



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería en Informática



TFG del Grado en Ingeniería Informática

**Sistema de reconocimiento
automático en arqueobotánica**



Presentado por Jaime Sagüillo Revilla
en Universidad de Burgos — 23 de febrero de 2017
Tutores: Álgvar Arnaiz González, José Francisco Díez
Pastor y Virginia Ahedo García



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería en Informática



Álvar Arnaiz González, José Francisco Díez Pastor profesores del departamento de Departamento de Ingeniería Civil, Área de Lenguajes y Sistemas Informáticos.

Expone:

Que el alumno D. Jaime Sagüillo Revilla, con DNI 12782524-K, ha realizado el Trabajo final de Grado en Ingeniería Informática titulado Sistema de reconocimiento automático en arqueobotánica.

Y que dicho trabajo ha sido realizado por el alumno bajo la dirección del que suscribe, en virtud de lo cual se autoriza su presentación y defensa.

En Burgos, 23 de febrero de 2017

Vº. Bº. del Tutor:

Vº. Bº. del co-tutor:

D. nombre tutor

D. nombre co-tutor

Resumen

Descriptores

Arqueobotánica, Fitolito, Reconocimiento de imagenes, Python

Abstract

Keywords

Índice general

Índice general	III
Índice de figuras	IV
Índice de tablas	V
Introducción	1
Objetivos del proyecto	2
Conceptos teóricos	3
3.1. Segmentación	3
3.2. Binarización	3
3.3. Thresholding	4
3.4. Descriptores visuales	4
3.5. Máquinas de vectores soporte	4
Técnicas y herramientas	6
Aspectos relevantes del desarrollo del proyecto	7
5.1. Procesamiento de imágenes	7
5.2. Clasificadores	8
Trabajos relacionados	13
Conclusiones y Líneas de trabajo futuras	14
Bibliografía	15

Índice de figuras

Índice de tablas

Introducción

Objetivos del proyecto

Conceptos teóricos

Para la comprensión de este proyecto será necesario la comprensión de algunos conceptos teóricos que introduciré en este apartado:

- Segmentación
- Binarización
- *thresholding*

3.1. Segmentación

La segmentación en el campo de la visión artificial, como se indica en la wikipedia, consiste en subdividir una imagen en varios pixeles u objetos. [4] Cuando segmentamos una imagen lo que pretendemos hacer es cambiar su representación para poder obtener de esta una mayor utilidad o cantidad de información.

En nuestro caso segmentamos la imagen para eliminar el fondo de ella y obtener así una imagen con solo su parte delantera. De esta manera eliminamos el ruido que existe en la imagen y a su vez la simplificamos reteniendo la parte de la imagen en la que se encuentran los objetos que nos interesan.

Posteriormente a ese paso nos interesa, como es obvio, dividir la parte delantera de la imagen resultante en objetos. Para de esta manera obtener cada uno de los objetos por separado de forma ideal.

3.2. Binarización

La binarización de una imagen consiste en la simplificación de los valores de cada pixel a 2 posibles valores, blanco o negro, representando el fondo y el frente de la imagen cada uno de ellos. De esta manera conseguimos una

simplificación muy significativa. Para ejemplificarlo de manera más visual estamos convirtiendo una imagen RGB en la que cada pixel tiene 3 valores y cada uno de estos valores puede variar desde el 0 al 255, lo cual es una explosión de combinaciones muy significativa, frente a una imagen binarizada en la que cada pixel tiene 2 posibles valores, 0 o 1. Como podemos imaginar una simplificación de la imagen de tal calibre facilita el procesamiento de la imagen enormemente.

3.3. Thresholding

Es el método mas simple para la segmentación de una imagen, pudiendose utilizar para la binarización de una imagen, como es nuestro caso. Consiste en reemplazar los píxeles por debajo de una determinada constante a píxeles negros y los que se encuentran por encima a píxeles blancos o viceversa.

