

Estética computacional aplicada a la industria automotriz

José Luis Crego Quirante

Resumen– La industria automotriz está cada vez más capacitada para ofrecer a sus clientes vehículos que no sólo cumplen con las prestaciones necesarias sino que también resultan estéticamente atractivos para el comprador. Este es un campo muy amplio y poco explorado ya que las características de un vehículo que lo hacen estéticamente atractivo son muchas y la variedad de gustos de los clientes también. Este trabajo tiene como punto de partida la creación de una base de datos de imágenes automóbiles de bajo contenido semántico (con mínima información sobre “marca”, “modelo”, “social target” etc.) categorizada de acuerdo a las preferencias de un grupo piloto de usuarios. El paso siguiente es la extracción de características de bajo nivel (brillos, color, contornos, etc.) para aplicar métodos de clasificación computacional.

Palabras clave– Estética computacional, DOF, MIDR, ROI, máscara binaria, valores extremos, keypoints, DoG, DBSCAN, regresión lineal, categorización de colores, NICE, HSV.

Abstract– The automotive industry is increasingly able to offer its customers vehicles that not only meet the required performance but are also aesthetically attractive to the buyer. This is a very broad field and little explored since the characteristics of a vehicle that make it aesthetically appealing are many and the variety of tastes of the customers as well. This work has as its starting point at the creation of a database of automobile images of low semantic content (with minimal information about “brand”, “model”, “social target. etc.) categorized according to the preferences of a group User pilot. The next step is the extraction of low-level features (brightness, color, contours, etc.) to apply computational classification methods.

Keywords– Computational aesthetics, DOF, MIDR, ROI, binary mask, maxima values, keypoints, DoG, DBSCAN, linear regression, colour categorization, NICE, HSV.

1 INTRODUCCIÓN

SE puede definir la experiencia estética como una combinación entre dos canales principales de recepción y codificación de información, mostrados en la figura[1]. El modelo de Redies [1], en resumen, nos dice que cuando observamos una imagen, una fotografía, una obra de arte, una prenda de ropa, etc. el proceso hasta obtener una experiencia estética se divide en 2 canales que operan paralelamente:

- Sensorial (belleza universal): referente a las teorías formalistas que nos dicen que existe una belleza

- E-mail de contacto: crego.joseluis@gmail.com
- Mención realizada: Computación
- Trabajo autorizado por: Carlos Alejandro Párraga, Computer Sciences Department (UAB) y Computer Vision Center (CVC)
- Curso 2016/17

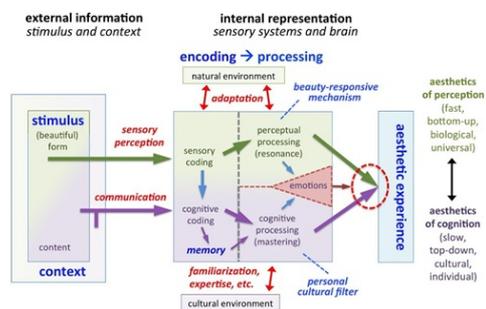


Fig. 1: Modelo del proceso de experiencia estética por Christoph Redies.

intrínseca en el proceso visual de cada uno, en nuestra percepción natural de la belleza como tal. Nuestro sentido visual recibe información de lo que estamos viendo (form) independientemente de lo que sea y lo procesa en un entorno sensorial (interpretación natu-

ral de los colores, líneas, esquinas, iluminación...) que produce una respuesta estética biológica.

- Cognitivo (contexto cultural): referente al dónde observamos el elemento a evaluar (contexto) y el qué estamos viendo (content). Esta información se procesa en las neuronas de nuestro cerebro (cognitive processing) junto a un filtro cultural personal (de dónde venimos, quiénes somos, época, recuerdos) y produce una respuesta estética cognitiva.

Se producirá una respuesta estética positiva final si tanto el procesamiento cognitivo como sensorial tienen una respuesta positiva **o simplemente existe una percepción sensorial positiva sin existencia de procesamiento cognitivo.**

2 METODOLOGÍA DE TRABAJO

2.1. Sistema de trabajo

Puesto que este proyecto es un estudio exploratorio o de investigación, no se sabía a ciencia cierta por donde poder avanzar con paso firme. Debido a esto no se ha seguido una metodología en concreto sino que se han aprovechado los resultados experimentales de donde poder extraer hipótesis coherentes y después tratar de extrapolarlas al conjunto global de coches. No obstante, y gracias a la ayuda del tutor he podido tener ese punto de vista necesario que da la experiencia para saber por dónde encaminar el proyecto en cada momento.

2.2. Herramientas utilizadas

A continuación se describen las herramientas utilizadas para el desarrollo de todas y cada una de las etapas del proyecto:

- MATLAB 2014 para el desarrollo del software.
- Documentación:
 - Plataforma online Overleaf para informes en LATEX.
 - Microsoft Project 2013 para la planificación del proyecto.
- Seguimiento de la evolución del proyecto:
 - Microsoft Office 2013.
 - Plataforma Mega/MEGASync para sincronizar documentos con el tutor.
 - Netbeans para actualizar los documentos HTML y CSS de la plataforma web para el dossier.
 - Plataforma online Bitbucket para la gestión de la configuración.
- Adobe Photoshop CS7 para la edición del dataset de imágenes de coches.
- Plataforma online Doodle para realizar encuestas.

3 ESTADO DEL ARTE Y OBJETIVOS

EL objetivo principal es encontrar cuáles son los descriptores que tienen mayor influencia en los usuarios a la hora de decidir una valoración estética de coches. Para conseguirlo se aplican técnicas y procedimientos computacionales que nos permiten extraer y clasificar información de los coches para después aplicar una clasificación. No obstante, la estética computacional es un campo emergente y del que no existen datos comprobados que argumenten un patrón o *modus operandi* para llegar a una experiencia estética positiva. Además, este campo se desarrolla para el análisis de las obras de arte, no existe actualmente ninguna aplicación al mercado automotriz. Por lo tanto este proyecto pretende ser un estudio exploratorio para llegar a conclusiones de gustos sobre coches, en función únicamente de la estética que se observa.

El proyecto consta de un software de extracción y estudio de características desarrollado exclusivamente para el propósito en cuestión que nos sirve para interpretar los gustos con más peso intentando responder a preguntas como ¿es este coche más bello que este otro? ¿Por qué? Para ello se trabaja únicamente sobre el estímulo sensorial que producen las imágenes de diferentes coches en el ser humano y se separan del canal cognitivo, que en este caso se han considerado la marca y modelo de los mismos editando las imágenes de muestra.

