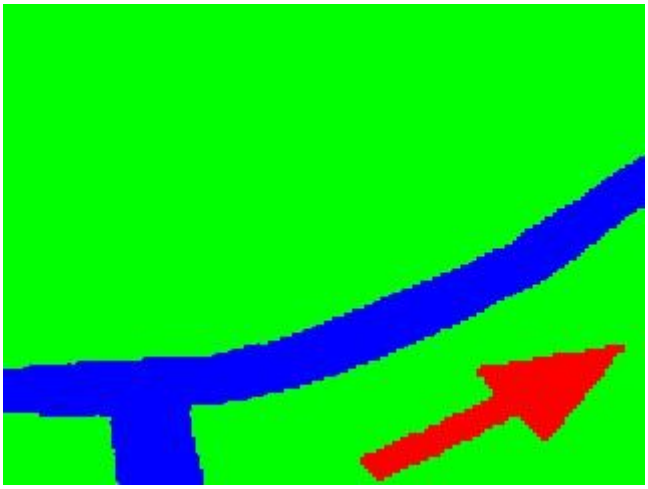
Breve explicación del código

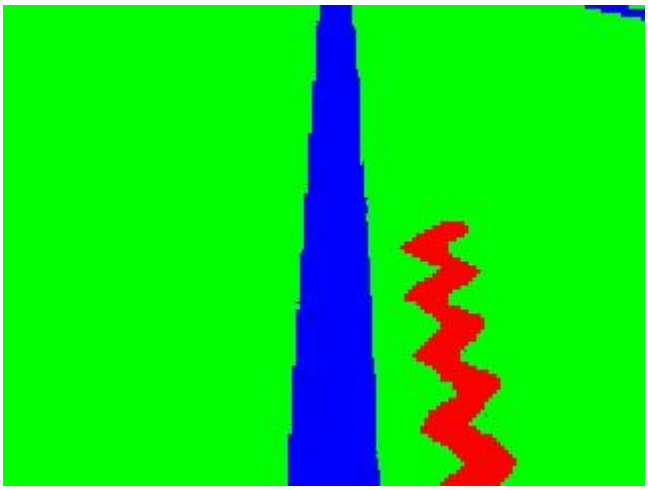
Tras crear y entrenar el clasificador utilizando la imagen de entrenamiento, se comienza a leer el vídeo que se va a segmentar.

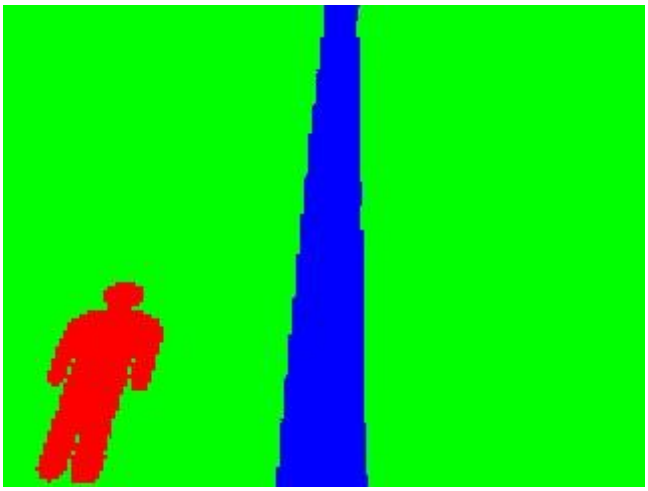
Para cada imagen, se normalizan sus valores RGB, se modifican sus dimensiones para adaptarla al formato de entrada del clasificador (una matriz de datos en la que cada fila es un dato y cada columna un valor del píxel para cada canal RGB) y se segmenta con el clasificador. Tras esto, se muestran la imagen original y la imagen segmentada.