Existen distintas maneras de llevar a cabo este proceso, siendo uno de lo más conocidos el método de Otsu. [5]

3.4. Descriptores visuales

Los descriptores visuales, o descriptores de características, son descripciones de las características visuales de los contenidos en imágenes o videos , en nuestro caso de imágenes, con el proposito de la detección de objetos [2]. El objetivo de los descriptores visuales es obtener la información que resulta significativa, eliminando a su vez la información que no lo es. Para basarnos en la información obtenida para clasificar los objetos en las clases de objetos que deseemos. Algunos ejemplos de características son la forma, el color o la textura.

Como se puede imaginar obtener las características a mano es una tarea complicada y que usualmente no funciona correctamente. Por ello utilizamos un método de extracción automática de características como es *Histogram of Oriented Gradients*, el cual se basa en los gradientes de la imagen para obtener si un determinado objeto se encuentra en la imagen [3].

3.5. Máquinas de vectores soporte

Las máquinas de vectores soporte, o SVM, son modelos de aprendizaje supervisados con algoritmos de aprendizaje asociados que analizan datos usados para tareas de clasificación o de regresión [1]. En nuestro caso este modelo se ve usado para tareas de clasificación, puesto que es lo que nos concierne en nuestra problemática.

Para que nuestra SVM sea capaz de clasificar los objetos le proveemos de un conjunto de entrenamiento compuesto por positivos y falsos positivos, es decir,

ejemplos de los objetos que nos interesan y otros objetos, respectivamente. A partir de esta información nuestra SVM será capaz de clasificar nuevos ejemplos en las categorías pertinentes.

Técnicas y herramientas

Aspectos relevantes del desarrollo del proyecto

5.1. Procesamiento de imagenes

Como primera aproximación al problema que nos concierne hemos escogido el procesamiento de imagenes mediante la librería Scikit-image para Python. Mediante esta herramienta trataremos de dar solución a nuestro problema siguiendo los siguientes pasos:

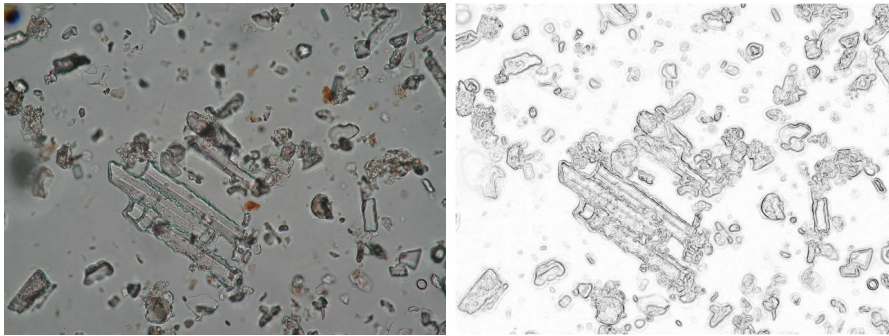
1. Convertimos la imagen a escala de grises
2. Segmentamos los objetos del fondo de la imagen
3. Obtenemos los distintos objetos de la imagen

Convertimos la imagen a escala de grises

La conversión de la imagen original (RGB) a escala de grises viene motivada porque para poder segmentar los objetos del fondo de la imagen mediante el método de *Thresholding* solo se puede partir de una imagen en escala de grises.

Segmentamos los objetos del fondo de la imagen

Una vez tenemos la imagen en escala de grises procedemos a transformar nuestra imagen en una imagen en blanco y negro o binarizada. Los motivos por los que binarizamos la imagen es para obtener una imagen que sea más significativa para nosotros y además este simplificada, para facilitarnos su procesamiento.



(a) Original

(b) Imagen en escala de grises

Scikit nos proporciona distintos métodos mediante los cuales podemos segmentar una imagen. A continuación vamos a ver el resultado aplicando distintos métodos, los cuales se van indicando en cada una de las figuras:

Obtenemos los distintos objetos de la imagen

Después de tener la imagen binarizada de la forma más apropiada posible probamos a segmentar los distintos objetos de nuestra imagen.