4 BASE DE DATOS DE IMÁGENES

4.1. Presentación del coche en la imagen

La primera etapa del proyecto se basa en la recopilación de información y enfoque del proyecto en cuanto a la estética y el cómo se interpreta nuestra percepción de la misma, así como el enfoque que le daremos al proyecto para poder obtener resultados fehacientes. Por lo tanto lo primero que debemos tener claro son las características que deben tener en común todas las imágenes de nuestro dataset. Para ello nos hemos basado en alguna técnicas estéticas que utilizan los fotógrafos [2]:

- Por un lado la técnica depth-of-field (DOF): esta técnica se basa en apurar al máximo el rango de la cámara al realizar la fotografía de un objeto, dejando con efecto suavizado el fondo o resto de la imagen y dando una especial importancia a aquello que se quiere resaltar. En nuestro caso, esta focalización recaerá sobre la región de la imagen correspondiente al coche. Para ello se ha creado un resalte y enfoque de bordes y detalles previo a todas las imágenes de los coches.
- Por otro lado, muchos fotógrafos de paisajes recomiendan tener en proporciones equiparables el objeto que se analiza, con un fondo, apareciendo éste objeto centrado en imagen y ocupando 3/4 partes del total, pero sin eclipsar nunca el total de la imagen. Por lo tanto, un requisito indispensable de nuestras imágenes será que nuestro coche aparezca centrado en imagen pero sin ocupar la totalidad de la misma.

4.2. Características de los coches

Con objetivo de acotar el dominio del problema, se ha creado un dataset de imágenes que resumen estrictamente las siguientes características:

- Perspectiva isométrica.
- Un único matiz principal de pintura.
- Tamaño de imagen de 1024x768 píxeles.
- Eliminación de ruido contextual y filtro cultural en la medida de lo posible según el planteamiento inicial del proyecto. Para ello:
 - Se han eliminado los logotipos de marcas.
 - Se ha eliminado cualquier señal que indique un modelo o posible modelo del coche. Por ejemplo, las letras “FR” en un coche indican que estamos ante un coche de marca SEAT.
 - Se ha eliminado de las imágenes de los coches matriculados el país de procedencia.
- Las marcas seleccionadas para ello han sido marcas que por lo general puedan pasar desapercibidas si se aplica el punto anterior a las imágenes y además respetando el top 10 del ránking de marcas de coches matriculados [3] según la Asociación española de Fabricantes de Automóviles y Camiones (ANFAC). Son las siguientes: Chevrolet, Chrysler, Dacia, Daihatsu, Infiniti, Lancia, Lexus, Lincoln, Mazda, Skoda, Subaru y Volvo.

4.3. Eligiendo colores

A la hora de seleccionar el color de los coches surge un problema importante: la variedad. En cuanto al mercado automovilístico, los colores preferidos en Europa occidental son comunes y se repiten continuamente, hasta el punto de llegar a un 92 % de variedad cromática entre sólo 6 colores [4]: blanco, negro, gris, plateado, azul y rojo (Moreno, 2016). El resto de colores se agrupan por colores naturales, y otras variantes. Para suplir esta falta de colores poco comunes se ha utilizado el algoritmo Multi Illuminant Dichromatic Reflection model (MIDR) [5]. Este algoritmo permite seleccionar el área de una imagen determinada y cambiar su color por otro distinto a original. No obstante es mucho más que eso, ya que conserva todos los brillos y saturación de color de la zona onde se intercambia el color y además permite recoger la información de la luz que reflejan varios objetos sobre la zona donde se realiza el cambio, manteniendo la influencia de colores sobre la misma. Véase la imagen [2] para apreciar el resultado de aplicar del algoritmo. Debido a que este algoritmo está basado en el espacio de color RGB, es más ajustado para aquellos colores que son cercanos a estos tres canales, motivo por el cual los colores elegidos a introducir han sido el cian, verde limón y rosa/fucsia en función de la saturación y brillo de la pintura del coche. El dataset definitivo de imágenes contiene de un total de 87 imágenes, divididas en tres tipos de coche (urbanos, sedan y crossover) alternando entre todos ellos características de coches comunes que podemos encontrar por la calle, características de coche clásico, y características deportivas, en

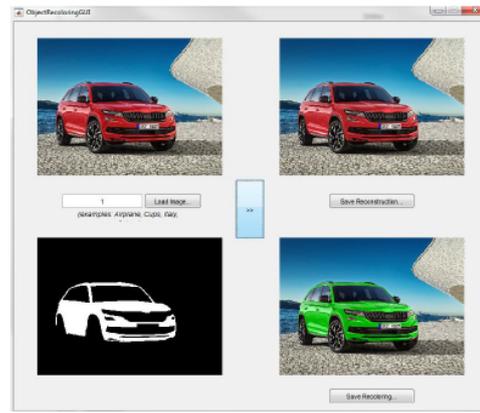


Fig. 2: Cambio de color con el algoritmo MIDR del coche con ID 73 (Skoda Kodiaq Sportline del año 2018).

un rango de 15 años de antigüedad, siendo el más antiguo del año 2003 y el más nuevo un prototipo que saldrá a la venta el año 2018. No obstante se ha centrado el tipo de coche en aquellos que se pueden observar habitualmente en el día a día, evitando así gamas de lujo o de competición.

5 ENCUESTAS

Las encuestas pretenden ser una herramienta para llegar a la gente y obtener sus gustos puramente estéticos dejando de lado el canal cognitivo y el filtro cultural.

5.1. Planteamiento de las encuestas

Para cumplir el propósito anterior, siguen el modelo que se muestra a continuación:

- Explicación inicial: *Esta encuesta tiene el propósito de conocer la opinión puramente estética sobre los automóviles que se observan en las imágenes. Por lo tanto NO existen respuestas correctas o incorrectas, simplemente existen gustos personales. Se pide encarecidamente dar una puntuación que se ciña a la frase “¿cuán bonito es el coche para mí”.*
- Preguntar por el país de nacimiento.
- Por cada imagen pedir una puntuación de la belleza de cada coche del 1 al 10 en intervalos de 1 punto, donde el 1 hace referencia a *horrible* y el 10 a *bellísimo*. Además, correspondiente a cada imagen se pide que si se cree haber reconocido el modelo o marca del coche votado, se describa el mismo seguidamente.
- El dataset de imágenes se dividió en tres conjuntos de coches en proporciones de 31, 32 y 32 cada uno, siempre mirando que hubiera la misma cantidad de coches urbanos, sedan y crossover, así como variedad de gamas y colores. Esto es debido a que se ha querido preservar la lucidez e interés con la que se vota cada coche, haciendo menos monótono y laborioso para el votante el procedimiento.

5.2. Resultados obtenidos

El total de encuestas realizadas ha sido de 253, de las cuales se han eliminado 22 respuestas, dejando el total a 231. Los motivos de estos descartes se describen aquí:

- Un requisito ha sido el de eliminar el filtro cultural, por lo tanto se han descartado aquellas respuestas cuyos encuestados procedan de un territorio no español.
- Se han considerado votaciones no válidas aquellas en las que la puntuación no varía, por ejemplo, puntuar todas las muestras de coches con una nota de 1.

