UNIVERSIDAD SEÑOR DE SIPÁN

FACULTAD DE INGENIERIA, ARQUITECTURA Y URBANISMO

ESCUELA ACADÉMICO PROFESIONAL DE INGENIERIA DE SISTEMAS

TESIS

PATRONES DE CONDUCTA FACIAL, PARA IDENTIFICAR ACCESOS INFORMÁTICOS NO AUTORIZADOS

PARA OPTAR TÍTULO PROFESIONAL DE INGENIERO DE SISTEMAS

Autor
Fray Luis Becerra Suárez
Asesor Metodológico
Mg. Victor Alexci Tuesta Monteza
Asesor Especialista
Ing. Heber Ivan Mejia Cabrera

Línea de Investigación
Ciencias de la Computación

Pimentel – Perú
2019

TESIS

PATRONES DE CONDUCTA FACIAL, PARA IDENTIFICAR ACCESOS INFORMÁTICOS NO AUTORIZADOS

Aprobación de la Tesis

MG. TUESTA MONTEZA VICTOR ALEXCI

Presidente del jurado de tesis

ING. MEJIA CABRERA HEBER IVAN

Secretario del jurado de tesis

Mg. BRUNO SARMIENTO JOSE MANUEL

Vocal del jurado de tesis

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

DEDICATORIA

No sería posible en empezar esta dedicatoria, sin dar gracias al creador de todas las cosas, el que me ha dado y sigue dándome sabiduría y fortaleza para continuar cuando a punto de caer eh estado; por ello, dedico primeramente mi trabajo a Dios.

De igual manera, dedico el presente trabajo a mis padres Aureliano Becerra Mendoza, Glenda C. Suarez Bardales y en especial a mis queridos abuelos Telmo Hernández Barrantes, Teonila Mendoza Espinoza, que, por su amor, trabajo y sacrificio en todos estos años, siempre eh contado con su apoyo incondicional tanto en la parte moral y económica para seguir adelante. Un anhelo que tanto han querido para mí, hoy reflejado en el presente trabajo.

Así como también, a mi esposa Jhoana E. Quiroz Hernández, y al más consentido de la casa, mi hijo Cristhian Fabian Becerra Quiroz, hermanos y demás familiares por brindarme su apoyo día a día en el transcurso de cada año de mi carrera universitaria.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

AGRADECIMIENTO

Deseo expresar mi más sincero agradecimiento al Ing. Victor Tuesta Monteza y al Ing. Miguel Ángel Vidaurre Flores, quienes además de transmitir su vocación investigadora, me orientaron y ayudaron constantemente en el desarrollo de los diferentes aspectos del presente trabajo. Y un agradecimiento muy especial al Ing. Ivan Mejia Cabrera por la plena confianza que siempre ha demostrado hacia mi persona.

A la plataforma virtual de aprendizaje en línea *www.coursera.org*, por incentivar la autoeducación de las personas ofreciendo diferentes cursos en las diferentes áreas de investigación, en especial en el área de visión por computador. No se necesita tener abundancia de recursos para seguir superándote y cumplir con tus metas.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

INDICE

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

INDICE DE FIGURAS

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

RESUMEN

El rostro humano, por naturaleza es una estructura muy compleja que varía en función del tiempo y de la raza étnica. Existen numerosas técnicas y métodos para detectar y reconocer rostros dentro de un sistema de reconocimiento facial (Ver Estado del arte).

El sistema de reconocimiento facial como cualquier otro sistema presenta diferentes ataques de seguridad. El más común es la presentación de una fotografía. Esto sucede porque en trabajos de investigación (Patrascu, 2016), (Juhong, Pintavirooj, 2017) los algoritmos de reconocimiento facial utilizan una imagen estática para identificar el rostro humano. La presentación de video de alta calidad es otro ataque muy común, así como también cuando se habla de personas gemelas, sumado a los factores ambientales que pueden afectar la correcta extracción de las características faciales (Sole, 2014).

Por este motivo, en el presente trabajo se propone un sistema que permita identificar un rostro real, con la finalidad de mejorar la seguridad en el reconocimiento facial en tiempo real basado en la identificación de patrones de conducta facial. El punto de partida empieza con la detección del rostro, para ello se evalúa dos técnicas de detección de objetos, filtros de haar y el histograma de degradados de orientación con un clasificador líneal SVM. Posteriormente, se procede con la detección de 68 puntos de referencia facial empleando el método de (Kazemi & Sullivan, 2014).

Estos puntos característicos, se evalúan utilizando vectores en el espacio y distancias para determinar diferentes patrones conductuales del rostro: parpadeo, sonrisa, boca abierta o cerrada. Dichos patrones conductuales, serán evaluados, con la finalidad de poder identificar un rostro real y poder brindar un nivel de seguridad extra en un sistema de reconocimiento facial.

Palabras clave: Visión Artificial, Procesamiento de Imágenes, Reconocimiento de patrones, reconocimiento facial

ABSTRACT

The human face, by nature, is a very complex structure that varies according to time and ethnic race. Today there are many techniques and methods to detect and recognize faces within a facial recognition system (See State of the art).

The facial recognition system, like any other system, presents different security attacks. The most common is the presentation of a photograph. This happens because in research works (Patrascu, 2016), (Juhong, Pintavirooj, 2017) the facial recognition algorithms use a static image to identify the user's facial face. The presentation of high-quality video is another very common attack, as well as when talking about twin people, added to the environmental factors that can affect the correct extraction of facial features, (Sole, 2014).

For this reason, in the present work, we propose a system that allows to identify a real face, with the purpose of improving the security in facial recognition in real time based on the identification of facial behavior patterns. The starting point begins with face detection, for which two object detection techniques, haar filters and the gradient histogram of orientation are evaluated with a linear SVM classifier. Subsequently, we proceed with the detection of 68 facial reference points using the method of (Kazemi & Sullivan, 2014).

These characteristic points are evaluated using vectors in space and distances to determine different behavioral patterns of the face: blink, smile, mouth open or closed. These behavioral patterns will be evaluated in order to identify a real face and provide an extra level of security in a facial recognition system.

Keywords: Artificial vision, image processing, pattern recognition, facial recognition

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

INTRODUCCIÓN

En la actualidad, las personas están conectadas al internet más que nunca. Según GloblaWebIndex, (2018) un usuario normal ocupa seis horas diarias conectado a internet desde algún dispositivo móvil o servicio electrónico, siendo las redes sociales lo que más se utiliza a diario. En estos servicios o recursos de internet, el usuario con o sin su consentimiento registra todo tipo de información. Información que se ha convertido en una fuente muy importante, donde personas, que, con alto grado de conocimientos de técnicas hacking e ingeniería social se valen para tomar tu información y utilizarlos con diferentes fines. Un claro ejemplo es donde afirma que la comisión federal de comercio de Estados Unidos abrió una investigación a Facebook ante las sospechas de brindar información de millones de usuarios a una empresa vinculada a la campaña del 2016 del presidente de EE.UU (El comercio, 2018).

Estos servicios o recursos, requieren de algún tipo de validación. El factor de autenticación más común es la validación de usuario y contraseña. A este factor se ha sumado un segundo factor basado en un código de seguridad o dato biométrico. Aun así, proteger la información con estos factores de autenticación no es suficiente, por lo que han surgido nuevas formas de autenticación basado en las características inherentes de la persona, siendo el tema de reconocimiento facial abordado en la presente investigación.

En los últimos años, el reconocimiento fácil ha sido una de las áreas con mayores investigaciones en comparación con las demás tecnologías biométricas. Según (Calderón, Ortega, 2016), afirma que las investigaciones en el año 2016 son de 608 y 1522 publicaciones en la base datos WoS y Scopus respectivamente.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Sin embargo, el reconocimiento facial como cualquier otro sistema de autenticación presenta vulnerabilidades que hasta la fecha aún no se ha podido obtener un 100% de efectividad. Según (Kazemi, Sullivan, 2014) esto se debe por diferentes factores como el exceso o ausencia de luz, presencia de oclusiones, ángulo de rotación del rostro entre otros. En (Sole, 2014), describe dos problemas que presenta el reconocimiento facial, que es la presentación de fotografías y la grabación de video del rostro facial en alta definición. Estos problemas de suplantación se producen porque los algoritmos de reconocimiento facial, por ejemplo, análisis de componentes principales y análisis de discriminante lineal, identifican al usuario a partir de una imagen de entrada y un clasificador entrenado.

Detectar el rasgo global de la cara y los rasgos locales (ojos, nariz, boca), presenta un gran esfuerzo cuando se trabaja en ambientes no controlados (Carzola 2016), así, como la extracción de puntos de referencia facial. En este caso, si el detector de rasgos globales y locales falla, también fallará la extracción de puntos de referencia facial.

Por ello, el presente trabajo estudia el problema, de determinar cuál es el mejor algoritmo para la detección de los rasgos globales y locales y la extracción de patrones de conducta facial en tiempo real, teniendo como principal objetivo la identificación de patrones de conducta facial usando técnicas de visión por computador.

En el estado de arte, se describen diferentes técnicas que abordan problemas semejantes al que se describe en esta investigación. En nuestro caso, se implementa el método de histograma de degradados orientados para la

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

detección de rasgos globales y locales del rostro y para determinar los puntos de referencia facial se implementa el método propuesto por (Kazemi,& Sullivan, 2014)

Para la detección de los rasgos globales y locales, se hace una comparación de dos métodos basado en filtros de Haar y el histograma de degradados orientados + un clasificador lineal SVM. El primer resultado que se obtiene es que el filtro de Haar al detectar rostros en ambientes no controlados presenta una precisión de 0.92 y el segundo método de Hog +svm presenta una precisión de 0.99

Cuando se evalúa el patrón de boca abierta y cerrada se obtiene una tasa de detección de 0.92 cuando el valor de la relación de aspecto de la boca es mayor que 0.1 y cuando el valor de la relación de aspecto de la boca es mayor que 0.3, se obtiene una tasa de detección de 0.34.

Para el patrón de sonrisa, al evaluar la triangulación CHN en un conjunto de 50 imágenes donde está presente únicamente el estado de sonrisa se obtiene un error de 0.02, pero cuando se evalúa en otro grupo de 50 imágenes con diferentes estados de ánimo presenta un error de 0.4

Sin embargo, se concluye que, con la implementación de tres patrones faciales de conducta, es suficiente para discernir entre un rostro real y una fotografía.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Capítulo I: Plan de Investigación

1.1 Situación Problemática

¿Cuál es la frecuencia de uso de cada una de nuestras cuentas en internet? y ¿cada cuánto tiempo, actualizamos nuestras contraseñas? , es natural que por alguna u otra razón te hayas registrado en algún servicio de internet, ya sea para descargar archivos, ver series o acceder a diferentes recursos; registrando todo tipo de información, desde ingresar cual es el color favorito, que es lo que te gusta; hasta ingresar tus datos personales como dirección, números telefónicos, cuentas bancarias, entre otros.

Con la aparición de las nuevas tecnologías, aumento de los ordenadores y dispositivos móviles, han hecho que, en la actualidad, las personas dependan cada vez más de estos, y presten menos atención a la información que registran y publican a través de estos sistemas, siendo las redes sociales lo que más utilizan las personas. Una noticia publicada en el diario El Comercio (2018), afirma que a la red social más utilizada (Facebook), se le ha abierto un proceso de investigación ante las sospechas de brindar datos de 50 millones de usuarios a una empresa vinculada a la campaña presidencial del 2016 de EE.UU. Esto es un claro ejemplo, que la información que registramos en internet es utilizada para diferentes fines lucrativos.

Toda esta información almacenada en la nube, se ha convertido en una valiosa fuente de información donde personas, que, con alto grado de conocimientos de técnicas hacking e ingeniería social se valen para tomar tus datos y poder utilizarlos a su antojo. Según (Eset, 2017), en su reporte anual, señala que los controles de seguridad (Figura 01), más implementados en Latinoamérica son el

antivirus (83%) el firewall (75%) y el backup de información (67%), dejando a las soluciones de doble factor de autenticación con un 11%. Esto es muy alarmante, pues las soluciones de doble factor de autenticación es el que está más vinculado con las aplicaciones que los usuarios finales acceden todos los días, y por desconocimiento de estos no son implementados.

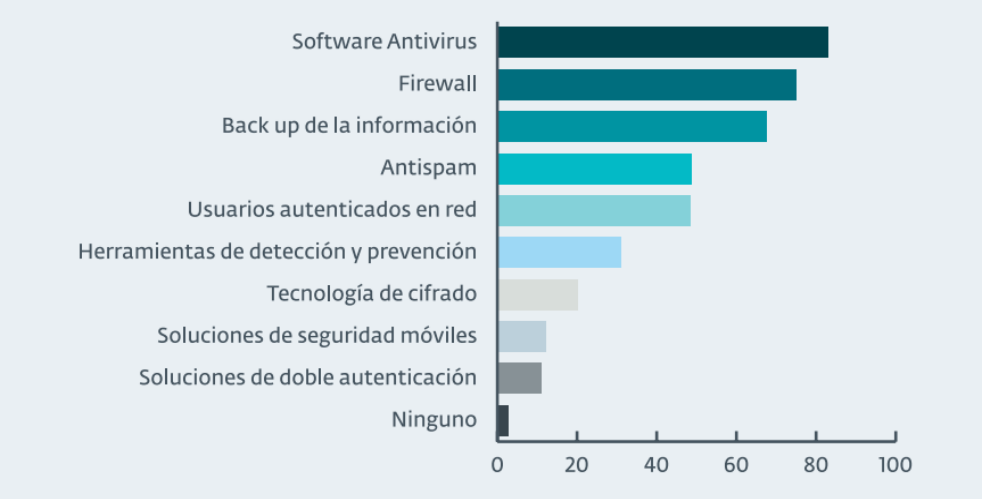


Figura 01:Controles de seguridad más implementados en Latinoamérica durante el 2016
Fuente: (Eset, 2017)

La mayoría de las personas utilizan servicios que requieren de algún tipo de validación de datos; uno de los factores más comunes de autentificación es la validación del nombre de usuario y contraseña para acceder a su información. Este factor de autenticación, es la más común dentro de los sistemas informáticos (Eset Security Report, 2017). Sin embargo, este tipo de validaciones se ha visto afectado por el incremento de ataques informáticos, sumado a las malas prácticas conductuales de las personas, como el uso de una contraseña única para varios servicios, basadas en contraseñas idénticas para uso personal y laboral, longitud de caracteres demasiado pequeñas, entre otras alternativas;

facilitando que personas no autorizadas, tengan control de la información de los usuarios.

Por esta razón, han surgido nuevas formas de autenticación, por ejemplo, el factor doble autenticación basado en un código de seguridad que puede ser un código de seguridad o un dato biométrico. Esta forma de autenticación está asociada a un dispositivo del usuario que puede ser un celular o token. La mayoría de las aplicaciones cotidianas como Facebook, Twitter, Linkedin, Google y otros poseen un doble factor de autenticación, que en la mayoría de los usuarios es desconocido y no es implementado.

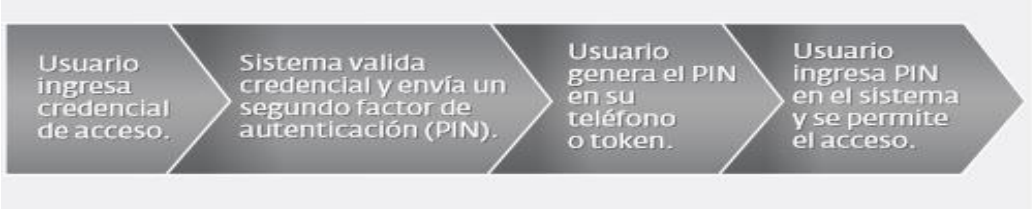


Figura 02: Etapas del factor de doble autentificación en sistemas de validación
Fuente: *(Eset, 2017)*

Otra forma de autentificación en el cual está basado la presente investigación, se basa en las características inherentes de la persona, que no requiere que el usuario ingrese algún dato para autenticarse, en conclusión, ser uno mismo su clave personal. Las características inherentes de la persona más utilizadas en estos sistemas es el rostro, iris, huella digital y mano. Dichos sistemas biométricos según el rasgo biométrico presentan diferentes ataques de seguridad. En el trabajo de Solé (2014), afirma que los sistemas basados en la detección facial, los ataques más habituales es la presentación de fotografías, ya sean originales o con pequeñas modificaciones o fotografías impresas sobre soportes bidimensionales. También afirma que otro ataque muy común es la

grabación de video de alta calidad. Además, este tipo de sistemas se ve comprometido cuando se habla de personas que son gemelos.

Vulnerar la seguridad de un sistema biométrico facial se convierte en una tare sencilla, teniendo en cuenta que las personas hoy en día, están conectados a internet más que nunca, especialmente en las redes sociales, donde suben fotografías en alta calidad, diferentes ángulos (posición de la cabeza y rostro), diferentes circunstancias y diferentes factores ambientales. Por ello Sole (2014), recomienda implementar la vida de muestra en dichos sistemas con el fin de evitar la presentación de biometría falsa.

En la literatura del reconocimiento facial, se puede encontrar diferentes técnicas que abordan estos problemas. Según Guevara (2017), agrupa estas técnicas en métodos basados en el conocimiento, en características invariantes, en plantillas y en apariencia. Dentro de estos métodos, existen algoritmos inspirados en el modelo biológico (modelo visual), por ejemplo, la wavelet de Morlet ofrece una aproximación a este modelo, sin embargo, su limitación, tal como describe Guevara (2017), es limitado y tendrá mal desempeño en imágenes con grandes dimensiones y en localizar rostros grandes. A comparación con las redes neuronales convoluciones, tal como describe Sudhakar, et all (2016), tiene un mejor rendimiento en comparación con el método anterior.

Ante los diferentes problemas descritos anteriormente, en el presente trabajo de investigación se centra en implementar nuevas técnicas de detección facial y análisis de movimiento especialmente en la localización de los puntos de referencia facial para detectar la ubicación de los ojos, la boca, distancias, entre otras. Además, se analizará diferentes patrones de conducta facial y poder

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

implementarlo en tiempo real para evitar la suplantación de identidad en el momento de reconocer un rostro real y poder discernir entre una fotografía.