Transformación divisoria

Transformación divisoria, o en inglés *Watershed segmentation*, es un algoritmo clásico en la segmentación de objetos en una imagen.

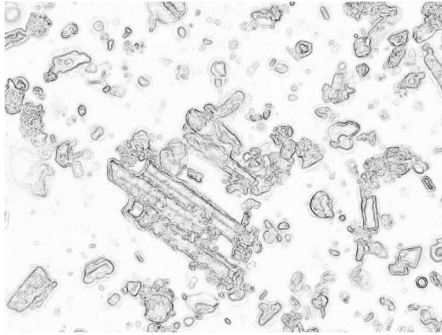
Durante las primeras pruebas la segmentación más interesante hasta el momento ha sido la que se muestra a continuación a partir del *Watershed segmentation* con marcado. Más allá de esta segmentación no se ha conseguido nada mejor.

5.2. Clasificadores

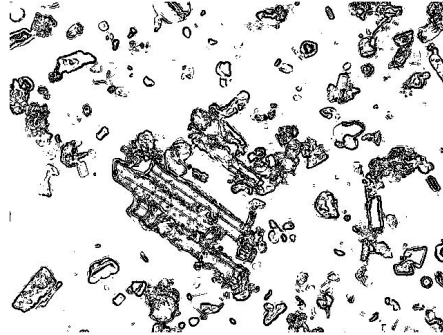
La aproximación mediante procesamiento de imágenes no parece la más adecuada visto los resultados obtenidos. Por ello vamos a realizar el estudio sobre una segunda aproximación mediante máquinas de vector soporte, para la tarea de clasificación, junto a Histogramas de los gradientes orientados, la cual es una técnica para la extracción automática de características.

El procedimiento para obtener el clasificador es el siguiente:

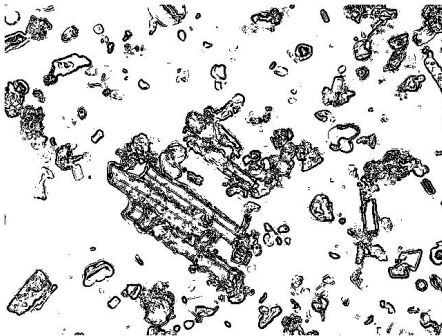
1. Crear un conjunto de entrenamiento de imágenes de caras que supongan positivos.
2. Crear un conjunto de entrenamiento de imágenes de no-caras que supongan falsos-positivos.



