

UNIVERSIDAD NACIONAL DEL COMAHUE
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE ELECTROTECNIA



DETECCIÓN DE PATRONES DE ATENCIÓN VISUAL EN LA
CONDUCCIÓN AUTOMOVILÍSTICA

Proyecto Integrador Profesional presentado por

CAMACHO MARINA DANIELA

Dirección de tesis: Ing. DARIO MENDIETA

Neuquén, 2019

Resumen

En este trabajo se desarrolla un prototipo que permite estudiar, mediante el uso de cámaras de visión, el comportamiento de un conductor de un automóvil. La extracción de información se realiza en una PC, mientras que las imágenes se adquieren con la cámara de un Smartphone. Específicamente, se determinan los patrones de atención visual en relación a si el usuario está observando la ruta por la cual circula el automóvil, basándose únicamente en visión monocular y utilizando técnicas de procesamiento de imágenes.

El procesamiento de imágenes es el conjunto de técnicas que se aplican a las imágenes con el objetivo de mejorar la calidad o bien facilitar la búsqueda de información. En este caso la meta final es obtener, a partir de una imagen origen, datos para desarrollar una aplicación específica. Esto se logra extrayendo ciertas características de la misma mediante operaciones del algoritmo.

En función de las mediciones y los datos obtenidos se determina la robustez del algoritmo y la calidad de detección de dirección de la mirada de una persona al volante.

Abstract

In this work, a prototype that allows to study the behavior of a car driver through the use of vision cameras is developed. The extraction of the information is done in a PC, while the images are acquired with a Smartphone camera.

Particularly, visual attention patterns are determined on relation to whether or not the user is watching the road in which the car circulates, based only on monocular vision and using images processing techniques.

The image processing is a set of techniques that are applied to images in order to improve their quality or facilitate the information searching. In this case, the final goal is to obtain data from a source image to develop a specific application. This is achieved by extracting certain characteristics of it through operations of the algorithm.

According to the measures and the data, the robustness of the algorithm and the quality of the direction of the gaze of a person behind the wheel is determined.

Índice

CAPITULO 1. Motivación y objetivos	6
1.1 Motivación	6
1.2 Objetivo general	6
1.3 Objetivos específicos.....	7
1.4 Organización de la Tesis	7
CAPITULO 2. Estado del Arte	8
2.1 Introducción	8
2.2 Localización de los ojos en la imagen.....	9
2.2.1 Proyección por Filas y Columnas	12
2.2.2 Detección mediante identificación de píxeles de piel.....	14
2.2.3 Algoritmo de Viola Jones	16
2.3 Estimación de la mirada	20
2.3.1 Extracción del Borde y el centro del Iris	22
2.3.2 Ajuste de la Elipse	25
2.3.3 Computo de la Normal del plano.....	27
2.4 Definición de la Medida de Desempeño	29
CAPITULO 3. Método Propuesto: Detección ocular	31
3.1 Introducción	31
3.2 Recopilación de Imágenes para un Dataset.....	31
3.3 Implementación de la Detección de ROI	32
3.4 Determinación del Centro del Iris	38
3.5 Implementación del Ajuste de la Elipse.....	41
3.6 Implementación del vector normal al plano.....	44
3.7 Detección de las esquinas de los ojos.....	45
3.8 Creación de Dataset.....	47
3.9 Puesta a Punto del Sistema	48

CAPITULO 4. Resultados	51
4.1 Condiciones Generales	51
4.1.1 Dataset I: Condiciones	51
4.1.2 Dataset I: Resultados	54
4.1.3 Dataset II: Condiciones.....	57
4.1.4 Dataset II: Resultados	60
4.1.5 Dataset III: Condiciones	65
4.1.6 Dataset III: Resultados.....	67
4.2 Discusiones.....	71
4.2.1 Discusión Dataset I	71
4.2.2 Discusión Dataset II.....	72
4.2.3 Discusión Dataset III	73
4.3 Conclusiones.....	74
Bibliografía	76

CAPITULO 1. Motivación y objetivos

1.1 Motivación

A pesar de que actualmente se encuentran vehículos autónomos en el mercado, se estima que aún falta mucho tiempo para que el hombre ceda su lugar como conductor. Según un estudio del MIT realizado en abril de 2017 a más de 3000 personas, el 48 % no estaría dispuesto a utilizar un auto completamente autónomo [1]. En este sentido, herramientas pasivas que colaboren con el manejo vehicular son más aceptadas por los conductores. Esto ha contribuido al auge en el desarrollo de software en el entorno automovilístico que se observa en el presente.

Una cámara y el hardware adecuado para procesar su información constituyen una potente herramienta para el desarrollo de aplicaciones relacionadas tanto con el confort del conductor como con la seguridad vial. Vehículos como el Cadillac ATS 2019, dotado de diversos sensores como un sistema LIDAR infrarrojo, radares y un conjunto de cámaras, pueden otorgar comodidad al conductor mediante los sensores de proximidad o simplemente a la hora de estacionar. Dichas tecnologías liberan al conductor de la tensión que estas maniobras presuponen. Respecto a la seguridad vial, un sistema que pueda emitir señales de aviso cuando el conductor no se encuentra atento al volante, incide directamente reduciendo la tasa de mortalidad por accidentes automovilísticos. Particularmente en nuestro país, fallecieron en promedio 20 personas por día durante el año 2018 [2].

1.2 Objetivo general

El objetivo general del proyecto integrador es desarrollar el prototipo de un sistema basado en una cámara monocular, que permita determinar la atención al volante de un conductor vehicular, en un ambiente controlado.

1.3 Objetivos específicos

Los objetivos específicos del presente proyecto integrador son los siguientes:

- El Estudio, evaluación e implementación de las posibles técnicas de detección de localización de cara y ojos en una imagen.
- Análisis de las técnicas de detección de la mirada.
- Armado de un conjunto de imágenes propio, para la puesta a punto del sistema.
- Utilizar la información obtenida en el dataset para determinar la robustez y desempeño del sistema.

1.4 Organización de la Tesis

El proyecto integrador se presenta en este documento mediante 4 capítulos:

- Capítulo 1: Se muestran las motivaciones y los objetivos.
- Capítulo 2: Se desarrolla el estado del arte, es decir, como se encuentra el desarrollo en cuanto al estudio de la localización de los ojos y la determinación de la dirección de la mirada en una imagen.
- Capítulo 3: Método propuesto: se presentan las técnicas utilizadas en este trabajo; cómo se obtiene el dataset de imágenes propuesto y a partir del mismo como se realiza la extracción de información y su correspondiente procesamiento.
- Capítulo 4: Se presentan los resultados obtenidos, se calcula la precisión y exactitud del algoritmo. Se presentan las posibles mejoras y posibles futuros desarrollos.

CAPITULO 2. Estado del Arte

2.1 Introducción

En la *Figura 1* se observan las distintas partes que componen un ojo humano, cada una de ellas con diferentes e importantes funciones. El iris, por ejemplo, es una estructura que separa las cámaras anterior y posterior del ojo. Su músculo dilatador aumenta el tamaño de la pupila y el músculo esfínter lo reduce. La pupila, por su parte, regula la cantidad de luz que entra en el ojo [3].

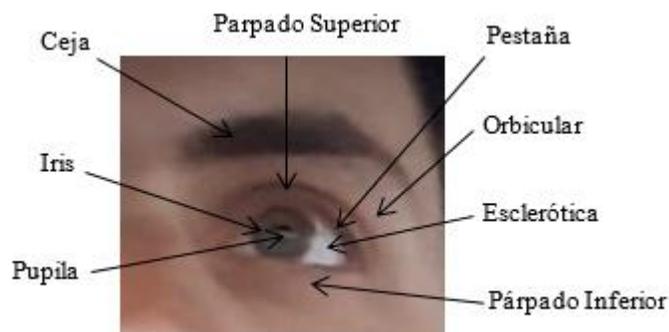


Figura 1: Las distintas partes que conforman un ojo humano.

La información sobre la localización ocular en una imagen tiene múltiples aplicaciones. Es muy utilizada, por ejemplo, en identificaciones biométricas y en diferentes interacciones humanas con computadoras.

Puntualmente, determinar la precisión y trayectoria de movimientos oculares puede proveer información sobre cómo los individuos localizan objetivos de interés. El nivel de atención es usualmente reflejado en la fijación visual, por lo tanto, la medición del tiempo de fijación puede ser usada como una medida del interés de una persona. También puede reflejar la duración de procesamiento de información cerebral: el total de la duración de una mirada por área del objetivo es una medida de la dificultad de recordar la información vista [3].

La detección y el seguimiento ocular han pasado a través de los años por numerosos desafíos, tales como oclusiones en el ojo, variabilidad en la escala y movilidad de la localización, además de sombras y luces en la imagen. Por otra parte, la detección de mirada requiere definir la Región de Interés o ROI, por sus siglas en inglés de *Region of Interest*; asociada a el/los ojo/s en cada

imagen que se procesa. Estas dificultades han desembocado en el estudio y desarrollo de diferentes técnicas con diversos enfoques.

2.2 Localización de los ojos en la imagen

Encontrar la ubicación y características inherentes a los ojos tales como esquinas, borde o centro del iris es fundamental para estimar la orientación de los mismos. Se debe, en primer lugar, detectar la existencia de ojos en la imagen para luego determinar si los mismos se encuentran cerrados. Existen diversas técnicas para localizarlos, las cuales pueden dividirse en cuatro grandes grupos:

- Técnicas basadas en la forma del ojo
- Técnicas basadas en las características del ojo
- Técnicas basadas en la apariencia del ojo
- Técnicas Híbridas

A su vez, los métodos basados en la forma del ojo pueden subdividirse en las siguientes categorías:

- Métodos de forma fija
- Métodos de forma deformable

El desarrollo de las estrategias con forma fija puede realizarse con el contorno o con puntos característicos del ojo y de la cara. Cuando el ojo está abierto, se encuentra identificado por los contornos de la pupila y del iris, estando los mismos limitados por el contorno de los párpados. Es por esto que muchas aplicaciones de seguimiento de la mirada solo utilizan la detección de características como el contorno del iris. Dependiendo del ángulo de visión y la dirección de mirada, este contorno será elíptico, por lo cual podría ser definido por cinco parámetros. Para encontrar el iris se puede utilizar una estrategia de votación, o bien un método de ajuste. En particular, las estrategias de votación extraen características recurriendo a una hipótesis y a un proceso de acumulación progresiva. Un ejemplo de esta técnica se desarrolla en el trabajo de Kyung-Nam Kim [4] en donde, bajo la hipótesis de que el iris posee un rango de intensidad

particular, se aplican sucesivos umbrales de distintos valores de intensidad en la imagen, con el objetivo de hallar el contorno del iris y su centro.

En cambio, los modelos de ajuste realizan una adaptación sobre alguna particularidad aprovechable del ojo, como podría ser una elipse aplicada al contorno del iris. Un ejemplo de la aplicación de este método se observa en el desarrollo de la transformada de Hough [5].

Estrategias más complejas y con figuras deformables necesitan un modelo del ojo mucho más detallado. Uno de los ejemplos más característicos de este tipo de detección es el modelo de plantilla deformable. Este consiste en dos parábolas que representan los párpados, y un círculo para el iris, tal como se puede apreciar en la *Figura 2*. La plantilla puede ser ajustada a la imagen mediante un proceso de optimización que involucra valles y picos de intensidad, así como también contornos de la imagen [6].

El valle de intensidad se corresponde con el iris, por lo cual el círculo de la plantilla se escala según dicha particularidad. Luego, los picos de intensidad se encargan de la orientación mientras que con los bordes de la imagen se termina de refinar el ajuste.

Las estrategias basadas en las características del ojo, en cambio, se enfocan en explorar las particularidades que puedan destacar al ojo con respecto al resto de la imagen. Las mismas pueden dividirse en la siguiente clasificación:

- Métodos basados en intensidad
- Métodos basados en respuesta a filtros

Los primeros utilizan la intensidad para detectar, por ejemplo, el iris del ojo; mientras que los segundos extraen información mediante filtros cuya finalidad es destacar alguna particularidad y eliminar la información no relevante. Dentro de la primera clasificación se encuentran, por ejemplo, los métodos que utilizan las reflexiones especulares como información para hallar la localización de los ojos. Las reflexiones especulares se definen como puntos brillantes que se producen en la córnea del ojo debido a una o muchas fuentes de luz brillante [7]. Entre las técnicas que utilizan la respuesta a filtros se pueden destacar aquellas que utilizan la transformada Wavelet de Gabor para detectar los bordes o la esclerótica que forma parte del globo ocular del ojo [8].

La apariencia del ojo también es una estrategia de detección eficaz que puede recurrir a diversos métodos como la distribución de color en la imagen, y las respuestas a filtros del ojo o sus

alrededores. A diferencia de las técnicas basadas en las características del ojo, los filtros utilizados en esta clasificación no hacen una selección del rasgo a utilizar.

Por último, se encuentran los métodos híbridos que combinan las ventajas de las diferentes estrategias para detectar los ojos.

En la *Figura 3* se puede observar un diagrama de los distintos métodos desarrollados en esta sección para la detección de la región ocular.

En las siguientes secciones se desarrollan en detalle algunas técnicas que se encuentran dentro de las clasificaciones observadas en la *Figura 3*.

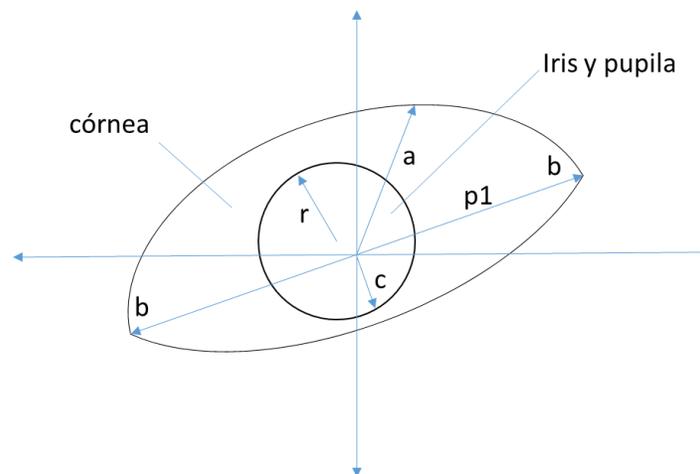


Figura 2: Se observa un modelo del ojo utilizado para la plantilla conformado por dos parábolas y una esfera. Imagen extraída de [6].

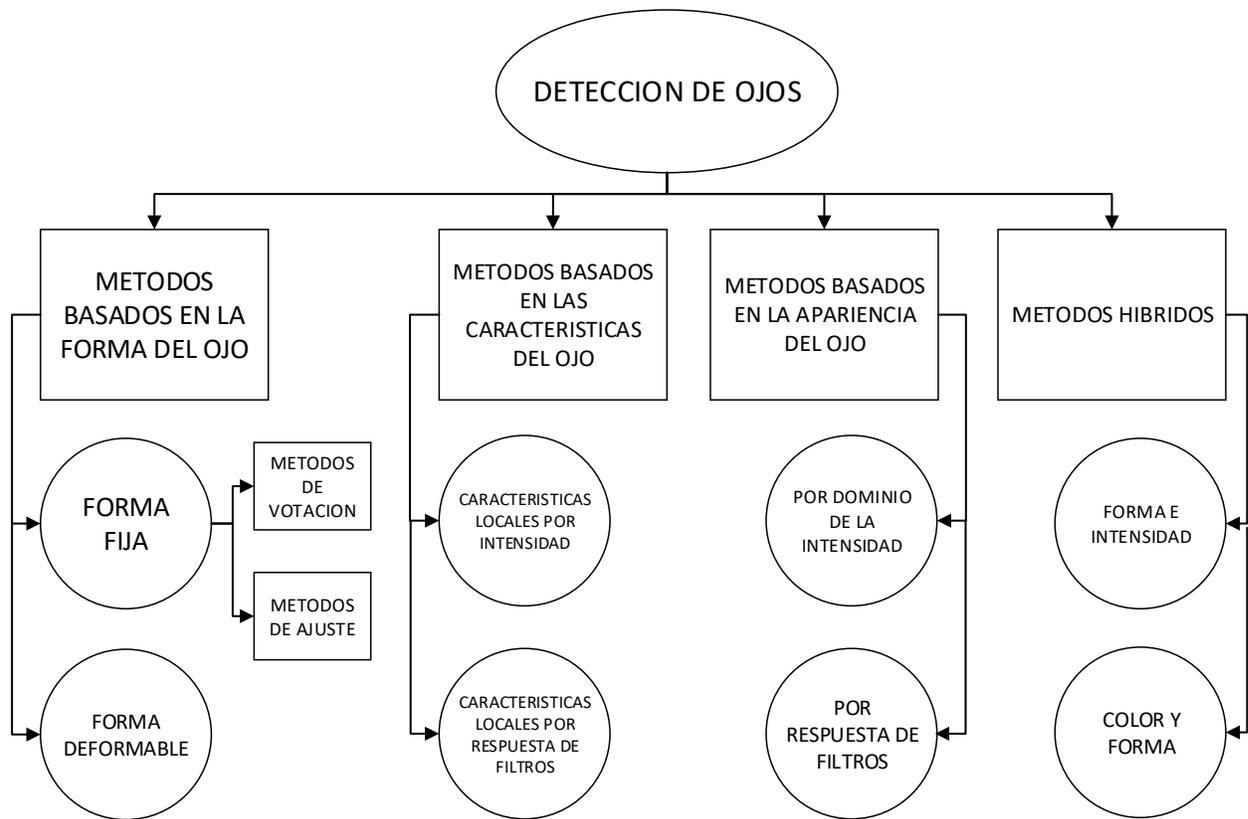


Figura 3: Esquema de los diferentes métodos de detección de la ROI de los ojos.

2.2.1 Proyección por Filas y Columnas

Un recurso práctico para detectar la ubicación de los ojos consiste en aplicar una técnica basada en la apariencia del ojo, la cual utiliza proyecciones obtenidas a partir de la imagen. Teniendo delimitada la ROI de la cara, se puede realizar una proyección horizontal (suma de las filas) de la imagen delimitada y binarizada. Luego se localizan los picos máximos: el primer máximo local en dirección descendente se considera que es la ceja, mientras que el segundo, los ojos. En función de esto puede delimitarse una caja de medida estándar con el objetivo de delimitar la región de los ojos. Las medidas estándar se obtienen a partir de medidas antropométricas de los ojos en relación con el tamaño de la cara, por lo cual se considera que el alto de la ROI de los ojos debería ocupar como máximo una tercera parte del largo de la cara. Esta estrategia, aunque es simple, se encuentra limitada por requerir un alto contraste entre la cara y su alrededor para que no se generen falsos picos.

Particularmente, Gines Garcia Mateos [9] propone en su trabajo una proyección integral horizontal y vertical de una imagen facial, lo cual permite una localización con mayor exactitud. La proyección integral consiste en la creación de un modelo en el cual se construye una suma de señales gaussianas dadas por la media y la varianza de proyecciones de la cara. En la *Figura 4* se muestran tres modelos de proyecciones integrales verticales y horizontales típicos.

Por otra parte Kun Peng [10] propone utilizar las proyecciones del gradiente de la imagen como característica para la detección de la ROI de los ojos. En la *Figura 5* se observa un esquema de los pasos realizados para obtener dicha localización.