1.2 Formulación del problema

Es importante que, durante el proceso de reconocimiento facial, se tenga muy claro dos etapas de desarrollo. La primera etapa y la más importante consiste en la detección del rostro. Tal como afirma *Cama (2015)*, para que un sistema de reconocimiento facial sea robusto es necesario tener en cuenta diferentes factores como iluminación, cercanía o lejanía del rostro, ángulo del rostro, presencia de oclusiones como la barba o gafas, entre otros. Estos factores, como explica Kazemi, Sullivan (2014), pueden afectar la correcta extracción de las características faciales. La segunda etapa consiste en reconocer un rostro, teniendo en cuenta la primera etapa y los diferentes clasificadores a utilizar para evaluar el rendimiento del sistema biométrico.

El rostro humano por naturaleza es una estructura muy compleja que varía en función del tiempo y de la raza étnica. En el estado de arte, se han propuesto diferentes técnicas de detección facial que no siempre se han obtenido buenos resultados positivos. El método, más implementado para realizar dicho trabajo, se basa en los filtros de haar, propuesto por Viola, Jones (2001). Sin embargo, Cama (2015), concluye que este método no tolera rotaciones (+- 15°), por lo que es necesario implementar otros métodos para mejorar la tasa de detecciones.

Detectar el rasgo global de cara en ambientes controlados, tienen una eficiencia computacional muy alta. Freire (2016) explica que estos algoritmos PCA, LDA, ICA, LBP de reconocimiento facial implementados se ven fuertemente afectados cuando se trabaja en ambientes no controlados. Es por ello que Carzola (2016),

al trabajar en ambientes no controlados obtiene un 87.5% de detección de rostros y un 64.28% en la detección de ojos. Determinar, cual es el mejor algoritmo para detectar el rasgo global (cara) y los rasgos locales (ojos, boca, nariz), implica un arduo trabajo, pero es muy importante dedicarle el mayor esfuerzo si se quiere conseguir buenos resultados.

En Blázquez (2013), se implementa un sistema para la detección y corrección de puntos característicos faciales, utiliza un total 78 distancias, implementa el análisis de componentes principales y obtiene una varianza de error entre 21.41% y 30.09% en ambientes no controlados. Diferentes librerías de procesamiento digital como FaceSDK, extrae 66 puntos de referencia facial en tiempo real, los resultados son aceptables en un 99.9% cuando se trabaja con rostros frontales, en cambio, si el rostro sufre una pequeña variación de rotación, FaceSDK cae en un 0% de detección.

Es por ello, que en el presente trabajo de investigación se presenta un esfuerzo por desarrollar una herramienta de software que, haciendo uso de un adecuado procesamiento digital en tiempo real de imágenes faciales y clasificación de patrones conductuales basado en puntos de referencia del rostro, se pretende conseguir como resultado el reconocimiento de un rostro real, evitando la suplantación de identidad al presentar una fotografía dentro del sistema. Ante esta descripción, surge la pregunta ¿Cuál es el mejor algoritmo para la detección de rostros y la extracción de patrones de conducta facial en tiempo real?

1.3 Delimitación de la investigación

El desarrollo de la presente investigación se realizó en la Universidad Señor de Sipan, teniendo un tiempo de desarrollo de abril – julio del 2018.

1.4 Justificación e importancia

El presente trabajo de investigación es necesario y útil porque está dentro de una de las líneas de investigación de la Escuela de Ingeniería de Sistemas, en el marco de “Tecnología de la información y comunicación”, teniendo como desarrollo en el campo de inteligencia artificial de procesamiento de imágenes y visión artificial.

El desarrollo de la biometría, especialmente en el campo de reconocimiento facial es una de las áreas más impulsadas en investigaciones. Según (Calderón, Ortega, 2016), realizan un estudio bibliométrico de las investigaciones publicadas en las bases de datos de WoS y Scopus en el periodo de 1996-2016, y determinan que, en el año 2016, se publicaron 608 y 1522 trabajos de investigación en WoS y Scopus respectivamente. Con esto se demuestra que el tema que se aborda en la presente investigación, tiene una gran base sólida en investigaciones. Además, existe diferentes foros de comunicación y recursos en internet para realizar consultas referentes al tema. Con este trabajo de investigación se pretende contribuir con un intrascendente conocimiento para mejorar la seguridad en el reconocimiento facial.

En la actualidad, diferentes empresas tecnológicas (FacePhi, Smowl, SmartGate, etc.) comercializan software de reconocimiento facial. Adquirir los servicios de estas empresas requiere de mucho dinero. Hoy, gracias a los avances tecnológicos, el auge de los equipos y el mejor equipamiento de los mismos en cuanto a rendimiento, cámaras, sensores, etc. permitirán aprovechar estas características, a fin de no tener que hacer una mayor inversión de la que ya se ha hecho con los equipos con que cuentan las personas.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Poniendo en evidencia lo anterior, el presente trabajo de investigación se centra en desarrollar una herramienta de software que pueda reconocer patrones de conducta facial usando procesamiento de imágenes en tiempo real y clasificadores para la detección y validación de un rostro real.

1.5 Limitaciones de la investigación

La principal limitación es el trabajar en ambientes no controlados, por lo que no se espera conseguir buenos resultados en la detección de rostro y extracción de puntos de referencia facial, después de haber consultado y analizado los resultados obtenidos en el estado de arte.

El reconocimiento facial comprende dos etapas (detección y reconocimiento de rostros). El presente trabajo se desarrolla en la primera etapa que es detección.

Los patrones de conducta facial que puede expresar la persona son muchos. Por ello, se va implementar en este proyecto el patrón de parpadeo, detección de boca abierta y cerrada y detección de sonrisas.

1.6 Objetivos

Objetivo General

Identificar patrones de conducta facial usando procesamiento de imágenes en tiempo real para identificar un rostro real.

Objetivos específicos

a. Seleccionar algoritmos de detección de puntos de referencia facial
b. Extraer puntos característicos faciales del rostro humano.
c. Implementar algoritmos de patrones de conducta facial.
d. Desarrollar software de patrones de conducta facial
e. Evaluar los algoritmos de patrones de conducta facial.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Capitulo II: Marco teórico

2.1 Antecedentes de la investigación

En el trabajo de investigación **"Sistema de control del estado de somnolencia en conductores de vehículos"** desarrollado por **López (2016)** emplea técnicas de visión por computador para detectar la somnolencia en los conductores de vehículos. Para ello detecta el rostro facial y los ojos utilizando filtros de Haar. Con la detección de los ojos y la implementación de dos clasificadores (ojo abierto y ojo cerrado), permite discernir el parpadeo de los ojos, orientación y dirección del rostro. Realiza una prueba en el día con 8843 frames y consigue un 98.74% de acierto para la detección de rostro. Para la detección de los ojos cerrados obtiene un 98.28% de aciertos. En la noche con 6501, consigue un 98.75% para detección facial y para los ojos cerrados un 98.41%. Dicho trabajo se toma como referencia para determinar el grado de orientación y dirección del rostro.

En el trabajo de investigación ***"Extracción de características morfológicas de rostro humano a partir de imágenes",*** propuesto por **Quintero, (2016),** Tiene como objetivo general extraer las características morfológicas para describir los rasgos faciales del rostro humano. Empieza con la detección del rostro y después define sub-regiones de interés para los ojos, cejas, nariz y boca utilizando el algoritmo de viola-jones. Estas sub-regiones son pre-procesadas teniendo en cuenta el siguiente orden: ajuste de intensidad de color, filtro gaussiano para el suavizado de los pixeles, conversión a espacio de color, segmentación de los rasgos encontrados aplicando binarización y operaciones morfológicas

(dilatación y erosión). Después de este proceso empieza a detectar 35 puntos faciales teniendo en cuenta las diferentes medidas antropométricas. Para validar el método propuesto, emplea 96 fotografías de la base de imágenes a color FERET. Para los rasgos (ojos, nariz y boca) obtiene una precisión de 98.95%, el 1.05% corresponde a una imagen que tenía mucha iluminación. Para la detección de los puntos faciales se obtiene una precisión de 68% para los ojos, 76% para las cejas, 78% para la nariz, 84% para la boca. Se concluye que se debe tener en cuenta más puntos, medidas y proporciones faciales, además hacer una comparación con puntos faciales antropométricos marcados por un especialista en el área, para determinar con mayor precisión que tan acertado ha sido el método desarrollado.

Según **Cama, (2015).** En el trabajo de investigación **"Prototipo computacional para la detección y clasificación de expresiones faciales mediante la extracción de patrones binarios locales"**, tiene como objetivo el rostro humano y reconocer las expresiones de emociones globales. Implementa un modelo para realizar la detección de rostro en una imagen de entrada. Para ello, realiza un pre-procesado en escala de gris y ecualización de histograma, con la finalidad de representar la imagen de entrada en un solo canal de color (gris). Luego implementa método de Viola-Jones para detectar el rostro. La región del rostro detectado, se hace un recorte de 65% en horizontal y 85% en vertical para eliminar características irrelevantes que no brinda información alguna de la expresión facial. Finalmente se realizar un escalado en dimensión de 70 x 90 pixeles. Para extraer las características faciales implementa el método de patrones binarios locales uniformes y el clasificador adaboost para el

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

entrenamiento y clasificación de las expresiones faciales. Al realizar el test en 300 imágenes basado en LBP con radio = 3, obtiene un máximo de 93.14% en la expresión de asombro y un mínimo de 83.14% en tristeza. Por ultimo se concluye que, para la detección de rostros, el algoritmo no tolera rotaciones (15 grados como mínimo), por lo que es necesario implementar otros métodos para mejorar la tasa de detección de rostros.

2.2 Estado del arte

Según ***Alshamsi, Kepuska, Men (2017),*** en su investigación, ***"Automated Facial Expression Recognition App Development on smart phones using cloud computing",*** aborda la detección de emociones humanas, presentando un nuevo método de reconocimiento facial basado en el modelo de nube, en combinación con el sistema de expresión facial tradicional. Como punto de partida empieza con la detección de la cara y luego redimensionando a un único tamaño. A partir de ello, se extrae los puntos de referencia facial y el centro de gravedad (COG), generando los conjuntos de entrenamiento para predecir las emociones humanas. Como clasificador utiliza un SVM y la técnica de matriz de confusión. Cuando se realiza las pruebas del método, se logra obtener una tasa de predicción de 96.3% utilizando los puntos de referencia facial y el método de centro de gravedad.

En el trabajo de ***Juhong, Pintavirooj (2017),*** denominado ***"Face recognition bassen on Facial Landmark detection",*** presenta un método para detectar puntos de referencia facial de los ojos, nariz y boca. Emplea los filtros de haar para localizar las características locales del rostro determinando por cada detección un ROI para los ojos, nariz y boca. Cada ROI detectado es

transformado a una imagen binaria y se aplica el método de proyección horizontal y vertical. Con esto logra detectar 7 puntos faciales y utilizando el análisis vectorial forma los trillizos de área y la invariancia geométrica asociada para identificar a las personas. Cuando realiza el test en imágenes con seis personas a distancia de 100 cm de la cámara, teniendo un error promedio entre 0.38 y 1.31. Llegando a la conclusión que el método propuesto demuestra resultados prometedores en cuanto al reconocimiento facial.

En el trabajo de (***Soukupová, Cech, 2016)*** denominado "***Real-Time eye blink detection using facial landmarks".*** Se desarrolla un algoritmo en tiempo real para detectar el parpadeo de los ojos en secuencia de video desde una cámara estándar. Para la localización de los puntos de referencia utiliza dos detectores CHEHRA e INTRAFACE. Con estos puntos, establece la función relación de aspecto del ojo que mide la distancia inter ocular. Para la clasificación, implemente un SVM lineal, para determinar el estado de parpadeo de los ojos.

El autor trabajó en ambientes no controlados. Con este fin, enfoca todo el trabajo en la extracción de los puntos de referencia del rostro utilizando dos detectores según estado del arte, CHEHRA e INTRAFACE, dos dataset de imágenes, el ZJU contiene 80 videos con un promedio de 136 fotogramas por video, los fotogramas tienen una resolución de 320 x 240. La segunda base de datos contiene 8 videos con un promedio de 11k fotogramas y una resolución de 640 x 480 px. Sin embargo, los resultados para una mejor interpretación se pueden analizar en el siguiente enlace https://vision.fe.uni-lj.si/cvww2016/proceedings/papers/05.pdf. Dicho trabajo se toma como

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

referencia por el método EAR para la detección del parpadeo de los ojos. Sirviendo de base para poder distinguir rostros reales.

En (***Sudhakar, Saberian, Li 2015),*** en su trabajo de investigación denominado ***"Multi-view Face Detection Using Deep Convolutional Neural Networks",*** tiene como objetivo principal detectar rostros en múltiples vistas. Ante este problema propone un metodo llamado Deep dense Face Detector (DDFD) que es capaz de detectar caras en todas las orientaciones utilizando un único modelo basado en redes neuronales de convolución profunda. Este método no es tan complejo, pero los resultados obtenidos soy muy altos. Para validar los resultados utiliza un conjunto de datos de 21k imágenes. Se obtiene un 0.999% cuando se reconoce una cara frontal mientras que las caras con más rotación en el plano tienen menos puntaje. De hecho, los puntajes disminuyen a medida que aumenta la rotación en el plano. Por último, se concluye que, estos resultados se puede mejorar aún más el rendimiento si se mejora las estrategias de muestreo y técnicas de aumento de datos.

Para (**Hsu, Peng, Chang 2014)** en su trabajo de investigación denominado **"Landmark Based Facial Component Reconstruction for Recognition Across Pose"**. Se desarrolla un método que reconstruye componentes faciales 3D a partir de componentes 2D. Dicho trabajo se desarrolla en dos etapas: reconstrucción de componentes y reconocimiento basado en componentes. En la fase de reconstrucción, se extraen regiones de componentes, utilizando los métodos segmentación facial por puntos de referencia de pose invariante y la reconstrucción 3D de componentes faciales usando el modelo de referencia Etnia y género (E-T). En la fase de reconocimiento, la pose de una imagen dada

se determina por un conjunto de puntos de referencia que guía la rotación de los componentes reconstruidos de manera que la imagen reconstruida puede alinearse con los componentes de la imagen original. Para realizar las pruebas, se utilizó la base de datos PIE (68 sujetos) y la base de datos Multi-Pie (249 sujetos), teniendo como resultado que el componente de reconstrucción 3D (E + G) mantiene su rendimiento a 67.5◦ y un coste computacional de 36 segundos. Llegando a la conclusión que la reconstrucción de la cara se construye mejor en modelos de referencia de la misma etnia y de género

Según (**Kazemi, Sullivan, 2014)** en su trabajo de investigación ***"One Millisecond Face Alignment with an Ensemble of Regression Trees"***, aborda el problema de la alineación de rostros para una sola imagen y estimar con precisión la posición de los puntos faciales. Para ello, utiliza una cascada de regresiones, donde representa (x, y) las coordenadas de los puntos faciales dentro de la imagen, almacenados en un vector S (shape). El punto crítico en esta etapa es que el regresor hace sus predicciones basadas en características como la intensidad de los pixeles. Para entrenar cada regresor, utiliza el algoritmo de gradiente de árbol con una suma de perdida de error cuadrático. Con este método, emplea dos líneas base, el primero en selección de características aleatorias (EF) y el segundo en la selección de características basas en correlación (EF +CB). El tiempo de ejecución en una sola imagen es constante, pero el entrenamiento depende de la cantidad de imágenes a entrenar. El método EF genera un error de 0.069 y el método (EF+CB) tiene un error de 0.59, trabajado con la base de imágenes HELEN. En la base de imágenes LFPW, EF tiene un error de 0.51 y el método (EF+CB) tiene un error

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

de 0.41. Por último, concluye que, el método propuesto es más rápido para reducir el error en comparación con el método EF.

2.3 Base teórico científicas

Se presentan los conocimientos o bases teóricas que serán empleadas en el trabajo de investigación.

2.3.1 Procesamiento digital de Imágenes

Debido a que este proyecto se enmarca en el campo de visión por computador, es necesario tener en cuenta algunos para la representación de imágenes

Formación de Imagen

"Una imagen puede ser definida como una función de dos dimensiones *F(x,y),* donde X,Y son coordenadas espaciales" (Gonzales R, 2002 p.1). Estas imágenes son matrices de puntos denominados pixeles. El pixel representa la menor unidad homogénea dentro de la imagen. Una imagen en color está formada por tres canales (rojo, verde, azul), por tanto, cada pixel estará representando por tres valores numéricos que están en un rango de 0 a 255. Esto significa que 0 representa la ausencia de un color y cuanto más se acerque a 255, más intenso será el color.

Características de un Pixel

El color que adquiere un pixel depende esencialmente de tres componentes:

- ***Color de la Luz**:* La luz es una forma de energía que ilumina las cosas, las hace visibles y se propaga mediante partículas llamadas fotones. Se caracteriza por sus componentes que se puede representar como una onda

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

con una determinada longitud o frecuencia. El ojo humano solo puede percibir un subconjunto de luz, conocido como el espectro visible.