A la hora de analizar los resultados, y para lograr eliminar el ruido contextual, se han considerado como no válidos aquellos votos cuyos modelos o marcas de coche hayan sido reconocidos. Una vez descartada esta información, se ha procedido a realizar la media de votos de cada coche. No obstante, la media no es un indicativo suficiente para saber si se está haciendo justicia con la variedad de los gustos sobre un coche, por lo tanto se han analizado las votaciones de cada uno de los coches realizando un análisis de su histograma[3], comprobando la variación entre votaciones. Se ha realizado para encontrar alguno de los siguientes posibles casos:

- Histogramas con una gran acumulación de votos en valores muy bajos que no se encuentren entre los coches peor valorados.
- Histogramas con una gran acumulación de votos en valores muy altos que no se encuentren entre los coches mejor valorados.
- Histogramas con un máximo de curvatura en un punto y gran acumulación de votos en puntos alejados de dicha media . Estos tipos de histogramas implican una varianza de votos muy grande y además muy específica entre diferentes gustos y que no se corresponden con su media aritmética.

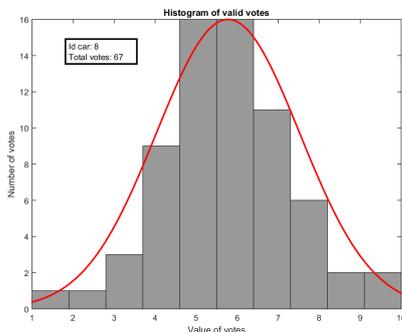


Fig. 3: Ejemplo de histograma de votaciones del coche id 8 (Dacia Sandero del año 2009).

No se han encontrado ninguna de estas tres combinaciones nombradas. Los coches mejor y peor valorados se muestran en el apéndice en las secciones A.1 y A.2 respectivamente.

6 REGIONES DE INTERÉS

Debido a que un coche tiene muchas características diferentes, se necesita estudiar varias zonas del coche independientemente. Para ello se ha desarrollado una pequeña herramienta que permite seleccionar el coche que queremos estudiar, y a continuación crear una máscara binaria sobre la zona de interés, es decir, una *region of interest* (ROI). En este proyecto se han definido por cada uno de los 87 coches 87 máscaras binarias para conjunto completo del coche con el motivo de estudiar los brillos locales, descartando así toda la información del background. Por otro lado, también se han creado otras 87 máscaras binarias con el propósito de analizar únicamente las zonas de pintura del coche de cara al estudio de colores; en total 174 máscaras binarias. En el apéndice A.3 se detallan las posibles opciones para la creación de estas máscaras.

7 ANÁLISIS DE LOS DETALLES BRILLANTES COMO CARACTERÍSTICA ESTÉTICA

La primera propiedad que se ha decidido investigar ha sido la del brillo localizado. Para ser más concretos, existen brillos en los coches que hacen que nos gusten. Si analizamos zonas como las llantas, bordes luminosos, faros, etc. se puede observar que son zonas donde empresas automovilísticas de coches ponen énfasis es resaltar, ya que llaman la atención por sus fuertes cambios lumínicos. Se pretende tratar de demostrar si realmente los contrastes de brillo en zonas localizadas de los coches es una característica que afecta positivamente a la experiencia estética en la percepción de los coches.

7.1. Extracción de mínimos máximos locales

7.1.1. Algoritmo de extracción de features

El algoritmo que se ha utilizado para captar los puntos principales de contraste local se fundamenta en unos pocos pasos principales [6]:

1. Crear una pirámide de escalado de imágenes a partir de una imagen inicial. Cada nueva escala de esta imagen tendrá las dimensiones de la anterior dividido por dos.
2. Calcular las derivadas horizontales y verticales de la imagen de cada escala I_x e I_y convolucionando la imagen original. Esto nos proporcionará una imagen en la cual se destacan los bordes de las imágenes, manteniendo no obstante gran parte de la información restante.
3. El resultado de cada una de estas imágenes se convoluciona de nuevo con las nuevas derivadas de Gaussianas, y así sucesivamente dentro de cada escala. De esta manera tenemos en una misma escala diferentes segmentaciones de imagen en función de la convolución realizada.
4. Una vez obtenidas las diferentes segmentaciones en diferentes escalas, pasamos a encontrar puntos máximos y mínimos en toda la imagen, que serán los puntos de interés (a partir de ahora *keypoints*) representativos de los cambios de contraste.

7.1.2. Creando escalas piramidales

En primera instancia se escoge la imagen original, y a través de un bucle se va referenciando la siguiente imagen redimensionando la imagen actual a su mitad. En este punto surge una pregunta: ¿cuántas veces hemos de redimensionar la imagen? En este caso, se están tratando imágenes iniciales de 1536x2048 píxeles ya duplicamos la resolución de la imagen inicial de 768x1024 píxeles con el objetivo de obtener más keypoints. Por lo tanto si dividimos entre 2 un total de 4 veces nos quedaría la última escala de 96x128 píxeles. Se ha considerado el número 4 de escalados como un número idóneo para este proyecto, ya que tenemos de este modo una gran diversidad de escalas representativas [7]. Cada conjunto de escalas de dimensiones de imágenes toman el nombre de octavas, siendo la primera octava la imagen original duplicadas sus dimensiones y la última y quinta octava la imagen escalada más pequeña.

7.1.3. Diferencia de Gaussianas

El segundo paso del algoritmo es el de convolucionar por cada octava, la imagen inicial con una matriz gaussiana. Este proceso es más comúnmente conocido como aplicar un filtro gaussiano. ¿Qué sucede cuando aplicamos un filtro gaussiano a una imagen? El resultado se puede apreciar en la figura[4]:



Fig. 4: Resultado de aplicar el filtro gaussiano varias veces a la imagen del coche Infiniti QX50 Concept, id 61.

Al analizar la imagen anterior se aprecia que el efecto es de desenfoque o *blurring*. En cada octava se realiza un blurring, y a la imagen siguiente de la misma octava se le aplica un blurring, pero en cada iteración sobre el blurring anterior, de forma incremental. De esta manera, cada vez aumenta más el desenfoque de la imagen. Para este proceso es necesario definir el número de veces que se aplicará este blurring dentro de una misma octava, para este proyecto se ha tomado un valor de 5 niveles, debido a que a partir del quinto nivel se pierde mucha información de la imagen original. Al pensar en lo anterior se nos viene una pregunta a la mente, ¿en qué medida hay que aplicar este efecto de desenfoque? La respuesta depende de σ . La función gaussiana depende en su mayor medida de su varianza. No obstante, en nuestro caso, este valor de sigma ha de ir evolucionando de una iteración a otra dentro de cada octava. La fórmula que describirá este valor de σ es la siguiente 1:

$$\sigma = \sigma_0 2^{(s + \frac{1}{S})} \quad (1)$$

, donde:

- s es el número de la iteración dentro de la octava actual.
- S es el número total de iteraciones por octava. Por lo tanto: $s = 0, 1, 2, \dots, S-1$.

- σ_0 es el valor inicial de sigma, es decir, el valor de sigma en la primera iteración del algoritmo. Este valor se recomienda para este proyecto que se encuentre entre 1 y 2 en función de resultados obtenidos.
- o es el número de octava en la que nos encontramos.