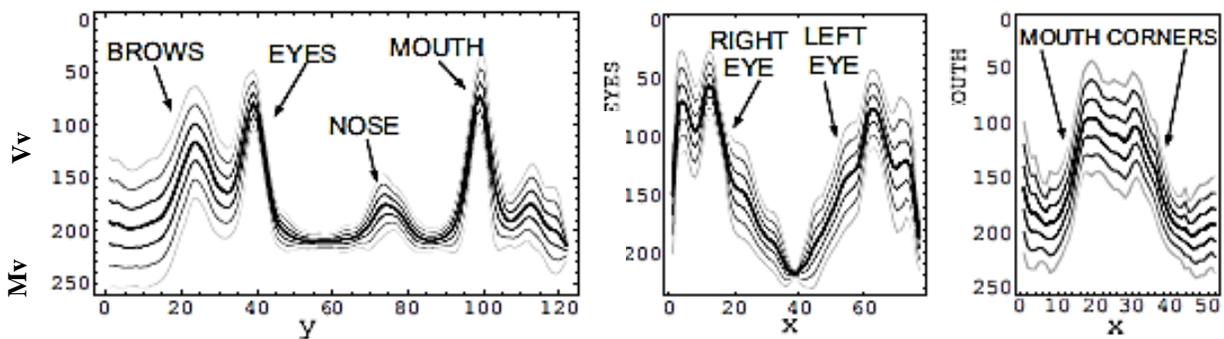


Figura 4: Se observan los distintos modelos de proyecciones de la imagen de una cara. El primer modelo es una proyección integral horizontal de la cara, el segundo es un modelo de proyección vertical de los ojos; mientras que el tercero es un modelo correspondiente a la boca. Imagen extraída de [9].

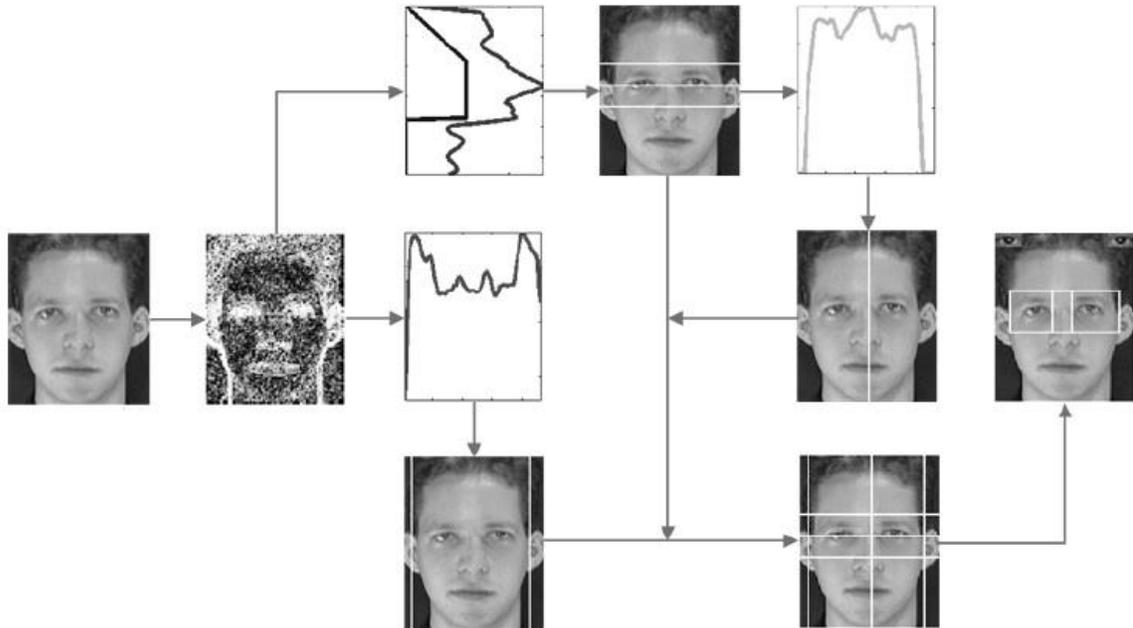


Figura 5: Esquema de la detección de la región de los ojos mediante la proyección horizontal y vertical del gradiente de la imagen. Imagen extraída de [10].

En primer lugar, se calcula el gradiente de la imagen junto con sus respectivas proyecciones verticales y horizontales. La información obtenida de dichas proyecciones se cruza para obtener la localización deseada. La detección del centro del iris se lleva a cabo mediante la aplicación de una plantilla deformable en la zona anteriormente delimitada.

2.2.2 Detección mediante identificación de píxeles de piel

Basándose en el color de la piel es posible extraer características que permitan ubicar la ROI de los ojos. Por ejemplo, Desire Sidibe en su trabajo [11] propone una estrategia híbrida que tiene como foco central la detección de píxeles de piel mediante distintos filtros en una imagen RGB normalizada.

Estos filtros son obtenidos de un modelo de distribución de color en el cual se utiliza una función de densidad de probabilidad gaussiana y elíptica. Los parámetros de dicha función son obtenidos de un set de entrenamiento de más de 100 imágenes de caras. En la *Figura 7* se observa el resultado de aplicar el modelo de color en una imagen de una persona mirando al frente.

Una vez que se detecta la región de los píxeles de piel, se escogen candidatos de posibles ojos dentro de esa región. En la *Figura 6* a continuación se observan los posibles candidatos a regiones de ojos procesadas.

Luego, utilizando características antropométricas de los ojos, es posible descartar candidatos indeseados. Esto se lleva a cabo ajustando una elipse en cada posible región de ojos, donde finalmente un par de los candidatos son elegidos si los mismos cumplen ciertas restricciones basadas en la ubicación y tamaño esperables.



Figura 6: A la izquierda la región de piel detectada y a la derecha los potenciales candidatos a región de ojos. Imagen extraída de [11].



Figura 7: Detección de píxeles de piel en una imagen de una cara. A la izquierda se observa la imagen original, mientras que a la derecha la región de piel detectada en color negro. Imagen extraída de [11].

2.2.3 Algoritmo de Viola Jones

El algoritmo de Viola-Jones (VJ) es una técnica de bajo costo de detección de caras y características faciales desarrollado por Paul Viola y Michael Jones en julio de 2001 [12]. Esta estrategia basada en la apariencia, realiza un procesamiento en donde se utiliza una imagen integral, un entrenamiento de Adaboost y una cascada de clasificadores.

Una imagen integral es nada más y nada menos que una nueva representación de la imagen, la cual contiene en cada punto (x,y) la suma de los píxeles a la izquierda de x y por encima de y ; tal como se representa en el esquema de la *Figura 8*. Esto logra que, al reducir de forma crítica la cantidad de operaciones sobre los píxeles, se extraigan rápidamente las características buscadas.

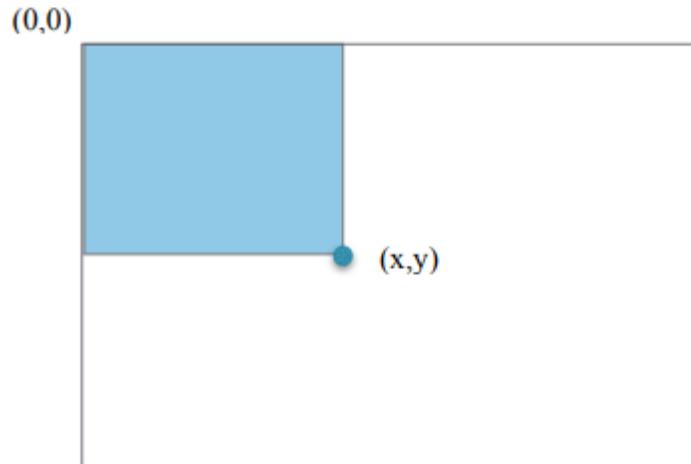


Figura 8: Representación de una imagen integral. El valor del punto (x, y) es la acumulación de intensidad de la región pintada. Imagen extraída de [13].

Teniendo en cuenta la definición de imagen integral, se puede construir la siguiente ecuación:

$$Ii(x, y) = \sum_{x' \ll x, y' \ll y} i(x, y)$$

Siendo $I(x, y)$ la imagen integral, e $i(x, y)$ la intensidad de la imagen original.

Una vez obtenida la imagen integral, el algoritmo aplica tres tipos de filtros, los cuales se pueden observar en la

Figura 9. A medida que se avanza en la implementación estos se hacen cada vez más complejos. El primer filtro se muestra en los ítems (a) y (b) y se compone de dos regiones rectangulares compuestas por otras dos más pequeñas que recorren la imagen. Los píxeles del área blanca se suman, mientras que los del área gris se restan, obteniendo de este modo una noción de la variación de intensidad entre ambas regiones. El segundo filtro, correspondiente al ítem (c), está compuesto por tres rectángulos más pequeños: se suman los resultados de las regiones exteriores y se resta el del medio. Finalmente, el último filtro del ítem (d) está compuesto por cuatro rectángulos, y computa la diferencia entre los pares de rectángulos diagonales.

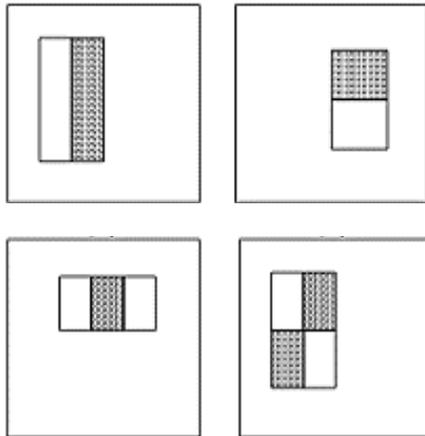


Figura 9: Filtros utilizados por el algoritmo para extraer información de la imagen integral. Imagen extraída de [12].

El objetivo de estos filtros es evaluar si existen determinados rasgos de la cara en la zona de detección. La suma y la resta de las regiones en los filtros devuelven un valor que se compara con un umbral determinado en el algoritmo para evaluar la existencia de la característica buscada.

Con el objetivo de seleccionar el mejor conjunto de filtros rectangulares que resuelvan el problema de detección de rostros, el algoritmo de Viola-Jones utiliza la función de clasificación AdaBoost [14]. El nombre de este algoritmo proviene de la abreviación de *adaptive boosting* y hace referencia a una estrategia de ponderar de manera diferente cada uno de los casos durante un entrenamiento recursivo. Este algoritmo selecciona en cada iteración el mejor filtro rectangular sobre todas las posibilidades, es decir aquel con mayor porcentaje de aciertos.

La *Figura 10* muestra de manera representativa uno de los filtros sobre la ventana de la imagen que es seleccionado por la función AdaBoost. El cuadro de borde rojo indica la ventana a procesar, los recuadros verdes indican que su valor de intensidad será sumado con signo positivo y el recuadro azul con signo negativo.

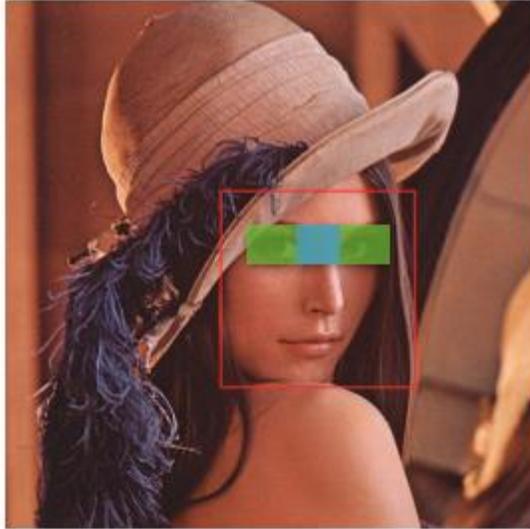


Figura 10: Ejemplo de un filtro rectangular aplicado en una imagen. Imagen extraída de [13].

Además de lo anterior, se aplican clasificadores en cascada lo cual sirve tanto para incrementar la selectividad en la detección como para disminuir los tiempos de cómputo. Se comienza aplicando clasificadores débiles (sencillos y poco precisos); y si el resultado es positivo se van aplicando clasificadores más complejos. Un resultado negativo en cualquiera de las etapas permite desechar al candidato a cara, ya que el objetivo es determinar rápidamente dónde puede aparecer un rostro en la imagen y reservar el procesado más complejo para cuando la probabilidad de detección haya aumentado. Cada uno de estos clasificadores es entrenado utilizando el ya mencionado algoritmo de AdaBoost. En la *Figura 11* se ilustra este proceso:

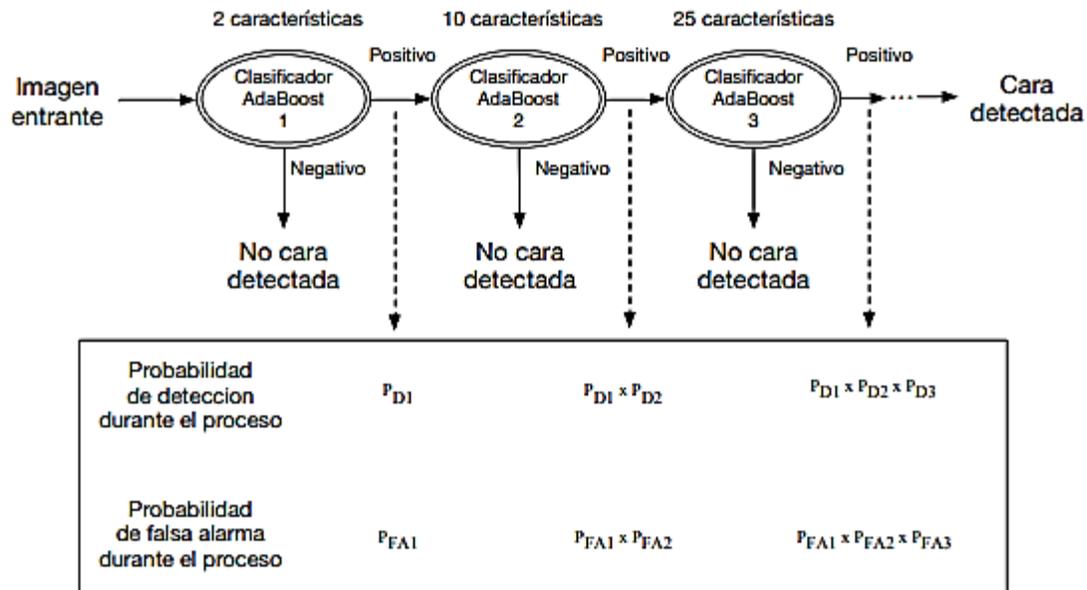


Figura 11: Proceso de clasificación en cascada. Imagen extraída de [13].

2.3 Estimación de la mirada

Una vez aplicada alguna de las técnicas desarrolladas en la sección 2.2 para extraer la información de la ubicación de la región de los ojos en la imagen completa, es posible entonces avanzar en un procesamiento enfocado en obtener la dirección de la mirada.

En este sentido, es preciso definir el seguimiento de la mirada (EGT por sus siglas en inglés de *Eye Gaze Tracking*) el cual consiste en medir y realizar un seguimiento del punto de fijación (PoG por sus siglas en inglés de *Point of Gaze*).

Un factor importante a tener en cuenta cuando se habla de EGT es la postura de la cabeza, ya que está ligada a la posición ocular. El movimiento de cabeza presupone una rotación y un cambio de escala; y es actualmente uno de los mayores desafíos a la hora de realizar el seguimiento de la mirada. Esto último conlleva a que sea necesario, ya sea directa o implícitamente, realizar un modelado de esta postura [15].

Todos los métodos de estimación de la mirada requieren parámetros obtenidos a partir de alguna calibración del sistema. Las mismas pueden distinguirse entre:

- Calibraciones de la cámara: donde se obtienen los parámetros intrínsecos de la misma.
- Calibraciones geométricas: se utilizan para determinar posiciones relativas de cámaras, fuentes de luz, entre otros.
- Calibraciones personales: donde se obtienen datos como la curvatura de la córnea, o el desfase angular entre el eje visual y el óptico.
- Calibraciones del mapeo de la dirección de la mirada: donde se obtienen parámetros relacionados a funciones del mapeo de la postura de los ojos con respecto al entorno.

Existen diversas clasificaciones aplicadas a los métodos utilizados para el rastreo de la mirada de un usuario, sin embargo, la mayoría de los mismos son basados en características del ojo. Estas estrategias utilizan la detección de características locales tales como contornos, esquinas de los ojos, o reflexiones especulares. La principal razón de que esta clase de estrategias sea la más utilizada está relacionado con la facilidad de extraer esas características [15]. A su vez estas técnicas pueden clasificarse en:

- Métodos basados en interpolaciones
- Métodos basados en modelados 3D

Los métodos basados en interpolaciones asumen que el mapeo entre las características de la imagen en 2D y las coordenadas de la mirada en 3D poseen una forma paramétrica particular. De este modo evitan de manera explícita calcular la intersección entre la dirección de mirada y el punto observado. La utilización de reflejos especulares es una de las herramientas más utilizadas en este tipo de estrategia.

Como ya se mencionó en la sección 2.2, debido a las propiedades reflejantes de la córnea del ojo, es posible utilizar fuentes de luz infrarroja con el propósito de captar o detectar el reflejo especular en la imagen. Cuando la mirada está dirigida a la luz, el reflejo especular se encontrará cerca del centro de la pupila, si en cambio la dirección de la mirada está apuntando a otro lugar, esa distancia aumentará. El vector formado entre el reflejo y el centro de la pupila puede ser utilizado para detectar y rastrear la dirección de la mirada, siempre que se considere que no hay rotación de cabeza. Considerando esta información, un mapeo lineal es un método viable para estimar el PoG. Sin embargo, se debe tener en cuenta que para ángulos grandes es necesario

realizar regresiones lineales o interpolaciones con el objetivo de compensar errores que surgen de no linealidades.

Las técnicas basadas en modelados 3D, en cambio, utilizan las estructuras geométricas físicas más comunes del ojo humano para calcular el vector 3D que representa la dirección de la mirada. En la sección 2.3.1 a continuación se desarrolla una estrategia para extraer características del ojo con la finalidad de relacionarlas luego con un modelo del ojo en 3D explicado en la sección 2.3.2. Finalmente, en la sección 2.3.3 se utilizan los resultados de las secciones anteriores para poder calcular el vector representativo de la dirección de la mirada utilizando basándose en técnicas de modelado en 3D.

2.3.1 Extracción del Borde y el centro del Iris

Extraer el borde del iris puede llegar a ser una técnica muy útil a la hora de pensar en estrategias que lleven de una manera robusta y eficiente a la detección de la mirada. En el trabajo de *R. Kothari; J. L. Mitchell* [16] se desarrolla, por ejemplo, un método de detección de los bordes del iris mediante el gradiente espacial de la imagen y un sistema de votación.

Dado que el iris se encuentra rodeado del blanco del globo ocular, hay en el ojo un cambio brusco de color, lo cual implica un alto gradiente. En particular un área circular negra o más oscura, como el iris, que se encuentra rodeado de un trasfondo más claro produce un campo saliente de gradientes, como puede observarse en la *Figura 12*.