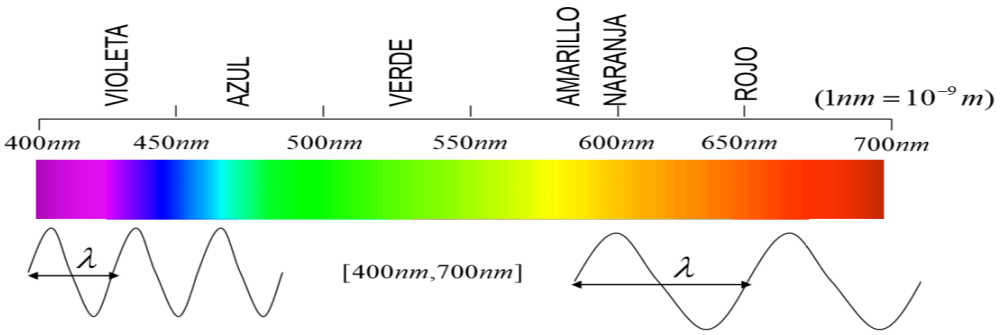


Figura 03: Espectro visible de la luz que puede percibir el ojo humano

Fuente: (Vanrell, sf)

- ***Material de la superficie**:* El tipo de material de la superficie determina las longitudes de onda que refleja y las que absorbe.
- ***Sensibilidad del dispositivo**:* Las cámaras tienen tres tipos de sensores que integran sobre diferentes longitudes de onda con el objetivo de cubrir todo el espectro visible.

Espacios de Color

"El color no existe y no es una característica de un objeto sino una apreciación subjetiva", (Isaac Newton, 1666). Según (Ford, Roberts, 1998), "un espacio de color es un método mediante el cual podemos especificar, crear, y visualizar el color".

Con los avances de la computación gráfica, transmisión de señal han surgido diferentes espacios de color. Podemos consultar los diferentes espacios de color en (Ford et al. 1998) y (Lucero, Saldaña, 2016). A continuación, se describe tres espacios de color más importantes y más utilizados en visión por computador para la detección facial.

Espacio de Color RGB

Tiene origen en aplicaciones de pantalla CRT. Según Vezhnevets, Sazonov, Andreeva (2003), afirma que es el espacio de color más utilizado en imágenes digitales. Sin embargo, RGB, presenta alta correlación entre canales, mescla de datos de cromancia y luminancia que hacen que no sea una buena opción para el análisis de color.

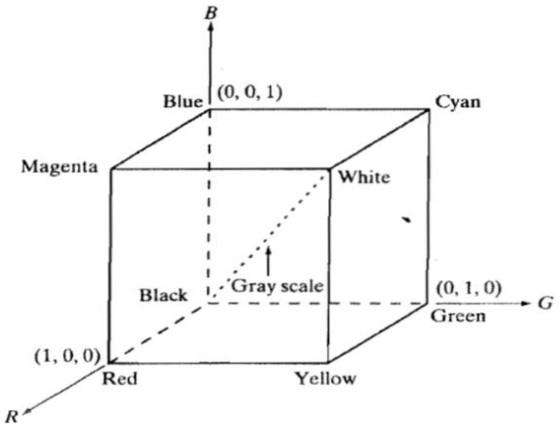


Figura 04: Espacio de color RGB representado en un sistema de coordenadas cartesianas

Fuente: (Gonzales Woods, 2008),

Espacio de Color YCbCr

El espacio de color YCbCr es una señal RGB no lineal codificado. Se construye como la suma ponderada de los valores RGB y dos valores de diferencia de color Cr y Cb que se forma restando la luminancia de los componentes rojo y azul RGB. El parámetro Y indica luminancia, Cb ubica el color entre el rojo y el verde y Cr indica la ubicación del color entre el rojo y el verde.

La simplicidad de la transformación y lo explícito de la separación de los componentes de luminancia (brillo) con la cromancia (color), hacen que este espacio sea atractivo para el modelamiento del color de la piel. (Vezhnevets,et al, 2003).

Espacio de Color HSV

Este modelo se obtiene de las siglas en inglés (Hue, Saturation, Value), que traducido significa tonalidad, saturación y valor. Según Huang (2018), este modelo es una transformación no lineal del espacio de color RGB. La ventaja de este modelo reside en la manera extremadamente intuitiva de especificar el color. (Ford et al. 1998).

Escala de Grises

"Una imagen en escala de grises está conformado únicamente por una capa, la cual muestra la intensidad de la luz que refleja cada objeto en tono de gris, que va a un rango de 0 a 255". (Callejas, 2016 p. 53).

Para convertir a escala de grises, podemos utilizar dos expresiones matemáticas:

$$(1)\ escala\_gris = Red * 0.3 + Verde * 0.59 + Azul * 0.11$$
$$(2)\ escala\_gris = (Rojo + Verde + Azul)/3$$

Histograma de Imagen

Es la representación gráfica de la distribución de una imagen, cuantificando el número de pixeles para cada valor de intensidad considerado. Los histogramas son la base para las diferentes técnicas de procesamiento digital de imágenes. Para calcular el histograma se usa el siguiente algoritmo:

- Imagen de entrada 'src' de 8 bits de origen.
- Calcular el histograma H para src.
- Normalizar el histograma de modo que la suma de los intervalos de histograma sea 255.
- Calcular la integral del histograma.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

$$H'_i = \sum_{0 \le j < i} H(j)$$

- Transformar la imagen usando H' como una tabla de búsqueda:

$$dst\ (x, y) = H'(src(x, y))$$

2.3.2 Reconocimiento de Patrones faciales

Descripción de un patrón

Un patrón es una entidad a la que se le puede dar un nombre y que está representada por un conjunto de propiedades con la finalidad de poder describir y clasificar (reconocimiento) de acuerdo a sus atributos. Según Vázquez (2014), los patrones los clasifica en patrones abstractos que representan una forma o modelo o ideas conceptuales, mientras que un patrón concreto representa la parte física del objeto. Ejemplos de patrones podemos mencionar a un vector de características de la cara de una persona, señales sonoras, peatones en una autopista, etc. Desde el punto de vista computacional, el reconocimiento de patrones es un tema muy extenso y a veces complicado si no apuntamos bien a donde queremos llegar

Patrones Faciales

Para las personas, el rostro es el principal rasgo discriminativo que permite discernir e identificar a simple vista a una persona. Posee una gran cantidad de características como son ojos, nariz, boca, etc. acompañado también de las expresiones faciales. Todas estas características han hecho que hoy en día, se desarrollen diferentes aplicaciones, ya sea para el control de acceso, sistemas de video vigilancia, celulares inteligentes, etc. por las características favorables

que presenta como baja intrusividad, gran poder discriminante, disponibilidad extendida (Eslava, 2013).

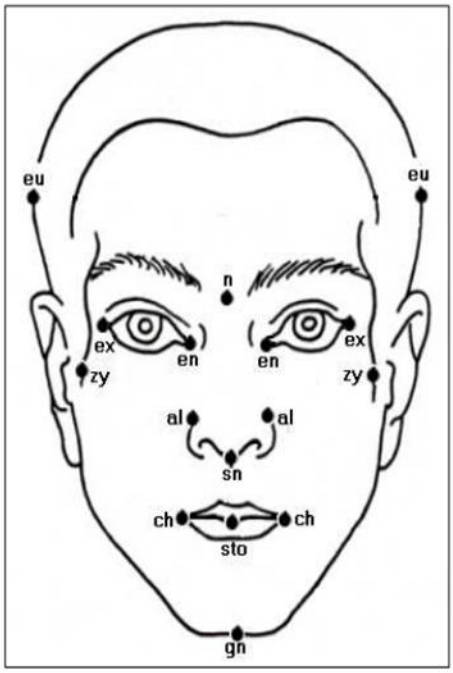
La emoción es el cimiento de la motivación humana y que la base de la emoción está en el rostro (Tomkins & McCarter, 1964). Según Matsumoto, D., Sung H, H., López, R., Pérez-Nieto M. (2008), afirman que la universalidad de la expresión facial son siete emociones: ira, desprecio, asco, miedo, alegría, tristeza, y sorpresa. Sin embargo, a estas emociones universales, se tiene que agregar otra expresión donde la persona refleja neutralidad o seriedad.

Puntos Característicos del Rostro

Antes de continuar con la descripción de este apartado, es necesario conocer los siguientes términos: cefalometria y antropometría. La *cefalometría* es el nombre que se le da a las medidas que se obtienen del cráneo humano y la *antropometría* es el estudio y las medidas del cuerpo humano.

En Prieto (2008), describe, que, el rostro desde el punto de vista frontal, posee puntos característicos distribuidos en las seis regiones del complejo facial (cabeza, cara, ojos, nariz, labios y boca, y orejas), un total de once puntos [Fig. 06]. Sim embargo, Gutierrez (2004), describe que, para el estudio de la simetría en las mismas regiones del complejo facial, utiliza la misma cantidad de medidas (once), pero los puntos ubicados en posiciones diferentes.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe



DESCRIPCION
Eurion → *eu*
Cigion → *zy*
Nasion → *n*
Gnation → *gn*
Subnasal → *sn*
Alar → *al*
Cheilion → *ch*
Estomion → *en*
Endocanto → *ex*
Labiale Superius → *ls*
Labiale Inferius → *li*

Figura 05: Puntos característicos faciales para una visa frontal

Fuente: (Prieto, 2008)

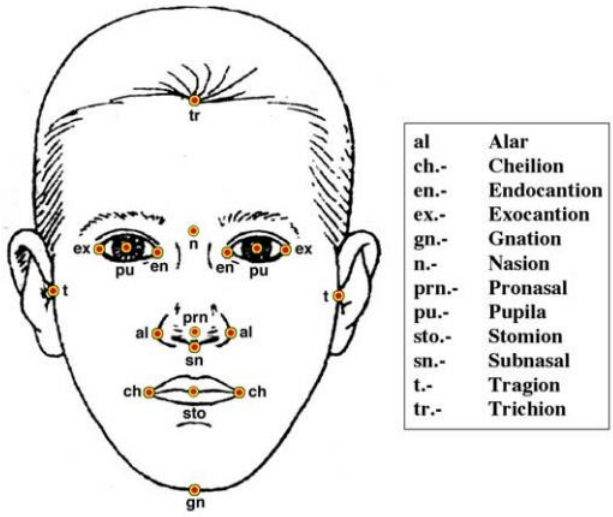


Figura 06: Definición de puntos faciales antropométricos del rostro humano

Fuente: (Gutierrez de Aguas, 2004)

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

2.3.2 Algoritmos y clasificadores

HOG (Histogram of Oriented Gradients)

Desarrollado por Dalal, Triggs (2005), es un descriptor muy popular para la detección de objetos. A continuación, se describe todo el proceso de cálculo descrito por Valveny, (2015).

Como punto de partida es necesario realizar el cálculo de gradiente, y para ello el gradiente se define como el cambio direccional en la intensidad de una imagen. Definido por dos valores: dirección y magnitud. Para encontrar el gradiente del pixel de color amarillo se procede de la siguiente manera.

0	**0**	**255**	**255**	**255**
0	**0**	**255**	**255**	**255**
0	**0**	**0**	**255**	**255**
0	**0**	**55**	**0**	**0**
0	**0**	**0**	**0**	**0**

$$dx = I(x+1, y) - I(x-1, y)$$
$$dx = 255 - 0 = 255$$
$$dy = I(x, y+1) - I(x, y-1)$$
$$dy = 255 - 55 = 200$$
$$Luego\ se\ calcula, direccion\ \sigma(x, y)\ y\ magnitud\ \ g(x, y)$$
$$\sigma(x, y) = arctang\frac{dy}{dx}$$
$$g(x, y) = \sqrt{dx^2 + dy^2}$$

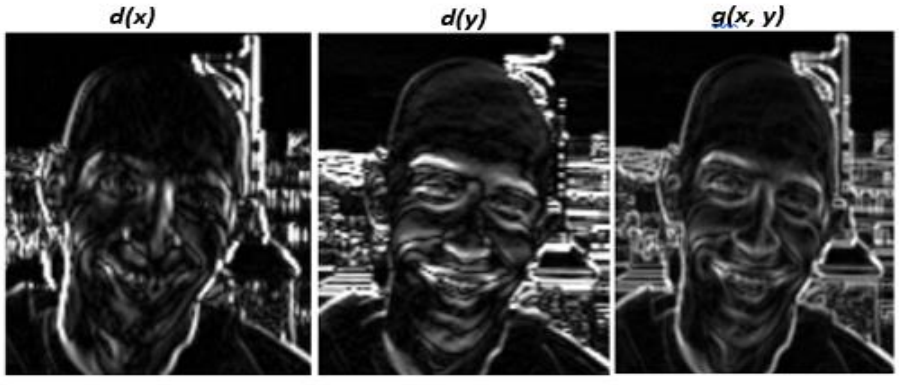


Figura 07: Resultada de imagen al aplicar el cálculo de gradiente

Fuente: Valveny, (2015).

El problema con esta representación a nivel de pixeles de gradiente es que difícilmente se puede utilizar junto con clasificador en un sistema de detección de objetos, además puede ser muy sensible a pequeñas variaciones tanto en la forma como en la localización del objeto.

Un clasificador, requiere la representación global de la imagen, para ello, el descriptor HOG, actúa en dos pasos diferenciales que son:

- División de la imagen en celdas de tamaño fijo, cuyos valores habituales suelen ser en 6 y 8 pixeles tanto en ancho como en alto.
- Calculo de un histograma de las orientaciones en cada celda.

Algoritmo Viola – Jones

El algoritmo de Viola Jones, (2001), está enfocado originalmente a la detección de caras, pero se ha demostrado que su utilidad es más amplia y se puede utilizar como base para detectar otro tipo de objetos. Este algoritmo utiliza como descriptor de la imagen las características de HAAR, imagen integral y como método de clasificación adaboost, temas que se describen a continuación.

Las características de HAAR se obtiene como resultado de aplicar por toda la imagen un conjunto de filtros básicos de HAAR [Fig. 09].

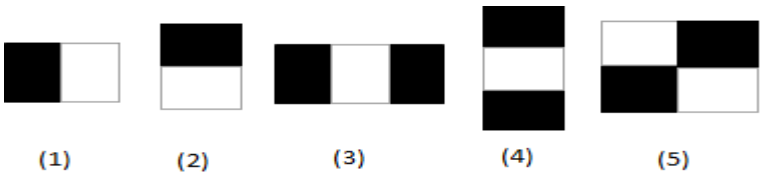


Figura 08: Filtros básicos de Haar

Fuente: *(Coursera, 2018)*

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Como se puede observar en la imagen anterior un filtro de HAAR es la combinación de varios rectángulos entre dos y cuatro adyacentes en forma horizontal como vertical. Los rectángulos pintados de color negro representan zonas con una contribución positiva al filtro, mientras que los rectángulos pintados de color blanco tendrán una contribución negativa para el filtro. Esto genera como resultado final del filtro la diferencia en la suma de los valores de intensidades de los pixeles entre zonas en negro y zonas en blanco. Estableciendo la formula quedaría de la siguiente manera:

$$Resultado\ del\ Filtro\ (RF) = \sum_{(x,y)\in N} I(x,y) - \sum_{(x,y)\in B} I(x,y)$$

Por ejemplo, si tenemos una imagen de 4x8 representados en su intensidad de pixeles, se pide calcular el filtro de haar que se muestra a continuación.

200	**200**	**100**	**100**	**200**	**200**	**100**	**100**
240	**120**	**50**	**70**	**100**	**100**	**80**	**90**
70	**200**	**100**	**100**	**200**	**150**	**30**	**100**
190	**200**	**50**	**50**	**100**	**200**	**100**	**50**

La solución quedaría de la siguiente manera:

$$Resultado\ del\ filtro = 650 - (300 + 280) = 70$$

Como ya explicamos anteriormente cada filtro se va aplicar a todas las posibles escalas en horizontal y vertical. Para ello, explicaremos este procedimiento utilizando como referencia la imagen del Ejemplo 01 sin valor de intensidad de cada pixel y el primer filtro básico de haar de la Imagen 01. Entonces, con los datos que se presentan en la dirección horizontal solo va a ser posible generar escalas en tamaños dos, cuatro, seis y ocho para este filtro. Hay que tener en

cuenta que el tamaño del rectángulo negro y blanco debe ser siempre el mismo tamaño

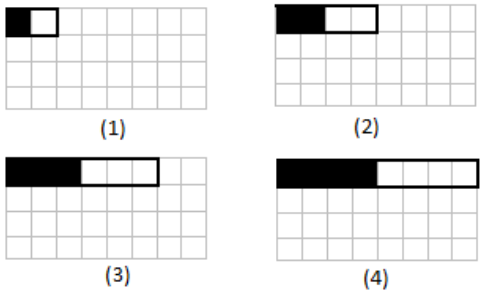


Figura 9: Filtro de Haar aplicado en forma horizontal. (1) Imagen con escala de tamaño dos. (2) Imagen con escala de tamaño cuatro. (3) Imagen con escala de tamaño seis. (4) Imagen con escala de tamaño ocho
Fuente: *(Coursera, 2018)*

Igualmente podemos escalar el filtro en la dirección vertical (Figura 03) y para cada nueva escala en vertical podemos volver a aplicar el mismo escalado que en horizontal. Al final obtendremos un total de 16 escalas que corresponden 4 en dirección horizontal y 4 en dirección vertical.