Llegados a este punto, ya tenemos todos los blurring realizados para cada una de las imágenes de cada octava. Ahora necesitamos aplicarlo sobre la imagen anterior, restando a ésta por cada uno de sus píxeles el valor de la imagen desenfocada del siguiente nivel[5], obteniendo así los píxeles que resaltan más. A esta operación aritmética entre imágenes con filtro gaussiano aplicado se le conoce como *Difference of Gaussians* (DoG). Este proceso responderá con valores muy cercanos al 0 formando una segmentación de las imágenes de cada blur level, omitiendo el resto de píxeles, ya que en la mayoría de píxeles se le restará su propio valor obteniendo un resultado igual a 0, es decir, un píxel negro. Para este caso en el que hay que realizar un proceso iterativo por varias escalas tratando imágenes pesadas es ideal, ya que la complejidad computacional de una resta de arrays bidimensionales es muy baja.

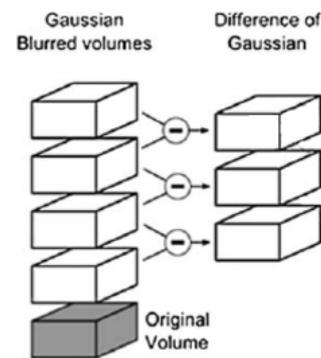


Fig. 5: DoG en perspectiva dentro de una misma octava.

7.1.4. Máximos y mínimos locales

Ya tenemos disponibles todas las imágenes con los bordes segmentados en cada octava. El siguiente paso es el de encontrar aquellos puntos específicos que destaquen por encima del resto. Por lo tanto, se buscarán ahora los máximos y mínimos locales. Esto conducirá a una distribución desigual de puntos de interés a través de la imagen, por ejemplo, los puntos se aglomerarán en regiones de mayor contraste entre brillos. El procedimiento es el que sigue: por cada imagen que no sea ni la primera ni la última dentro de una misma octava y por cada píxel dentro de dicha imagen que no se encuentre ni en la primera ni última fila, ni tampoco en la primera ni última columna:

1. Comparamos el valor del píxel actual con sus 8 vecinos más cercanos en la misma escala de la octava actual. Si dicho valor es mayor a todos ellos o menor a todos ellos pasamos al siguiente punto, sino evaluaremos el siguiente píxel. Conservamos la información de si es mayor o menor al resto.
2. Comparamos el valor del píxel actual con sus 9 vecinos cuya posición es la equivalente en la escala superior

dentro de la misma octava. Si dicho valor es mayor a todos ellos (en caso de que en el paso 1 sea mayor a sus vecinos en dicha escala) o menor a todos ellos (en caso de que en el paso 1 sea menor a sus vecinos en dicha escala) pasamos al siguiente punto, sino evaluaremos el siguiente píxel.

3. Comparamos el valor del píxel actual con sus 9 vecinos cuya posición es la equivalente en la escala inferior dentro de la misma octava. Si dicho valor es mayor a todos ellos (en caso de que en los pasos 1 y 2 sea mayor a sus vecinos en dicha escala) o menor a todos ellos (en caso de que en los pasos 1 y 2 sea menor a sus vecinos en dicha escala) este píxel será un **keypoint**. Si es el último píxel de la imagen finalizaría aquí el procedimiento, sino seguiríamos examinando el siguiente píxel.

En resumen, los máximos y mínimos de la diferencia de las imágenes gaussianas se detectan comparando un píxel con sus 26 vecinos en regiones 3x3 en las escalas actual y adyacente [6]:

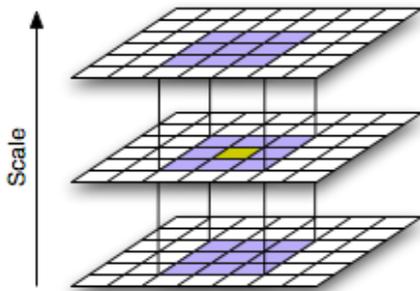


Fig. 6: Píxel que se compara (amarillo) y sus 26 vecinos a comparar (lila).

7.2. Refinamiento de puntos

En este punto del desarrollo cobra sentido el hecho de tener diferentes octavas de imágenes, ya que se informarán como puntos de interés en la imagen original los puntos obtenidos en cada una de sus escalas, de manera que se identifican cambios de contraste tanto si la imagen es pequeña, mediana o grande, haciendo que el algoritmo adquiera robustez.

7.2.1. Umbral de contraste

Aún teniendo todos los puntos máximos y mínimos localizados por nuestro algoritmo no todos aportan información necesaria. Existen píxeles que sí son mayores que sus vecinos pero sin embargo en relación a la imagen no son lo suficientemente definitivos como para considerarlos valores extremos. Estos puntos los podemos identificar de una manera muy simple: creando un umbral de contraste. El valor asignado al umbral que se menciona se recomienda que sea: [0.01, 0.05] según resultados empíricos. El procedimiento se muestra en un pequeño código anidado en el apéndice A.4.

7.2.2. Repetitividad de información en los keypoints

Utilizando este algoritmo se busca obtener puntos determinados que representen la imagen como tal, por sus contrastes de intensidad locales. Por lo tanto, el siguiente paso es el de eliminar los puntos que provengan de seguir la línea que defina un borde, quedándonos sólo con aquellos puntos que identifiquen una zona dentro de la imagen. Aquí se introduce el algoritmo de detección de bordes de Harris [?], que se utiliza para encontrar un umbral de borde y detectar si un keypoint es el resultado de un borde o no: si es así, lo descartamos. Este método está basado en el detector de esquinas y bordes de Harris y eliminará la información repetible como, por ejemplo, 20 puntos que están definiendo el mismo borde en línea recta. Explicaremos cómo en unos pasos:

1. En primer lugar, se calculan las primeras derivadas sobre el eje X y el eje Y. En computación estas derivadas no son más que matrices cuyos vectores son $[-1 \ 0 \ 1]$ y $[-1 \ 0 \ 1]^T$ respectivamente convolucionadas con la imagen en cuestión, es decir, nuestra imagen resultante en cada iteración al aplicar DoG.
2. A continuación se calculan las segundas derivadas, elevando al cuadrado las primeras derivadas y aplicándoles a posteriori un filtro gaussiano de 6x6.
3. Observemos ahora la imagen de la figura [7]. Harris denominó a α como el espacio de respuesta del valor de la segunda derivada en X y β en Y. Esto viene a

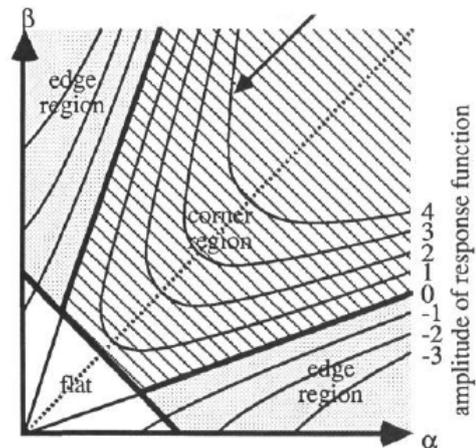


Fig. 7: Espacio de auto-correlación en función de α y β .

decir que dependemos de los valores retornados por α y β para saber si estamos ante una zona plana, un borde o una esquina.

