kwiznia

[Dirección de correo electrónico]

Descripción breve

[Dibujar su lector con un resumen de la participación. Normalmente es un breve resumen del documento.   
Cuando esté listo para agregar contenido, haga clic aquí y empiece a escribir.]

[Título del documento]

[Subtítulo del documento]

# Parte 1

Algoritmo implementado

Para la parte uno de la práctica se ha implementado una clase llamada practica4.py que consta de los siguientes métodos:

* Detect: Este método se encarga de buscar los diferentes rectángulos donde cree que está la imagen. Para poder realizar esto, se utilizó el método *cascade.detectMultiScale*con el fin de detectar los diferentes objetos, de diferentes tamaños en la imagen. Para esto se utilizó un factor de escala de 1.2. Finalmente esto, devuelve una lista vacía, en el caso de que no encuentre ningún rectángulo que se ajuste a esos parámetros, o una lista con los rectángulos en caso contrario.
* draw\_rects: Se ha implementado como método auxiliar, para ser utilizada en clases posteriores con el fin de permitir pintar los diferentes rectángulos.
* detectorMatriculas: este método se encarga de la detección de matriculas con el detector de Haaris. El propósito de este método es ir leyendo las diferentes imágenes guardadas en una variable llamada *cascade\_file* y mediante la función *detect* (explicada anteriormente) intentar detectar los diferentes rectángulos que conforman la imagen. Una vez detectados cada uno de estos rectángulos, se dibujan los mismos junto con la imagen.
* pintarContornos: Este método es el encargado de pintar los contornos del rectángulo en la imagen. Para ello utiliza la función *rectangle* de openCV.
* filtrarContornos: