

UNIVERSITAT OBERTA DE CATALUNYA



**TRABAJO FINAL DE MÁSTER**

# **CLASIFICACIÓN AUTOMÁTICA DE CUADROS**

Autor: Pablo Belmonte Hernández

**JUNIO - 2011**

# ÍNDICE

	Página
1 INTRODUCCIÓN .....	1
2 ESTADO DEL ARTE .....	2
3 HERRAMIENTAS .....	4
4 ESTILOS ARTÍSTICOS .....	5
5 DESARROLLO .....	7
5.1 Creación de la base de datos .....	7
5.2 Extracción de características de las imágenes .....	8
5.3 Clasificación SVM .....	11
5.4 Ensayos .....	12
6 CÓDIGO .....	13
7 CONCLUSIONES Y TRABAJOS FUTUROS .....	20
8 BIBLIOGRAFÍA .....	21

## 1. INTRODUCCIÓN

La visión por computador está experimentando un gran avance en los últimos años, debido en gran parte al aumento de la potencia de los ordenadores. Se está investigando en muchas áreas, como biomédica, industrial, aeroespacial, seguridad, robótica...

Una de las principales líneas es el reconocimiento automático. Se está avanzando mucho sobre algoritmos para el reconocimiento de caras, ojos, señales de tráfico, matrículas, personas etc. Sobre el reconocimiento automático de obras artísticas no se ha trabajado mucho.

En los últimos años están creciendo las galerías de arte online y los museos más importantes están empezando a digitalizar sus obras. Los algoritmos para el reconocimiento de obras de arte podrían ayudar en esta tarea.

Este proyecto intentará clasificar obras de arte según su estilo. Para ello se creará una base de datos de cuadros. Seguidamente se desarrollará un software que al introducir un nuevo cuadro, lo clasifique en el estilo al que corresponda.

## 2. ESTADO DEL ARTE

### CLASSIFICATION AND INDEXING OF PAINTINGS BASED ON ART MOVEMENTS [1]

En este trabajo se recoge el proceso de extracción de las características de unas determinadas obras de arte de la época del clasicismo, impresionismo o cubismo, así como el desarrollo de un sistema capaz de clasificarlas según su movimiento artístico.

Para mostrar las características y el contenido de las imágenes se han establecido seis diferentes dimensiones. Una vez se han analizado las seis dimensiones, se pone en común y así se deduce a qué movimiento artístico pertenece, de forma eficiente.

En el diseño del clasificador, el reconocimiento estadístico de patrones se explota por clasificadores bayesianos, k-NN y SVM. Gracias a este método, se consigue una exactitud en la clasificación de más del 90%, mientras que el menor rendimiento se obtiene gracias al k-NN. Este sistema también ofrece una eficiente y eficaz base de datos, obtenidos a base de consultas y búsqueda por indexación de los cuadros, con sus seis dimensiones, y ofrece un programa de aplicación para los museos y centros de exposiciones.

Este paper es un trabajo muy parecido al proyecto de fin de master a realizar. Consiste en extraer unas características de unas obras de arte y luego clasificarlas según el movimiento al que pertenezcan. Se pueden clasificar en 3 movimientos por lo que la clasificación es un poco reducida pero obtienen unas buenas precisiones a la hora de clasificarlas.

## STOCHASTIC MODELING WESTERN PAINTINGS FOR EFFECTIVE CLASSIFICATION [2]

En una de las mayores herencias culturales, la pintura clásica occidental ha tenido siempre un papel relevante en la Historia de la Humanidad. Mientras que la clasificación de imágenes es uno de los temas más publicados, poca atención se le ha dado a la automática clasificación de éstas, aspecto que se podría convertir en la clave técnica de lo que podríamos llamar la moderna biblioteca, el moderno museo o la moderna galería de arte digital.

Este *paper* se centra en estudiar la clasificación automática de una amplia colección de pinturas desde un nuevo punto de vista, desde un marco de análisis nuevo. Con este marco, divesas características visuales se integran de manera efectiva para mejorar la precisión en el proceso de identificación. A lo largo de este artículo se evalúa también el propio método y sus competidores.

Los resultados empíricos de un estudio indican que este nuevo método de estudio goza de gran superioridad sobre el resto de procesos de clasificiación de pinturas, en muy distintos aspectos.

El presente *paper* es similar a nuestro tema de investigación. La técnica empleada es muy parecida a la que se desarrollará en el proyecto fin de master. El objetivo es clasificar unas obras de arte según el estilo al que pertenecen. Para ello, primero se extraen las características y luego las clasifica mediante redes neuronales.