4. A continuación Harris indicó dos fórmulas para definir umbrales de relación entre las porciones del mapa de la imagen anterior, véase las fórmulas 2, 3 y 4.

$$Tr = \alpha + \beta \quad (2)$$

$$Det = \alpha\beta \quad (3)$$

$$R = Det - kTr^2 \quad (4)$$

La tercera fórmula indica R como un valor de umbral que determinará si estamos ante un punto que es un borde ($R < 0$), una esquina ($R \gg 0$) o un plano ($R \approx 0$). Llegados hasta aquí, aplicaremos este procedimiento a cada uno de nuestros keypoints de la DoG actual.

5. En nuestro proyecto, determinaremos si un punto es la continuación de un borde asignando una relación de radio de distancia entre α y β tal que la primera será el valor de mayor magnitud y la segunda el valor de menor en un radio de distancia. Si estamos diciendo que α depende de β y viceversa, pero una es mayor a otra, entonces podremos expresarlo de la siguiente manera:

$$\alpha = r\beta \quad (5)$$

, donde r será la relación de distancia que guarda una respecto a la otra. Si se desarrolla un poco la ecuación anterior, obtenemos el resultado siguiente:

$$\frac{Tr^2}{Det} = \frac{(\alpha + \beta)^2}{\alpha\beta} = \frac{(r\beta + \beta)^2}{r\beta^2} = \frac{(r + 1)^2}{r} \quad (6)$$

, por lo tanto:

$$\frac{Tr^2}{Det} = \frac{(r + 1)^2}{r} \quad (7)$$

La cantidad $(r + 1)^2/r$ está en un mínimo cuando los dos valores propios son iguales y aumenta con r . Por lo tanto, para comprobar que la relación de curvaturas principales está por debajo de cierto umbral, r , sólo necesitamos comprobar esto:

$$\frac{Tr^2}{Det} < \frac{(r + 1)^2}{r} \quad (8)$$

, de tal manera que sólo consideraremos válidos aquellos puntos que cumplan la inecuación anterior, descartando los que no lo hagan, asignándoles así un valor de 0 o píxel negro. En cuanto al valor que debemos utilizar para r , para este proyecto se ha variado entre 5 y 15 (valores empíricos) para lograr resultados ideales.

En la imagen de la figura[8] se observa el resultado de todos los filtros de keypoints:

7.3. Agrupación de keypoints basada en la densidad

El último paso de todo este procedimiento es el de informar sobre aquellas zonas en las que se aglomeran una cantidad de keypoints más grande, es decir, aquellas zonas donde la densidad de puntos es mayor. Para este propósito introducimos en el desarrollo el *Density Based Spatial Clustering of Applications with Noise* (DBSCAN) [9]. Éste es un algoritmo basado en métodos de densidad, hace crecer regiones con una alta densidad en grupos y descubre grupos con forma arbitraria. El algoritmo DBSCAN funciona de la siguiente manera:

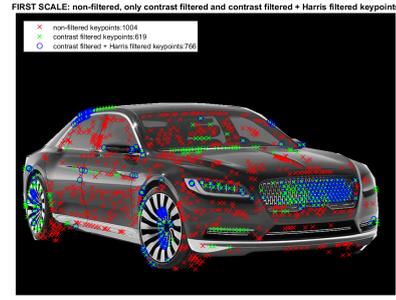


Fig. 8: Keypoints descartados por el umbral de contraste (X rojo), keypoints descartados por el filtro de bordes de Harris (X verde) y keypoints resultantes tras aplicar el filtro de umbral de contraste y de bordes de Harris (O azul).

1. Busca clústeres (agrupaciones) comprobando en el ϵ -vecindario de cada punto, donde ϵ será un valor de distancia en radio alrededor de cada uno de los keypoints.
2. Si dentro del vecindario de un punto p hay más de $MinPts$ puntos, un nuevo clúster con p como centroide es creado.
3. Iterativamente se recolectan los puntos que son directamente alcanzables desde los centroides.
4. El proceso termina cuando no se pueden añadir nuevos puntos a ningún clúster.

Por lo tanto, DBSCAN necesitará de la definición previa de dos parámetros: ϵ y $MinPts$, siendo éste último el mínimo número de puntos que se consideran como agrupación o clúster. En función de la distancia ϵ' a la que se encuentra un punto p' se pueden dar tres casos[10]:

- Dicho keypoint se encuentra dentro de la distancia máxima ϵ , pero no hace grupo con varios $MinPts-1$ keypoints, sólo con uno por lo tanto se considerará un keypoint borde perteneciente al grupo.
- El keypoint se encuentra dentro de un ϵ -vecindario, por lo tanto estará dentro del clúster.
- La distancia entre dicho keypoint y todos los keypoints alrededor suyo es mayor a ϵ , por lo tanto este punto dejará de considerarse como keypoint y no pertenecerá a ningún clúster. Estos keypoints se llaman *keypoints ruido*[9]:

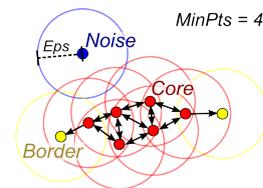


Fig. 9: Agrupación DBSCAN: keypoints ruido (azul), clustered (rojos) y borde (amarillos).

Si definimos ϵ como:

$$\epsilon = (\text{imageWidth})k \quad (9)$$

, siendo k un factor multiplicativo del ancho total de la imagen, conseguimos un *aspect ratio* aproximado en función de las dimensiones de la imagen que estamos tratando. Variando el número de puntos mínimos necesarios para definir un clúster, obtenemos los resultados de la figura [10], donde a medida que aumenta el número de puntos que definen un clúster se eliminan puntos ruidosos que no aportan información en conjunto.

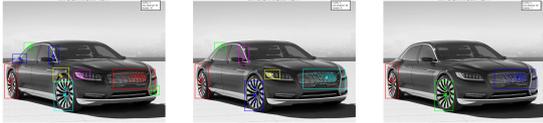


Fig. 10: Resultados para $\epsilon = 92$ según fórmula (9), variando minPts en 3, 5 y 9 de izquierda a derecha respectivamente.