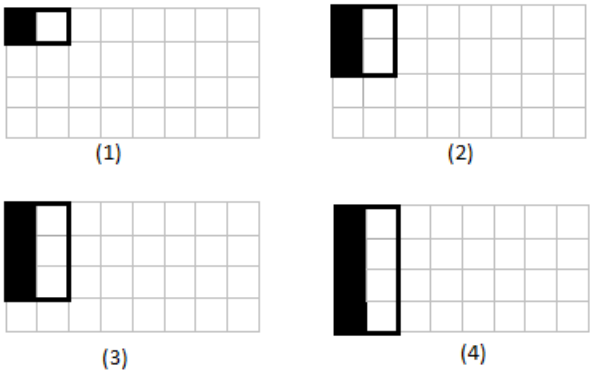


Figura 10:Filtro de Haar aplicado en forma vertical
Fuente: *(Coursera, 2018)*

Hay que tener en cuenta que para cada una de estas escalas se aplica a todas las posibles posiciones de la imagen. Para este ejemplo y el filtro de un tamaño de 2x4 se obtendrá un total de 15 posiciones que corresponde al desplazamiento

tanto en horizontal como en vertical [Fig. 11). De esta forma la descripción global de toda la imagen es la concatenación de todas estas características

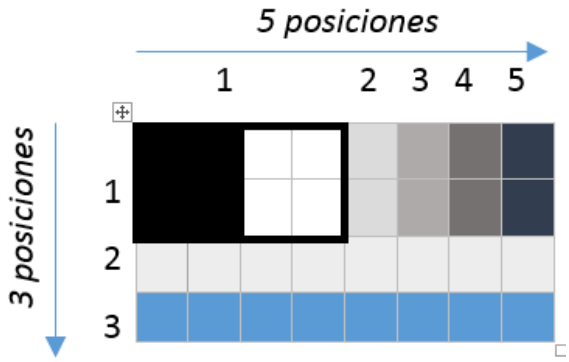
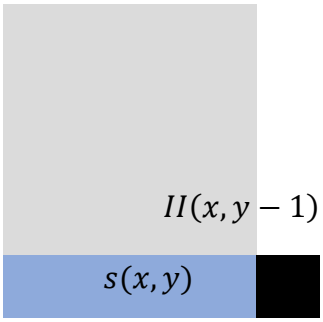


Figura 11: Posiciones que recorre el filtro de haar de 2x4 en una imagen de 4x8 pixeles.
Fuente: *(Coursera, 2018)*

Imagen Integral

Es simplemente una transformación de la imagen original que da como resultado una nueva imagen del mismo tamaño en el que el valor de cada pixel corresponde a la suma de todos los pixeles situados a la derecha y arriba en la imagen original permitiendo calcular rápidamente la suma de todos los pixeles de cualquier rectángulo en la imagen [Figura 13].



$$II(x,y) = II(x,y-1) + s(x,y)$$

$$s(x,y) = \sum_{(x,y)\in N} I(x',y) = s(x-1,y) + I(x,y)$$

Figura 12: *Calculo de la imagen integral*
Fuente: *(Coursera, 2018)*

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Cascada de clasificadores

Es una combinación secuencial de clasificadores donde una sola imagen es reconocida como cara si todos los clasificadores lo aceptan. El objetivo de cada nivel es alcanzar un índice de rendimiento (falsas detecciones vs detecciones correctas) determinando con el mínimo número de características posibles. Asi, el primer clasificador va a recibir todas las ventanas de una imagen. Todas aquellas que rechace el primer clasificador, quedaran descartadas. Las que este clasificador acepte se pasaran como entrada al segundo clasificador repitiendo este proceso hasta llegar al último clasificador.

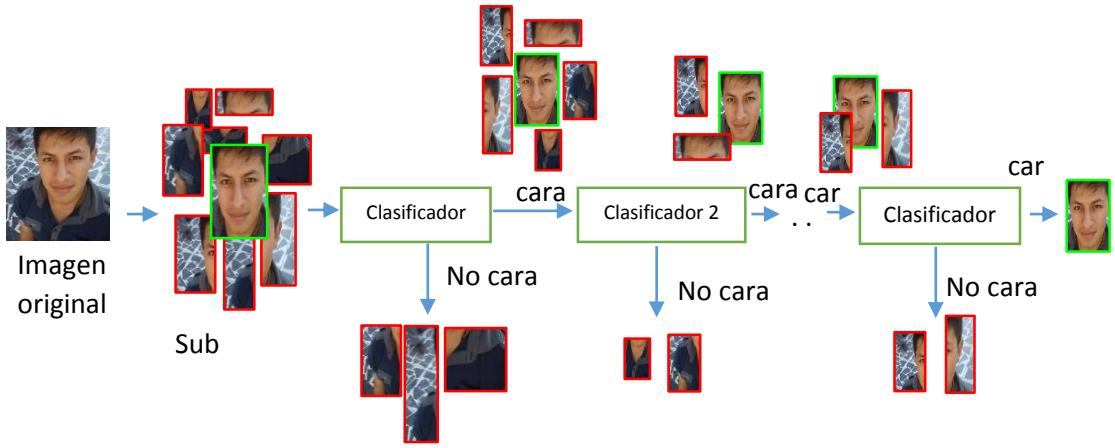


Figura 13: *Proceso iterativo de la cascada de clasificador*
Fuente: *Elaboración propia*

Local Binary Patterns (LBP)

Según describe Pietikäinen, Zhao, Hadid, Ahonen, (2006), LBP es un operador diseñado originalmente para la descripción de texturas, invariante ante cambios de iluminación y traslación. Sin embargo, otros autores afirman que “LBP es un operador que transforma una imagen en una matriz o una imagen de etiquetas de enteros que describen el aspecto de pequeña escala de la imagen” (Pietikäinen, et al. 2011 p.13).

LBP - Básico

Supongamos que tenemos una imagen con dimensiones de 7x7 [Fig.15], y que cada cuadrado representa un pixel. Entonces, si queremos hallar el valor del código LBP para el pixel central, ese cálculo se basará en primer lugar definir la vecindad del pixel y luego ir comparando la intensidad de gris de cada pixel con el pixel central y por último asignar el valor según la siguiente formula.

$$Valor\ bit\ b = \begin{cases} 1, & valor\ del\ vecino \geq valor\ central \\ 0, & en\ caso\ contrario \end{cases}$$

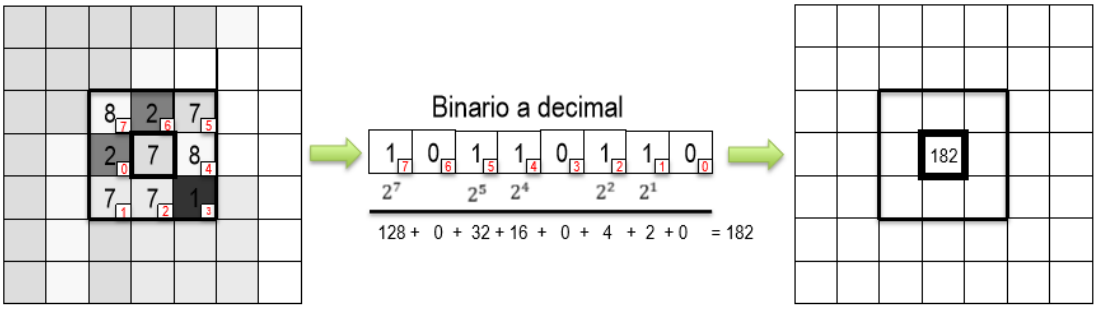


Figura 14: Calculo de LBP Básico

Fuente: *(Coursera, 2018)*

LBP - Uniforme

Según Pietikäinen et al. (2011) considera que un patrón binario local es uniforme si su medida de uniformidad (U: número de transiciones entre 0 y 1 en la representación binaria) tiene como valor 2. En la explicación de Lopez (2016), afirma que el LBP básico trabaja con valores en un rango de 0 a 255. Valores que al momento de trabajar con imágenes reales aparecen patrones que tienen muy poca probabilidad de darse. Por esta razón se decide reducir el número de patrones. Por ejemplo, los patrones 111111111 (0 transiciones), 01110000 (2 transiciones) se consideran como uniformes. Sin embargo, los patrones

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

11001011 (4 transiciones) y 01010011 (6 transiciones) no se consideran como uniformes.

Entonces a los patrones con U=0 ó U=2, se reasigna un código de patrón individual. A los demás patrones se les reasigna el mismo código (pasan a ser indistinguibles).

Support Vector Machines (SVM)

Según describe Valveny, Vilariño, (2015), define los siguientes conceptos básicos.

Clasificador lineal, por lo tanto, la solución siempre va hacer un hiperplano basado en margen máximo a partir de los vectores de soporte.

2.3.4 Sistema Biométrico

Evaluación de un sistema biométrico

“Evaluar es estimar aptitudes y rendimiento de un sistema” (DRAE). Por lo tanto, y como también lo indica K.R. Sharp, una evaluación completa de un sistema biométrico deberá prestar igual atención a todos los aspectos, desde la adquisición de datos hasta la integración del sistema.

Para poder evaluar la fiabilidad de un sistema biométrico, se debe tener en cuenta el concepto de Ground Truth. Término que es usado en varios campos para hacer referencia a la información proporcionada por la observación directa en contraposición a la información proporcionada por inferencia. En aprendizaje computacional se usa para referirse al resultado correcto que debería dar un clasificador.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

El método de evaluación de un sistema biométrico es indiferente. En este caso el método que utilizaremos y por las condiciones que se plante se ha optado por utilizar la matriz de Confusión.

Matriz de Confusión: Es la herramienta básica que permite visualizar el nivel de confusión de un clasificador.

Supongamos que estamos evaluando la clase Persona [Fig.16], el nombre de las instancias reales y resultado de clasificación es indiferente, según sea el ejemplo a evaluar. Para nuestro caso esta descrito como Persona y No Persona.

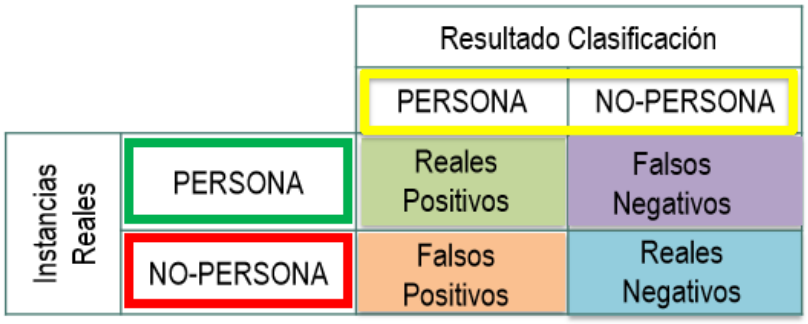


Figura 15: Matriz de Confusión y sus indicadores para evaluar la clase persona
Fuente: *(Coursera, 2018)*

Los cuadros con fondo de color verde, morado, anaranjado y celeste representan los indicadores. A continuación, se describe cada uno de ellos.

- **Reales Positivos:** Representan los candidatos que han sido correctamente clasificados como persona.
- **Falsos Positivos:** Representan a los candidatos que han sido incorrectamente clasificados como persona.
- **Falsos Negativos:** Representan a los candidatos personas que han sido incorrectamente clasificados por el clasificador.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

- **Reales Negativos:** Representan a los candidatos que han sido correctamente clasificados como no personas.

Supongamos que tenemos un conjunto de candidatos (persona, silla, auto, gato). Para nuestro ejemplo el resultado del clasificador lo representaremos dentro de un circulo y el resultado al aplicar la matriz de confusión se representará en la tabla 01.

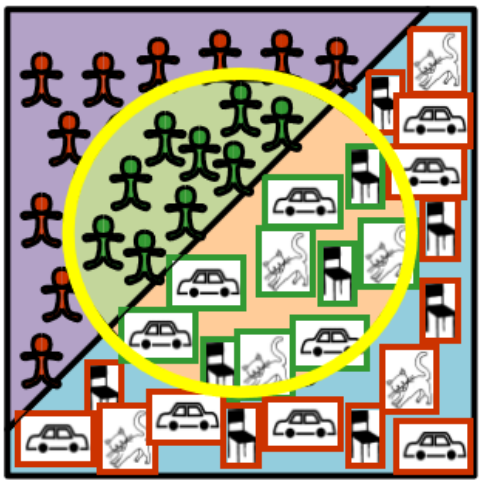


Figura 16: Conjunto de candidatos a evaluar.
Fuente: *(Coursera, 2018)*

Tabla 1: Resultados obtenidos después de aplicar la matriz de confusión

		Resultado Clasificación	
		PERSONA	**NO - PERSONA**
Instancias Reales	**PERSONA**	9	10
	NO - PERSONA	10	15

Fuente: Elaboración propia

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Como se puede apreciar, el procedimiento es muy sencillo. Sin embargo, obtener estos resultados no es suficiente. Es por ello que a partir de la matriz de confusión que se ha construido se definirán las siguientes medidas.

Nota: En cada medida descrita se ha utilizado los datos de la tabla 1.

- **Exactitud (*Accuracy*):** Mide la exactitud entre el resultado global de la exactitud y la clasificación exacta. Su expresión matemática es la siguiente.

$$Exactitud = \frac{Reales\ Positivos + Reales\ Negativos}{Predicciones\ totales}$$

$$Exactitud = \frac{9 + 15}{44} = 0.55$$

- **Precisión (*Precission*):** Mide la calidad de respuestas positivas del clasificador:

$$Precision = \frac{Reales\ Positivos}{Reales\ Positivos + Falsos\ Positivos}$$

$$Precision = \frac{9}{9 + 10} = 0.47$$

- **Sensibilidad (Sensitivity o True Positive Rate):** Evalúa la eficiencia en la clasificación de todos los elementos que son de la misma clase.

$$Sensibilidad = \frac{Reales\ Positivos}{Reales\ Positivos + Falsos\ Negativos}$$

$$Sensibilidad = \frac{9}{9 + 10} = 0.47$$

- **Especificidad (*Specificity o 1-False Positive Rate*):** Evalúa la eficiencia en la clasificación de todos los elementos que no son de la misma clase.

$$Especificidad = \frac{Reales\ Negativos}{Reales\ Negativos + Falsos\ Positivos}$$

$$Especificidad = \frac{15}{10 + 15} = 0.6$$

2.3.5 Tecnologías de desarrollo

Python

Es un lenguaje de programación de alto nivel y de código abierto, creado a finales de los ochenta por Guido Van Rossum. Tiene una filosofia multi-paradigma, soporta programación orientada a objetos, programación imperativa, programación funcional. Desde los últimos años este lenguaje de programación no ha hecho nada más que crecer y crecer ganando adeptos en todo el mundo y es que hoy en día puedes programar casi cualquier cosa en este lenguaje. La documentación y las nuevas actualizaciones están disponibles en https://www.python.org/.

OpenCV

Es una biblioteca de código abierto con licencia BSD. Cuenta con diferentes algoritmos de visión por computador implementados. Tiene una estructura modular, que significa que el paquete incluye varias bibliotecas compartidas o estáticas. Comprende 8 módulos: core, imgproc, video, calib3d, featured2d, objdetect, highgui, gpu. Podemos encontrar más información en https://opencv.org/.

DLIB

Contiene una amplia gama de algoritmos de aprendizaje automático. Todo diseñado para ser altamente modular, rápido de ejecutar y fácil de usar a través de una API limpia y moderna de C++. Se utiliza en una amplia gama de aplicaciones que incluyen robótica, dispositivos integrados, teléfonos móviles y grandes entornos informáticos de alto rendimiento. Toda la documentación esta disponible en http://dlib.net/ml.html.

***PIL* (Python Imaging Library)**

Añade capacidades de procesamiento de imágenes para el intérprete de Python. Esta biblioteca es compatible con muchos formatos de archivo y proporciona un potente procesamiento de imágenes y capacidades gráfica. La versión gratuita actual es PIL 1.1.7, compatible con Python 1.5.2 y versiones posteriores incluidas 2.5 y 2.6. No hay versión oficial para Python 3.x

2.4 Definición de términos básicos

a) **Vida de Muestra:** Es la detección de si la muestra adquirida y comparada proviene de un tejido vivo o no. Restringe la presentación de biometría falsa.

b) **Biometría:** Ciencia que se dedica al estudio del cuerpo humano como medio de identificación y autentificación del sistema.

c) **Reconocimiento de patrones:** es la ciencia que se encarga de la descripción y clasificación (reconocimiento) de objetos, personas, señales, representaciones, etc.

d) **Procesamiento de imágenes:** Campo que se basa en la transformación de imágenes para obtener otra de más calidad o mejor acondicionada para la posterior extracción de información.

e) **Ambiente no controlado:** se dice que un ambiente es no controlado cuando no se tiene control de ninguna variable del entorno. Llámese variables de entorno a factores ambientales como luz, oscuridad, etc. variabilidad de pose, presencia de oclusiones, etc.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Capitulo III: Marco Metodológico

3.1 Tipo y diseño de la investigación

Tipo de investigación

La presente investigación es del tipo Tecnológica. En este tipo de investigación, la realidad se puede cambiar y el hombre puede llevar acabo tal intervención. Determinar afirmaciones particulares que puede ser operativa o ejecutable, sirve como base para presentar, un método propio en cuanto al reconocimiento facial y poder contribuir con un pequeño granito de conocimiento en esta línea de investigación.

Diseño de investigación

El diseño de la investigación es Cuasi-Experimental, porque dentro de la ejecución del proyecto, existen variables que no se puede tener control absoluto, pero se pretende tener el mayor control posible. En este proyecto, la presencia excesiva de luz o ausencia de ella, juega un punto crítico para la detección y extracción de características faciales.

3.2 Población y Muestra

Población

16 algoritmos de extracción de puntos de referencia facial (Anexo I) y 05 algoritmos de detección de rostro (Anexo II)

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Muestra

Por conveniencia se va a emplear, para la detección de rostro 02 algoritmos (Viola & Jones; descriptor HOG), y para la extracción de puntos de referencia facial el algoritmo propuesto por (Kazemi & Sulivan, 2014)

3.3 Hipótesis

El algoritmo de Histograma de gradientes orientados y el modelo de apariencia activa, presentará mejores resultados en seguridad para la detección del rostro y extracción de patrones de conducta facial en tiempo real.

3.4 Variables

Variable independiente

Algoritmo de reconocimiento de patrones de conducta facial.

Variable dependiente

Seguridad en el reconocimiento de patrones de conducta facial

3.5 Operacionalización

Variable Independiente	Dimensiones	Indicadores	Técnicas e instrumentos de recolección de datos
Algoritmo de reconocimiento de Patrones de conducta facial	Cantidad de Imágenes	Tiempo de procesamiento	Documentación Muestra

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Variable Dependiente	Dimensión	Descripción	Indicadores	Técnicas e instrumentos de recolección de datos
Seguridad en el reconocimiento de patrones de conducta facial	Sensibilidad		$RP = reales\ positivos$ $FN = falso\ negativo$ $S = \frac{RP}{RP + FN}$	Técnicas de observación
	Precisión (precission)	Calidad de respuesta positiva del clasificador	$RP = reales\ positivos$ $FP = falsos\ positivos$ $P = \frac{RP}{RP + FP}$	

3.6 Métodos, técnicas e instrumentos de recolección de datos

Los métodos a utilizar dentro de la investigación son:

- *Métodos empíricos:* Extracción de los puntos de referencia facial. Para ello se determina un total de 68 puntos.
- *Métodos de la observación:* Observación para evaluar los puntos de referencia facial y determinar los patrones de conducta con la finalidad de reconocer un rostro real.
- *Métodos de medición:* Los resultados obtenidos se validarán para determinar el grado de precisión y exactitud del método que se propone a desarrollar en la presente investigación.