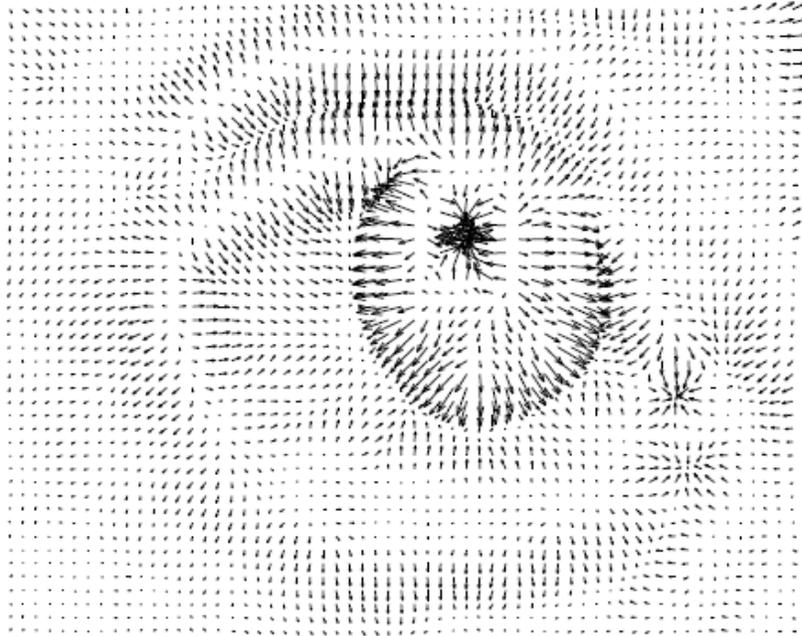


Figura 12: Gradientes de la imagen de un ojo. Se observa en la zona del iris flechas salientes que indican el sentido del gradiente en esa zona. Imagen extraída de [16].

Finalmente, el borde del iris puede ser fácilmente detectado filtrando los gradientes de mayor magnitud y que cumplen con determinado ángulo y sentido.

Adicionalmente si se trazan líneas contrarias a la dirección del gradiente; debido a la forma aproximadamente circular del iris; estas tendrían que interceptarse en un punto que debería coincidir aproximadamente con el centro de la pupila. Bajo este principio, se divide la imagen en “cajas” de 5x5 píxeles y se construyen acumuladores, los cuales cuentan la cantidad de veces que una línea pasa por ellos. Se supone entonces que la caja más votada o con más acumulaciones será la correspondiente al centro del iris. En la *Figura 13* se observa un esquema de las cajas acumuladoras.

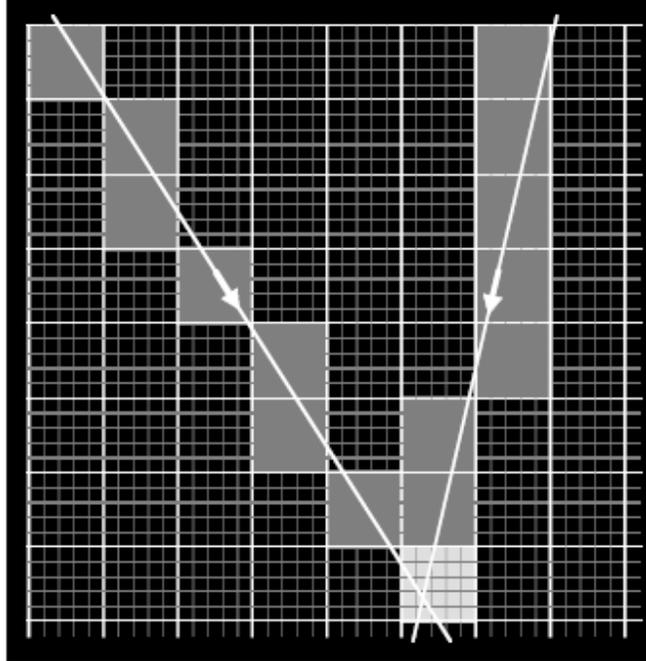


Figura 13: Se observan las cajas de 5x5 píxeles. Si la línea pasa por ellas, las cajas acumulan votos. A mayor brillo en la caja, mayor votación. En la intersección de las dos líneas puede observarse un máximo de votación. Imagen extraída de [16].

Luego, se realiza una matriz con el valor de cada caja y se procede a hacer la selección de dos de ellas, las cuales representarían a los dos iris. Para la elección se recurre a la minimización de la siguiente ecuación:

$$J = (b_{ij} - b_{i'j'})^2 + \lambda(i - i')^2$$

Donde:

i, j : fila y columna correspondiente al centro de la primera caja

i', j' : fila y columna correspondiente al centro de la segunda caja

b_{ij} : la acumulación de la caja de la posición (i, j) y λ : controla el peso relativo de cada criterio.

Esto quiere decir que es preciso encontrar aquellas cajas que tengan un valor acumulativo parecido, y a la vez estén horizontalmente alineadas.

2.3.2 Ajuste de la Elipse

Teniendo en cuenta que el iris es aproximadamente circular, si se proyecta un círculo con una arbitraria pose en tercera dimensión se obtendrá una elipse. Debido a esto, las típicas técnicas de detección de iris que utilicen operadores de borde circular no son lo suficientemente robustas.

La pose de un ojo puede ser definida por la postura de un limbo o contorno circular que es descrito por su punto central L y el vector normal g del plano que lo contiene. Este limbo puede ser obtenido del contorno visible elíptico del iris, mediante una estrategia que es denominada como *Limbus-based eye pose estimation* [17].

Para desarrollar esta técnica es necesario realizar un modelo 3D del ojo en el cual el globo ocular es representado como una esfera de radio R , como puede observarse en la *Figura 14 a)*. Por otra parte, el iris se posiciona en el frente del globo ocular y su contorno es un anillo circular de radio r , como se denota en la *Figura 14 b)*.

La distancia del centro del globo ocular al plano del iris es d (*Figura 14 d)*), dándose la relación:

$$R^2 = r^2 + d^2$$

Se puede definir la línea de la mirada (*eye gaze*), como la línea que conecta el centro del globo ocular con el centro del iris. Como los radios R y r pueden ser considerados constantes, d será aproximadamente una constante con una desviación estándar muy pequeña.

Según el modelo planteado, los bordes visibles que se usan para ajustar la elipse están entre los puntos formados por la intersección entre el iris y los párpados, los cuales se indican en la *Figura 14 c)* como $U1L1$ y $U2L2$.

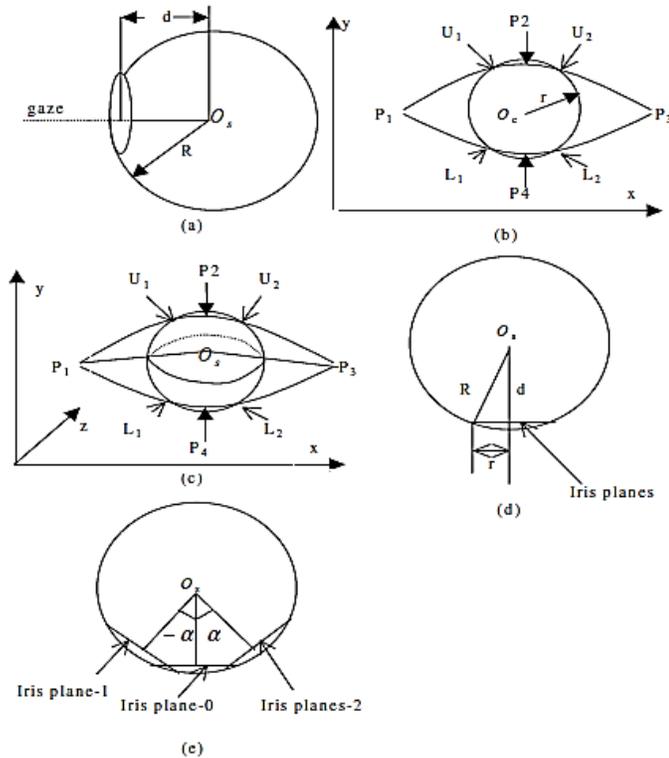


Figura 14: Modelado del ojo humano. a) Se presenta el radio R del globo ocular y la posición del iris. b) Se observa el radio del iris el cual es considerado esférico. c) $U1L1$ $U2L2$ representan las curvas de la parte visible del iris. d) Se esquematiza la relación entre los radios. e) Se observan las posibles rotaciones del iris. Imagen extraída de [17].

Si es posible extraer parte de los bordes visibles del iris, se puede entonces ajustar una elipse definiendo una medida de error. Eusebio de la Fuente López [18] utiliza, por ejemplo, un algoritmo que involucra detección de elipses denominado RANSAC, por sus siglas en inglés *RAN*dom *SAM*ple *CON*sensus. Esta técnica selecciona de forma aleatoria muestras de “ n ” puntos según la cantidad de parámetros que el modelo necesite (en el caso de una elipse $n = 3$); y luego se construye el conjunto de consenso de acuerdo a un umbral de proximidad prefijado. Después de K iteraciones, si el número de puntos del mejor conjunto encontrado es superior a un umbral preestablecido, entonces el modelo se asume como válido.

Otros ajustes se realizan mediante un algoritmo de mínimos cuadrados para encontrar la elipse que más se ajusta a un conjunto de puntos (x, y) en dos dimensiones [19]. Paul L. Rosin, por otra parte, utiliza una técnica basada en acumular parámetros de elipses ajustadas a un subconjunto de datos, ordenando, y seleccionando el promedio de cada parámetro [20].

Una elipse puede expresarse mediante una ecuación de la forma:

$$Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0$$

Donde:

$$A = a^2(\sin \alpha)^2 + b^2 (\cos \alpha)^2$$

$$B = 2(b^2 - a^2) \sin \alpha \cos \alpha$$

$$C = a^2 (\cos \alpha)^2 + b^2(\sin \alpha)^2$$

$$D = -2Ax_c - By_c$$

$$E = -Bx_c - 2Cy_c$$

$$F = Ax_c^2 + Bx_c y_c + Cy_c^2 - a^2 b^2$$

Siendo,

a : Radio de la elipse relacionado al eje x

b : Radio de la elipse relacionado al eje y

α : ángulo de rotación de la elipse

x_c, y_c : Coordenadas del centro de la elipse

2.3.3 Computo de la Normal del plano

Contando con el ajuste de la elipse al contorno del iris, para poder continuar con la estimación de la mirada es necesario pasar a un modelado 3D del ojo. En este sentido, se podría considerar que la circunferencia en perspectiva que representa al iris se encuentra contenida en un plano π . De esta forma, la dirección de la mirada puede ser aproximada mediante el vector normal \vec{n}_π del plano π .

Kanatani K. en su trabajo sobre interpretación de cónicas en 3D aplicadas a las imágenes [21], realiza un desarrollo sobre cómo calcular el vector normal \vec{n}_π . También es útil considerar lo explicado en el paper de Eusebio de la Fuente López [18], el cual utiliza el trabajo de Kanatani para realizar una aplicación en imágenes de una máquina industrial. Esta máquina posee una forma aproximadamente elíptica y encontrando la normal es posible detectar su posición.

Con el objetivo de calcular el vector \vec{n}_π , se define la matriz de covarianza asociada a una elipse como:

$$Q = \begin{bmatrix} A & \frac{B}{2} & \frac{D}{2f} \\ \frac{B}{2} & C & \frac{E}{2f} \\ \frac{D}{2f} & \frac{E}{2f} & \frac{F}{f^2} \end{bmatrix}$$

Donde

f : Distancia focal de la cámara utilizada.

Si la elipse representa la proyección de un círculo, luego, los autovalores de la matriz Q λ_1, λ_2 y λ_3 deben cumplir la siguiente condición:

$$\lambda_3 < 0 < \lambda_1 \leq \lambda_2$$

Sean U_2 y U_3 los autovectores unitarios asociados a sus respectivos autovalores λ_2 y λ_3 . Entonces la normal del círculo viene dada por:

$$\vec{n}_\pi = \pm \sqrt{\frac{(\lambda_2 - \lambda_1)}{(\lambda_2 - \lambda_3)}} U_2 \pm \sqrt{\frac{(\lambda_1 - \lambda_3)}{(\lambda_2 - \lambda_3)}} U_3 \quad (1)$$

Dado que los signos de los autovectores U_2 y U_3 son arbitrarios, y que generalmente $\lambda_1 \neq \lambda_2$, entonces habrán dos soluciones a la interpretación 3D.

Esta dualidad puede ser resuelta con diversos métodos, en uno de ellos Jian-Gang Wang aplica el cálculo de las esquinas 3D cuando se utiliza el procesamiento de un solo ojo [17]. Cuando el procesamiento es aplicado en ambos ojos utiliza una constante que se obtiene a partir de las dimensiones antropométricas de los mismos. Haiyuan Wu, por otra parte, realiza una minimización del cuadrado del producto vectorial de la normal de ambos ojos para encontrar la normal correcta [22].

2.4 Definición de la Medida de Desempeño

Con la finalidad de determinar la calidad de un algoritmo se puede, entre otras técnicas, calcular la precisión y sensibilidad del mismo.

La precisión de un algoritmo se define como la cantidad de identificaciones positivas correctas sobre la cantidad total de detecciones, para lo cual se utiliza la siguiente ecuación:

$$Precision = \frac{VP}{VP + FP} \quad (2)$$

Donde VP son los Verdaderos Positivos, y FP son los Falsos Positivos.

Por otra parte, la sensibilidad se define como la cantidad de detecciones verdaderas sobre la cantidad total de elementos (suma de los que fueron detectados más los que no fueron detectados):

$$Sensibilidad = \frac{VP}{VP + FN} \quad (3)$$

Donde FN son los falsos negativos.

En la *Figura 15* se observa un esquema en donde se encuentran representadas las definiciones antes desarrolladas.

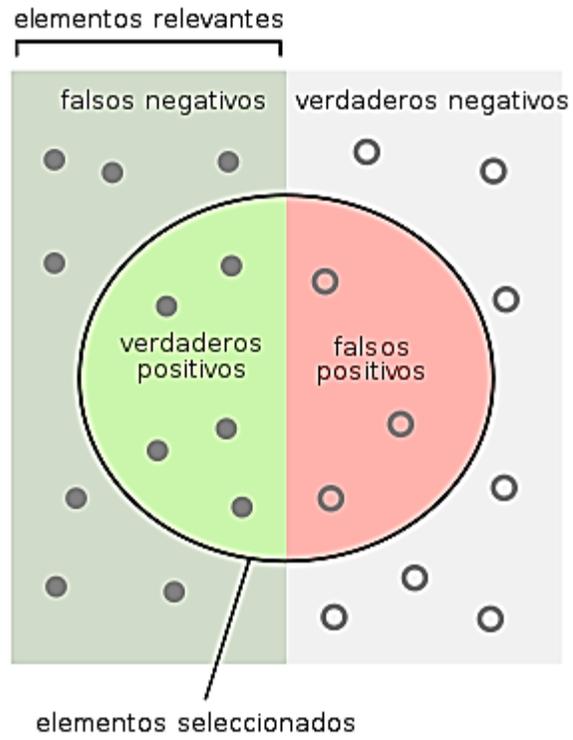


Figura 15: Esquema representativo de la clasificación de detecciones. Imagen extraída de [23]

CAPITULO 3. Método Propuesto: Detección ocular

3.1 Introducción

Las técnicas implementadas en este trabajo son variadas y apuntan a un procesamiento que se pueda aplicar a una imagen obtenida desde un Smartphone o webcam, sin necesidad de una fuente de luz externa, soportes, múltiples cámaras o imágenes de alta resolución. Se inicia con una ubicación global de la cara, para luego comenzar a recortar la ROI hasta extraer la información necesaria para poder ajustar una elipse y hallar el vector normal correspondiente. El algoritmo se encuentra desarrollado de forma que solo funciona si en todas las imágenes hay una única cara; y los ojos se encuentran dentro del rango de visión de la cámara.

Este desarrollo se realiza en un software denominado MATLAB (MATrix LABoratory). El mismo consiste en un sistema de cómputo numérico que ofrece un entorno de desarrollo integrado (IDE) con un lenguaje de programación propio (lenguaje M).

3.2 Recopilación de Imágenes para un Dataset

Los dataset utilizados en este trabajo se componen de tres videos en el entorno de un automóvil, donde el usuario mira hacia diferentes puntos, pero no se encuentra manejando. Las imágenes de estos videos fueron obtenidas con la cámara de un Smartphone cuidando tener una iluminación controlada, evitando luces y/o sombras. Los datos de dicha cámara pueden ser observados en la *Tabla 1*.

Información Smartphone	
Modo de imagen	GRGB
Megapixels	15.9 MP
Resolución de cámara web	4608×3456
Relación de aspecto	1.7

Tabla 1: Características de la cámara del Smartphone utilizado para la adquisición de imágenes en el entorno de un automóvil.

Con motivo de extraer los parámetros intrínsecos de la cámara, se realiza una calibración mediante una herramienta de MATLAB denominada *Camera Calibration Toolbox*. Este proceso se observa en la *Figura 16*.

Los resultados de dicha calibración se encuentran en la *Tabla 2*.

Calibración Cámara	
Distancia Focal	[982,3660 987,9941]
Distorsión Radial	[0.0039 -0,1017]
Error Distancia Focal	[9,3784 9,4867]

Tabla 2: Parámetros Intrínsecos de la cámara obtenidos mediante la herramienta de Matlab.

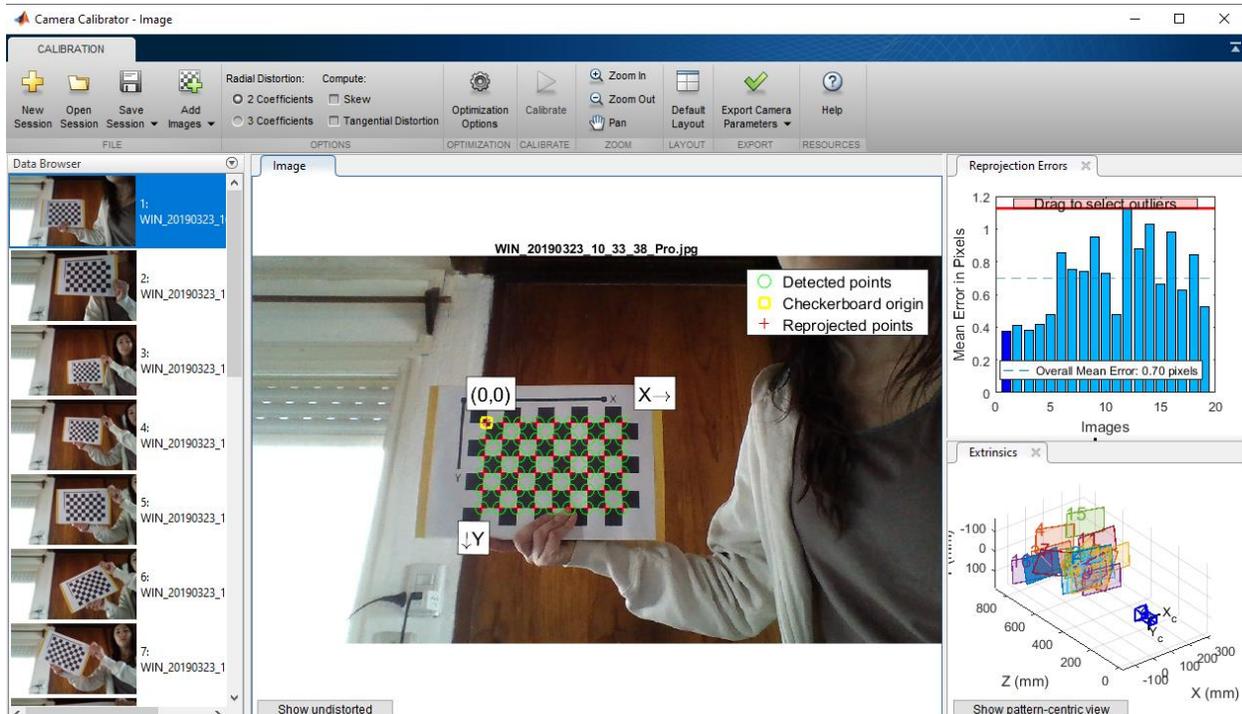


Figura 16: Captura de pantalla de la calibración de cámara realizado con la finalidad de obtener los parámetros intrínsecos correspondientes.