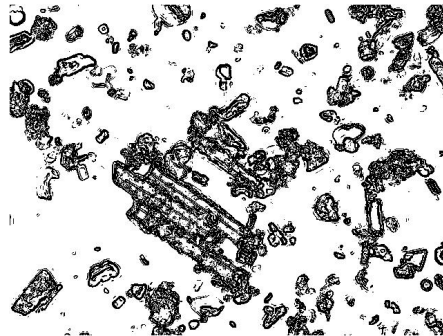
(a) Imagen en escala de grises



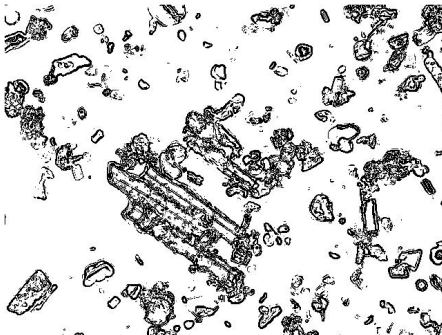
(b) Método de Otsu



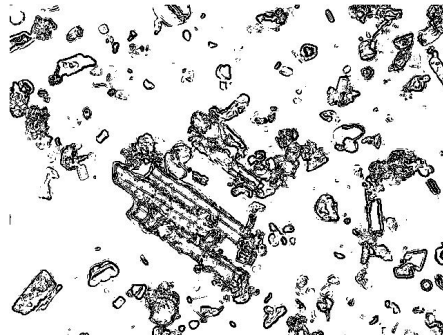
(c) Método de Otsu



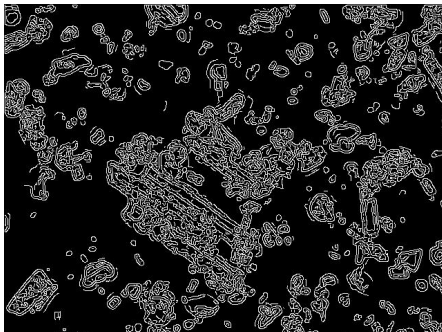
(d) Método de Yen



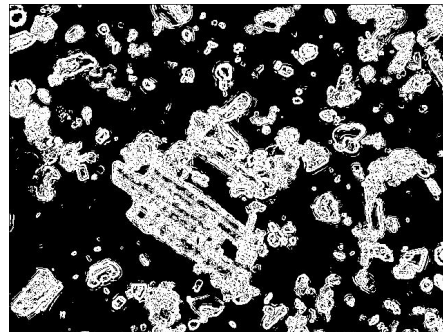
(e) Método de Li



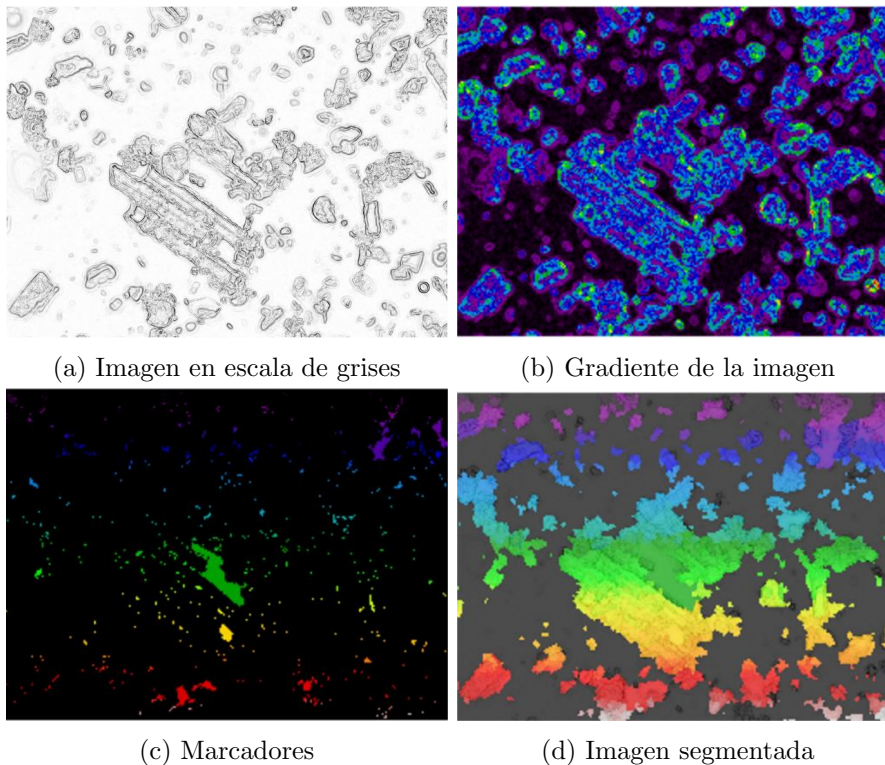
(f) Método de ISODATA



(g) Método basado en bordes



(h) Método adaptativo



3. Extraer las características de HoG del conjunto de entrenamiento.
4. Entrenar el clasificador SVM.

Finalizado el entrenamiento de la SVM, ya tenemos nuestro clasificador listo para enviarle nuevas imágenes y que sean clasificadas.

Reconocimiento de imágenes en nuevas imágenes

Para el reconocimiento de objetos en nuevas imágenes deberemos de llevar a cabo los 2 siguientes pasos:

1. Pasar una ventana por toda la imagen, comprobando si la ventana contiene un objeto.
2. Si existe solapamiento en la detección del objeto, muy común en el uso de este tipo de clasificadores, se deben de combinar dichos solapamientos en uno único.