### 3. HERRAMIENTAS

Para el desarrollo del presente proyecto se ha usado el entorno de desarrollo Eclipse instalado sobre Ubuntu 10.10. El programa se ha desarrollado en C++ por lo que ha sido necesario instalar el complemento CDT de eclipse. [3]

Como librería de visión para trabajar con las imágenes, se ha usado Opencv, elegida por su potencia, flexibilidad y popularidad. [4]

Las imágenes han sido descargadas de diferentes páginas webs. Estando detallados los enlaces de cada imagen en la base de datos.

## 4. ESTILOS ARTÍSTICOS

Para el presente proyecto se han elegido 4 estilos artísticos: impresionismo, romanticismo, cubismo y constructismo.

### **Impresionismo**

El impresionismo es un movimiento pictórico francés de finales del siglo XIX que apareció como reacción contra el arte académico. [5]

Los paisajes son el tema en torno al que gira este movimiento pictórico. Se caracteriza por la eliminación del color negro al eliminarse las sombras muy oscuras, y del color blanco puro.

Las características más comunes del impresionismo son el uso de los colores puros sin mezcla, la saturación y abundancia de color, contraste cromático, abundancia de colores claros y brillantes y la casi ausencia de negros. Los pintores tiende a pintar libremente, de forma suelta y despojada, utilizando pequeñas pinceladas fragmentadas.

### **Romanticismo**

El Romanticismo en la pintura se da en la primera mitad del siglo XIX, y surge con una nueva sensibilidad caracterizada por conceder un valor primordial al sentimiento y al individuo. [6]

En la pintura romántica lo que más resalta es la exaltación del colorido. La pincelada es libre, viva y llena de expresividad. Desaparece la línea frente al color. Se recupera la potencia sugestiva del color, liberándose las formas y los límites excesivamente definidos.

La luz tiene una gran importancia y se cuidan sus gradaciones. Las composiciones están marcadas por las líneas curvas y los gestos dramáticos.

Destaca por la variedad de temas como el paisaje, las revoluciones políticas, los desastres o lo fantástico.

## **Cubismo**

Los orígenes del Cubismo giran en torno a 1907, fecha en la que Picasso concluye *Las Señoritas de Avignon*, que será el punto de partida. [7]

Se caracteriza por la representación anti realista, plana y bidimensional de un objeto visto desde múltiples ángulos.

Fragmenta las formas y el espacio en planos interrelacionados por el color y las líneas. Carece de profundidad espacial, y tiene austeridad cromática. Emplea figuras geométricas como triángulos, cuadrados, rectángulos y cubos. Los colores utilizados suelen ser apagados, y la luz es escasa. Los temas principales de los cuadros cubistas suelen ser los retratos y las naturalezas muertas.

## **Constructismo**

El constructivismo surgió en el siglo XX en Rusia, y es el movimiento artístico que incorpora a sus obras los conceptos de espacio y de tiempo a fin de conseguir formas dinámicas. [8]

Entre las características de estilo está el uso de los colores naranja, rojo, azul, amarillo, negro y blanco (generalmente siempre en los mismos tonos), la constante alusión a elementos modernos que simbolizan el progreso, las estructuras geométricas y las formas pesadas.

La obra se abre por todas partes hacia el espacio y consta de elementos, frecuentemente transparentes, de formas geométricas, lineales y planas. Hace hincapié en lo abstracto, relacionado con la industria y la técnica, y es geométrico y funcional. Utiliza materiales que permitan que el espacio penetre en la escultura, como alambre, madera, vidrio, yeso y plástico.



## 5. DESARROLLO

Para el desarrollo del proyecto se han seguido los siguientes pasos:

1. Creación de la base de datos
2. Extracción de características de los cuadros
3. Clasificación mediante SVM
4. Ensayos

### 5.1 Creación de la base de datos

Para la creación de la base de datos se han buscado cuadros de 4 estilos artísticos diferentes: romanticismo, cubismo, impresionismo y constructismo. En un principio, se empezaron a buscar en galerías de arte online, así como en las páginas webs de los museos pero la calidad con la que se podían descargar era bastante pobre. Por ello se buscaron directamente desde google obras con una buena resolución.