3.3 Implementación de la Detección de ROI

Para poder extraer la información necesaria en las siguientes etapas, es de gran importancia poder delimitar de manera precisa el área de interés o ROI de los ojos en la imagen analizada.

En la *Figura 17* se muestra un diagrama de flujo de los pasos utilizados en el algoritmo propuesto para conseguir la detección de la cara y de los ojos.

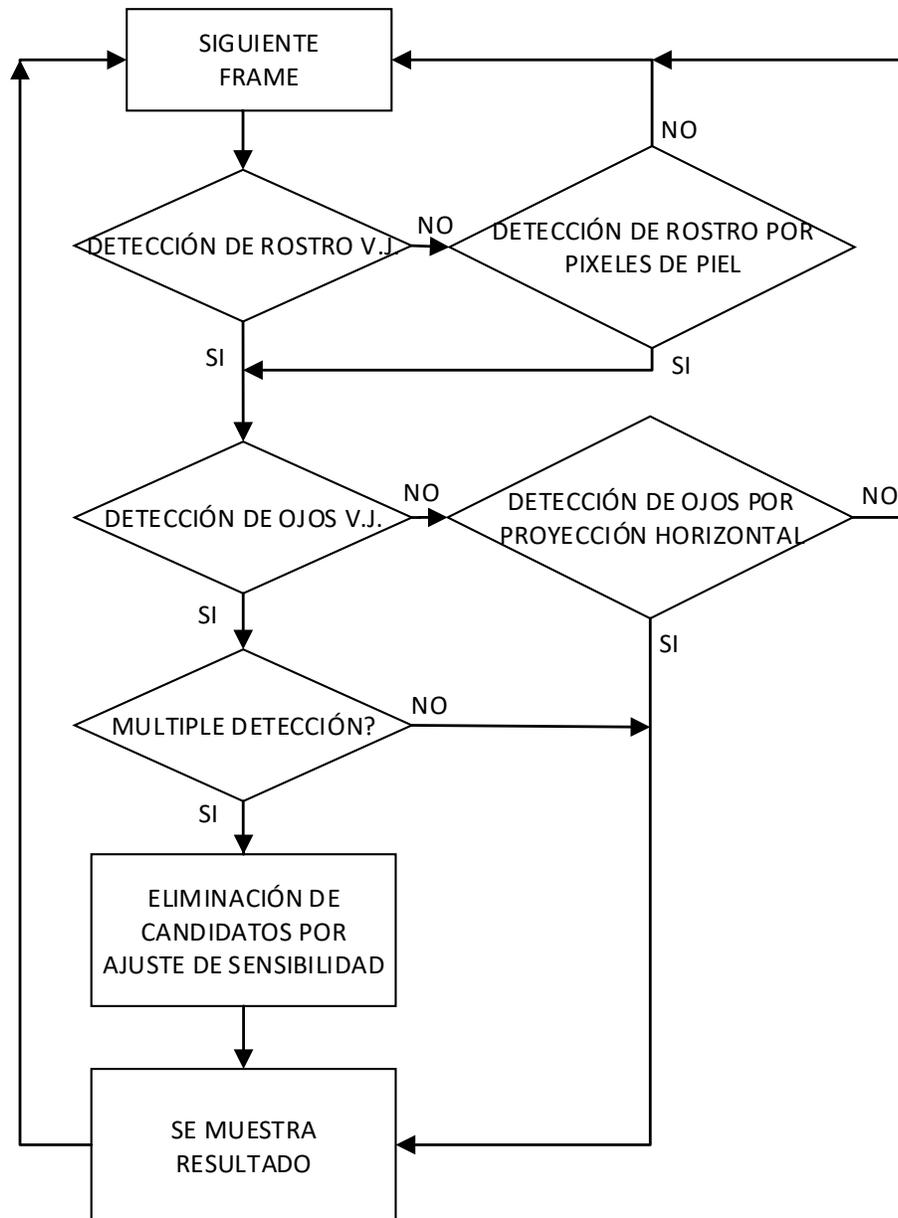


Figura 17: Diagrama de flujo del algoritmo desarrollado. Se observan técnicas adicionales en caso de que el algoritmo de Viola Jones no funcione. Bajo el supuesto de que hay una sola cara en la imagen, se eliminan las múltiples detecciones.

Como se observa en la *Figura 17*, para este trabajo se decidió emplear el algoritmo de Viola Jones en dos etapas. En la primera, se realiza la detección del rostro, y si este fue exitoso se recorta esa sección. En la segunda etapa, sobre el rostro detectado, nuevamente se aplica el algoritmo de Viola Jones con una configuración que permita extraer la ROI correspondiente al par de ojos.

Si al aplicar el algoritmo hay más de un candidato, la sensibilidad del mismo se disminuye con el objetivo de aumentar la precisión en las detecciones. En la *Figura 18* se observan múltiples detecciones, pero bajando la sensibilidad se logra descartarlas.

Si aun reduciendo la sensibilidad las detecciones no se limitan a un solo resultado, entonces se utiliza otro algoritmo alternativo encargado de elegir el candidato correcto. El mismo se basa en detectar píxeles de piel en cada una de las posibles caras, tal como se discute en la sección 2.2.2. Luego, la opción elegida será aquella que posea mayor cantidad de píxeles de piel.

Otro posible escenario es que no se detecte ninguna cara, en ese caso la sensibilidad del algoritmo se aumenta hasta encontrar al menos una detección. Si aun así no hay resultado, teniendo en cuenta la premisa de que siempre en la imagen habrá una cara y será única; se aplica otro método de localización basado nuevamente en los píxeles de piel. En este caso, al resultado se le aplica una operación morfológica de apertura para limpiar píxeles aislados y unificar regiones. Finalmente, de todos los candidatos encontrados se elige como verdadero a aquel que posea mayor cantidad de píxeles.

En la *Figura 19* se observa el resultado de aplicar dichos métodos de detección de cara.

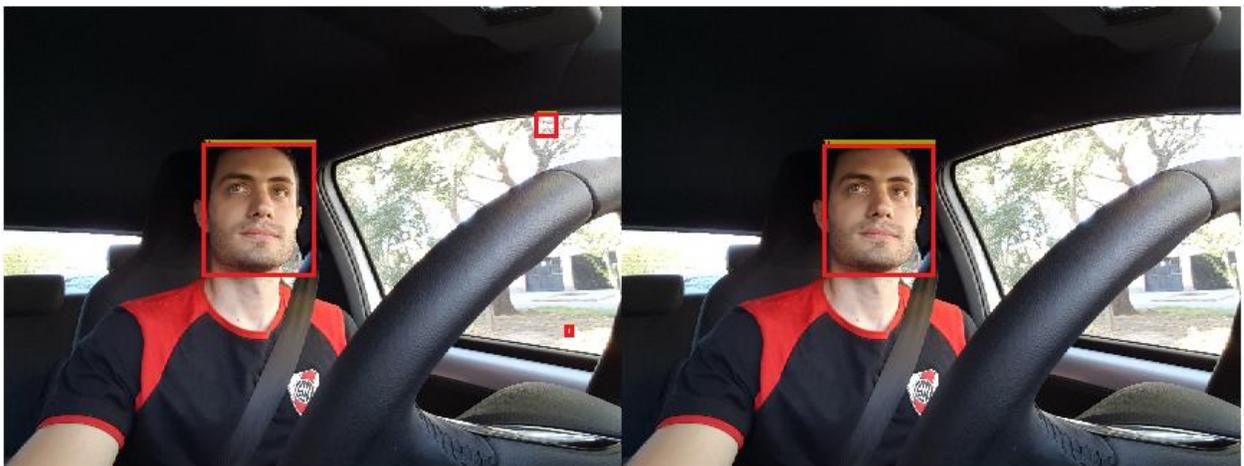


Figura 18: En la imagen de la izquierda se observan tres detecciones de cara. Bajando la sensibilidad del algoritmo se obtiene la imagen de la derecha.

Detección de cara

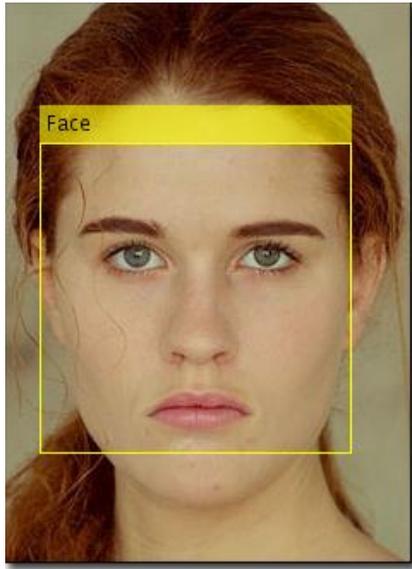


Imagen Recortada



Figura 19: Se observa cómo la región de interés facial es delimitada.

Un procesamiento similar es el que se utiliza para detectar la ROI de los ojos: utilizando la imagen delimitada de la cara, se procede a aplicar nuevamente el algoritmo de Viola Jones, pero esta vez para detectar el par de ojos.

De nuevo, si hay más de un candidato se baja la sensibilidad del algoritmo con el objetivo de reducir las detecciones. En la *Figura 20* puede observarse un ejemplo de esta secuencia:

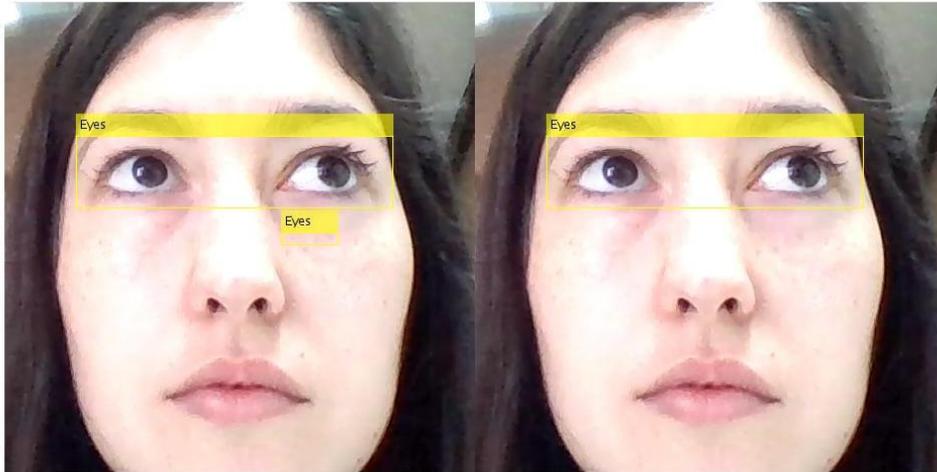


Figura 20: Luego de aumentar la sensibilidad del algoritmo de Viola Jones, se logra descartar las detecciones falsas.

Si lo anterior no funciona, se recurre a un algoritmo alternativo para elegir los candidatos a ojos correctos. El mismo consiste en evaluar el tamaño y la ubicación de los candidatos: aquel candidato que posea un tamaño adecuado y que se encuentre por arriba de la mitad de la cara es el escogido.

En el caso de que el algoritmo de Viola Jones no detecte ningún par de ojos también se recurre a otro procesamiento: el mismo consiste en delimitar el contorno de la cara utilizando una vez más el recurso de detectar píxeles de piel para luego realizar una proyección horizontal. Este procesamiento puede observarse en la *Figura 21*. Tal como se explica en la sección 2.2.1, el primer pico detectado en la proyección horizontal se considera que es la ceja, mientras que el segundo pico la zona de los ojos. Luego, utilizando una medida aproximada del tamaño de la región de los ojos con respecto al de la cara se delimita la ROI de los ojos. Finalmente, en la *Figura 22* se observa el resultado de aplicar este algoritmo.

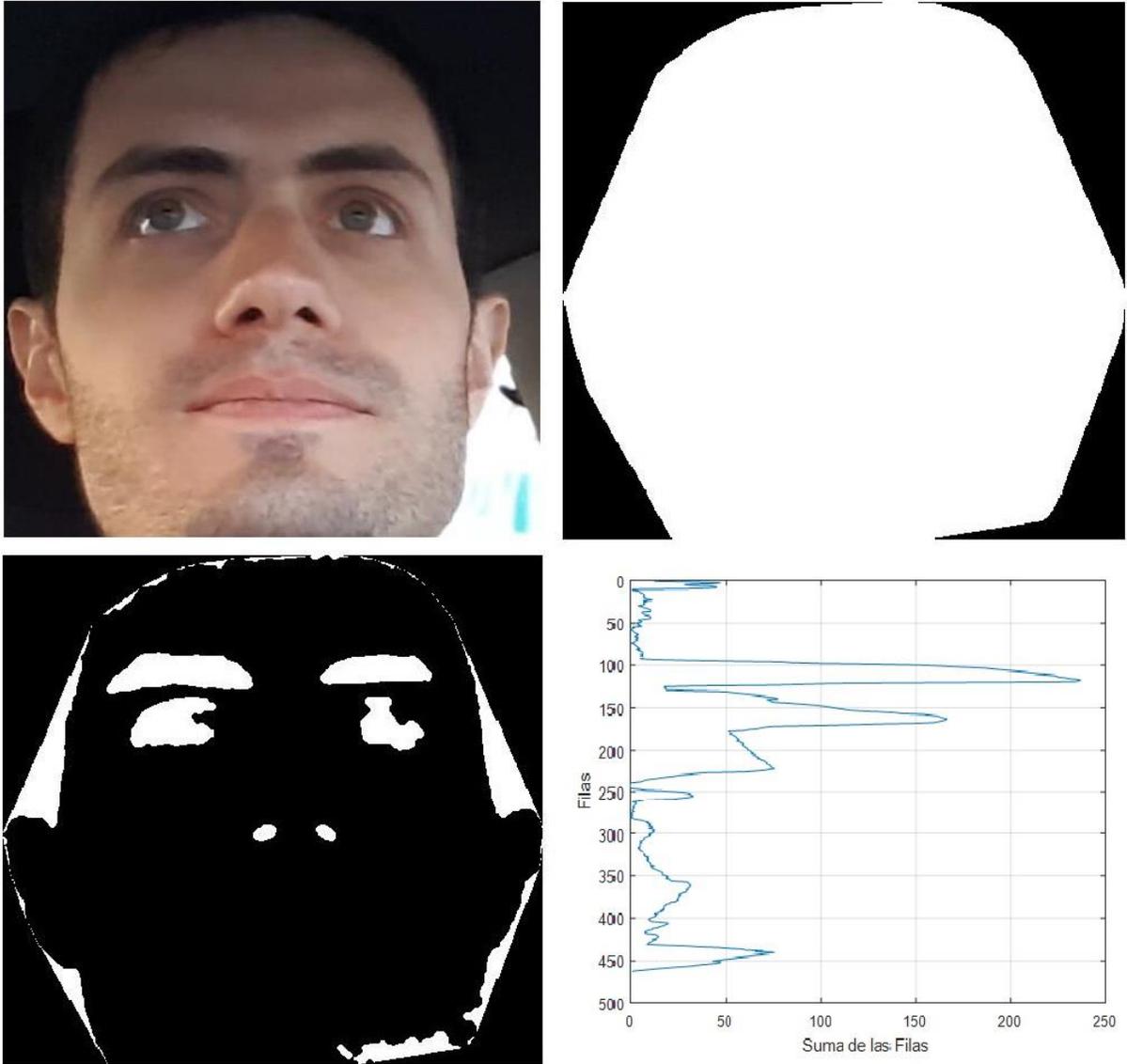


Figura 21: Arriba se observa la imagen del rostro, y a la derecha la máscara hallada mediante la detección de píxeles de la cara. Abajo se observa la imagen binarizada y con la máscara ya aplicada que elimina el contorno indeseado. Al costado la proyección horizontal de dicha imagen. Se pueden observar los picos máximo correspondientes a las cejas y los ojos.

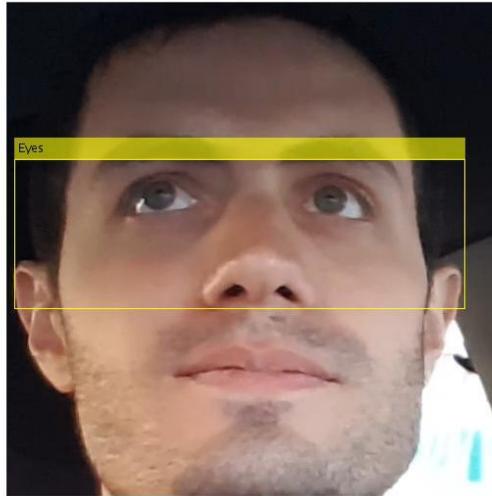


Figura 22: Resultado final del procesamiento: la detección de la Región de Interés correspondiente a los ojos.

Con el objetivo de bajar el costo computacional se utiliza información del frame anterior, siempre que sea posible. Esto se realiza presuponiendo que la cara se ha movido muy poco de lugar, por lo tanto, la región de interés del rostro es la misma que la del frame anterior. Si al aplicar el algoritmo de Viola Jones para hallar los ojos estos no se detectan, entonces se procede a calcular la ROI de la cara, ya que la misma podría haberse corrido con respecto a las coordenadas anteriores.

3.4 Determinación del Centro del Iris

Una vez delimitada la región de interés de los ojos, es posible comenzar con la detección del iris de cada ojo. Para ello, se aplica el método desarrollado en la sección 2.3.1 que se basa en el gradiente de la imagen. En la *Figura 23* se observa el resultado de calcular el gradiente en la imagen. Luego los mismos se filtran, dejando solamente aquellos de mayor magnitud.



Figura 23: Arriba, la imagen presenta un gradiente representado con pequeñas flechas azules. Abajo, solamente los gradientes que superan un umbral determinado son representados.

Para trazar las líneas con dirección contraria al gradiente se utiliza el algoritmo de Bresenham [24]. El resultado puede observarse en la *Figura 24*: se aprecia una mayor concentración de líneas en la zona donde se encuentra el iris del ojo.

Luego, como se expuso anteriormente, la imagen se divide en “cajas” de 5x5 píxeles y se construyen acumuladores, los cuales cuentan la cantidad de veces que una línea pasa por ellos. Se supone que, bajo este principio, la caja más votada o con más acumulaciones será la correspondiente al centro del iris. Finalmente, el centro del iris es determinado tal como se puede observar en la *Figura 25*.