7.4. Repercusión de los keypoints en la valoración estética

En este momento hay que contrastar la información extraída. Se intuye a raíz de pruebas y comparativas entre los coches del conjunto peor valorado y mejor valorado que, la cantidad de keypoints pertenecientes a éste último grupo es bastante mayor que la cantidad de keypoints de coches poco estéticos. Por lo tanto se procede a la extracción de keypoints para cada uno de los 87 coches del dataset. Para llevar a cabo el estudio de correlación entre la cantidad de puntos y la valoración estética de los mismos se ha utilizado la técnica de regresión lineal, que nos permite saber la inferencia que tiene una propiedad en otra dentro de un conjunto de datos compartido por ambas propiedades. Como variable dependiente se toman las votaciones ordenadas de menor a mayor. Del mismo modo, ordenados en función de la votación y como variable independiente se toman la cantidad de keypoints de cada coche. Se puede observar en

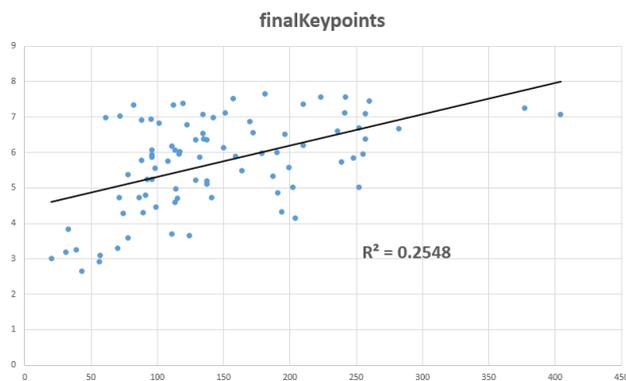


Fig. 11: Eje de abscisas: número de keypoints por coche, eje de ordenadas: rating de cada coche. Resultados obtenidos para: $\epsilon=92$ según fórmula (9), $\text{minPts}=7$, $\sigma=1$, $r=8$, $\text{contrastThreshold}=0.3$.

la figura [11] que el coeficiente de determinación del gráfico es de aproximadamente 0.25. Este coeficiente informa

sobre la proporción de variación de la variable Y (rating) que es explicada por la variable X (cantidad de keypoints). Con estos datos se puede concluir que aproximadamente el **25 % de la respuesta estética a los coches es debida a la cantidad de puntos de contraste de intensidades locales agrupadas en función de su densidad**. En el apéndice A.5 se puede ver la máscara binaria utilizada para el resultado de las imágenes.

8 ESTUDIO DE COLORES

Partiendo de la base de los resultados observados en la gráfica anterior, se ha centrado la investigación en el estudio de los *outliers* o valores atípicos. Estos valores hacen referencia a dos coches con ID 50 y 51, que tiene una nota de 4 aproximadamente, pero sin embargo se observa que ambos superan la media de keypoints en aproximadamente el 40 %. Esta observación nos quiere decir que para estos dos coches existe una o varias propiedades que impide(n) que sean valorados como coches estéticos.

8.1. El matiz de los coches

Estos dos coches considerados como valores atípicos son los mostrados en la figura [12]. En primera instancia destaca que ambos son el mismo modelo pero con el color modificado para rosa y verde. Sin embargo, accediendo a la base de datos de coches y sus respectivas votaciones, la imagen original es la del coche con ID 48 y es de color rojo, con una puntuación de 5,72. Además, estos dos coches pertenecen al grupo de los 15 coches peor valorados. Este dato nos hace pensar que el factor principal que influye en estos dos coches para que su puntuación merme es su matiz.



Fig. 12: *Outliers* con impacto estético negativo, coches con ID 50 y 51 respectivamente a izquierda y derecha.

Con motivo de estudiar el color que aparece en cada uno de los coches, es una tarea muy difícil incluso para el ojo humano clasificar algunos colores en una categoría, etiquetarlos dentro de un mismo matiz. Para solucionar este problema se ha contado con la ayuda de un algoritmo muy preciso [10] y lo más importante, basado en la percepción humana de los colores. Se trata del algoritmo de categorización de colores fundamentado en los elipsoides de los neurosensores del color (NICE). El algoritmo NICE recibe como entrada la imagen a tratar, y en función de los colores detectados categoriza cada uno de los píxeles de la imagen según su probabilidad de pertenecer a uno de los 11 siguientes matices: verde, azul, púrpura, rosa, rojo, naranja, amarillo, marrón, gris, blanco y negro. Para llevar a cabo su utilización se han usado las máscaras binarias correspondientes a las ROI de la pintura de los coches. El algoritmo NICE, como salida ofrece un vector de 11 matrices, donde cada una de las matrices hace referencia a cada uno de los

11 colores categorizados. Estas matrices tienen las mismas dimensiones que la imagen de entrada y sus píxeles son un valor con rango [0,1] indicando la probabilidad de pertenecer a la dimensión del vector actual de la matriz dentro del vector relacionada con el color. Para más información sobre el algoritmo NICE y su ejecución consultar el apéndice A.6.

Tras evaluar el color de cada uno de los coches del grupo de peor valorados y mejor valorados se han obtenido los resultados de la figura[13]. De esta comparativa extraemos

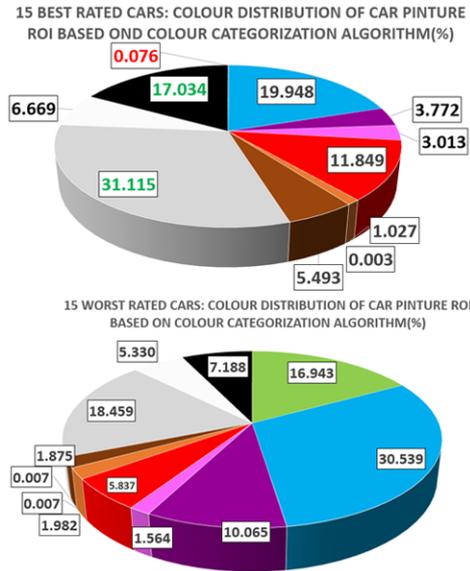


Fig. 13: Comparación de la categorización de colores de los 15 coches peor y mejor valorados.

dos conclusiones principales:

- En primer lugar se observa que la probabilidad del color verde en los coches estéticos es del 0,076 %, mientras que en los coches que han tenido un rechazo estético ocupa un lugar importante con un 17 % aproximadamente. Desgraciadamente no existen los suficientes coches con un índice de probabilidad de matiz verde significativo como para poder medir el impacto que genera en el total del dataset de coches. No obstante, considero estas estadísticas como prueba suficiente para afirmar a modo de conclusión que **el matiz verde en la pintura de los coches repercute ejerciendo una respuesta estética negativa.**
- En segundo lugar se observa un incremento considerable de los colores faltos de matiz como el negro y el gris. Al prestar atención al porcentaje que suman los colores negro, blanco y gris en los coches peor valorados, éste asciende a 31 %, mientras que en los coches mejor valorados es de un 55 % aproximadamente, más de la mitad. Es un indicativo claro de que este tipo de matices atrae mucho estéticamente, sin embargo no son resultados suficientemente claros como para diagnosticar una relación estética.