Se realizó una hoja de cálculo con los siguientes campos:

- Fichero: Nombre del fichero del cuadro
- Nombre: Nombre de la obra
- Estilo: Estilo al que pertenece la obra
- Autor: Autor de la obra
- Año: Año de creación del cuadro
- Ancho: Ancho en píxeles de la obra
- Alto: Alto en píxeles de la obra
- Tamaño: Tamaño en Mega píxeles del cuadro
- Enlace: Enlace de internet donde se puede descargar la obra

Algunos campos no se han podido rellenar debido a la falta de información de algunas obras sobretodo las más nuevas.

La base de datos consta de 76 obras pertenecientes a 4 estilos artísticos: romanticismo, impresionismo, cubismo y constructismo.

## 5.2 Extracción de características de los cuadros

Para este trabajo se han extraído 6 características diferentes de cada imagen. Con ello se obtiene un vector de características de 6 dimensiones. Las características son:

$\mu_1$ : Porcentaje de colores oscuros

$\mu_2$ : Coeficiente del gradiente

$\mu_3$ : Número de máximos locales y globales en el histograma de luminancia

$\mu_4$ : rango de color al que corresponde el pico del histograma de luminancia

$\mu_5$ : Después de partir la imagen en 9 bloques iguales, desviación del nivel medio de gris de cada bloque y el nivel medio de gris de la imagen completa.

$\mu_6$ : Desviación de la distribución del nivel de gris de la distribución de Gauss

La luminancia de una imagen en color se define como:

$$Y = 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B$$

Hay mucha relación entre la luminancia de una imagen y su nivel de gris. En este trabajo la luminancia se representa con bits (256 niveles de gris) y los píxeles cuya luminancia sea menor de 64 son considerados píxeles oscuros.

$$\mu_1 = \frac{\text{número de píxeles oscuros}}{\text{número de píxeles de la imagen}}$$

$$\mu_2 = \text{Normalizado} \left( \sum_{i=1}^r \sum_{j=1}^c \sqrt{(f_{ij})_x^2 + (f_{ij})_y^2} \right)$$

Donde r y c son las filas y las columnas de la imagen y  $(f_{ij})_x$  es el gradiente en la dirección x del pixel ij.

$$\mu_3 = \text{número} ( |H(i)| > \text{thr} )$$

Donde H(i) es el valor del histograma para el nivel i de gris y thr es un umbral que en nuestro caso ha sido la mitad del máximo valor del histograma. Seguidamente se normaliza dividiéndolo entre 256.

$$\mu_4 = \text{max}_i |H(i)|$$

Es el máximo valor del histograma.

$$\mu_5 = \sum_{i=1}^9 (mean_i - mean)$$

Para hallar  $\mu_5$  hay que dividir la imagen en 9 bloques de igual tamaño. A continuación se halla la media de cada bloque y se resta con la media de toda la imagen. Siendo *mean*

$$mean = \frac{\sum_{i=1}^r \sum_{j=1}^c f_{ij}}{rc}.$$

Por último se halla  $\mu_6$

$$\mu_6 = skew = \frac{\sum_{i=1}^r \sum_{j=1}^c \left( \frac{f_{ij} - mean}{\sigma} \right)^3}{rc}$$

$$\sigma = \frac{\sqrt{\sum_{i=1}^r \sum_{j=1}^c (f_{ij} - mean)^2}}{rc - 1}$$

Una vez extraídas las características de las imágenes de prueba, se guardan en un fichero y en una matriz.

Cada imagen tendrá un vector de tamaño 6 con las características como el que se muestra a continuación:

$\mu = (0.071011, 0.054311, 0.484375, 0.008500, 141.772339, -0.065191)$

### 5.3 Clasificación de imágenes mediante SVM

Support Vectors Machines (SVM) son un conjunto de algoritmos de aprendizaje supervisado desarrollados en los laboratorios AT&T. Están relacionados con problemas de clasificación y regresión. [9]

Dado un conjunto de puntos pertenecientes a varios grupos, el algoritmo SVM construye un modelo capaz de predecir a que grupo pertenece un nuevo punto.

OpenCv ya tiene implementados los algoritmos de SVM en la librería ml.h.

Para usarlos tenemos que crear 3 matrices:

- Matriz con las características de los datos de entrenamiento. En este caso es una matriz de 76 filas y 6 columnas (76 cuadros)
- Vector con el estilo al que pertenece cada cuadro de la matriz anterior: Vector que contiene números del 1 al 4 dependiendo del estilo al que pertenezca cada cuadro.
- Matriz de entrada: Matriz con las características de los cuadros que queremos clasificar. En este caso es de 16 filas y 6 columnas.