Para la recolección de datos se empleará las siguientes técnicas:

- *Observación:* Consiste en analizar las imágenes faciales obtenidas de la muestra existente, identificando y clasificando los puntos característicos de cada imagen para el análisis posterior que se verá reflejado en la implementación del software
- *Documentación:* Se evaluará los algoritmos más eficientes para hacer el reconocimiento de imágenes faciales en tiempo real. Esto se tomará en función de investigaciones anteriores, analizando los resultados que hayan obtenido.

3.7 Procedimiento para la recolección de datos

Observación

Analizar los puntos de referencia facial para determinar los patrones de conducta facial, por ejemplo, la distancia inter ocular del ojo para determinar el parpadeo. O también, el radio de aspecto de la boca para determinar si una boca esta abierta o cerrada. Y así, diferentes medidas que se deben analizar para estables un vector característico de patrones de conducta facial.

Análisis documental

Sirve como referencia para establecer que algoritmos son más eficientes, y poder implementar en nuestro sistema de acuerdo a la realidad que se plantea.

Muestra

Conjunto de imágenes adquiridas en ambientes no controlados. Sirve como base para establecer el comportamiento producto de la implementación de los algoritmos a evaluar

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

3.8 Plan de análisis estadístico de datos

Para evaluar cada algoritmo y método a desarrollar en la presente investigación, se tomará como referencia la matriz de confusión.

Para ello se describe las siguientes formulaciones matemáticas.

- **Sensibilidad:** Evalúa la eficiencia en la clasificación de todos los elementos que son de la misma clase.

$$Sensibilidad = \frac{Reales\ Positivo}{Reales\ Positivos + Falsos\ Negativos}$$

Donde:

- Reales positivos: representan los candidatos que han sido correctamente clasificados.
- Falsos Negativos: Representan a los candidatos personas que han sido incorrectamente clasificados por el clasificador

- **Precisión:** Calidad de respuesta positiva del clasificador.

$$Precisión = \frac{Reales\ Positivos}{Reales\ Positivos + Falsos\ Positivos}$$

Donde:

- Falsos positivos: representan a los candidatos que han sido incorrectamente clasificados por el clasificador.

3.9 Principios éticos

La presente propuesta de investigación se realiza siguiendo los criterios éticos que debe tener todo investigador que se tomarán en cuenta. La expresión “principios éticos básicos” se refiere a aquellos conceptos generales que sirven como justificación básica para los diversos principios éticos

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

y evaluaciones de las acciones humanas. Entre los principios básicos seguidos en el desarrollo de nuestra propuesta de investigación tenemos: Responsabilidad social Rectitud.

3.10 Criterios de rigor científico

La presente propuesta de investigación se realiza siguiendo los juicios científicos establecidos, estos permiten garantizar la calidad de la propuesta de investigación.

Así, seguimos la coherencia metodológica durante el desarrollo de la propuesta de la investigación, realización apropiada del muestreo de datos, los cuales son al azar para ser totalmente imparcial en el recojo de datos. Los datos recolectados se basan en puntos de referencia facial extraídos, los cuales son tomados como datos principales.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Capitulo IV: Análisis e interpretación de los resultados

4.1 Detección de rostro

Para la detección del rostro se ha construido la matriz de confusión tomando como referencia los datos del Anexo III, utilizando la base de imágenes en ambientes no controlados que se encuentra en la carpeta ***"imgBD"***

Tabla 2: Matriz de confusión para el algoritmo de Viola & Jones

	ROSTRO	ROSTRO
ROSTRO	179	21
ROSTRO	74	0

Fuente: Elaboración propia

Tabla 3: Matriz de confusion para el algoritmo de Hog + svm

	ROSTRO	ROSTRO
ROSTRO	199	1
ROSTRO	0	0

Fuente: Elaboración propia

A partir de ambas matrices construidos, se calcula la exactitud y precisión, teniendo en cuenta las siguientes formulaciones matemáticas.

$$Exactidud\ (E) = \frac{Reales\ positvos + Reales\ negativos}{Predicciones\ totales} \qquad 4.1.1$$

$$Precisión\ (P) = \frac{Reales\ positivos}{Reales\ positivos + falsos\ positivos} \qquad 4.1.2$$

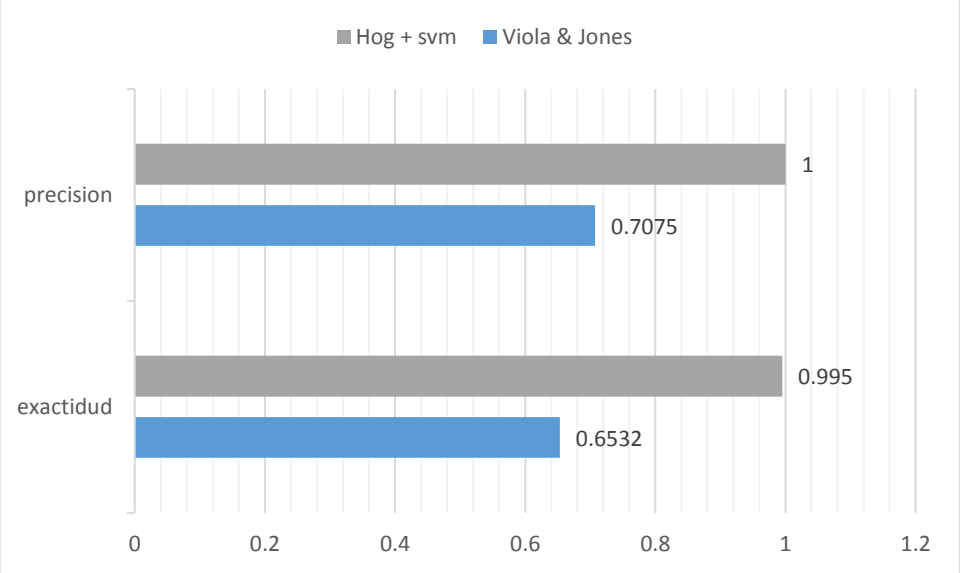
Calculamos estas medidas para el algoritmo de Viola & Jones con los datos de la Tabla 2.

$$E = \frac{179+0}{179+21+74+0} = 0.6532 \text{ y } P = \frac{179}{179+74} = 0.7075$$

Lo mismo, para el algoritmo de HOG +SVM con los datos de la Tabla 3.

$$E = \frac{199+0}{179+1+0+0} = 0.995 \text{ y } P = \frac{179}{179+74} = 1$$

Gráfico 1: Medidas de evaluación para el algoritmo de Viola & Jones y Hog+svm



Fuente: Elaboración propia

Del ***Gráfico 1*** se puede observar que el algoritmo Viola & Jones tiene una precisión de 0.71 y una exactitud de 0.65 en comparación al algoritmo de Hog+svm que presenta una precisión de 1 y una exactitud de 0.99. Por lo tanto, se concluye que el mejor algoritmo para la detección de rostros en ambientes no controlados es Hog+svm.

4.2 Patrones de conducta facial

4.2.1 Detección de parpadeo

En el caso del parpadeo de los ojos, se ha evaluado la relación de aspecto del ojo (EAR), considerando que el parpadeo es el cierre y apertura rápida de los parpados. Con este propósito, se ha realizado la captura de dos videos: cada video con presencia únicamente de ojos abierto y ojos cerrados por separado.

a) Video con ojos abiertos

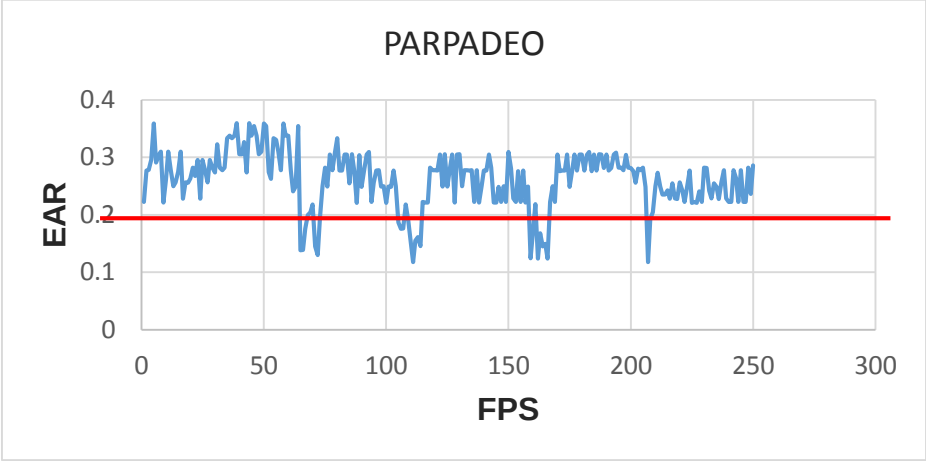
Por ello, se ha evaluado un video de 250 frames. En la Tabla 4, se muestra un resumen del Anexo 4 de cada frame con su respectivo valor EAR

Tabla 4: Valor EAR obtenido para cada frame evaluado

FPS	EAR
1	0.22222222
2	0.27709819
3	0.27827004
4	0.29617438
5	0.35867438
6	0.2907014
7	0.3036867
8	0.31006327
9	0.22154264

Fuente: Elaboración propia

Gráfico 2: Dispersión de datos del parpadeo de ojos en una secuencia de video



Fuente: Elaboración propia

En el Grafico 2, se muestra la detección de parpadeo (ojo abierto y cerrado), en una secuencia de video. En este caso, se presenta 4 parpadeos, para todos los frames por debajo de la línea roja.

4.2.2 Detección de boca abierta y cerrada.

Se utiliza el mismo principio de la detección de parpadeo. En este caso se propone la relación de aspecto de la boca (abreviado en MAR, que traducido en ingles significa: relation aspect mouth). Hay dos tipos de experimentos realizados. Un experimento realizado en una secuencia de video (Sección a) y

el otro experimento realizado en un conjunto de 50 imágenes en ambiente no controlado (Sección b).

a) Experimento en secuencia de video

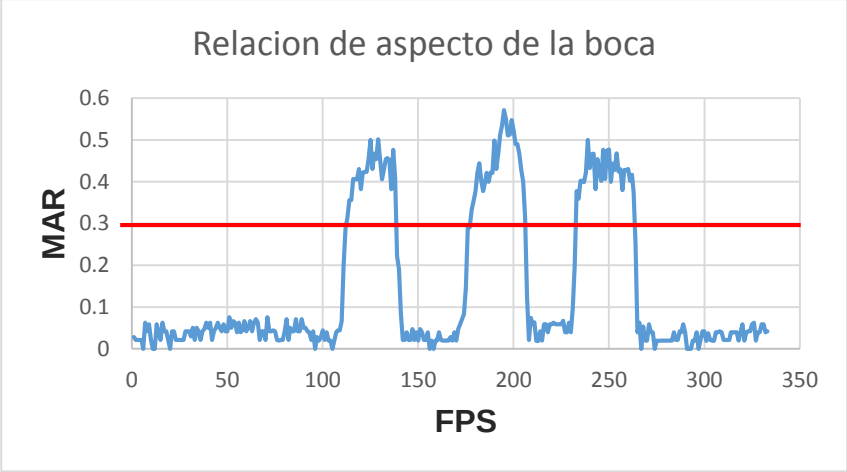
Se ha tomado una secuencia de video de 333 frames. En la table 5 se muestra un resumen del Anexo V, para cada frame con su valor MAR, respectivamente.

Tabla 5: Valor MAR, obtenido para cada frame

FPS	MAR
42	0.020833333
43	0.041666667
44	0.050296116
45	0.0625
46	0.050296116
47	0.041666667
48	0.058925565
49	0.041666667

Fuente: Elaboración propia

Gráfico 3: Boca abierta y cerrada en secuencia de video



Fuente: Elaboración propia

En el Gráfico 3, se muestra la presencia de 3 veces se ha abierto la boca, para todos los valores superiores a 0.3, mientras que los valores inferiores a 0.3 representa el estado de boca cerrada. Ahora veamos como es el comportamiento

cuando se detecta boca abierta en imágenes. Para ellos, se ha construido una base de imágenes de 50 elementos.

b) Experimento en imágenes

Se ha construido una base de imágenes de 50 elementos. Cada imagen es adquirida en ambientes no controlados y diferentes tamaños. El rostro presenta diferentes posiciones. A continuación, se muestra una tabla, que es un resumen del anexo V. Para efectos de prueba, se ha evaluado 3 valores: 0.1, 0.2, 0.3. Al hacer la comparación con el valor MAR, se establece lo siguiente. Si MAR es mayor que los valores establecidos, entonces existe la presencia del estado de boca abierta que se representa con el valor 1. Y cuando MAR es menor que los valores establecidos, el clasificador detecta la presencia de boca cerrada, el cual se denota con 0.

Tabla 6: Detección de boca abierta con la relación de aspecto de la boca (MAR).

img	MAR	>0.3	>0.2	>0.1
1	0.26715342	0	1	1
2	0.17692623	0	0	1
3	0.6700202	1	1	1
:	:	:	:	:
48	0.22053956	0	1	1
49	0.17670428	0	0	1
50	0.23085433	0	1	1
	TOTAL:	17	33	45

Fuente: Elaboración propia

El resultado al hacer las pruebas se ha trabajado con dos valores, tal como se puede apreciar en la tabla 6, que muestra el resumen del anexo VII. En este caso el valore 0 representa estado neutral y 1 representa un estado de sonrisa.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Tabla 7: Tabla de valores para detectar boca abierta y cerrada.

FRAME	MAR >0.3	MAR >0.1
1	0	1
2	0	1
3	1	1
4	1	1
4	0	0
5	1	1
6	0	0

Fuente: Elaboración propia

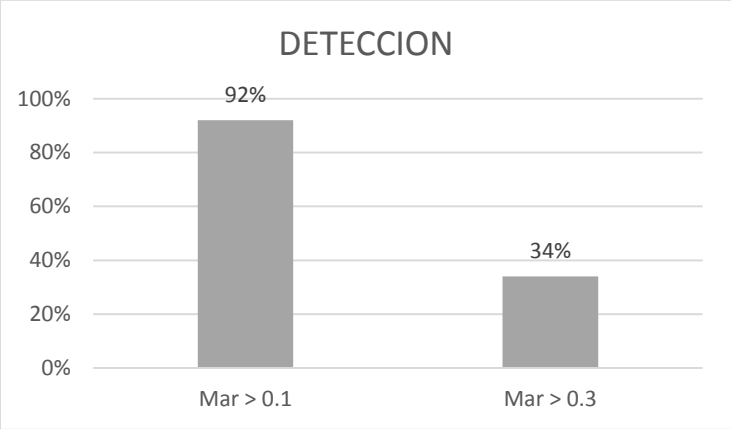


Gráfico 4: Detección de boca abierta aplicado en imágenes

Fuente: Elaboración propia

Como se puede apreciar en el grafico 4, el valor de MAR superior a 0.3 la detección es mínima cuando se hace el test en imágenes, algo que no ocurre en la secuencia de video. Sin embargo, cuando se trabaja con un valor superior a 0.1 la tasa de detección sube a 92%, pero en secuencia de video, este valor genere muchos falsos positivos. En conclusión, el valor a tomar depende de las circunstancias en que se desarrolla el proyecto.

Detección de sonrisa

Para evaluar la detección de sonrisa, se ha considerado un fragmento de video de 255 frames. En la Tabla 8, se muestra un resumen de los datos del Anexo VI.

Tabla 8: Tabla de valores obtenidos mediante la triangulación CHN.

FPS	Ángulo
1	45.82
2	48.06
3	44.95
4	46.3
5	44.98
6	45.22
7	46.26
8	48.06

Fuente: Elaboración propia

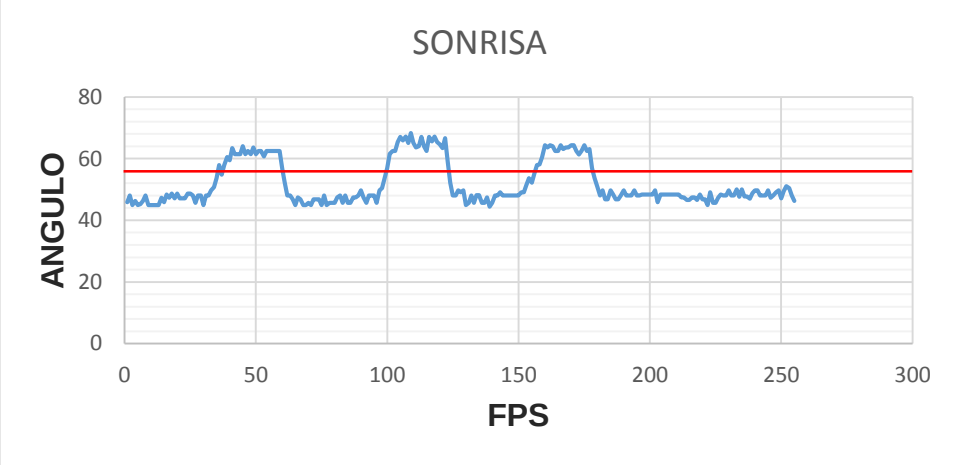


Gráfico 5: Detección de sonrisa en secuencia de video

Fuente: Elaboración propia

En el grafico 4 se muestra la presencia de 3 sonrisas, teniendo en cuenta que el ángulo con valores superiores a 54° representa una sonrisa y valores inferiores a 54° representa al rostro en su forma neutral.

Luego se hace una prueba en un conjunto de 100 imágenes, donde 50 imágenes representan a rostros únicamente sonriendo y 50 imágenes representan a rostros con diferentes estados de ánimo.