8.2. Brillo y saturación

Algo que llama la atención a la hora de observar los coches más recientes incorporados al mercado es la variación

de brillos en la pintura, aún más si entramos en un concesionario y miramos cada uno de ellos: los coches están situados estratégicamente debajo de unos focos que hacen resaltar sus cualidades. Por otro lado, *Karen B. Schloss y Stephen E. Palmer* [11] realizaron un tan laborioso como interesante y determinante estudio en que se determina que, en cuanto a la combinación de colores por pares, *las preferencias estéticas crecen cuando dos colores pertenecen al mismo matiz variando siempre su saturación y/o brillo.* Esto hace referencia al concepto de la armonía entre colores [12], siendo mayor la disonancia de gusto cuanto más se aleja un color de sus cercanos en la representación de matices. Este es el punto actual donde se encuentra el proyecto, tratando de encontrar algún indicio para poder comprobar si el concepto de la armonía dentro de la pintura de los del coche influye para su valoración estética.

En primer lugar se ha tratado de estudiar la variación de la saturación y del brillo como elementos comunes, ya que normalmente en las zonas menos saturadas suele coincidir un nivel más alto de brillo debido al tipo de pintura. Para ello se han separado los conjuntos de coches peor y mejor valorados. El procedimiento ha sido el siguiente:

1. Por cada uno de los coches, independientemente de su matiz, se ha procedido a transformar su imagen del espacio RGB al espacio HSV. El espacio HSV es ideal para esto, ya que descompone sus 3 canales en matiz(Hue), saturación (Saturation) y Brillo (Value). Estos valores se encuentran en un rango de normalización [0,1]: para la saturación 0 es nada saturado, por ejemplo un azul celeste muy claro, y 1 sería un color sobresaturado, por ejemplo un color cyan. En cuanto al brillo, 0 sería un color prácticamente negro y 1 sería un color prácticamente blanco debido a su brillantez.
 2. A continuación, por cada uno de los píxeles de cada imagen se ha extraído su valor de saturación y brillo.
 3. Cada uno de estos valores de cada píxel se ha operado según la fórmula(10), de manera que se guarda una información de correlación entre ambas características.
- $$corr = \sqrt{saturation^2 + value^2} \quad (10)$$
4. Finalmente, se ha procedido a calcular la desviación estándar de todos los valores resultantes de esta correlación, tratando así de explicar si hay una relación directa en la variación entre el brillo y la saturación.

Los resultados obtenidos, como se observa en la figura[14], no han sido como se esperaba. El coeficiente de determinación sentencia que no se pueden extraer resultados concluyentes, ya que un 5 % es un porcentaje muy bajo. Sin embargo, es posible afirmar que **no existe una relación estética directa para la variación entre la saturación y el brillo en las pinturas de los coches del dataset.** Aún así esta investigación sigue abierta explorando estos datos.

9 CONCLUSIONES

Respecto al trabajo realizado creo que el resultado ha sido muy positivo. He aprendido mucho en todo el proceso de estudio, tanto a la hora de plantear la investigación creando

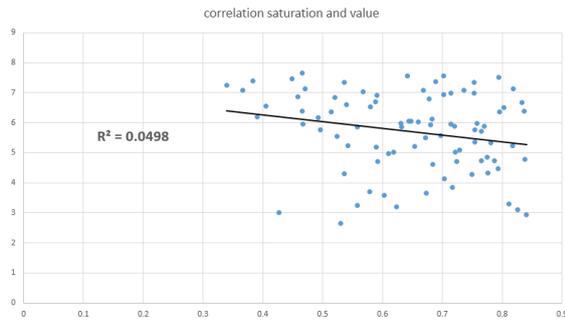


Fig. 14: Eje de abscisas: desviación de la *corr* por coche, eje de ordenadas: rating de cada coche.

el dataset de imágenes como a la hora de tratar de buscar indicios para las propiedades de las imágenes y del método científico para llevar a cabo todo ello.

En cuanto a los resultados obtenidos puedo concluir que el contraste local intenso de brillos en zonas concretas de los coches tiene un impacto estético importante. Por otro lado, el verde y el rosa son colores que no tienen una respuesta estética positiva en la pintura de los coches. Asimismo, también se puede afirmar que no existe una relación directa entre brillo y saturación para la valoración estética.

AGRADECIMIENTOS

En primer lugar agradecer toda la paciencia y dedicación que ha tenido mi tutor, Carlos Alejandro Párraga, por ayudarme a plantear el proyecto, encaminar las etapas dudosas y ser la voz de la experiencia que me ha guiado a tomar decisiones. En segundo lugar a todas las personas que me han aguantado explicando mis teorías sobre coches, en especial mi familia y mi pareja. Y para finalizar a todas y cada una de las 253 personas que contestaron las encuestas y se tomaron esos 10 minutos de molestias para que este trabajo fuera posible.

REFERENCIAS

- [1] C. Redies. Combining universal beauty and cultural context in a unifying model of visual aesthetic experience. *Frontiers in human neuroscience*, 2015.
- [2] Dhiraj Joshi, Ritendra Datta, Elena Fedorovskaya, Quang-Tuan Luong, James Z. Wang, Jia Li, and Jiebo Luo. Aesthetics and emotions in images. *IEEE Signal processing magazine*, September 2011.
- [3] Asociación española de Fabricación de Automóviles y Camiones (ANFAC). Las matriculaciones de turismos en mayo alcanzan 126.411 unidades. Nota de prensa. 1 de junio del 2017.
- [4] S. Moreno. Ranking de los colores de coches más populares de 2016. Documento online: <http://www.lavanguardia.com/motor/rankings/20161212/412258136562/ranking-colores-populares-coche-2016.html>. Accedido el 3 de abril del 2017.

- [5] S. Beigpur, J. van de Weijer. Object Recoloring based on Intrinsic Image Estimation. Centre de Visio per Computador, Computer Science Department. *IEEE International Conference on Computer Vision*. 2011.
- [6] R. Szeliski. *Computer Vision: Algorithms and Applications*. Springer, 2010.
- [7] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 2004.
- [8] T. Tuytelaars and K. Mikolajczyk. Local Invariant Feature Detectors: A Survey. *Foundations and Trends in Computer Graphics and Vision*, Vol. 3, No. 3, (2007) 177-280.
- [9] M. C. Montes. Clustering: Clasificación no supervisada. gráficas estadísticas y minería de datos con python. Technical report, Centro de Investigaciones Energéticas Medioambientales y Tecnológicas, 2013.
- [10] Párraga CA, Akbarinia A (2016) NICE: A Computational Solution to Close the Gap from Colour Perception to Colour Categorization. *PLoS ONE* 11 (3): e0149538. doi:10.1371/journal.pone.0149538.
- [11] Karen B. Schloss and Stephen E. Palmer. Aesthetic response to color combinations: preference, harmony, and similarity. Springer, 2010.
- [12] Stephen Westland, Kevin Laycock, Vien Cheung, Phil Henry and Forough Mahyar. *Colour Harmony*. School of Design, University of Leeds, Leeds LS2 9JT, UK. *Colour-journal*, January 2007.