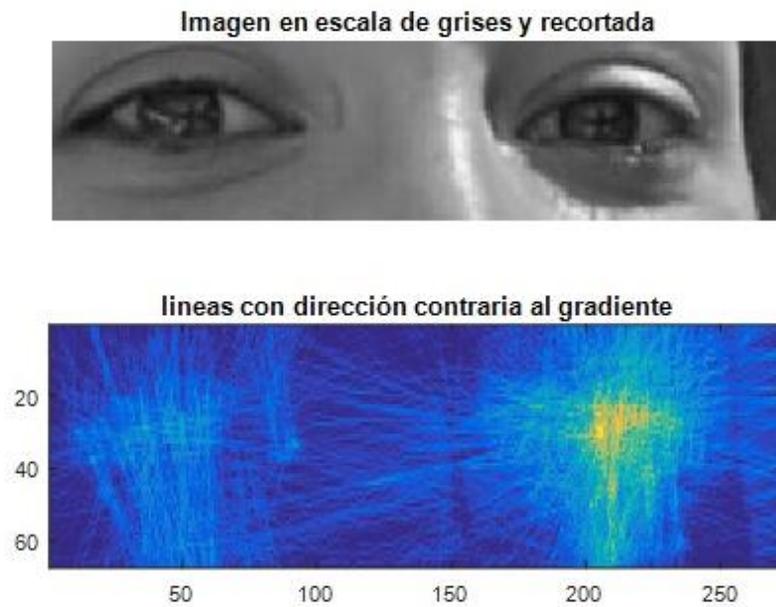


Figura 24: Arriba, la imagen recortada y en escala de grises. Abajo, líneas con dirección contraria al gradiente encontrado. Se observa una intersección mayor de las líneas en la zona del iris.

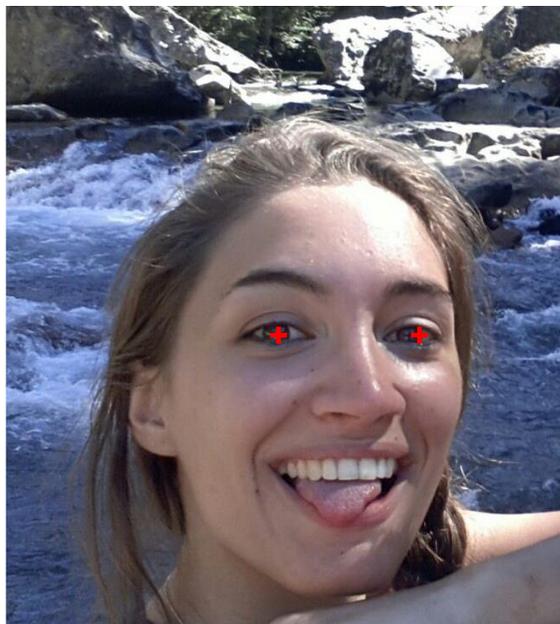


Figura 25: El centro del iris es estimado luego de trazar las líneas contrarias al gradiente y calcular las cajas acumulativas.

3.5 Implementación del Ajuste de la Elipse

En esta sección se desarrollan los siguientes procesamientos:

- Detección de la ROI de cada ojo por separado
- Construcción de una máscara a partir de los gradientes del iris
- Detección de bordes
- Ajuste de la elipse

Los candidatos a centro de los iris se pueden utilizar para determinar la ROI del ojo derecho e izquierdo. Esto se logra utilizando una distancia determinada por medidas antropométricas, en este caso 3/10, en relación al tamaño de la ROI de ambos ojos. El resultado de este proceso se puede observar en la *Figura 26*.

El siguiente paso en esta etapa es la detección de los bordes que definen el iris. Esto puede lograrse filtrando los gradientes con el objetivo de extraer solamente aquellos que se encuentran en el borde del iris.

Se toman entonces los gradientes de cada imagen de los ojos, reteniendo aquellos que cumplen con una cierta magnitud y dirección. La magnitud del gradiente indica el contraste entre píxeles, lo cual es deseable debido al contraste del iris con el globo ocular; y la dirección (preferentemente horizontal) permite obtener aquellos gradientes correspondientes al borde del iris.

Teniendo en cuenta la ubicación de los gradientes, para los bordes del lado izquierdo del centro de cada ojo solamente se consideran aquellos que tengan un ángulo de entre 160 y 240 grados, mientras que para el lado derecho entre 290 y 360 grados. Finalmente se realiza un filtrado de los gradientes de forma iterativa: en primer lugar, se estima un rango del posible radio del iris en función del tamaño de la imagen, utilizando una vez más medidas antropométricas del iris. Habiendo descartado algunos gradientes, se vuelve a calcular el centro del iris utilizando la técnica de líneas y matriz de votación antes mencionada, obteniendo el resultado que se observa en la *Figura 27*. A partir de las posiciones de los gradientes se realiza una imagen binarizada la cual se dilata en forma vertical y horizontal. En la *Figura 28* se observa la dilatación horizontal de la imagen binarizada.



Figura 26: Delimitación de la ROI de los ojos izquierdo y derecho.

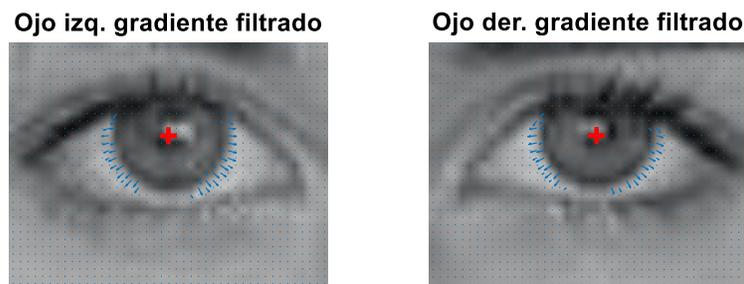


Figura 27 : Filtrado de gradientes y determinación del centro del iris.

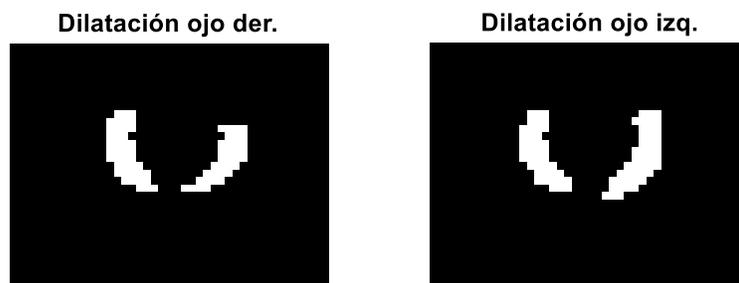


Figura 28: Dilatación de imagen binaria de los bordes de los ojos hallada en función del gradiente filtrado.

En forma paralela, se extraen los bordes de la imagen del ojo mediante la técnica de Canny [25], como se puede denotar en la *Figura 29*. Finalmente, la imagen binarizada es utilizada como una máscara para obtener los bordes del iris, tal como se puede ver en la *Figura 30*.

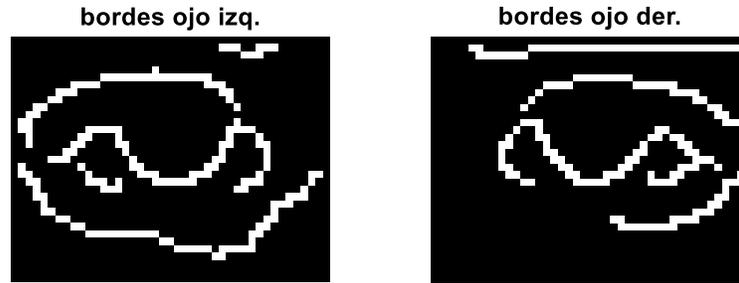


Figura 29: Bordes de los ojos izquierdo y derecho.

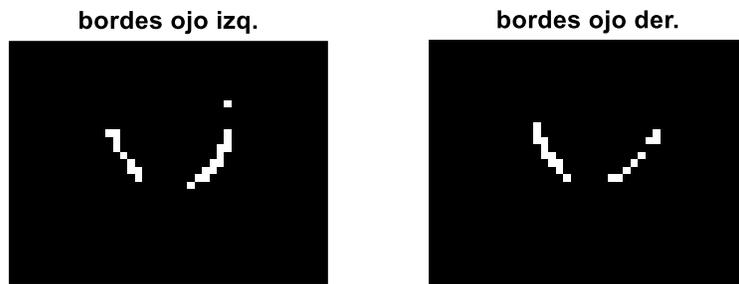


Figura 30: bordes filtrados de los ojos izquierdo y derecho respectivamente.

Los bordes obtenidos se utilizan para ajustar una elipse en ambos ojos buscando el menor error posible. Tal como se menciona en la sección 2.3.2, hay diversas estrategias para realizar este procesamiento. En este trabajo en particular, se elige un algoritmo que utiliza mínimos cuadrados para encontrar la elipse que más se ajusta a un conjunto de puntos (x, y) en dos dimensiones, debido a su simplicidad y efectividad.

En la Figura 31 se puede observar el resultado de la aplicación de estos pasos: el ajuste de la elipse en la imagen.



Figura 31: Ajuste de la elipse en la imagen recortada de la zona ocular.

Con el objetivo de eliminar detecciones erróneas, se aplican diferentes criterios que la elipse encontrada debe cumplir:

- La excentricidad de la elipse (la relación entre el radio mayor y menor) debe ser mayor a 0,5.
- El radio mayor de la elipse debe estar entre los límites establecidos por el promedio de los radios de las elipses encontradas en los frames anteriores.

Si la elipse pudo ser ajustada en uno de los ojos, pero no así en el otro, se toma la misma elipse para ambos ojos.

3.6 Implementación del vector normal al plano

Tal como se explicó en la sección 2.3.3, la línea que pasa por el centro del iris y el centro del globo ocular es justamente la dirección de la mirada que se necesita encontrar. Se puede considerar entonces a la elipse encontrada como un círculo en perspectiva, por lo cual se recurre a la ecuación (1) para calcular la normal. Como se observa en la *Figura 32*, el cálculo de la normal deriva en dos posibles resultados.



Figura 32: Se observan las dos normales encontradas para la elipse ajustada en el iris del ojo izquierdo. Una de las normales se encuentra en color amarillo, mientras que la otra, en color rojo.

3.7 Detección de las esquinas de los ojos

Con motivo de elegir la dirección de la mirada correcta, se requiere de un punto de referencia en la imagen que sea lo suficientemente robusta. La estrategia utilizada en este caso implica calcular las esquinas de los ojos utilizando un ajuste polinomial sobre los párpados de los ojos [26].

En primer lugar, se pasa la imagen del ojo del RGB al espacio de color denominado HSV, por sus siglas en inglés *Hue, Saturation, Value* [27]. Esto se realiza debido a que la esclerótica del ojo presenta una saturación baja, por lo cual es más fácil de detectar en ese espacio de color. A la matriz de saturación se le aplica un filtro horizontal de mediana unidimensional, seguido por una ecualización de intensidades de la imagen. En la *Figura 33* se observan los resultados de los pasos antes mencionados. Como se busca el contorno del ojo, utilizando las coordenadas de los píxeles de la elipse correspondiente al ajuste del iris, se elimina el iris de la imagen ecualizada. Luego se le aplica un filtro para detectar las partes más oscuras de la imagen. Este procesamiento puede observarse en la *Figura 34*.

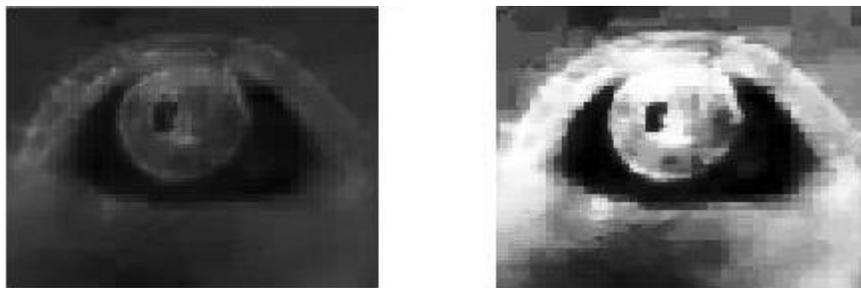


Figura 33: En la imagen de la izquierda se observa el canal de saturación. En la derecha se observa la imagen filtrada y ecualizada.

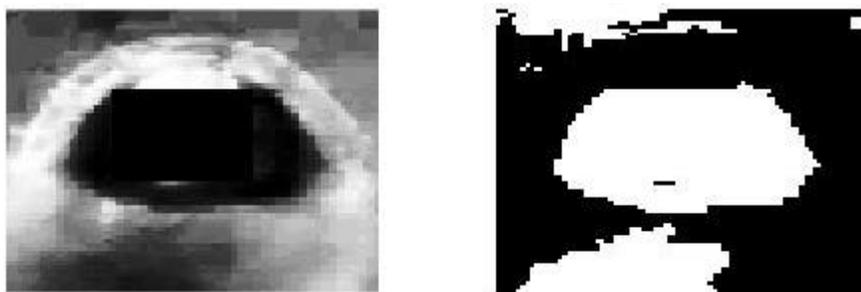


Figura 34: En la izquierda se observa la imagen con la mayor parte del iris eliminado. En la derecha, la imagen se binariza.

Mediante una función de Matlab, la cual encuentra componentes conectados en la imagen binaria, se selecciona la región con mayor cantidad de píxeles. Luego, con el algoritmo de Canny se extrae el contorno, y se realiza un ajuste polinomial sobre el contorno superior e inferior, tal como se puede observar en la *Figura 35*. Los puntos en los que ambas funciones polinómicas se cruzan se consideran como las esquinas de los ojos, o por lo menos puntos cercanos a las mismas.

Tomando como referencia la distancia del centro del iris a las esquinas, se determina cuál de las normales es la correcta.

En la *Figura 36* se muestra el resultado de aplicar este procesamiento.



Figura 35: Ajuste de un polinomio de segundo grado sobre el contorno superior e inferior.



Figura 36: En rojo se observan las esquinas del ojo izquierdo obtenidas luego del procesamiento realizado. En amarillo se observa la normal elegida a partir de la distancia de las esquinas al centro del iris.

3.8 Creación de Dataset

Se construyen tres dataset a partir de tres videos realizados en distintas condiciones. El armado de los mismos tiene como finalidad:

- Ajustar el algoritmo desarrollado, llevando la precisión y sensibilidad a valores aceptables.
- Medir el desempeño de este algoritmo a partir de los resultados obtenidos.

En resumen, esta data servirá para el ajuste/calibración de la herramienta diseñada y la evaluación de su desempeño.

En los videos, el usuario debe sostener la mirada por varios segundos en una serie de posiciones definidas en el interior de un automóvil, las cuales son:

- Espejo retrovisor izquierdo
- Espejo retrovisor frontal
- Camino
- Espejo retrovisor derecho

Este procedimiento tiene como finalidad conocer la categoría verdadera observada para poder ajustar el algoritmo y mejorar luego su desempeño.

3.9 Puesta a Punto del Sistema

Sobre los dataset de imágenes adquiridos se implementan todas las técnicas comentadas en las secciones anteriores de detección de cara, de ojos y de mirada. Una vez realizado esto, se ejecuta un ajuste del algoritmo, es decir, se definen las regiones de proyección del vector normal.

Como primer paso, se descartan aquellas imágenes en donde se detecta que el ojo está cerrado, además de las transiciones entre una posición y otra. De los frames que quedan, se selecciona de forma aleatoria un 60% del total, los cuales son utilizados para la calibración o ajuste del sistema. El resto de las imágenes se utiliza para la validación del desempeño del algoritmo.

Con los frames de calibración ya seleccionados, se calcula el promedio de las normales en cada posición y, en función del signo positivo o negativo del resultado, se eliminan aquellas direcciones de mirada que debido a un error del algoritmo poseen un sentido contrario.

A partir de esta información, puede establecerse una región de representación para cada una de las categorías e identificar hacia donde está dirigida la atención visual del usuario.

Para definir dichas regiones se puede recurrir a dos metodologías:

- Regiones rectangulares
- Regiones definidas por SVM, por sus siglas en inglés *Support Vector Machine*

Las regiones rectangulares se definen por medio de las coordenadas de los máximos y mínimos puntos de cada categoría. Es una técnica sencilla, sin embargo, cuando dos grupos de puntos se entremezclan o están muy juntos esta estrategia no es eficaz.

El método de SVM, en cambio, utiliza un conjunto de algoritmos desarrollados por Vladimir Vapnik [28]. Se recurre a esta estrategia con la finalidad de delimitar de manera correcta las regiones en el plano xy que definen cada categoría. Se trata de una técnica supervisada, dado que se conoce la clase o etiqueta de cada dato o muestra de entrada, el cual sería en este caso cada normal calculada. Se entrena entonces un modelo que posee la hipótesis de que es posible separar los datos mediante hiperplanos lo más eficientemente posible. El hiperplano a encontrar o “vector soporte” se construye entre los puntos de ambas clases que se encuentren más cercanos entre sí. De esta forma, las nuevas muestras pueden ser clasificadas a una u otra clase utilizando al vector soporte hallado mediante el modelo construido.

En la *Figura 37* se observa un esquema del modelo de SVM: dos grupos de puntos representados por círculos azules y cuadrados rojos se separan por el hiperplano que representa la mayor separación posible entre los mismos.

En este trabajo se utiliza un algoritmo SVM multiclase, debido a que se cuenta con cuatro categorías o grupos de muestra.

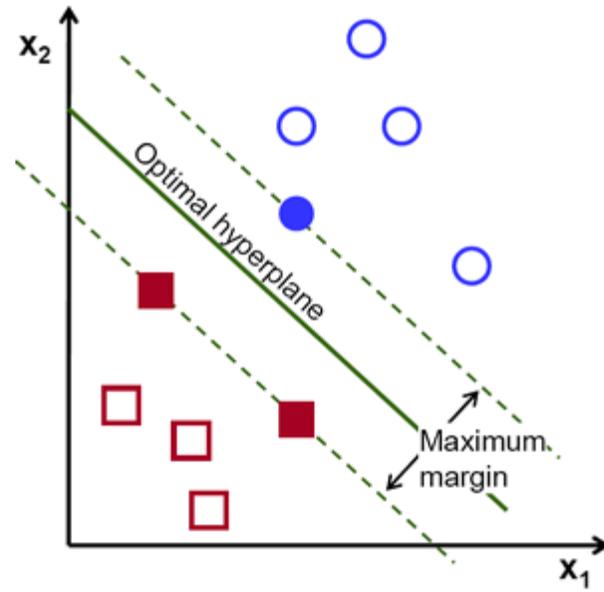


Figura 37: Esquema de funcionamiento del algoritmo de SVM. Se halla el hiperplano óptimo, es decir, aquel que posee el margen máximo entre los grupos de puntos. Imagen extraída de [29].

CAPITULO 4. Resultados

4.1 Condiciones Generales

El método propuesto de detección se analiza sobre los tres dataset descritos en la sección 3.8, y su desempeño es evaluado según el cálculo de la precisión y la sensibilidad. Como ya se menciona anteriormente, de los frames considerados como útiles un 60% será empleado para el ajuste del algoritmo (calibración) y el resto para evaluación del desempeño.

Se opta entonces por medir el algoritmo por etapas o categorías; utilizando las ecuaciones (2) y (3) de la sección 2.4.