En la tabla 8, muestra el resumen del anexo VII, donde 0 representa estada neutral y 1 representa el estado de sonrisa.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Tabla 9: Tabla con valores de detección de sonrisa en dos conjuntos de 50 imágenes.

Img	Rostros con diferente estado de animo	Rostros únicamente con sonrisa
1	0	1
2	1	1
3	1	1
4	0	1
4	0	1
5	0	1
6	0	1

Fuente: Elaboración propia

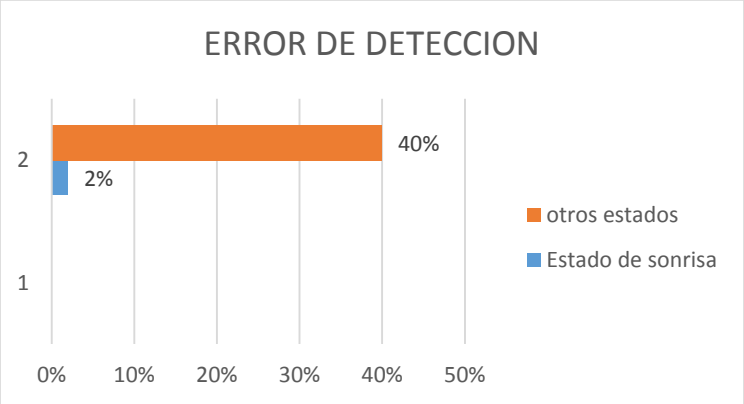


Gráfico 6: Error de detección en un conjunto de 100 imágenes

Fuente: Elaboración propia

En el grafico 6 se muestra un error de 2% cuando se trabaja en un conjunto de imágenes donde esta únicamente presente el estado de ánimo de sonrisa. Cuando se evalúa en un conjunto de 50 imágenes donde está presente otros estados de ánimo, presenta un error de 40%

Capítulo V: Propuesta de investigación

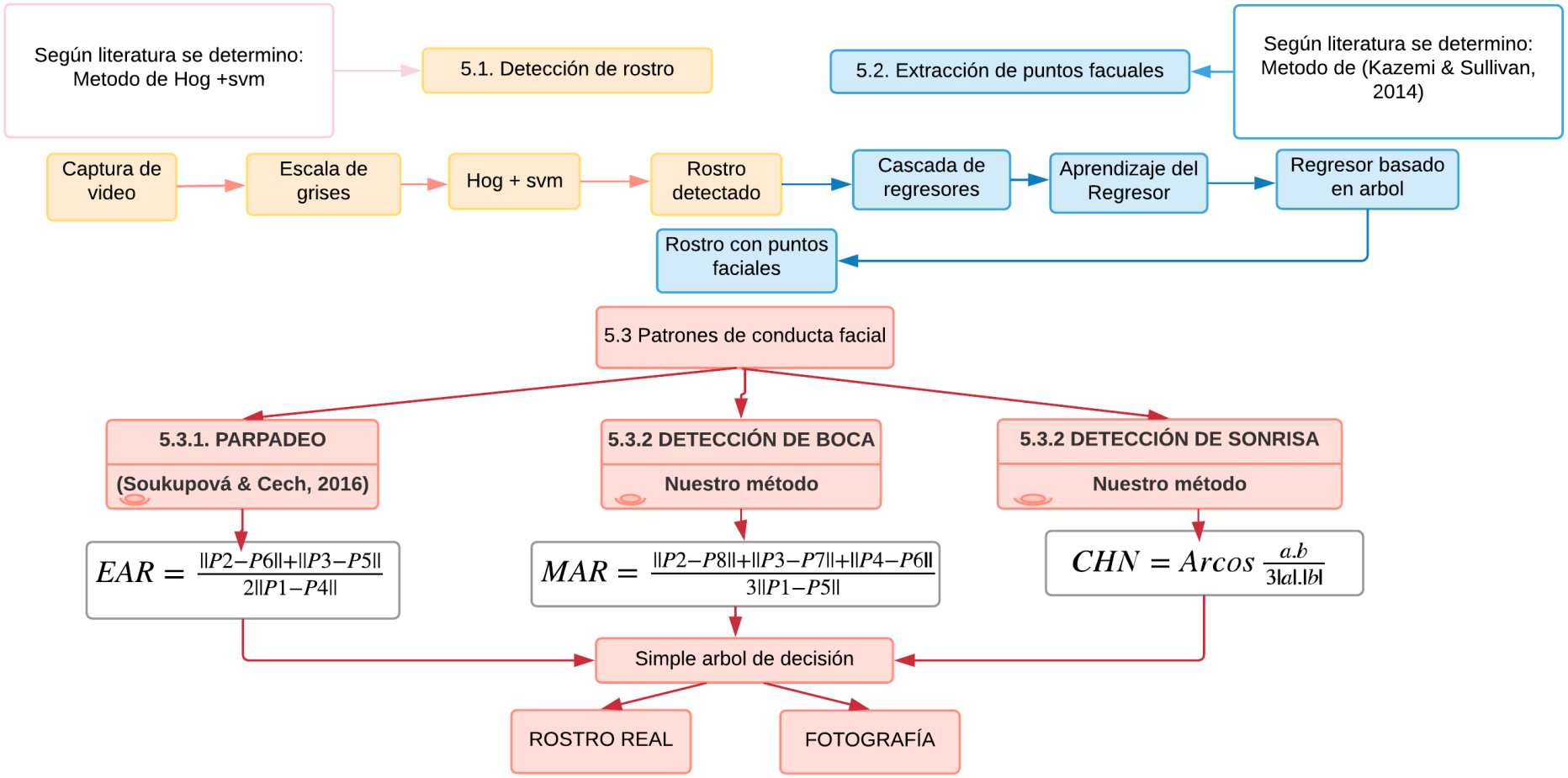


Figura 17: Método propuesto para el desarrollo de la investigación
Fuente: Elaboración propia

5.1 Detección de rostro

La detección de rostro es considerada como la etapa más importante dentro del reconocimiento facial. En esta etapa se ha elaborado la *Tabla 05* con diferentes algoritmos de detección de rostros a evaluar. Se ha tenido como indicador de evaluación la *precisión*, que mide la calidad de respuestas positivas del clasificador.

Tabla 10: Algoritmos de detección de rostros

	Autor	Algoritmo	Precisión
1	García, Gonzales, Alegre, Fidalgo, 2017).	Viola & Jones	0.86
2	García, Gonzales, Alegre, Fidalgo, 2017).	**HOG +SVM en DLIB**	**0.99**
3	García, Gonzales, Alegre, Fidalgo, 2017).	MTCNN	0.93
4	(Ueno, 2003).	Segmentación por color de piel	0.97
5	(Scott, Chun, Boonping, s.f)	Template matching + segmentación por color de piel	0.89

Fuente: Elaboración propia

Al evaluar los algoritmos de la tabla 05, se observa que el método de HOG + SVM en Dlib obtiene una precisión de 0.99 en comparación con los otros algoritmos. Por esta razón, para llevar a cabo la detección de rostro se ha

utilizado el algoritmo descrito anteriormente. Y para el desarrollo de todo este proceso se resumen en [Fig. 18]

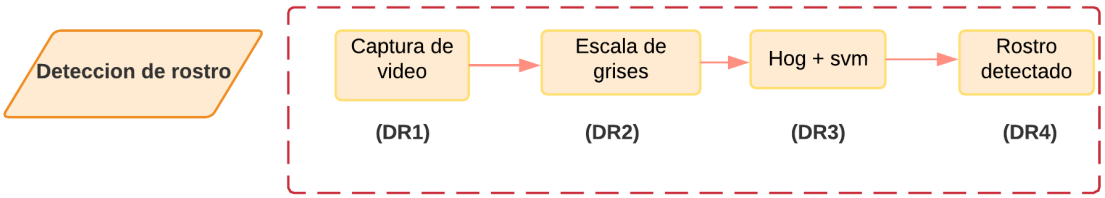


Figura 18: Diagrama de secuencia para la detección de rostro
Fuente: Elaboración propia

DR1. Captura de video

Dado que el sistema a desarrollar es en tiempo real, tenemos que hacer la captura de video desde una cámara. Se ha utilizado la cámara incorporada en un equipo HP Pavilion 14 que trabaja a 30 fps con una resolución de 1280 x 720 pixeles. A continuación, crearemos un script "Apfacial.py" de Python 2.7 para realizar el proceso de captura de video.

```
1. import cv2
2. class Apfacial:
3. def __init__(self, video_source=0):
4.     self.vid = cv2.VideoCapture(video_source)
5.     if not self.vid.isOpened():
6.           raise ValueError("Error al abrir origen de webcam",
7.                            video_source)
8. def get_frame(self):
9.     ret, frameFPS = self.vid.read()
10.    frame = cv2.cvtColor(frameFPS, cv2.COLOR_BGR2RGB)
11.    return ret, frameFPS
```

La **línea 1** maneja la importación de la librería de procesamiento de imágenes Opencv 2.4. En la **línea 2** se implementa la clase *Apfacial* que tiene un constructor definido en la **línea 4-7.**

El constructor contiene un parámetro: *vid* para crear un objeto VideoCapture que recibe un parámetro con valor 0. El valor 0 es porque se está trabajando con la cámara integrada del equipo. En caso que fuese una cámara externa, se pasa un valor de 1. En la **línea 8-11** se declara la función get_frame que retorna el frame de la captura de video cuadro por cuadro.

DR2. Escala de Grises

En el apartado **2.3.1** de dicho proyecto se describe las diferentes formas de convertir una imagen RGB a escala de grises. Por conveniencia, se ha utilizado la forma más común es utilizar el promedio de los valores RGB. Siguiendo con el script "Apfacial.py", en la función get_frames(), se agrega la **línea 11,** encargada de transformar una imagen RGB a escala de grises con la función *cvtColor*() de Opencv. Este proceso se realiza con la finalidad de reducir la intensidad de luz en cada cuadro de imagen.

```
8. def get_frame(self):
9.     ret, frameFPS = self.vid.read()
10.    frame = cv2.cvtColor(frameFPS, cv2.COLOR_BGR2RGB)
11.    frame_grey = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
12.    return ret, frameFPS
```

DR3. HOG + SVM

El método HOG (Dalal, Triggs, 2005), utilizados principalmente para describir la forma y el aspecto de un objeto. Consta de diferentes etapas que se describe en el algoritmo 01, tomando como referencia la explicación dada por Valveny (2015).

Algoritmo 01: Calculo general del descriptor HOG

1. Inicio
2. Leer imagen escala de grises
3. Calculo de gradiente
4. Calculo de histograma de gradiente
5. Calculo del descriptor HOG
6. Fin

Cálculo de gradiente.

El gradiente se define como un cambio de dirección en intensidad definido por dos valores: dirección y magnitud. Supongamos que tenemos una imagen de entrada (Fig. 20), en escala de grises de tamaño 5 x 5.

0	0	255	255	255
0	0	255	255	255
0	0	0	255	255
0	0	55	0	0
0	0	0	0	0

Figura 19: Imagen en escala de gris de tamaño 5x5

Fuente: Elaboración propia

Para realizar el cálculo de gradiente del pixel sombreado de amarillo, se procede de la siguiente manera:

a) Calcular la diferencia de intensidad en el eje x.

$$dx = I(x+1, y) - I(x-1, y) \qquad 5.1\ (1)$$

$$dx = \ 255 - 0 = 255$$

b) Calcular la diferencia de intensidad en el eje y

$$dy = I(x, y+1) - I(x, y-1) \qquad 5.1\ (2)$$

$$dy = \ 255 - 55 \ = 200$$

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

c) Hallar la dirección $\sigma(x,y)$

$$\sigma(x,y) = \text{arctang}\frac{dy}{dx} \quad \text{5.1 (3)}$$

$$\sigma(x,y) = \text{arctang}\frac{200}{255} = 38.11$$

d) Hallar la magnitud $g(x,y)$

$$g(x,y) = \sqrt{dx^2 + dy^2} \quad \text{5.1 (4)}$$

$$g(x,y) = \sqrt{255^2 + 200^2} = 324.08$$

El resultado se puede observar en [Fig. 21], donde d(x) representa la diferencia de intensidad en el eje x, d(y) representa la diferencia de intensidad en el eje y, por último, g(x, y) representa la magnitud de gradiente obtenido.

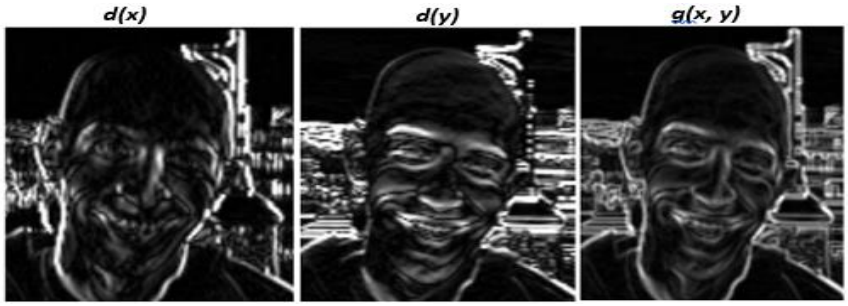


***Figura 20**:* Resultado de imagen al aplicar el cálculo de gradiente
Fuente: (Rosebrok, s.f)

Esta representación, tal como explica (Valveni, 2015), difícilmente se puede utilizar junto con un clasificador en un sistema de detección de objetos, por esta razón, es necesario convertir esta información local a nivel de pixeles en una representación global de toda la imagen en forma de vector de características, que capture la forma global del objeto.

Cálculo de histograma de gradiente

Según Valveni, (2015), explica que el descriptor actúa de la siguiente manera.

- División de la imagen en celdas de tamaño fijo. Valores habituales suelen ser en 6 y 8 pixeles tanto en ancho como en alto. Y para cada una de estas celdas se obtiene un histograma de las orientaciones de los gradientes en dicha celda. Un ejemplo según explica (Rosebrok, s.f.) seria si tuviéramos una imagen de 128x128 y definiéramos nuestra pixels_por_celda como 4 x 4, tendríamos así 32 x 32 = 1024 celdas.
- División del rango de orientaciones en un número de intervalos fijo. El número de orientaciones controla el número de bandejas en el histograma resultante. El ángulo de gradiente está dentro del rango [0, 180] (sin signo) o [0, 360] (con signo), siendo el gradiente sin signo el más implementado.

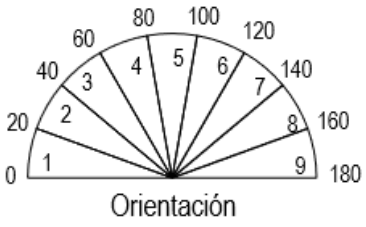


Figura 21: División del rango de orientaciones en un numero de intervalos fijos.
Fuente: Valveny, (2015).

- Cada pixel con su magnitud y orientación de cada celda se debe asignar a un intervalo en función de la orientación del gradiente.
- Y por último, acumular la magnitud de cada gradiente de todos los pixeles asignados a un intervalo.

Matemáticamente se puede expresar de la siguiente manera.

$$h(k) = \sum_{(x,y \in C)} \omega_k(x,y) g(x,y) \qquad 5.1\ (5)$$

Para un escenario simplificado:

5.1 (6)

$$\omega_k(x,y) = \begin{cases} 1 & si\ (k-1)\delta\theta \leq \theta(x,y) < k\delta\theta \\ 0 & en\ caso\ contrario \end{cases}$$

Y para una interpolación en orientación

$$\omega_k(x,y) = max(0.1 - \frac{\theta(x,y) - \theta_k}{\delta\theta}) \quad 5.1\ (7)$$

Cálculo del descriptor HOG.

Según Valveni (2015), resume este proceso en los siguientes parámetros: tamaño de celda, signo del gradiente, número de intervalos histograma de orientaciones y el número de celda en cada bloque. Este vector característico está representado de la siguiente manera.

$$hog = (x_1, \ldots, x_{n)} \quad 5.2.\ (8)$$

donde:$n = n°\ bloques * n°\ celda/bloque * n°\ intervalos\ de\ histograma$

$$n°\ bloques = n°celdas - n°celda/bloque + 1$$

Por ejemplo, queremos calcular el descriptor HOG para una imagen de 64 x128 teniendo los siguientes parámetros: tamaño de celda de 8x8, n° celda por bloque 2x2, orientación del histograma sin signo. ¿Cuál es la dimensión final del descriptor HOG?

Solución:

Hallar cant. de celdas en eje x e y

X = 64 /8 = 8

Y = 128/8 = 16

Hallar bloques imagen en eje x e y

X = 8 – 2 + 1 = 7

Y = 16 -2 +1 = 15

Dimensión final de HOG es.

HOG = 7*15*4*9 = 3780

***Figura 22**:* Transformación de imagen en escala de gris al descriptor HOG
Fuente: Elaboración propia

Hasta el momento, se ha descrito el funcionamiento del descriptor HOG, como plantillas rígidas que se escanean sobre una imagen. Ahora se debe detectar la ubicación del rostro y para ello se hace uso de las máquinas de vectores de soporte (SVM), teniendo como referencia la explicación de (Rosebrock, 2014), que se detalla a continuación:

1. Clasificar las muestras positivas del P de los datos de entrenamientos de objetos para detectar y extraer la descripción Hog de cada muestra.
2. Clasificar muestras negativas N que no contenga ninguno de los objetos a detectar. Tener en cuenta que N siempre debe ser mayor que P.
3. Entrenar un SVM lineal con las muestras P y N
4. Por cada imagen y todas las escalas posibles de las muestras P y N, se aplica la técnica de ventana deslizante. Cada ventana con su descriptor HOG, es entrenado con el clasificador dado. Si el clasificador detecta ventanas que no pertenecen al clasificador, representara a los falsos positivos.