APÉNDICE

A.1. Tabla de los 15 coches mejor valorados

ID	Modelo (<i>diferencias</i>)	Puntuación
24	Chevrolet Impala	7.0889
31	Infiniti M35H GT	7.3956
33	Lexus IS 350 F Sport	7.569
34	Lexus IS US Version	7.6596
35	Lexus IS	7.3415
38	Lincoln Continental	7.5696
39	Lincoln MKZ	7.2593
43	Mazda 6	7.125
46	Subaru WRX	7.5205
61	Infiniti QX50	7.4615
64	Lexus RX450H	7.0816
65	Lexus NX	7.338
68	Lincoln MKC	7.0769
70	Mazda CX5 (2013)	7.1194
71	Mazda CX5 (2017)	7.3678

A.2. Tabla de los 15 coches peor valorados

ID	Modelo (diferencias)	Puntuación
5	Chrysler Akino	3
6	Chrysler GEMPeapod	3.194
10	Daihatsu Sirion (gris)	3.6526
11	Daihatsu Sirion (rojo)	3.3008
12	Lancia Ypsilon	3.8375
17	Skoda Citigo	3.7037
18	Subaru Pleo	2.6554
19	Dacia Sandero (rosa)	3.5918
20	Dacia Sandero (verde)	4.2857
22	Daihatsu Sirion (verde)	3.0977
23	Daihatsu Sirion (cyan)	2.9255
26	Dacia Logan (gris)	3.2439
27	Dacia Logan (azul)	4.3125
50	Volvo S40 (rosa)	4.1429
51	Volvo S40 (verde)	4.3182

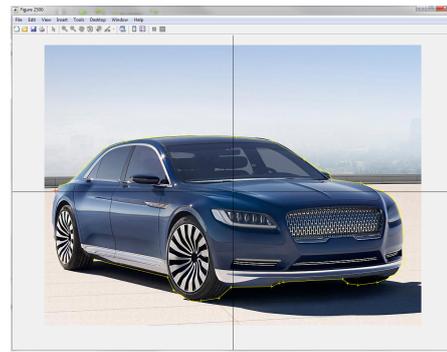
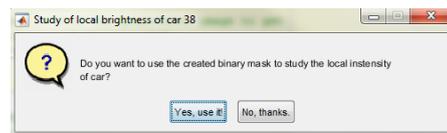


Fig. 18: Ejemplo de creación de máscara binaria para el coche con id 38.



A.3. Interfaz gráfica para las ROI

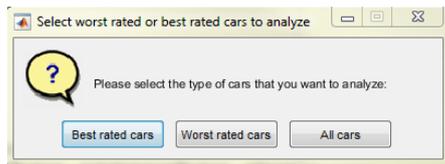


Fig. 15: Selección del conjunto de coches.

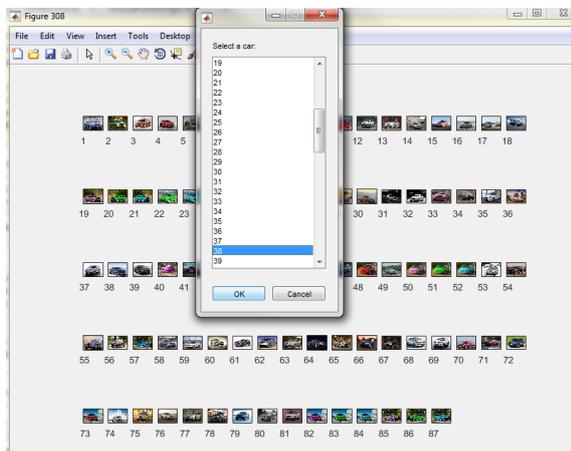


Fig. 16: Selección del coche a estudiar.

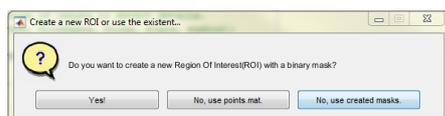


Fig. 17: Selección de opciones de máscara. De izquierda a derecha: crear una nueva ROI, cargar la última ROI guardada en una matriz *points.mat* o escoger la máscara creada anteriormente para el coche seleccionado.

Fig. 19: Selección de opción: analizar los brillos locales utilizando la máscara binaria, o en su defecto analizar la imagen completa incluyendo el background.

A.4. Código del detector de Harris

```

getPathCars.m | filterKeypoints.m | carSelection.m | choosePointsOrigin.m | choo
% Detecting points from line points:
% -----
% Here the Harris edge detection algorithm is used to find an edge
% threshold and to detect whether a keypoint is the result of an
% edge or not. If so, we discard it. This method will discard the
% repeatable information.

% x derivative matrix
dx = [-1 0 1; -1 0 1; -1 0 1];
% y derivative matrix
dy = dx';

% Getting the derivatives of image.
Ix = conv2(currentLevel, dx, 'same');
Iy = conv2(currentLevel, dy, 'same');

% 6x6 Gaussian filter.
sigma = 1;
g = fspecial('gaussian', max(1,fix(6*sigma)),sigma);

% Second it gets the second derivatives of image.
Ix2 = conv2(Ix.^2, g, 'same');
Iy2 = conv2(Iy.^2, g, 'same');

% Finally it creates the formula to calculate the values to decide
% if a keypoint is under threshold or not.

% Ratio curvature for Harris corner detector.
r = 10;
trace = Ix2 + Iy2;
det = Ix2 .* Iy2;
curvature = (trace.*trace)./det;
threshold = ((r+1)^2)/r;

% r = det - k*tr^2
% If a keypoint comes from a line, it will be discarded.
for k=1:numtimes
    x1=x(k);
    y1=y(k);
    if (curvature(x1,y1)>threshold)
        extremeValuesHarris(x1,y1)=0;
    end
end
    
```

Fig. 20: Código de descarte de falsos píxeles extremos referentes a bordes.

A.5. ROI coche ID 38

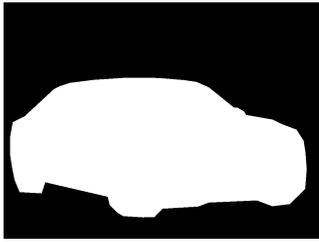


Fig. 21: Máscara binaria de la ROI del coche con ID 38 de los resultados de las imágenes.

A.6. Resultado NICE

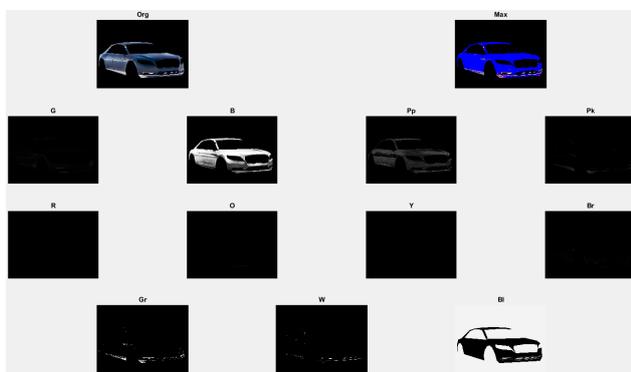


Fig. 22: Resultado del algoritmo NICE para la categorización de colores del coche con ID 38.