A continuación, se enumeran los contextos y resultados observados con los tres dataset creados en el entorno de un automóvil. Cada uno de ellos posee distintas condiciones, y se le aplican distintas técnicas con el objetivo de medir el algoritmo desarrollado. Con el propósito de representar de una manera óptima los resultados, se realiza la siguiente asociación de colores, lo cual se aplica en todos los gráficos:

- Posición 1: Puntos en amarillo, dirección de la mirada en Espejo Retrovisor Izquierdo
- Posición 2: Puntos en verde, dirección al camino
- Posición 3: Puntos en rojo, Espejo retrovisor Frontal
- Posición 4: Puntos en verde, Espejo Retrovisor Derecho

4.1.1 Dataset I: Condiciones

El Video I es realizado en el entorno de un automóvil con iluminación controlada, tal como se observa en la *Figura 38*. La cámara es colocada en el vidrio del parabrisas, al costado derecho del volante.



Figura 38: Imágenes por posición extraídas del Video I, realizado en el entorno de un automóvil, con iluminación controlada. Debido a la utilización de la cámara frontal del Smartphone, el video se encuentra espejado.

En el video, el usuario mira 3 segundos en cada una de las categorías definidas en la sección 3.8, con un movimiento de cabeza controlado, y cuidando siempre que el iris se encuentre dentro del rango visible de la cámara [30]. A su vez, el movimiento del iris en cada categoría es casi nula, y se realizan pocos pestañeos. Por cada posición se obtienen aproximadamente 100 imágenes, de las cuales el algoritmo descarta aquellas en las que detecta que el usuario tiene los ojos cerrados, quedando 70 frames en promedio.

En la *Figura 39* a continuación, se observa a la izquierda el seguimiento en píxeles del centro del ojo izquierdo mientras el ojo se va trasladando de posición en posición. El sistema de referencia se encuentra ubicado en coordenadas de la imagen, estando el origen en el ángulo superior izquierdo. A la derecha de la imagen se observa la proyección de un vector 3D unitario sobre el plano de la cámara.

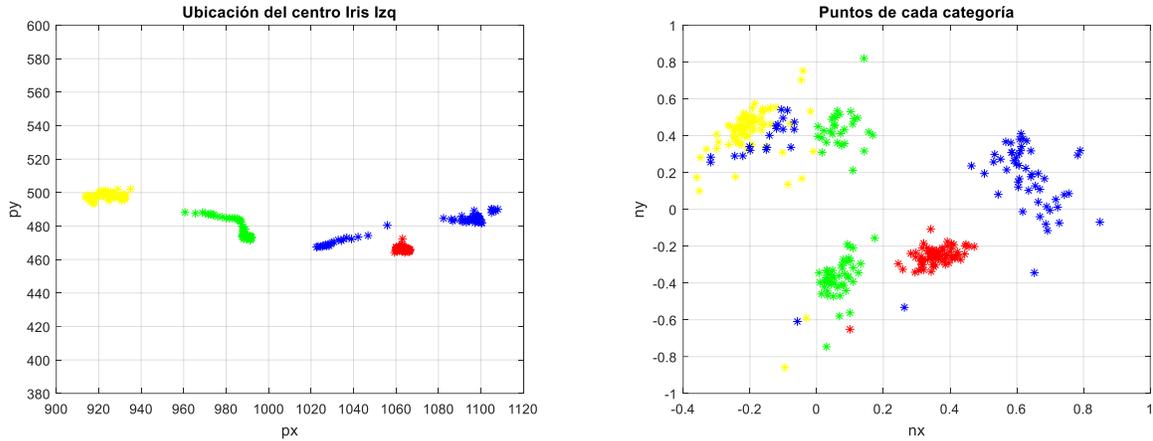


Figura 39: A la izquierda se observa gráfico de la posición del centro del iris del ojo izquierdo en píxeles de la imagen completa. A la derecha, las normales resultado de la calibración realizada en los diferentes puntos característicos de un automóvil.

De las 70 imágenes por posición, se seleccionan de forma aleatoria un 60% para extraer la información necesaria para delimitar la región correspondiente a cada posición. Tal como se explica en la sección 3.8, las normales con signos diferentes a los promedios por categoría se eliminan y luego se recurre a la delimitación por medio de regiones rectangulares, tal como se observa en la Figura 40. Este procesamiento se efectúa para realizar el ajuste/calibración del algoritmo.

La dispersión que se observa en la Figura 40 cuando se mira al camino (puntos en verde) se debe a que se está mirando al frente, con lo cual la elipse es casi esférica y causa que el cálculo de la dirección de la mirada oscile entre dos resultados. Para poder diferenciar estos resultados de la categoría de espejo retrovisor izquierdo (puntos en amarillo) se utiliza la excentricidad de la elipse y la distancia de las esquinas al centro de la elipse como datos adicionales [31].

Las 30 imágenes por posición restantes se utilizan para testear el algoritmo de identificación de mirada.

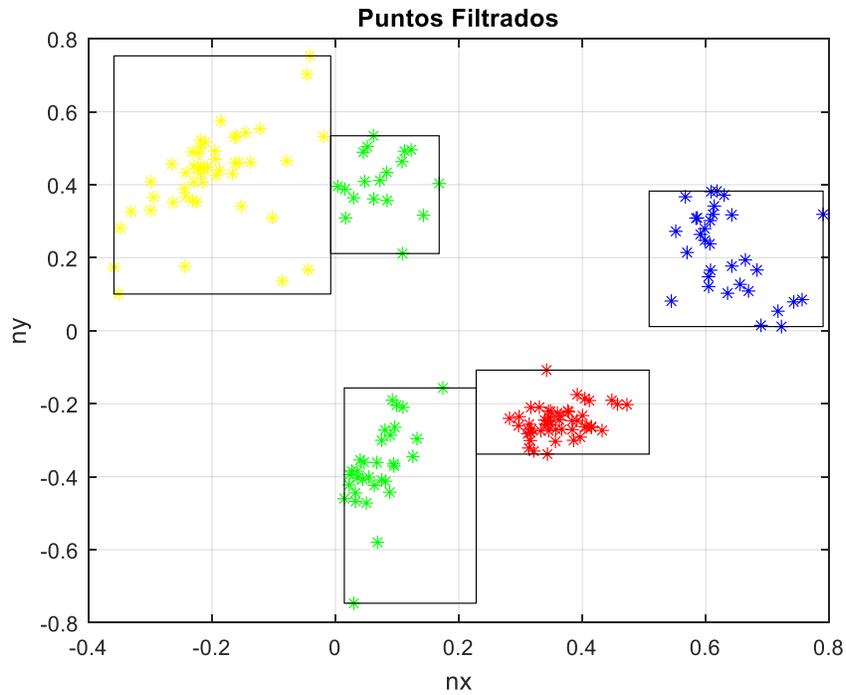


Figura 40: Regiones definidas para cada posición. Las normales que poseían distinto signo que el promedio calculado fueron eliminadas.

4.1.2 Dataset I: Resultados

Utilizando las imágenes del dataset que no se emplearon para la calibración, se calcula en primer lugar la sensibilidad del algoritmo. En el caso de la posición 1, espejo retrovisor derecho, y posición 2, se obtuvo el siguiente resultado observado en la *Figura 41*.

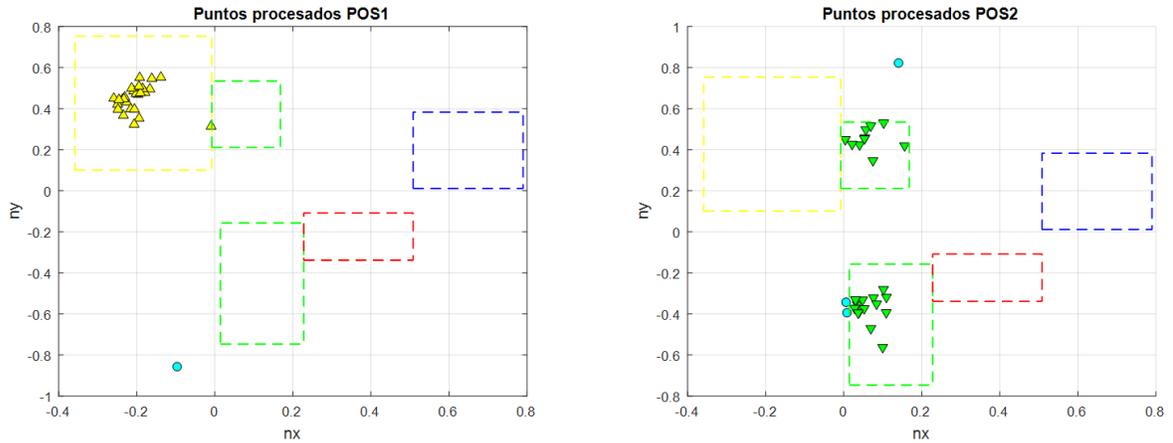


Figura 41: Puntos procesados de la posición 1 (izquierda), posición 2 (derecha). Los círculos celestes son aquellas que no pudieron ser relacionadas con ninguna posición. En línea punteada se observan las regiones de las distintas posiciones.

Luego, se consigue el siguiente resultado:

$$S1_{POS1} = \frac{VP}{VP + FN} \times 100\% = \frac{26}{26 + 1} \times 100\% = 96,29\%$$

$$S1_{POS2} = \frac{VP}{VP + FN} \times 100\% = \frac{25}{25 + 3} \times 100\% = 89,28 \%$$

Donde $S1_{POS1}$ es la sensibilidad asociada a la posición 1 y $S1_{POS2}$ es la sensibilidad asociada a la posición 2, del Video I.

Para la categoría 3, y la categoría 4 correspondiente a cuando se está mirando al espejo retrovisor frontal y al espejo retrovisor derecho, se obtuvieron los resultados observados en la Figura 42:

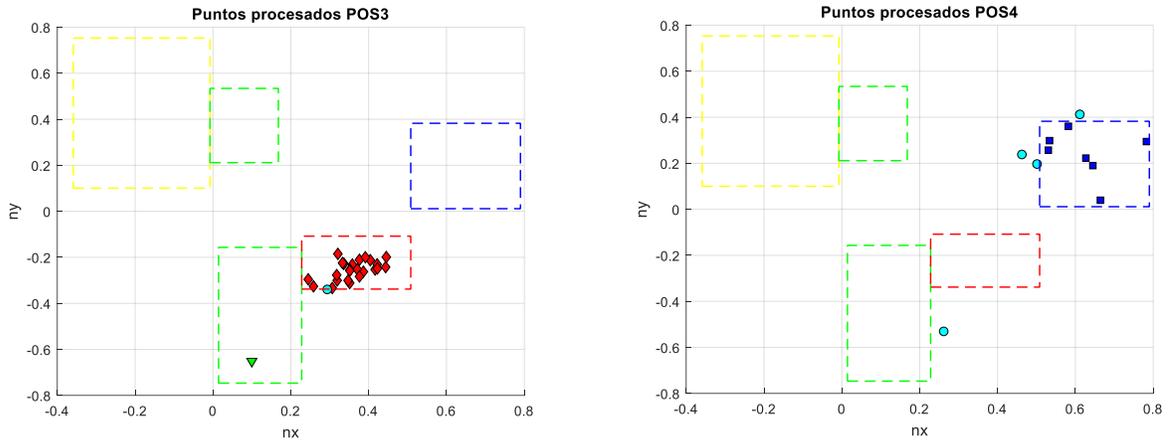


Figura 42: Puntos procesados correspondientes a la dirección del espejo retrovisor frontal y el espejo retrovisor derecho. Los rombos rojos son asociados a la posición 3, mientras que los cuadrados azules a la posición 4. El triángulo verde invertido se asocia a la posición 2, mientras que el círculo celeste es una normal que no fue clasificada en ninguna posición.

Luego, la sensibilidad de la posición 3 $S_{1_{POS3}}$ y de la posición 4 S_{POS4} estarán dadas por:

$$S_{1_{POS3}} = \frac{VP}{VP + FN} \times 100\% = \frac{25}{25 + 2} \times 100\% = 92,59 \%$$

$$S_{POS4} = \frac{VP}{VP + FN} \times 100\% = \frac{7}{7 + 4} \times 100\% = 63,63 \%$$

Con el propósito de calcular la precisión del algoritmo de la detección de la mirada, se realiza un gráfico con todas las normales de validación juntas. En el mismo se pueden apreciar algunos resultados que son clasificados en una categoría errónea, los cuales constituyen los falsos positivos necesarios para calcular la precisión.

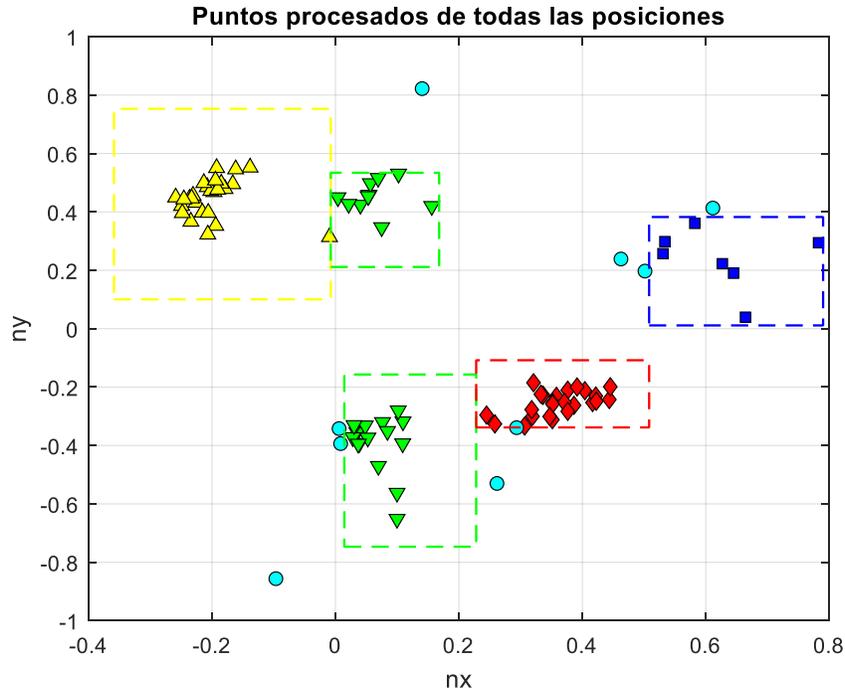


Figura 43: Se observan todas las normales utilizadas para la validación del algoritmo.

Finalmente, se calcula la precisión de cada posición:

$$P1_{POS1} = \frac{VP}{VP + FP} \times 100\% = \frac{26}{26 + 0} \times 100\% = 100\%$$

$$P1_{POS2} = \frac{VP}{VP + FP} \times 100\% = \frac{25}{25 + 1} \times 100\% = 96,15\%$$

$$P1_{POS3} = \frac{VP}{VP + FP} \times 100\% = \frac{25}{25 + 0} \times 100\% = 100\%$$

$$P1_{POS4} = \frac{VP}{VP + FP} \times 100\% = \frac{7}{7 + 0} \times 100\% = 100\%$$

4.1.3 Dataset II: Condiciones

Debido a que el Video I es demasiado controlado en cuanto a movimiento de cabeza y del iris del usuario en cada posición, se decide realizar otro video en donde la persona efectúe mayor cantidad de pestañeo, rotación de cabeza y movimiento del iris [32]. También se opta por cambiar la cantidad de tiempo por categoría a 5 segundos, lo cual permite mayor cantidad de frames para

realizar tanto la calibración como la validación. La cámara esta vez es colocada atrás del volante, quedando de frente al usuario. Debido a que el Smartphone se encontraba invertido cuando se realizó la filmación el video resultó dado vuelta, por lo cual hubo que utilizar un software para rotarlo, lo ocasionó pérdida de calidad de imagen.

De este nuevo dataset se obtienen 620 imágenes, las cuales se procesan en el algoritmo desarrollado en esta tesis [33]. Dejando de lado los frames de transición y aquellos en donde los ojos se encuentran cerrados, se pudieron detectar en promedio 100 normales por categoría; excepto cuando el usuario se encuentra mirando al espejo retrovisor derecho (posición 4) ya que, debido a una excesiva rotación de cabeza, solo se pudieron procesar la cantidad de 6 normales. A causa de esto, dicha categoría es eliminada de la medición.

En la *Figura 45*, a la izquierda, se observa la posición del centro del iris del ojo izquierdo en las imágenes a medida que pasa de posición en posición. A la derecha de la imagen, se observan las normales procesadas por cada categoría. Se observan muchas normales con dirección identificada erróneamente, lo cual ocasiona que se entremezclen los grupos.

Al igual que con el video anterior, se realiza un muestreo aleatorio de un 60% de la cantidad total de normales obtenidas, con la finalidad de utilizarlas para determinar las regiones. Con las normales muestreadas se realiza un promedio, y se descartan aquellos resultados que tengan un signo diferente. Luego se delimitan los espacios por medio de la técnica de regiones rectangulares, lo cual puede observarse en la *Figura 46*. Se observa que en el caso de la posición 2 (camino) y la posición 3 (espejo retrovisor frontal), esta técnica de separación no es eficaz, ya que deja muchas normales de una posición y de otra encuadrados en una categoría errónea. Debido a lo anterior, se decide aplicar un modelo SVM, tal como se explica en la sección 3.8, con el objetivo de mejorar la delimitación de las regiones que definen las cuatro categorías. Los resultados de aplicar ambas técnicas son plasmados en la sección 4.1.4 a continuación.



Figura 44: Se observa una imagen por posición extraídas del Video II, realizado en el entorno de un automóvil. Debido a la utilización de la cámara frontal del Smartphone, el video se encuentra espejado.

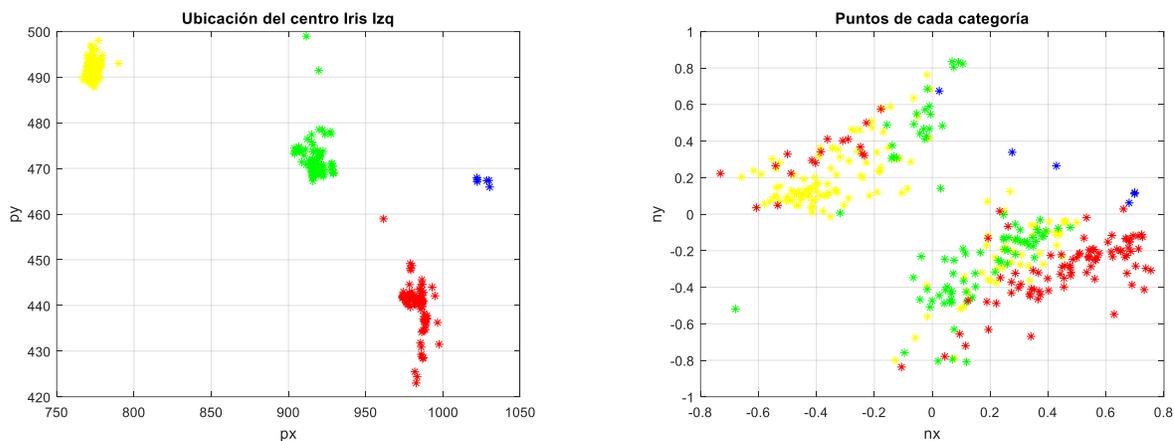


Figura 45: A la izquierda se observa la ubicación del centro del iris del ojo izquierdo en cada categoría definida. A la derecha, las normales de todas las categorías utilizados para realizar la calibración del sistema, sin haber filtrado aquellas normales identificados con signo erróneo.