5. Todas las muestras de falsos positivos, que se han detectado en el paso 4, se debe ordenar según la probabilidad deseada, y volver a entrenar el clasificador utilizando la muestra N
6. Hasta aquí, se debe tener un clasificador entrenado listo para ser aplicado a un conjunto de datos de pruebas

DR4. Rostro detectado

Para llevar a cabo este proceso, se ha implementado el script "Apfacial.py" desarrollado en el lenguaje de Python 2.7, acompañado de diferentes librerías, tal como se describen a continuación.

Se ha importado las librerías *Dlib*, que contiene lo algoritmos necesarios para la localización de puntos faciales. La librería *numpy* para el tratamiento de datos con algebra línea. La librería *imutils* que contiene una serie de funciones básicas para el procesamiento de imágenes, tal como se muestra desde la **línea 1-6**.

```
1. from __future__ import division
2. from imutils import face_utils
3. from scipy.spatial import distance as dist
4. import dlib
5. import numpy as np
```

Establecemos nuestra clase *Apfacial(),* con la función *init(),* que recibe como parámetro el valor de origen de video. En la **línea 14,** se carga el modelo de predictor de 68 puntos faciales, luego, en la **línea 15-16** se inicializa el detector facial HOG +SVM proporcionado por la librería DLIB.

```
12.class Apfacial:
13.def __init__(self, video_source=0):
14.    self.predictor_path = "shape_predictor_68_face_landmarks.dat_2"
15.    self.detector   = dlib.get_frontal_face_detector()
```


Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

```
16.    self.predictor  = dlib.shape_predictor(self.predictor_path)
17.    self.vid = cv2.VideoCapture(video_source)
18.    if not self.vid.isOpened():
19.          raise ValueError("Error al abrir origen de webcam",
20.                           video_source)
```

Dentro de la función ***get_frame,*** en la **línea 45** se procede a leer cada frames del video almacenado en frameFPS. Cada frame recibido ***(línea 47)*** es transformado a escala de grises mediante la función *cv2.COLOR_BGR2GRAY* de opencv. Ahora, para determinar la ubicación del rostro se logra en la **línea 49,** mediante la función ***detector()*** que devuelve el cuadro de ubicación en el eje (x,y), para cada rostro detectad0.

La **línea 51,** recorremos cada una de los rostros detectadas y se dibuja un rectángulo delimitador del rostro en la **línea 52.**

```
44.def get_frame(self):
45.    ret, frameFPS = self.vid.read()
46.    frame = cv2.cvtColor(frameFPS, cv2.COLOR_BGR2RGB)
47.    frame_grey = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
48.    frame_resized = self.resize(frame_grey, 120, 230, 1)
49.    detections = self.detector(frame_resized, 1)
50.
51.    for k,d in enumerate(detections):
52.    cv2.rectangle(frame, (int(d.left()/ratio),
53.                                  int(d.top()/ratio)),
54.                                  (int(d.right()/ratio),
55.                                  int(d.bottom()/ratio)),
56.                                   (0, 255, 0), 1)
57.    shape = self.predictor(frame_resized, d)
58.
59.    return ret,frame, self.ear_ojos, self.mar_boca, self.sonrisa
```

Todo este proceso se ve reflejado en la siguiente figura.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

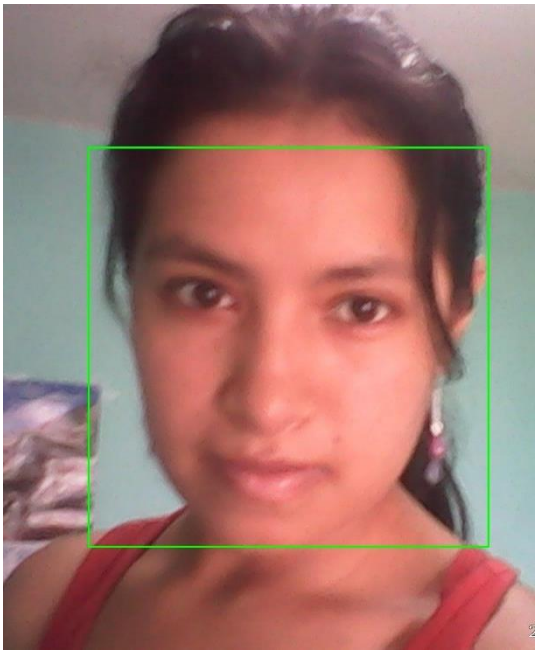


Figura 23: Detección de rostro mediante HOG +SVM
Fuente: Elaboración propia

5.2 Extraer Puntos característicos faciales

Se ha realizado un resumen de los diferentes métodos de extracción de puntos característicos faciales, tal como se muestra en la Tabla 10. Para determinar el método a utilizar, se ha tenido en cuenta lo siguiente: que detecte un total de 68 puntos faciales a nivel de ojos, cejas, nariz, boca y contorno de cara; que el método este entrenado en ambientes no controlados; que el método sea robusto ante las rotaciones de rostro; pero lo mas importante, que exista una gran variedad de documentación para ser implementado. Por esta razón, se ha optado por utilizar el método de (Kazemi & Sullivan, 2014), pues reúne todas las características que se ha mencionado anteriormente.

Tabla 11: Algoritmos para la extracción de puntos característicos faciales

AÑO	AUTOR	ALGORITMO	PUNTOS
2014	Zhang, Z., Zhang, W., Liu, J., Tang, X.	Multi View	13
2014	Kazemi & Sullivan	Ensemble of Regression Trees	68
2013	Xiang Yu; Junzhou Huang; Shaoting Zhang; Wang Yan; Dimitris N. Metaxas	Cascade Deformable Shape model	65
2013	Tadas Baltrusaitis; Peter Robinson; Louis-Philippe Morency	Constrained Local Neural Field model (CLNF)	65
2013	Yi Sun, Xiaogang Wang, Xiaoou Tang	Deep Convolutional Network Cascade	5
2013	Belhumeur, P.N., Jacobs, D.W., Kriegman, D.J., Kumar, N.	Bayesian Approach	68
2013	Xuehan Xiong , de la Torre, F.	Supervised Descent Method	29
2012	Xudong Cao, Yichen Wei, Fang Wen, Jian Sun	Explicit Shape Regression	49
2012	Y Tong, X Liu, FW Wheelerb, PH Tub,	Semi-Supervised learning	5 ó 87
2012	Dantone, M., Gall, J., Fanelli, G., Van Gool, L	Conditional Regression Forests	33
2012	(Sukno, et al, 2012	Combinatorial Search and Shape Regression	10
2012	Xiangxin Zhu, Ramanan, D	Tree-structured models	8
2011	Rapp, V., Senechal, T., Bailly, K., Prevost, L	Multiple Kernel Learning	61
2008	David Cristinacce, Tim Cootes	Constrained local models (CLM)	17
2008	De la Torre, F., Minh Hoai Nguyen	Parameterized Kernel Principal Component Analysis (PKPCA	22
2007	Yuchi Huang, Quinshang Liu, DimitrisMetaxas	Component based Deformable Mode	46

Fuente: Elaboración propia

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

El método de (Kazemi & Sullivan, 2014), presenta el siguiente esquema de trabajo, tal como se puede apreciar en [Fig.24]

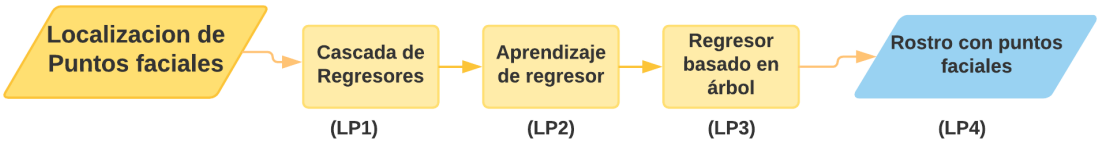


Figura 24: Diagrama de secuencia para la localización de puntos faciales del rostro
Fuente: Elaboración propia.

LP1. Cascada de Regresores

Se introduce el vector $S = (x_1^T, x_2^T, x_3^T, \dots x_p^T)$, que se conoce como shape, es el que contiene todos los *p* puntos de referencia en *I* (imagen). *Se usa* $S^{(t)}$ para estimar la denotación concurrente de *S.* Cada regresor r_t predice la actualización del vector de la imagen. Denotado matemáticamente como:

$$S^{(t+1)} = S^{(t)} + r_t * (I, S^{(t)})$$

El punto crítico de este regresor es que hace sus predicciones basado en características de intensidad de valores de los pixeles.

LP2. Aprendizaje de regresor en cascada.

Los datos generados por la cascada de regresores (I1, S1), ..., (In, Sn), representa el rostro de la imagen en S_i. Y para el aprendizaje de la primera función r_0, se crea un entrenamiento de datos triples del rostro de la imagen. Todos los datos generados, se utiliza el gredient tree boosting con una suma de perdida de error cuadrático. El algoritmo se representa en la siguiente imagen.

Algoritmo: Aprendizaje de regresor en cascada

Se tiene un entrenamiento de datos $\{I_\pi, S_i^{(t)}, \Delta S_i^{(t)})\}N_{i=1}$ y el radio de aprendizaje (shrinkage factor) 0 < *v* < *1*

1. Inicializar

$$f_0(I, S^{(t)}) = \arg min \sum_{i=1}^{N} ||\Delta S_i^{(t)} - \gamma||^2$$

2. Para k = 1, …, K:
   (a) Asignar para i=1, …., N

$$r_{IK} = \Delta S_i^{(t)} - f_{k-1}(I_{\pi,} S_i^{(t)}$$

   (b) Ajuste de regresión al árbol para la etiqueta r_{ik}, dado una función de regresión débil $g_k(I, S^{(t)})$.
   (c) Actualizar

$$f_k(I, S^{(t)}) = f_k - 1(I, S^{(t)}) + v g_k(I, S^{(t)})$$

3. Salida $r_i(I, S^{(t)}) = f_k(I, S^{(t)})$

LP3. Regresor basado en árbol

Cada nodo del regresor del árbol, se hace una decisión en función del thresholding de diferencia entre las intensidades de dos pixeles. Cada uno de estos pixeles están en la posición u y v cuando se definen en el sistema de coordenadas, donde cada imagen se puede deformar a la forma media en función de la estimación de la forma actual antes de extraer las características faciales.

Entonces, por cada nodo del regresor de árbol, se hace una aproximación a la función subyacente con una función constante por partes donde un vector constante es adecuado para cada nodo de la hoja. Y para hacer el entrenamiento, se genera aleatoriamente un conjunto de divisiones candidatas

LP4. Rostro con puntos faciales

Para obtener las coordenadas de los puntos faciales, se sigue trabajando con el script *"Apfacial.py".* Dentro de la función *get_frame(),* se ha agregado la **línea 57,** donde se aplica el detector de referencia facial a la región de la cara devolviendo un objeto *shape*, que se ha convertido a una matriz Numpy de tipo entero con una dimensión de 68 elementos en la **línea 58.**

```
44.def get_frame(self):
45.    ret, frameFPS = self.vid.read()
46.    frame = cv2.cvtColor(frameFPS, cv2.COLOR_BGR2RGB)
47.    frame_grey = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
48.    frame_resized = self.resize(frame_grey, 120, 230, 1)
49.    detections = self.detector(frame_resized, 1)
50.
51.    for k,d in enumerate(detections):
52.    cv2.rectangle(frame, (int(d.left()/ratio),
53.                                  int(d.top()/ratio)),
54.                                  (int(d.right()/ratio),
55.                                  int(d.bottom()/ratio)),
56.                                   (0, 255, 0), 1)
57.    shape = self.predictor(frame_resized, d)
58.    coords = np.zeros((68,2), dtype= int)
59.
60.    for i in range(0,68):
61.          coords[i]= (shape.part(i).x, shape.part(i).y)
62.          cv2.circle(frame, (int(shape.part(i).x/ratio),
63.                             int(shape.part(i).y/ratio)), 1,
64.                              (255,255, 0), thickness=2)
65.
66.
67.    return ret,frame, self.ear_ojos, self.mar_boca, self.sonrisa
```

Para la **línea 60-62** se recorre el objeto coords y se dibuja una serie de círculos de tamaño 1x1, visualizando cada punto de referencia facial, tal como se muestra en la siguiente imagen.

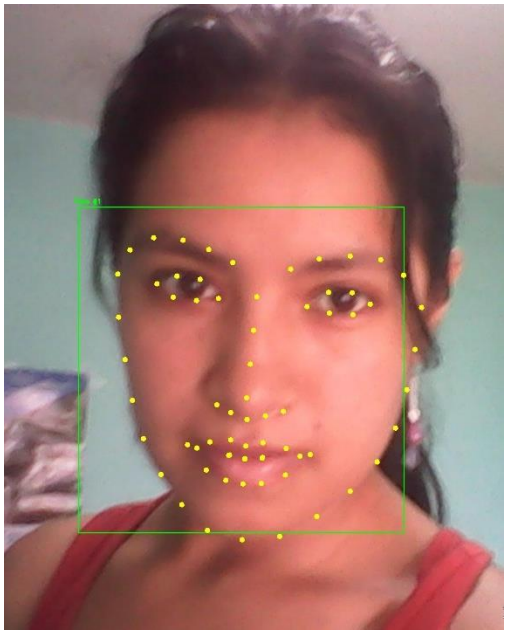


Figura 25. Estimación de los puntos de referencia facial.
Fuente: Elaboración propia

5.3 Patrones de conducta Facial

Según (Pineda, 2017) explica que un patrón representa un conjunto de características, formando una determinada clase con un comportamiento en particular con la finalidad de realizar diferentes tareas como discriminativas, predictivas o explicativas utilizando diferentes algoritmos conocidos como clasificadores.

5.3.1 Detección de parpadeo

Algoritmo 02: Detector de parpadeo de ojos

1. Inicio
2. Leer coordenadas faciales
3. Asignar valor: ojo izquierdo (p36, p37, p38, p39, p40, p41) y ojo derecho (p42, p43, p44, p45, p46, p47)
4. Calcular EAR ojo izquierdo
5. Calcular EAR ojo derecho
6. EAR = [(EAR ojo izquierdo) + (EAR ojo derecho)]/2
7. Si EAR > 0.20
   Estado = ojo abierto

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

8. Else
9. Estado = ojo cerrado
10. fin

El algoritmo 02, se ha implementado en el script “Coordenadas.py” que recibe las coordenadas X e Y de la ubicación de los puntos faciales. El código fuente del script se muestra a continuación.

```
1. import math
2. from scipy.spatial import distance as dist
3.
4. def puntosOjoIzquierdo©:
5.       p36, p37, p38, p39, p40, p41 = 0,0,0,0,0,0
6.       p36 = c[36]
7.       p37 = c[37]
8.       p38 = c[38]
9.       p39 = c[39]
10.      p40 = c[40]
11.      p41 = c[41]
12.      x1 = dist.euclidean(p37, p41)
13.      x2 = dist.euclidean(p38, p40)
14.      x3 = dist.euclidean(p36, p39)
15.      ear_izquierdo = (x1+x2)/(2.0 * x3)
16.      return ear_izquierdo
17.
18.def puntosOjoDerecho©:
19.      p42, p43, p44, p45, p46, p47 = 0,0,0,0,0,0
20.      p42 = c[42]
21.      p43 = c[43]
22.      p44 = c[44]
23.      p45 = c[45]
24.      p46 = c[46]
25.      p47 = c[47]
26.      x1 = dist.euclidean(p43, p47)
27.      x2 = dist.euclidean(p44, p46)
28.      x3 = dist.euclidean(p42, p45)
29.      ear_derecho = (x1+x2)/(2.0 * x3)
30.      return ear_derecho
```


Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

En la **línea 1-2,** se importa las librerías de procesamiento matemático para el cálculo de diferentes funciones matemáticas a utilizar. Desde la **línea 4-16**, se ha definido la función *puntosOjoIzquierdo(), que recibe como parámetro* una matriz numpy.zeros(). La **línea 5,** se declara 6 puntos inicializados en ceros. Desde la **línea 6-11,** cada variable declarada es asignada con un valor del parámetro de entrada a la función. En la **línea 12-14,** se realiza el análisis vectorial para calcular la distancia entre los puntos. A continuación, se describe el proceso de análisis vectorial.

Según (Alva, s.f), "un vector es una magnitud con dirección, sentido y modulo o longitud".En el plano cartesiano, un vector se puede representar tomando un punto de partida P (origen del vector), a partir de él, realiza un desplazamiento paralelo al eje x e y.

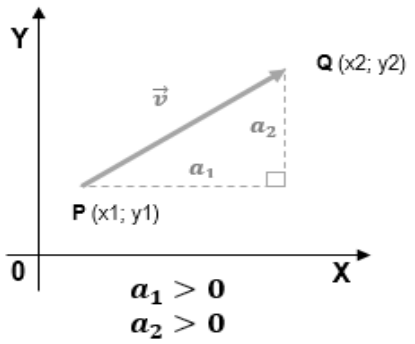


***Figura 26**: Representación de un vector en el plano cartesiano*
Fuente: (Alva, s.f)

En la figura anterior $\vec{v}$ es un vector de origen P(x1, y1) y extremo Q(x2, y2) con un desplazamiento hacia la derecho en el eje X y un desplazamiento hacia arriba en el eje Y. El sentido para $\vec{v}$ es de P hacia Q y se representa $\vec{v} = \overrightarrow{PQ}$. Las

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

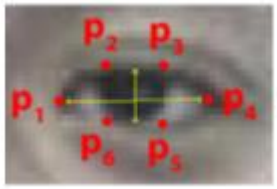
componentes de $\vec{v}$ son sus proyecciones (positivas o negativas, según sentido) sobre los ejes coordenados:

$$\vec{v} = (x2 - x1\,;\, y2 - y1)$$

El módulo, norma o longitud del vector $\vec{v}$ es la distancia entre un punto P y Q.

$$\overrightarrow{|a|} = \sqrt{(x2 - x1)^2 + (y2 - y1)^2} \qquad 5.3\ (1)$$

Para terminar, en la **línea 15**, se declara la variable ear_izquierdo que recibe un valor utilizando la formular de relación de aspecto del ojo EAR [Fig. 29], establecida por (Soukupová, Cech, 2016).



$$\text{EAR} = \frac{\|p_2 - p_6\| + \|p_3 - p_5\|}{2\|p_1 - p_4\|}$$

Figura 27:Relación de aspecto del ojo propuesto por (Soukupová, Cech, 2016)

La **línea 18-30,** ya no se entra en detalle, pues, tiene la misma forma de trabajo tal como se ha descrito anteriormente. En [Fig. 34], se muestra el resultado al detectar el parpadeo con su respectivo valor para cada estado de los ojos.