### 5.4 Ensayos

Se han introducido 76 obras de arte de las 4 categorías mencionadas para crear el modelo SVM. A continuación se han introducido 16 obras (4 de cada categoría) de las cuales ha clasificado correctamente 14 (87,5%).

Los resultados son parecidos a los dados en el paper “Classification and indexing of paintings based on art movements”, donde clasificaban cuadros de 3 estilos diferentes. Aquí se han clasificado entre 4 estilos y el porcentaje de aciertos es algo menor como cabía de esperar.

## 6. CÓDIGO FUENTE

```
//=====
// Name      : pfm.cpp
// Author    : Pablo Belmonte
// Version   :
// Copyright : Your copyright notice
// Description : Hello World in C++, Ansi-style
//=====

#include <iostream>
#include <stdlib.h>
#include <stdio.h>
#include <math.h>
#include <cv.h>
#include <highgui.h>
#include "imagen.h"
#include <ml.h>

using namespace std;
using namespace cv;

int main(int argc, char *argv[]) {
    float array_trainData[76][6];
    float array_entrada[16][6];
    FILE *fp;
    fp = fopen("datos.txt", "w");
    imagen I;
    IplImage* img2 = 0;
    IplImage* img3 = 0;
    char arch[20]="imagenes/";
    char arch2[30]="imagenes_entrada/";
    int altura,anchura,i,r;
    float *u5_u6;
    float u1,u2,u3,u4;
    for(i=1;i<77;i++){
if(i<10){
    arch[9]=i+48;
    arch[10]='.';
    arch[11]='j';
    arch[12]='p';
    arch[13]='g';
    arch[14]='\0';
}
if(i>9){
    arch[9]=i/10+48;
    arch[10]=i%10+48;
    arch[11]='.';
    arch[12]='j';
    arch[13]='p';
    arch[14]='g';
    arch[15]='\0';
}

    img2=I.abrir_imagen(arch);
    img3=I.abrir_imagen_gris(arch);
}
```

```

altura    = img2->height;
anchura   = img2->width;

u1=I.get_u1(img2,altura,anchura);
u2=I.get_u2(img3,altura,anchura);
u3=I.get_u3(img3,altura,anchura);
u4=I.get_u4(img3,altura,anchura);
u5_u6=I.get_u5_u6(img3,altura,anchura);
I.escribir(fp,u1,u2,u3,u4,u5_u6[0],u5_u6[1]);

array_trainData[i-1][0]=u1;
array_trainData[i-1][1]=u2;
array_trainData[i-1][2]=u3;
array_trainData[i-1][3]=u4;
array_trainData[i-1][4]=u5_u6[0];
array_trainData[i-1][5]=u5_u6[1];

}

//SVM

for(i=1;i<17;i++){
if(i<10){
    arch2[17]=i+48;
    arch2[18]='.';
    arch2[19]='j';
    arch2[20]='p';
    arch2[21]='g';
    arch2[22]='\0';
}
if(i>9){
    arch2[17]=i/10+48;
    arch2[18]=i%10+48;
    arch2[19]='.';
    arch2[20]='j';
    arch2[21]='p';
    arch2[22]='g';
    arch2[23]='\0';
}
img2=I.abrir_imagen(arch2);
img3=I.abrir_imagen_gris(arch2);
altura    = img2->height;
anchura   = img2->width;
printf("\n archivo: %s",arch2);

u1=I.get_u1(img2,altura,anchura);
u2=I.get_u2(img3,altura,anchura);
u3=I.get_u3(img3,altura,anchura);
u4=I.get_u4(img3,altura,anchura);
u5_u6=I.get_u5_u6(img3,altura,anchura);
    array_entrada[i-1][0]=u1;
    array_entrada[i-1][1]=u2;
    array_entrada[i-1][2]=u3;

```





```

    int escribir(FILE *fichero, float u1, float u2, float u3, float u4, float
u5, float u6);

```

```
};
```

```
#endif /* IMAGEN_H_ */
```

```
/*
```

```
 * imagen.cpp
```

```
 *
```

```
 * Created on: 02/04/2011
```

```
 * Author: pablo
```

```
*/
```

```
#include "imagen.h"
```

```
imagen::imagen() {
    // TODO Auto-generated constructor stub

```

```
}
```

```
imagen::~imagen() {
    // TODO Auto-generated destructor stub

```

```
}
```

```
IplImage* imagen::abrir_imagen(char* nombre){
    IplImage* img = 0;
    int altura, anchura, anchura_fila, canales;
    uchar *data;
    int i, j;
    img=cvLoadImage(nombre);
    if(!img){
        printf("No se ha podido cargar la imagen\n");
        exit(0);
    }