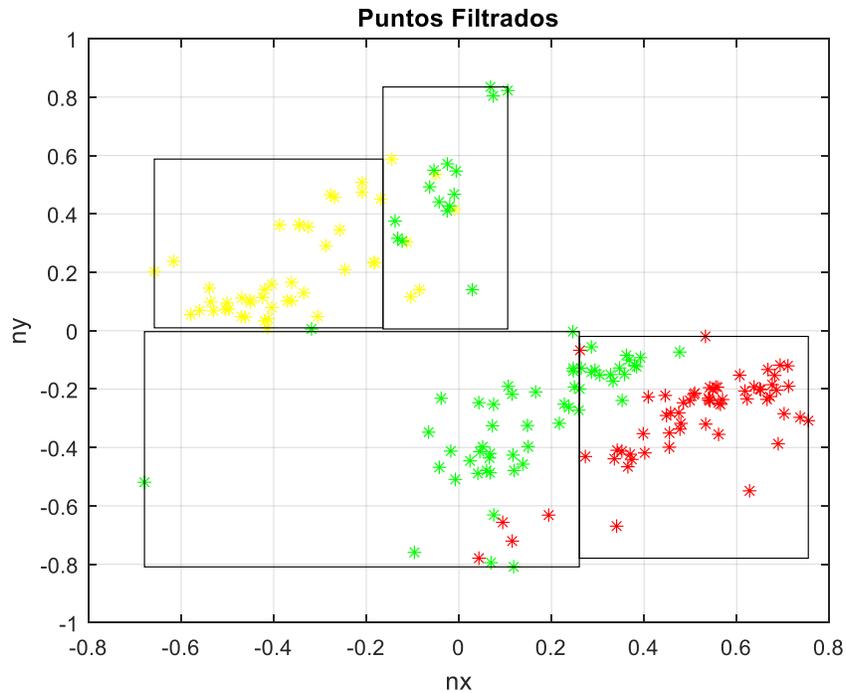


Figura 46: Se observan las normales filtradas por posición y con regiones delimitadas por medio de la técnica con rectángulos. La categoría correspondiente al espejo retrovisor derecho es eliminada de la medición debido a las pocas normales detectadas.

4.1.4 Dataset II: Resultados

Empleando las normales que no se usaron para la calibración, se calcula la sensibilidad del algoritmo tanto para la técnica de regiones rectangulares como para el modelo SVM. En el caso de la posición 1, espejo retrovisor derecho, se obtuvo el siguiente resultado observado en la *Figura 47*.

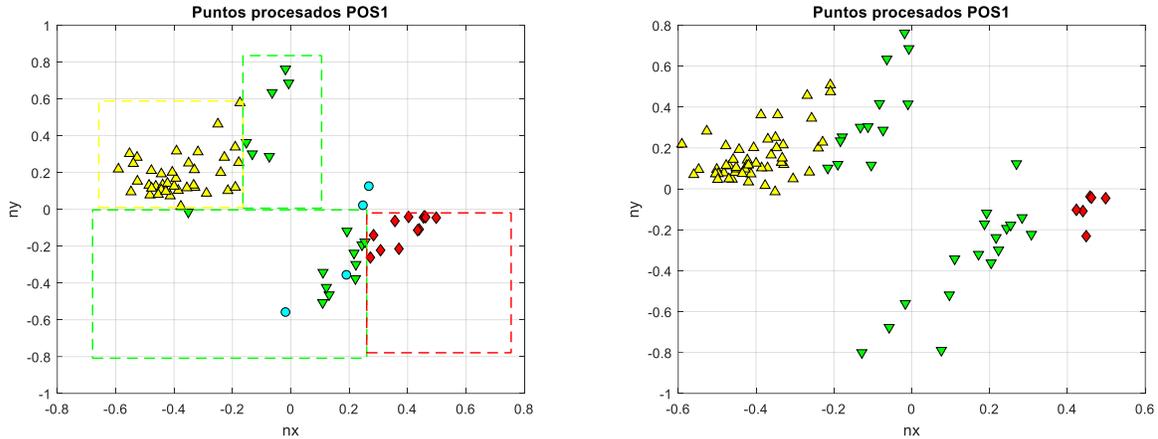


Figura 47: A la izquierda se observan los puntos procesados de la posición 1 mediante la técnica con rectángulos. En línea punteada se observan las regiones de las distintas posiciones. A la derecha, los puntos procesados con la técnica de SVM.

Luego, las sensibilidades asociadas se encuentran dadas por:

$$S2rec_{POS1} = \frac{VP}{VP + FN} \times 100\% = \frac{39}{39 + 33} \times 100\% = 54,16 \%$$

Donde $S2rec_{POS1}$ es la sensibilidad del Video II, de la posición 1 con la técnica de regiones definidas por rectángulos.

$$S2svm_{POS1} = \frac{VP}{VP + FN} \times 100\% = \frac{58}{58 + 28} \times 100\% = 67,44 \%$$

Donde $S2svm_{POS1}$ es la sensibilidad del Video II, de la posición 1 con la técnica del modelo SVM.

En la Figura 48 se observa el resultado de aplicar el algoritmo con la dirección de la mirada hacia la posición 2 utilizando la delimitación por rectángulos a la izquierda de la figura, y la técnica de SVM a la derecha.

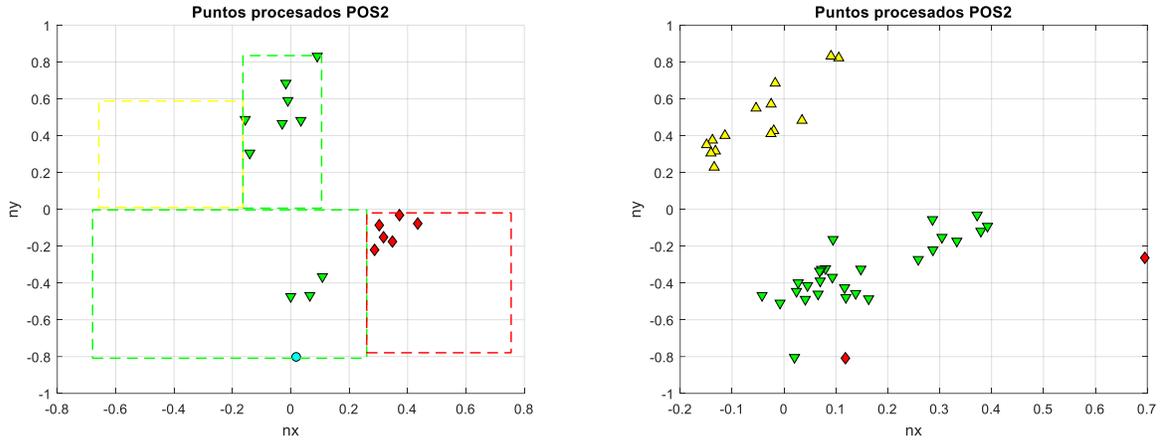


Figura 48: Resultados de aplicar el algoritmo con normales de la posición 2. A la derecha se observa el procesamiento utilizando la técnica con rectángulos. A la izquierda, la implementación con la estrategia de SVM.

Las sensibilidades obtenidas con ambas estrategias resultan:

$$S2rec_{POS2} = \frac{VP}{VP + FN} \times 100\% = \frac{10}{10 + 7} \times 100\% = 58,82\%$$

$$S2svm_{POS2} = \frac{VP}{VP + FN} \times 100\% = \frac{27}{27 + 16} \times 100\% = 62,79\%$$

Donde $S2rec_{POS2}$ es la sensibilidad asociada al Video II de la posición 2 con el método de la definición de regiones mediante rectángulos, y $S2svm_{POS2}$ es la sensibilidad asociada a la técnica de SVM.

Para la categoría 3, correspondiente a cuando se está mirando al espejo retrovisor frontal, se obtuvieron los resultados observados en la Figura 49.

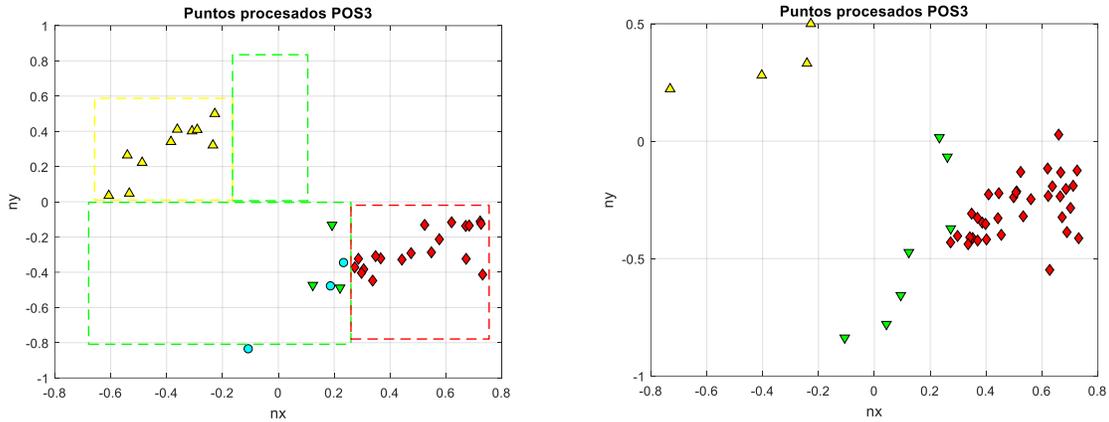


Figura 49: Procesamiento de las normales correspondientes a la posición 3 (espejo retrovisor frontal). A la izquierda se observa el resultado con la técnica de regiones rectangulares, y a la derecha con la estrategia de SVM.

Las sensibilidades logradas con esta técnica se obtienen mediante las siguientes ecuaciones:

$$S2rec_{POS3} = \frac{VP}{VP + FN} \times 100\% = \frac{19}{19 + 16} \times 100\% = 54,28\%$$

$$S2svm_{POS3} = \frac{VP}{VP + FN} \times 100\% = \frac{35}{35 + 11} \times 100\% = 76,08 \%$$

En donde $S2rec_{POS3}$ es la sensibilidad del video II de la posición 3 utilizando la técnica de rectángulos, y $S2svm_{POS3}$ es la sensibilidad empleando el método de clasificación de SVM.

Con la finalidad de calcular la precisión del algoritmo de la detección de la mirada, se realiza un gráfico con todas las normales de validación clasificadas mediante ambas técnicas, lo cual se observa en la *Figura 50*.

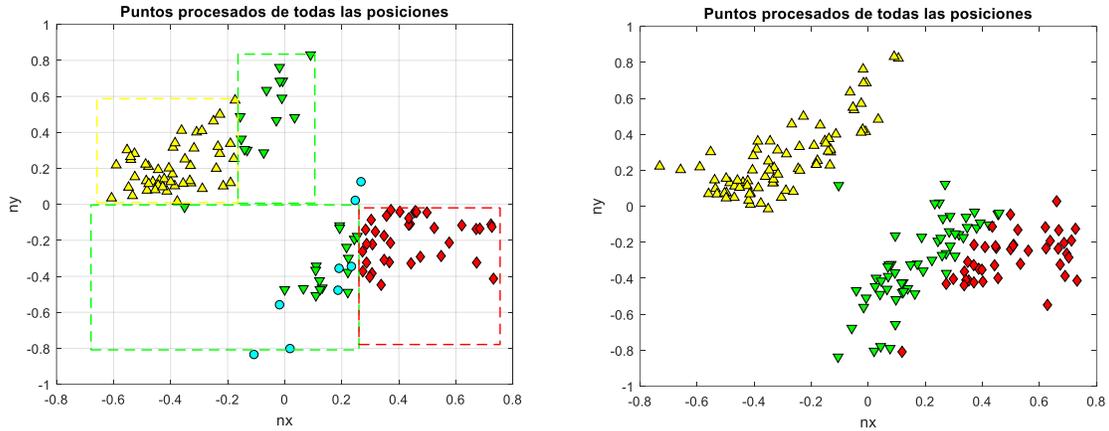


Figura 50: Todas las normales de validación procesadas. Izquierda: mediante la técnica de clasificación de regiones rectangulares, derecha: utilizando el método de SVM.

Las precisiones asociadas a cada posición son las siguientes:

$$P2rec_{POS1} = \frac{VP}{VP + FP} \times 100\% = \frac{39}{39 + 10} \times 100\% = 79,59 \%$$

$$P2rec_{POS2} = \frac{VP}{VP + FP} \times 100\% = \frac{10}{10 + 20} \times 100\% = 33,33 \%$$

$$P2rec_{POS3} = \frac{VP}{VP + FP} \times 100\% = \frac{19}{19 + 18} \times 100\% = 51,33 \%$$

Donde $P2rec_{POS1}$, $P2rec_{POS2}$ y $P2rec_{POS3}$ son las precisiones asociadas a cada categoría obtenidas del procesamiento mediante regiones rectangulares.

Luego, con la técnica de SVM se obtuvo:

$$P2svm_{POS1} = \frac{VP}{VP + FP} \times 100\% = \frac{58}{58 + 18} \times 100\% = 76,31 \%$$

$$P2svm_{POS2} = \frac{VP}{VP + FP} \times 100\% = \frac{27}{27 + 30} \times 100\% = 47,36 \%$$

$$P2svm_{POS3} = \frac{VP}{VP + FP} \times 100\% = \frac{35}{35 + 7} \times 100\% = 83,33 \%$$

Donde $P2rec_{POS1}$, $P2rec_{POS2}$ y $P2rec_{POS3}$ son las precisiones asociadas a cada categoría y al algoritmo de clasificación con SVM.

4.1.5 Dataset III: Condiciones

Debido a que las condiciones del Video II en cuanto a posición de la cámara y calidad de imagen no eran las más adecuadas para un funcionamiento eficiente del algoritmo, se decidió realizar otro dataset de imágenes. El video III posee movimiento de cabeza, oclusiones y movimiento del iris del ojo, pero una posición de la cámara a la derecha del volante, tal como se observa en la *Figura 51*, con la finalidad de en todas las poses ambos ojos se encuentren dentro del rango visible [34].

De este dataset se obtuvieron 660 imágenes, de las cuales, descartando las transiciones entre posiciones y situaciones de ojos cerrados, se obtuvieron 100 normales en promedio por categoría procesadas por el algoritmo [35].

En la *Figura 52* se observa, a la izquierda, la ubicación del iris del ojo izquierdo para cada posición, mientras que a la derecha la proyección de las normales procesadas en un plano 2D.



Figura 51: Frame extraído del Video III, en donde se observa una posición de la cámara hacia la derecha del volante.

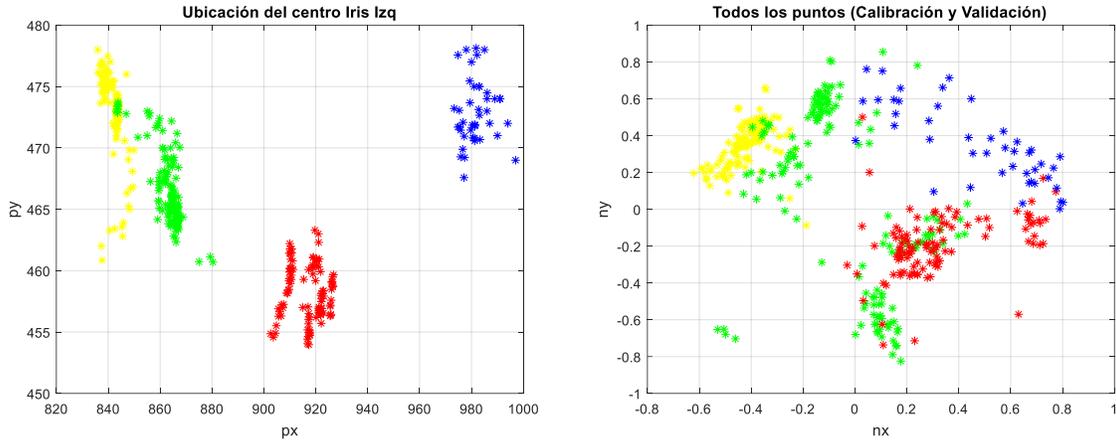


Figura 52: Izquierda: ubicación del centro del iris izquierdo cuando el usuario se encuentra fijando la mirada en cada una de las categorías. Derecha: Todas las normales procesadas por el algoritmo. Las posiciones se encuentran diferenciadas por color.

Al igual que con los dataset de imágenes de los videos I y II, se realiza un muestreo aleatorio, y por cada categoría se extrae un 60 % en promedio de la cantidad total, las cuales se utilizarán para la calibración, es decir la definición de las regiones. Luego, por cada posición se calculan los promedios y se descartan aquellas normales con signo distinto al del promedio hallado. El resultado de este muestro y de la eliminación de las normales con signo erróneo puede observarse en la *Figura 53*.

En este caso, se opta por construir un modelo SVM para la separación de los grupos de puntos.

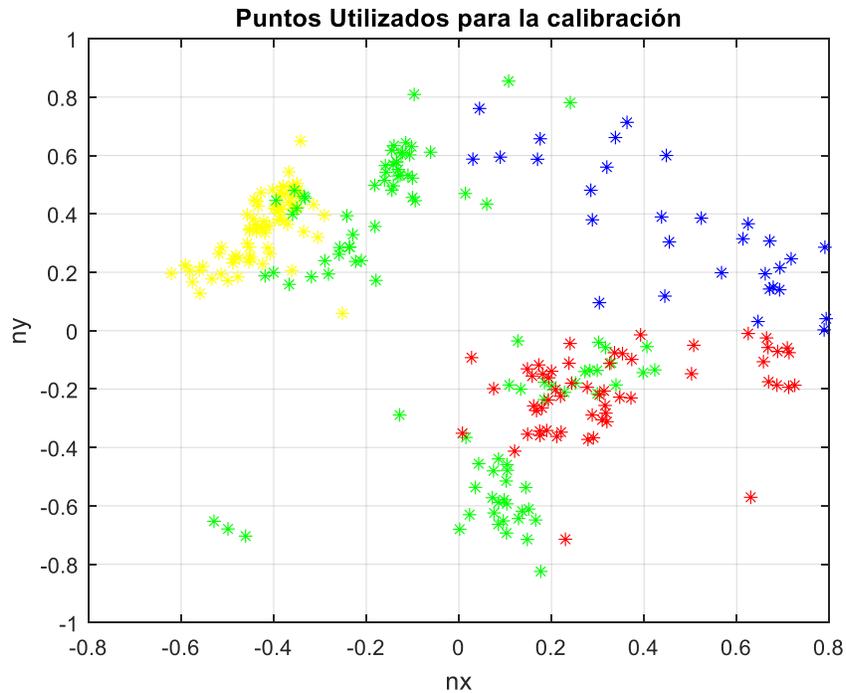


Figura 53: Normales utilizadas para la calibración, es decir, para la elaboración del modelo SVM.