En cuanto al primer paso, lo único a lo que nos referimos es que para el procesado de nuevas imágenes, una ventana del mismo tamaño que las

imágenes con las que ha sido entrenado el clasificador recorre toda la imagen comprobando si cada una de las porciones de la imagen contiene el objeto deseado.

El problema de este método es que nos reconocerá muchos más objetos de lo que realmente tenemos en la imagen. Debido a que la ventana recorre linealmente la imagen, por lo que reconocerá el mismo objeto varias veces en ventanas muy próximas, como podemos deducir que es totalmente lógico, ya que esas ventanas serán muy similares y por lo tanto tendrán unas características muy parecidas entre sí. Por ello se aplica el segundo paso sobre los objetos reconocidos, que es la eliminación del solapamiento de objetos mediante la técnica de *Non-Maximum suprresion*.

Como previa experimentación con esta metodología, vamos a entrenar el clasificador para el reconocimiento de caras. Y en función de la efectividad del método sobre las caras tomaremos una serie de conclusiones sobre las que decidamos si llevar a cabo esta solución de nuestro problema.

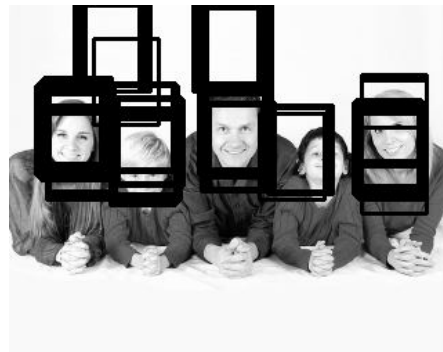
Aplicación sobre el reconocimiento de caras

A modo de aclaración de las explicaciones anteriores y para que podamos apreciar los resultados, vamos a ver algunas de las imágenes resultantes del clasificador.

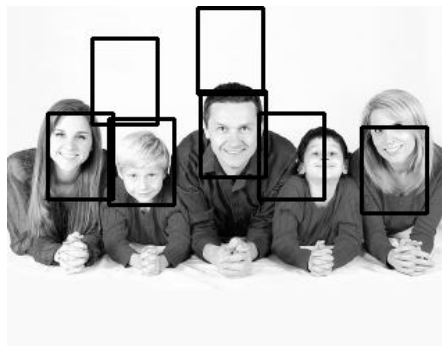
Como explicábamos anteriormente, una vez tenemos nuestro clasificador le enviamos una nueva imagen, como podría ser la presentada en la figura 5.4a. A partir de esta imagen el clasificador nos permitirá obtener las ventanas en las que detecta una cara, como vemos en la figura 5.4b. Podemos apreciar que existe más de una ventana alrededor de cada cara. Y finalmente, tras aplicar el método *Non-Maximum suprresion* obtenemos el resultado final mostrado en la figura 5.4c.



(a) Imagen original



(b) Imagen después de aplicarla nuestro clasificador



(c) Imagen tras aplicarla *Non-Maximum suprresion*

Trabajos relacionados

Conclusiones y Líneas de trabajo futuras

Bibliografía

- [1] Wikipedia. Support vector machine — Wikipedia, the free encyclopedia, 2016. [Online; accedido 22-Febrero-2017].
- [2] Wikipedia. Visual descriptor — Wikipedia, the free encyclopedia, 2016. [Online; accedido 22-Febrero-2017].
- [3] Wikipedia. Histogram of oriented gradients — Wikipedia, the free encyclopedia, 2017. [Online; accedido 22-Febrero-2017].
- [4] Wikipedia. Image segmentation — Wikipedia, the free encyclopedia, 2017. [Online; accedido 10-Febrero-2017].
- [5] Wikipedia. Otsu's method — Wikipedia, the free encyclopedia, 2017. [Online; accedido 8-Febrero-2017].