```

```

    altura    = img->height;
    anchura   = img->width;
    anchura_fila = img->widthStep;
    canales   = img->nChannels;
    data      = (uchar *)img->imageData;
    return (img);

```

```
}
```

```
IplImage* imagen::abrir_imagen_gris(char* nombre){
    IplImage* img = 0;
    img=cvLoadImage(nombre,0);
    if(!img){
        printf("\nNo se ha podido cargar la imagen %s", nombre);
        exit(0);
    }

```

```

    return (img);
}

float imagen::get_u1(IplImage* imag, int alto, int ancho){
    int i=10,j=20,dark=0;
    float y;
    float u1;
    CvScalar s;
    for(i=0;i<alto;i++) for(j=0;j<ancho;j++){
        s=cvGet2D(imag,i,j);
        y=0.299*s.val[2]+0.587*s.val[1]+0.114*s.val[0];
        if(y<64)
            dark++;
    }
    u1=dark/float(alto*ancho);
    return(u1);
}

float imagen::get_u2(IplImage* imag, int alto, int ancho){

    int i=10,j=20,dark=0;
    float y,suma=0;
    float u2;
    CvScalar s,s2,s3;
    for(i=0;i<alto-1;i++) for(j=0;j<ancho-1;j++){
        s=cvGet2D(imag,i,j);
        s2=cvGet2D(imag,i,j+1);
        s3=cvGet2D(imag,i+1,j);
        suma=suma+sqrt((s2.val[0]-s.val[0])*(s2.val[0]-s.val[0])+
(s3.val[0]-s.val[0])*(s3.val[0]-s.val[0]));
    }
    u2=suma/float(alto*ancho*256);
    return(u2);
}

float imagen::get_u3(IplImage* imag, int alto, int ancho){

    int bins = 256,suma=0;
    int hsize[] = {bins};
    float max_value = 0, min_value = 0,u3=0;
    float value;
    int normalized;
    float xranges[] = { 0, 256 };
    float* ranges[] = { xranges };
    CvHistogram* hist;
    IplImage* planes[] = { imag };
    hist = cvCreateHist( 1, hsize, CV_HIST_ARRAY, ranges,1);
    cvCalcHist( planes, hist, 0, NULL);
    cvGetMinMaxHistValue( hist, &min_value, &max_value);
    for(int i=0; i < bins; i++){
        value = cvQueryHistValue_1D( hist, i);
        if(value>(max_value*0.5)){
            suma++;
        }
    }
    u3=float(suma)/256.0;
    return(u3);
}

```

```
}

```

```
float imagen::get_u4(IplImage* imag, int alto, int ancho){
    int bins = 256,suma=0;
    int hsize[] = {bins};
    float u4;
    float max_value = 0, min_value = 0,u3=0;
    float value;
    int normalized;
    float xranges[] = { 0, 256 };
    float* ranges[] = { xranges };
    CvHistogram* hist;
    IplImage* planes[] = { imag };
    hist = cvCreateHist( 1, hsize, CV_HIST_ARRAY, ranges,1);
    cvCalcHist( planes, hist, 0, NULL);
    cvGetMinMaxHistValue( hist, &min_value, &max_value);
    u4=max_value/float(ancho*alto);
    return(u4);
}

```

```
int imagen::escribir(FILE *fichero, float u1, float u2, float u3, float u4, float
u5, float u6){
    fprintf(fichero, "\n%f %f %f %f %f %f", u1, u2, u3, u4, u5, u6);
    return(1);
}

```

```
float* imagen::get_u5_u6(IplImage* imag, int alto, int ancho){
    int i, j;
    float u5_u6[2];
    float mean[9], mean_t, u5=0, sigma=0, u6=0;
    float suma=0, suma2[9], cont[9], pix=0;
    CvScalar s, s2, s3;
    for(i=0; i<9; i++){
        suma2[i]=0;
        cont[i]=0;
    }

    for(i=0; i<alto; i++) for(j=0; j<ancho; j++){
        s=cvGet2D(imag, i, j);
        suma=suma+s.val[0];

        if(i<=alto/3 && j<=ancho/3){
            suma2[0]=suma2[0]+s.val[0];
            cont[0]++;
        }
        if(i<=alto/3 && j>ancho/3 && j<=2*ancho/3){
            suma2[1]=suma2[1]+s.val[0];
            cont[1]++;
        }
        if(i<=alto/3 && j>2*ancho/3){
            suma2[2]=suma2[2]+s.val[0];
            cont[2]++;
        }
        if(i>alto/3 && i<=2*alto/3 && j<=ancho/3){
            suma2[3]=suma2[3]+s.val[0];
            cont[3]++;
        }
    }
}