4.1.6 Dataset III: Resultados

Empleando las normales que no se usaron para la calibración, se calcula la sensibilidad del algoritmo, teniendo en cuenta que para la clasificación se utiliza la técnica de SVM. En el caso de la posición 1, espejo retrovisor derecho, y la posición 2, se obtuvo el siguiente resultado observado en la Figura 54:

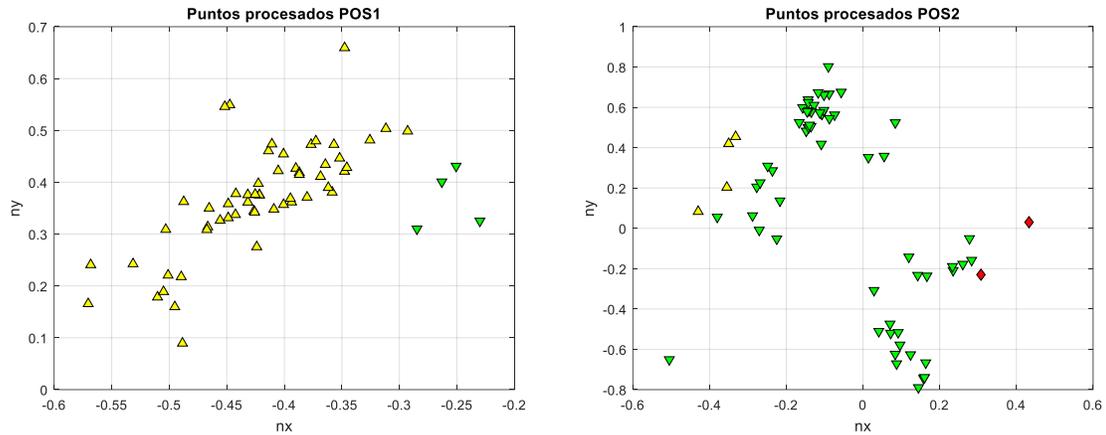


Figura 54: Resultado del procesamiento con las normales de la posición 1 (izquierda) y la posición 2 (derecha). Los triángulos amarillos se encuentran asociados a la posición 1, mientras que los triángulos verdes a la posición 2.

Se calcula la sensibilidad asociada a la posición 1 como se muestra a continuación:

$$S3_{POS1} = \frac{VP}{VP + FN} \times 100\% = \frac{56}{56 + 4} \times 100\% = 93,33\%$$

Donde $S3_{POS1}$ es la sensibilidad asociada al video III, posición 1.

La sensibilidad asociada a la posición 2 puede calcularse por medio de la expresión:

$$S3_{POS2} = \frac{VP}{VP + FN} \times 100\% = \frac{59}{59 + 6} \times 100\% = 90,76\%$$

Donde $S3_{POS2}$ es la sensibilidad del video III asociada a la posición 2.

En la Figura 49 a continuación se observa el resultado del procesamiento de la posición 3 y de la posición 4:

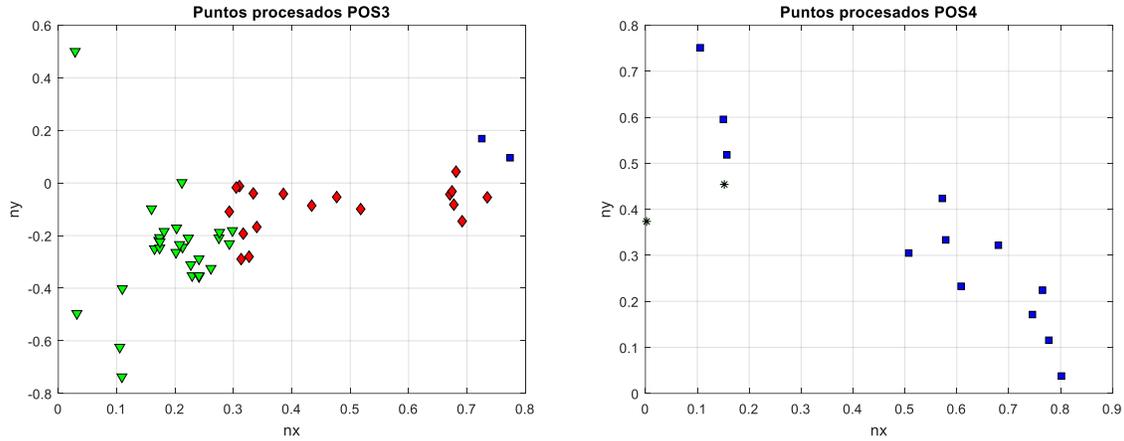


Figura 55: Normales procesadas correspondientes a la postura 3 (izquierda) y la postura 4 (derecha). Los rombos rojos representan las normales de la categoría 3, mientras que los cuadrados azules representan la categoría 4.

Luego, la sensibilidad de esta posición se calcula por medio de la siguiente ecuación:

$$S3_{POS3} = \frac{VP}{VP + FN} \times 100\% = \frac{18}{18 + 30} \times 100\% = 37,50\%$$

$$S3_{POS4} = \frac{VP}{VP + FN} \times 100\% = \frac{12}{12 + 2} \times 100\% = 85,71\%$$

Donde $S3_{POS3}$ es la sensibilidad del Video III asociada a la categoría 3, y $S3_{POS4}$ la sensibilidad de la postura 4.

En la Figura 56 se observan las normales de todas las posiciones utilizadas para validar el algoritmo.

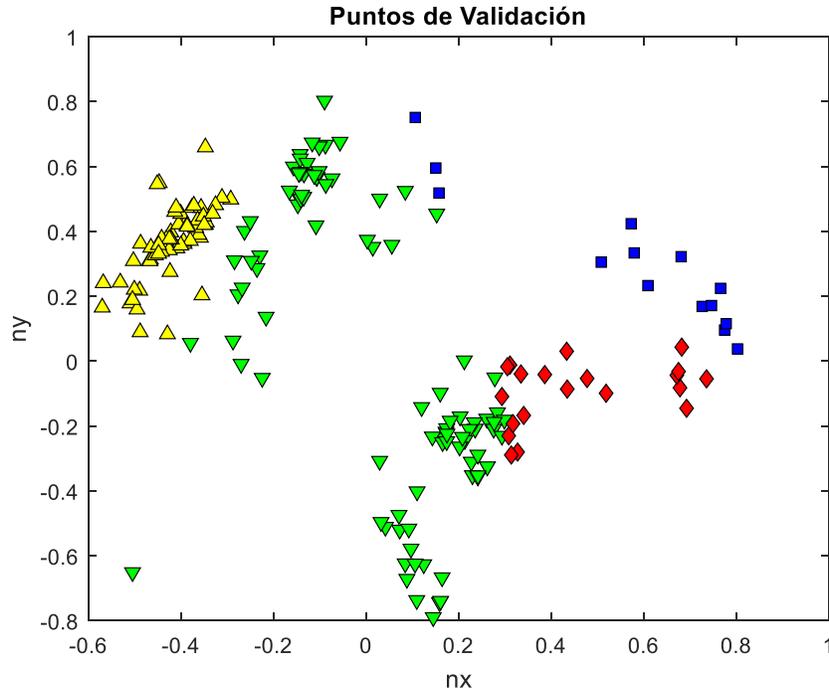


Figura 56: Se observan todas las normales utilizadas para la validación del algoritmo de detección de la mirada.

Las precisiones asociadas a cada categoría se calculan mediante las ecuaciones:

$$P3_{POS1} = \frac{VP}{VP + FP} \times 100\% = \frac{56}{56 + 4} \times 100\% = 93,33 \%$$

$$P3_{POS2} = \frac{VP}{VP + FP} \times 100\% = \frac{59}{59 + 34} \times 100\% = 63,44 \%$$

$$P3_{POS3} = \frac{VP}{VP + FP} \times 100\% = \frac{18}{18 + 2} \times 100\% = 90 \%$$

$$P3_{POS4} = \frac{VP}{VP + FP} \times 100\% = \frac{12}{12 + 2} \times 100\% = 85,71 \%$$

Donde $P3_{POS1}$, $P3_{POS2}$, $P3_{POS3}$ y $P3_{POS4}$ son las precisiones de cada posición del Video III.

4.2 Discusiones

Debido a que cada video refleja condiciones distintas en cuanto a posición de la cámara, calidad de imagen, tiempo en cada pose y rotación de cabeza y de iris, es preciso desarrollar una discusión para cada dataset tal como se realiza en las siguientes secciones.

4.2.1 Discusión Dataset I

El dataset obtenido del Video I posee pocos pestañeos por parte del usuario, el movimiento de cabeza y movimiento del iris en cada pose es mínimo, y esto se ve reflejado en los resultados del procesamiento del algoritmo de detección de la mirada. Tal como se observa en la *Figura 40*, los grupos de puntos que representan a las normales en distintas direcciones de la mirada se encuentran claramente separados y la definición de las regiones mediante rectángulos funciona de manera adecuada. En la *Tabla 3* se pueden observar los resultados de los cálculos de la precisión y de la sensibilidad por categoría.

VIDEO 1	Precisión	Sensibilidad
Posición 1	100 %	96,29%
Posición 2	96,15 %	89,28 %
Posición 3	100%	92,59 %
Posición 4	100%	63,63 %

Tabla 3: Resultados obtenidos del procesamiento del Dataset I.

Las sensibilidades halladas se encuentran en un promedio del 90%, excepto en la posición 4 (espejo retrovisor derecho) en donde la sensibilidad hallada es del 63,33 %, tal como se observa en la *Tabla 3*. La baja sensibilidad con respecto a las otras poses se debe a la rotación de cabeza que realiza el usuario para lograr mirar en esa dirección, lo que provoca que solo se vea una parte reducida del iris como se observa en la *Figura 57*. Debido a que parte del ojo no es visible en la imagen, la detección de esquinas tampoco funciona de forma correcta, lo que ocasiona muchos errores a la hora de realizar la desambiguación de las normales.



Figura 57: Frame extraído del Video I. Se observa que la posición de la cabeza y la de los ojos hace que parte del iris no se vea.

4.2.2 Discusión Dataset II

Tal como se explica en la sección 4.1.3, el video II posee una baja en la calidad de la imagen y movimientos más naturales. Otra característica particular de este dataset de imágenes es la posición de la cámara, más hacia el frente del usuario. Esto ocasiona que se detecten muy pocas normales cuando la dirección de la mirada es hacia el espejo retrovisor derecho, lo cual puede observarse en la *Figura 45*.

A causa de que los grupos de puntos esta vez no se encuentran espaciados, la delimitación por medio de rectángulos ya no es eficaz, como puede observarse en la *Figura 46*. Los grupos de puntos de las distintas posiciones se entremezclan debido a la mala calidad de la imagen y a la posición de la cámara. Es por esto que se opta por probar en paralelo con el algoritmo de clasificación de SVM.

Si bien se encuentran muchas detecciones erróneas, como puede observarse en la *Figura 44* o la *Figura 48*, la precisión y la sensibilidad mejoran un 10 % en promedio en la mayoría de los casos utilizando el modelo SVM, tal como se puede apreciar en la *Tabla 4*. Se detectaron muy

pocas normales asociadas a la posición 4 debido a una excesiva rotación de cabeza, tal como se observa en la *Figura 58*.

VIDEO 2	Precisión rec	Sensibilidad rec	Precisión SVM	Sensibilidad SVM
Posición 1	79,59 %	54,16 %	76,31 %	67,44 %
Posición 2	33,33 %	58,82%	47,36 %	62,79 %
Posición 3	51,33 %	54,28%	83,33 %	76,08 %

Tabla 4: Resultados del desempeño del algoritmo utilizando el dataset de imágenes del Video II. Se observan las precisiones y sensibilidades asociadas a la técnica con rectángulos y el método de SVM.



Figura 58: Debido a una rotación de cabeza mayor a la permitida por el algoritmo, en la posición 4 no se pudo detectar la dirección de la mirada.

4.2.3 Discusión Dataset III

En este dataset de imágenes se vuelve a reubicar la cámara hacia la derecha del volante, y la calidad de imagen mejora con respecto al Video II. Debido a que se observó una mejora en los

resultados utilizando el algoritmo de clasificación de SVM, se opta por utilizar este método. En la *Tabla 5* se observan las precisiones y sensibilidades asociadas a este dataset:

VIDEO 3	Precisión	Sensibilidad
Posición 1	93,33 %	93,33%
Posición 2	63,44 %	90,76%
Posición 3	90 %	37,50%
Posición 4	85,71 %	85,71%

Tabla 5: Resultados del desempeño del procesamiento realizado con el Video III.

A pesar de que los puntos asociados a las posiciones se encuentran entremezclados, como se observa en la Figura 53, los resultados de las precisiones y las sensibilidades calculadas superan el 80 % en la mayoría de los casos. Únicamente en la posición 3 se observa una sensibilidad de 37,50 %. Este resultado se debe a que las posiciones 2 y 3 se encuentran demasiado cerca, es decir, que la pose del ojo no es muy diferente de una posición a otra, con lo cual los puntos de ambas categorías se mezclan ocasionando una baja sensibilidad.

4.3 Conclusiones

Se consiguió en este trabajo, implementar el prototipo de un sistema que permite detectar hacia donde está mirando el conductor en un automóvil, en cuatro categorías definidas.

Los dataset de imágenes utilizados fueron variados en cuanto a calidad de imagen, controlabilidad de entorno y posición de la cámara. Puede afirmarse en base a los resultados observados en las secciones 4.1.2, 4.1.4 y 4.1.6 que la causa de mayor error en el algoritmo está asociado a una rotación excesiva de la cabeza, al punto que parte del ojo quede fuera del rango de la cámara. Este es un problema a tener en cuenta dado que la rotación de la cabeza es un movimiento frecuente cuando se conduce un automóvil. La baja calidad de imagen también interfiere en la precisión y sensibilidad del algoritmo, tal como pudo comprobarse en la sección 4.1.4.

Como futura mejora se propone mejorar los tiempos computacionales del algoritmo con el objetivo de que el mismo pueda trabajar en tiempo real. Como parte de la mejora, se podría realizar una búsqueda inicial del rostro y los ojos, que se volvería a ejecutar solo si ocurre una pérdida de

los ojos por más de dos frames. Además de lo anterior, también se podría desarrollar una aplicación de Android para Smartphone con el algoritmo desarrollado en tiempo real.

Para poder utilizar este procesamiento con un usuario que se encuentre manejando, se necesitaría un aporte con una segunda cámara, o bien una cámara móvil que siga el movimiento de la cabeza con el objetivo de que la mirada no se salga fuera del rango visible de la misma. En este caso, sería necesario un modelado o una detección de la postura de la cabeza.

Bibliografía

- [1] H. Abraham, B. Reimer and B. Seppelt, "Consumer Interest in Automation: Preliminary," 2017.
- [2] Luchemos por la Vida, "http://www.luchemos.org.ar/es/estadisticas/muertosanuales/muertos-argentina-durante-2018," [Online].
- [3] R. Ferreruela, "La Vision y el Ojo," *Instituto ILO Oftalmología*, 2007.
- [4] K.-N. Kim and R. S. Ramakrishna, "Vision-Based Eye-Gaze Tracking for Human Computer Interface," *IEEE SMC'99 Conference Proceedings. 1999 IEEE International Conference on Systems, Man, and Cybernetics*, pp. 324-329, 2011.
- [5] A. S. Hassanein, S. Mohammad and M. Sameer, "A Survey on Hough Transform, Theory, Techniques and Applications," 2015.
- [6] A. Yuille, P. Hallinan and D. Cohen, "Feature extraction from faces using deformable templates," *International journal of computer vision*, pp. 99-111, 1992.
- [7] D. W. Hansen and Q. Ji, "In the Eye of the Beholder: A Survey of Models for Eyes and Gaze," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2009.
- [8] A. Ray, N. Mahapatra, S. S. Das and A. Mishra, "Iris Recognition Using Gabor Filter and SURF Feature Detection Technique," *IUP Journal of Information Technology*, 2018.
- [9] G. G. Mateos, "Refining Face Tracking with Integral Projections," *International Conference on Audio-and Video-Based Biometric Person Authentication*, pp. 360-368, 2003.
- [10] K. Peng, L. Chen, S. Ruan and G. Kukharev, "A Robust Algorithm for Eye Detection on Gray Intensity Face without Spectacles," *Journal of Computer Science & Technology*, 2005.
- [11] D. Sidibe, P. Montesinos and S. Janaqi, "A simple and efficient eye detection method in color," *International Conference Image and Vision Computing New Zealand 2006*, 2006.

- [12] P. Viola and M. Jones, "Robust Real-Time Face Detection," *International Journal of Computer Vision*, 2001.
- [13] C. B. Heredia, "Implementación del Algoritmo de Detección de Caras de Viola y Jones sobre una FPGA," *Tesis de Licenciatura*, 2014.
- [14] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, 1997.
- [15] C. Nitschke, A. Nakazawa and H. Takemura, "Corneal Imaging Revisited: An Overview of Corneal Reflection Analysis and Applications," *IPSJ Trans. Computer Vision and Applications*, 2013.
- [16] R. Kothari and J. Mitchell, "Detection of eye locations in unconstrained visual images," *Proceedings of 3rd IEEE International Conference on Image Processing*, 1996.
- [17] J.-G. Wang, E. Sung and R. Venkateswarlu, "Eye Gaze Estimation from a Single Image of One Eye," *IEEE International Conference on Computer Vision*, 2003.
- [18] E. d. I. F. López and F. M. Trespaderne, "Detección Eficiente de Elipses en Imágenes. Aplicación al Posicionamiento 3D de un Robot Industrial.," *Revista Iberoamericana de Automática e Informática Industrial RIAI*, 2012.
- [19] A. Fitzgibbon, M. Pilu and R. Fisher, "Direct least square fitting of ellipses," *IEEE Transactions on pattern analysis and machine intelligence*, pp. 476-480, 1999.
- [20] P. Rosin, "Ellipse fitting by accumulating five-point fits," *Pattern Recognition Letters*, pp. 661-669, 1993.
- [21] K. Kanatani and W. Liu, "3D Interpretation of Conics and Orthogonality," *CVGIP: Image understanding*, 1993.
- [22] H. Wu, Q. Chen and T. Wada, "Conic-Based Algorithm for Visual Line Estimation from One Image," *Sixth IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 260-265, 2004.
- [23] Wikipedia,
 "https://es.wikipedia.org/wiki/Sensibilidad_y_especificidad_(estadística),"
 2019. [Online].

- [24] Wikipedia, "https://es.wikipedia.org/wiki/Algoritmo_de_Bresenham," [Online].
- [25] J. Canny, "A Computational Approach to Edge Detection," *IEEE Transactions on pattern analysis and machine intelligence*, pp. 679-698, 1986.
- [26] G. Santos and H. Proença, "A Robust Eye-Corner Detection Method for Real-World Data," *2011 International Joint Conference on Biometrics (IJCB)* , pp. 1-7, 2011.
- [27] Wikipedia, "https://es.wikipedia.org/wiki/Modelo_de_color_HSV," [Online].
- [28] J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio and V. Vapnik, "Feature selection for SVMs," *Advances in neural information processing systems*, pp. 668-674, 2001.
- [29] Y. G. García, "Algoritmos SVM para problemas sobre big data," *Tesis de Maestría*, 2013.
- [30] M. Camacho. Video I sin Procesar., "<https://youtu.be/l7UIE6jx0RA>," [Online].
- [31] M. Camacho. Video I Procesado, "<https://youtu.be/GCkH3jEWI-Q>," [Online].
- [32] M. Camacho. Video II sin Procesar, "<https://youtu.be/AiUHfnRpC04>," [Online].
- [33] M. Camacho. Video II Procesado, "<https://youtu.be/WUggWKcj9dA>," [Online].
- [34] M. Camacho. Video III sin Procesar, "<https://youtu.be/TWmHxZ69RJA>," [Online].
- [35] M. Camacho. Video III Procesado, "<https://youtu.be/MjnzF5mY3Eo>," [Online].