Resultados obtenidos

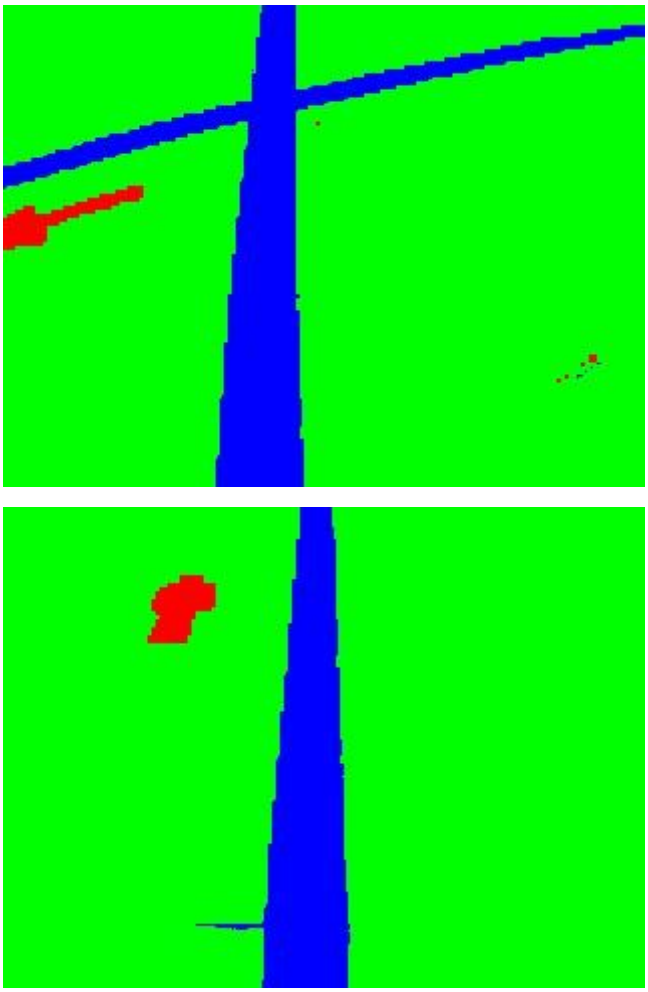
Ejemplos de imágenes bien segmentadas:





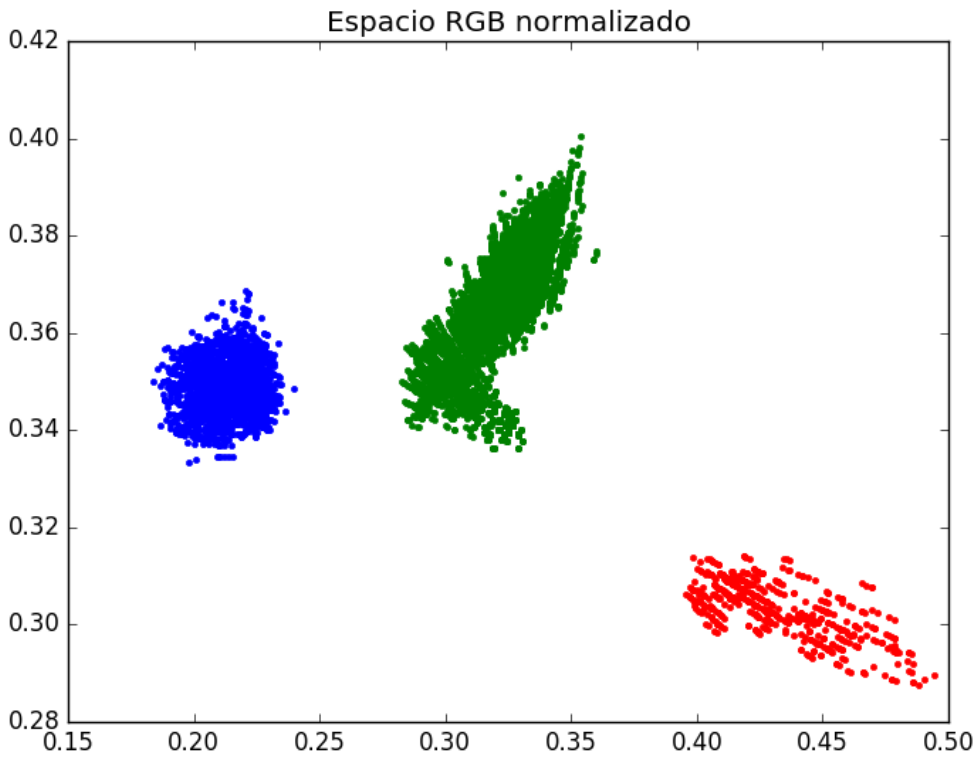


Ejemplos de imágenes mal segmentadas:



En general, la segmentación parece ser invariante a la iluminación, aunque existen algunos casos en los que el algoritmo confunde los píxeles del fondo.

En la siguiente imagen se puede ver la distribución de los datos de entrenamiento:



Las clases están bien definidas, por lo que el posible problema puede estar en que los datos de entrenamiento no son lo suficientemente representatvos.

Tiempos de ejecución

El tiempo de entrenamiento del clasificador oscila entre los 8 y los 24 milisegundos.

El tiempo de segmentación medio de una imagen es de unos 31 milisegundos.

Conclusiones

Los resultados obtenidos parecen indicar que el clasificador presenta un buen rendimiento para este problema. No obstante, para reducir el número de errores, es necesario experimentar con otros datos de entrenamiento.

La mayor ventaja de este algoritmo de segmentación es el tiempo de segmentación, que permite procesar un vídeo a 24 fotogramas por segundo en tiempo real.