```

```

        if(i>alto/3 && i<=2*alto/3 && j>ancho/3 && j<=2*ancho/3){
            suma2[4]=suma2[4]+s.val[0];
            cont[4]++;
        }
        if(i>alto/3 && i<=2*alto/3 && j>2*ancho/3){
            cont[5]++;

            suma2[5]=suma2[5]+s.val[0];
        }
        if(i>2*alto/3 && j<=ancho/3){
            suma2[6]=suma2[6]+s.val[0];
            cont[6]++;
        }
        if(i>2*alto/3 && j>ancho/3 && j<=2*ancho/3){
            suma2[7]=suma2[7]+s.val[0];
            cont[7]++;
        }
        if(i>2*alto/3 && j>2*ancho/3){
            suma2[8]=suma2[8]+s.val[0];
            cont[8]++;
        }
    }

    for(i=0;i<9;i++){
        mean[i]=suma2[i]/cont[i];
        pix=pix+cont[i];
    }
    mean_t=suma/float(ancho*alto);
    for(i=0;i<9;i++){
        u5=u5+abs(mean[i]-mean_t);
    }

    for(i=0;i<alto;i++) for(j=0;j<ancho;j++){
        s=cvGet2D(imag,i,j);
        sigma=sigma+((s.val[0]-mean_t)*(s.val[0]-mean_t));
    }

    sigma=sigma/float(alto*ancho-1);
    sigma=sqrt(sigma);
    for(i=0;i<alto;i++) for(j=0;j<ancho;j++){
        s=cvGet2D(imag,i,j);
        u6=u6+(((s.val[0]-mean_t)/sigma)*((s.val[0]-mean_t)/sigma)*((s.val[0]-
        mean_t)/sigma));
    }
    u6=u6/float(ancho*alto);
    u5_u6[0]=u5;
    u5_u6[1]=u6;
    return(u5_u6);
}

```

## 7. CONCLUSIONES Y TRABAJOS FUTUROS

El presente trabajo es capaz de clasificar cuadros correspondientes a 4 estilos artísticos diferentes con una precisión buena. Para ello extrae 6 características de cada obra y los clasifica mediante SVM.

Hay dos líneas en las que se puede mejorar: aumentar la precisión de clasificación o aumentar el número de estilos artísticos a clasificar.

Para aumentar la precisión pueden extraerse más características que diferencien mejor unos estilos de otros. Así mismo se pueden introducir más estilos artísticos y probar con otros clasificadores como puede ser K—nearest Neighbour, Bayes, redes neuronales, etc.

## 8. BIBLIOGRAFÍA

- [1] O. Icoglu, B. Günsel, S. Sariel, “Classification and Indexing of paintings based on art movements”, Maslak-Instanbul, Turkey, <http://www.ehb.itu.edu.tr/~bgunsel/mspr>
- [2] J. Shen, “Stochastic modeling western paintings for effective classification ”, Pattern Recognition , April 2008 Singapore
- [3] Página web: <http://www.eclipse.org/cdt/>
- [4] Opencv, manuales y tutoriales: <http://opencv.willowgarage.com/wiki/>
- [5] Pagina web: <http://arte.laguia2000.com/pintura/caracteristicas-del-impresionismo-i>
- [6] Pagina web: <http://www.slideshare.net/tomperez/pintura-del-romanticismo>
- [7] Pagina web: <http://www.pucpr.edu/facultad/pperez/101/caracgencubismo.htm>
- [8] Pagina web: <http://www.todacultura.com/movimientosartisticos/constructivismo.htm>
- [9] Documentación SVM:
- [http://opencv.willowgarage.com/documentation/cpp/support\\_vector\\_machines.html](http://opencv.willowgarage.com/documentation/cpp/support_vector_machines.html)
  - <http://littlesaiph.blogspot.com/2010/05/support-vector-machine-tutorial-with.html>
  - [http://www.cognotics.com/opencv/docs/1.0/ref/opencvref\\_ml.htm](http://www.cognotics.com/opencv/docs/1.0/ref/opencvref_ml.htm)
  - Chih-Chung Chang and Chih-Jen Lin, LIBSVM : a library for support vector machines, 2001. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.