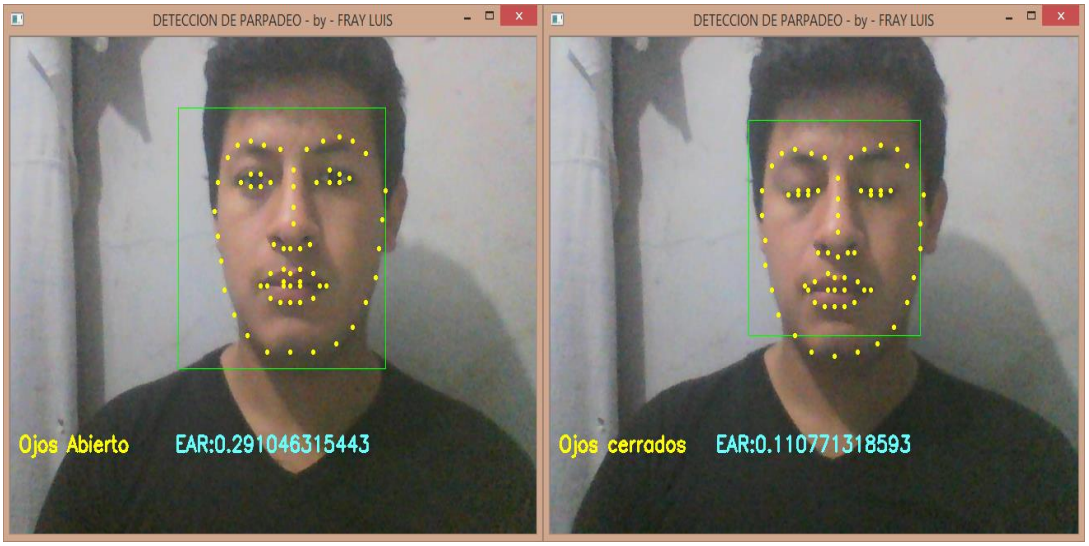


***Figura 28**: Detección de parpadeo en tiempo real con EAR*
Fuente: elaboración propia

Detección de boca abierta

La detección de boca abierta, el desarrollo este sujeto al algoritmo 03, que se muestra a continuación.

Algoritmo 03: Detector de boca abierta y boca cerrada

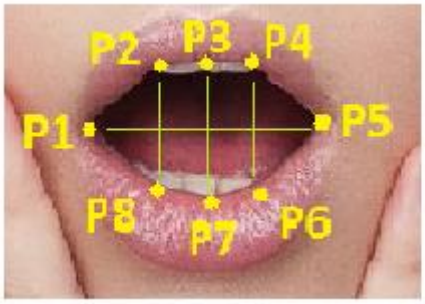
1. Inicio
2. Leer coordenadas faciales
3. Asignar valor: boca (p60, p61, p62, p63, p64, p65, p66, p67)
4. Calcular MAR
5. Si MAR > 0.30
   Estado = boca abierta
6. Else
   Estado = boca cerrada
7. fin
8.

Para evidenciar la implementación del algoritmo, en el mismo script "Coordenadas.py", se implementa la función *puntosBoca().*

```
41. def puntosBoca():
42.    p60, p61, p62, p63, p64, p65, p66, p67 = 0,0,0,0,0,0,0,0
```

```
43.    p60 = c[60]
44.    p61 = c[61]
45.    p62 = c[62]
46.    p63 = c[63]
47.    p64 = c[64]
48.    p65 = c[65]
49.    p66 = c[66]
50.    p67 = c[67]
51.
52.    x1 = dist.euclidean(p61, p67)
53.    x2 = dist.euclidean(p62, p66)
54.    x3 = dist.euclidean(p63, p65)
55.    x4 = dist.euclidean(p60, p64)
56.    mar  = (x1+x2+x3)/(3.0 * x4)
57.    return mar
```

La estructura de la función *puntosBoca()* es similar a la función *puntosOjoIzquierdo().* Por este motivo, no se entra en detalle a explicar línea por línea. Sin embargo, en la **línea 56,** teniendo como referencia el trabajo de (Soukupova, Cech, 2016), proponemos la relación de aspecto de boca [Fig. 33], que evalúa cuatro distancias en ancho y alto para determinar el estado de una boca abierta o cerrada.



$$MAR = \frac{|p2 - p8| + |p3 - p7| + |p4 - p6|}{3 * |p1 - p5|}$$

Figura 29: Relación de aspecto de la boca (MAR)
Fuente: elaboración propia

Detectar cuando una boca está abierta se puede establecer con un valor de 0.15. Este valor es aceptado cuando la detección de boca se realiza en una imagen estática. Sin embargo, cuando se trabaja en tiempo real, que es nuestro caso,

con un valor de 0.15 se obtiene una tasa de falso positivos demasiado alto. Para disminuir los falsos positivos es necesario incrementar el valor arrojado por MAR. Por ello se ha optado por trabajar con un valor de 0.30, [Fig.38].

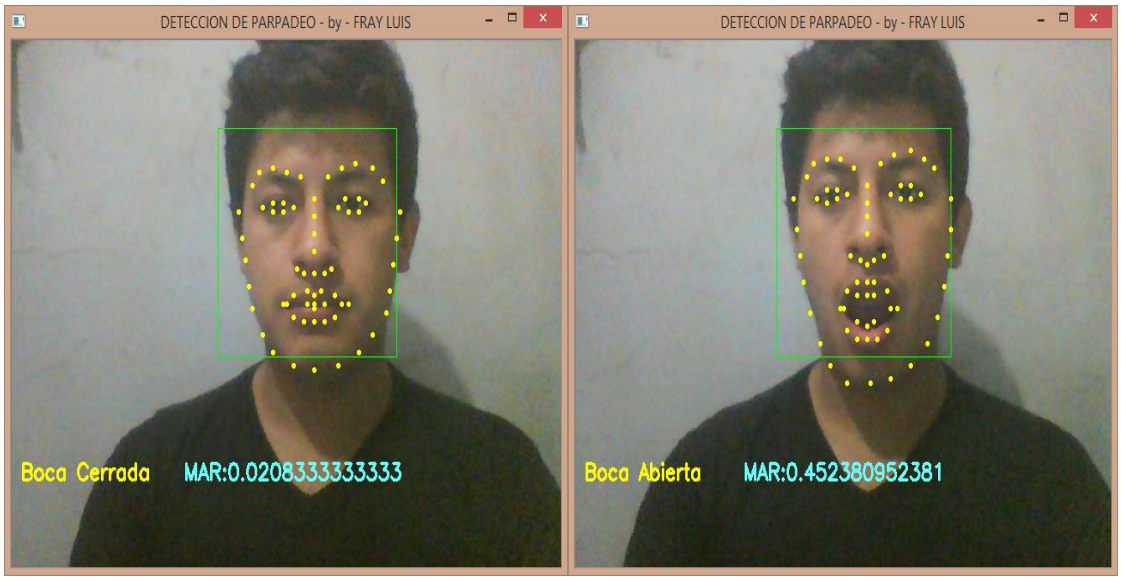


Figura 30: Detección de boca abierta en tiempo real con MAR.
Fuente: elaboración propia

Detección de Sonrisa

El estado de alegría (sonrisa) es una de las siete emociones universales (sorpresa, miedo, asco, ira, alegría, tristeza) definidas por Paul Ekman. Según (Castillo, 2015) define que el estado de alegría es una sensación dichosa de placer y bienestar. Esto se refleja cuando la comisura de los labios está hacia atrás y arriba, la boca puede estar abierta o no abierta, con o sin la exposición de los dientes, presencia de pata de gallo cuando las mejillas están levantadas.

Según la caracterización anterior, se define el siguiente algoritmo.

Algoritmo 04: Detector de sonrisa

1. Inicio
2. Leer coordenadas faciales
3. Establecer triangulación CHN
4. Calcular distancias de triangulación CHN

5. Calcular ángulo entre distancias
6. Si ángulo > 54°
   Estado = Sonriendo
7. Else
   Estado = Neutral
8. fin

Para la implementación del algoritmo 04, se sigue trabajando con el script "Coordenadas.py". Se ha implementado la función *puntosSonrisa(),* tal como se puede apreciar en el siguiente fragmento de código.

```
60.def puntoSonrisa©:
61.    p27x, p27y, p48x, p48y, p57x,p57y =0,0,0,0,0,0
62.    p27x = c[27][0]
63.    p27y = c[27][1]
64.    p48x = c[48][0]
65.    p48y = c[48][1]
66.    p54x = c[54][0]
67.    p54y = c[54][1]
68.    coordenada1 = (c[27][0],c[27][1])
69.    coordenada2 = (c[48][0],c[48][1])
70.    coordenada3 = (c[57][0],c[57][1])
71.    dist1 = dist.euclidean(coordenada1, coordenada2)
72.    dist2 = dist.euclidean(coordenada1, coordenada3)
73.    p1 = (p48x - p27x) *(p54x - p27x) + (p48y - p27y) *(p54y - p27y)
74.
75.    angulo = round(math.acos(p1/(dist1*dist2))*180/math.pi,2)
76.
77.    return angulo
```

En la **línea 60,** se declara la función *puntosSonrisa(),* que tiene como parámetro c, que representa un array numpy de coordenadas (x,y) de tipo entero con una longitud de 68.

Hasta el momento, tenemos los 68 puntos faciales del rostro, pero no sabemos que puntos se necesita para detectar la sonrisa. Según la caracterización de (Castillo, 2015), establecemos la ***triangulación CHN*** que consiste en tomar los dos puntos extremos de la boca (Chalion) y el punto medio entre las cejas

(Nasion). Recordar que estos términos están documentados en la sección ***2.3.2*** del presente proyecto. Según la documentación de (OpenFace, 2018), los puntos que vienen a representar la triangulación CHN es el punto 27, 48 y 54.

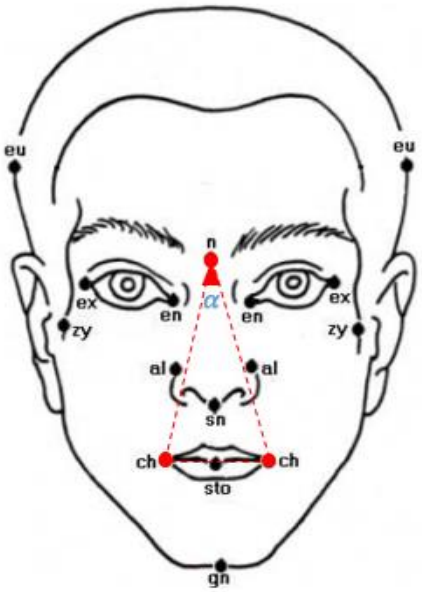


Figura 31: *Triangulación CHN de puntos para clasificar la emoción de alegría*

Por este motivo en la **línea 61,** declara 6 variables inicializadas es cero. Desde la **línea 62-67,** se asigna cada valor de las coordenadas x e y de cada punto definido. En la **línea 68-70,** se crea tres variables que representan las nuevas coordenadas de la triangulación CHN. Con estas nuevas coordenadas se realiza el cálculo de distancias, tal como se muestra en la **línea 71-72.**

La **línea 73,** se realiza el cálculo del producto escalar de vectores, también conocido como producto interno, de dos vectores $\vec{m} = (m1; m2)$ y $\vec{n} = (n1; n2)$ que se denota por $\vec{m}.\vec{n}$ y se evalúa como:

$$\vec{m}.\vec{n} = m1 * n1 + m2 * n2 \qquad 5.3\ (2)$$

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Por último, en la **línea 75** se realiza el cálculo cosenos para encontrar el ángulo entre dos vectores. A continuación, se describe la explicación propuesto por (Alva, s.f)

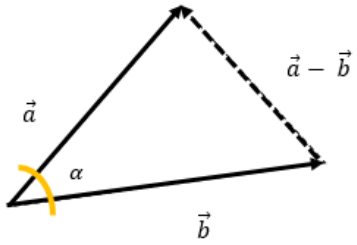


***Figura 32**: Representación de vectores*
Fuente: (Alva, s.f)

Sea α la medida del ángulo que determinan dos vectores $\vec{a}$ y $\vec{b}$. Entonces usando ley de cosenos entre las longitudes $|\vec{a}|$, $|\vec{b}|y\,|\vec{a}-\vec{b}|$ de los lados del triángulo tenemos:

$$|\vec{a}-\vec{b}|^2 = |\vec{a}|^2 + |\vec{b}|^2 - 2|\vec{a}||\vec{b}|Cos\alpha$$
$$(\vec{a}-\vec{b}).(\vec{a}-\vec{b}) = |\vec{a}|^2 + |\vec{b}|^2 - 2|\vec{a}||\vec{b}|Cos\alpha$$
$$\vec{a}.(\vec{a}-\vec{b}) - \vec{b}.(\vec{a}-\vec{b}) = |\vec{a}|^2 + |\vec{b}|^2 - 2|\vec{a}||\vec{b}|Cos\alpha$$
$$\vec{a}.\vec{a} - \vec{a}.\vec{b} - \vec{b}.\vec{a} + \vec{b}.\vec{b} = |\vec{a}|^2 + |\vec{b}|^2 - 2|\vec{a}||\vec{b}|Cos\alpha$$
$$|\vec{a}|^2 - \vec{a}.\vec{b} - \vec{a}.\vec{b} + |\vec{b}|^2 = |\vec{a}|^2 + |\vec{b}|^2 - 2|\vec{a}||\vec{b}|Cos\alpha$$
$$-2(\vec{a}.\vec{b}) = -2|\vec{a}||\vec{b}|Cos\alpha$$
$$Cos\alpha = \frac{\vec{a}.\vec{b}}{|\vec{a}||\vec{b}|} \Longrightarrow \alpha = Arcos\,\frac{\vec{a}.\vec{b}}{|\vec{a}||\vec{b}|}$$

Recordar que la **línea 75,** el valor del ángulo se transforma de radianes a grados y se redondea a dos decimales. Cuando el valor del ángulo es mayor a 54º, se considerad como una sonrisa y valores menores se considera como una expresión en estado neutro. (Fig. 38)

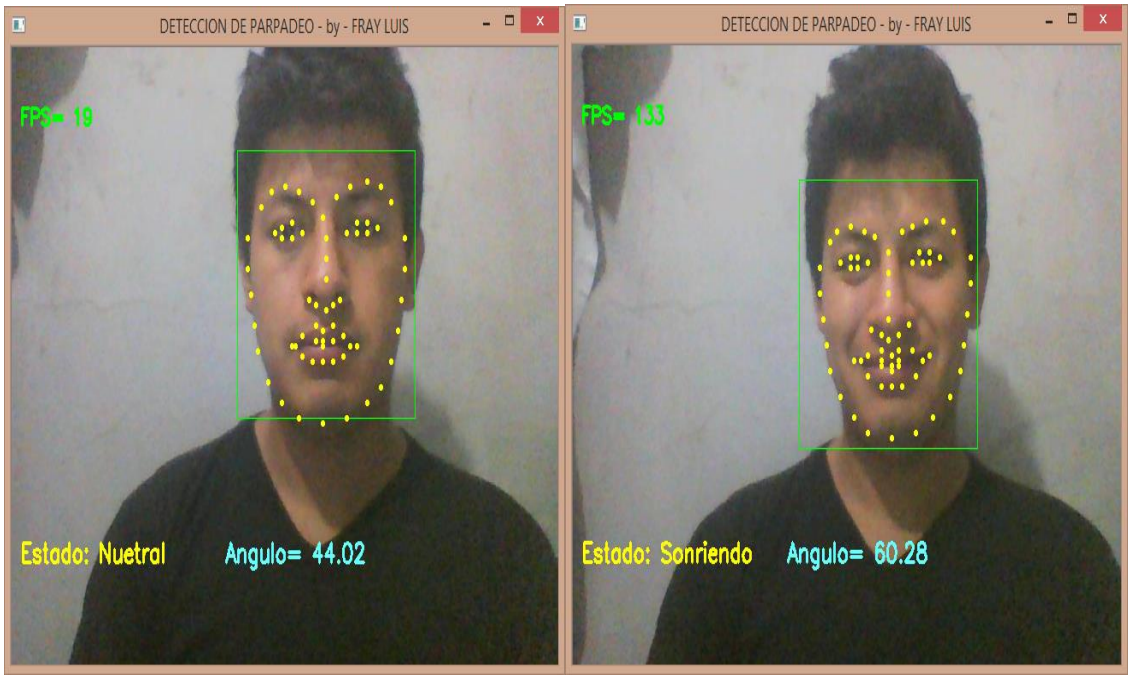


***Figura 33**:* Triangulación CHN para la detección de sonrisa
Fuente: elaboración propia

5.4 Sistema implementado

Para el desarrollar el prototipo, se ha utilizado en lenguaje de programación de Python 2.7, debido a que existe una mejor documentación para trabajar con las librerías de Opencv, Dlib y otras librerias para el tratamiento de datos.

Tabla 12: Software usado para la construcción del prototipo

DESCRIPCIÓN	VERSIÓN
Sistema operativo	Windows 8.1 S.L de 64 bits
Python	2.7
Opencv	2.4.6
Dlib	19.13
Tkinter (gui)	-
Imutils	-
Numpy	-
Pil (python imaging library)	- 1.1.7

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

El prototipo desarrollado consiste en dos ventanas. Primero se ha optado por el diseño del mockup, seguido por la descripción del caso de uso y por último, el desarrollo del mockup.

Ventana 01: Muestra información con la imagen del logo de la universidad donde el tesista a estudiado, así como también el nombre de la tesis y el autor

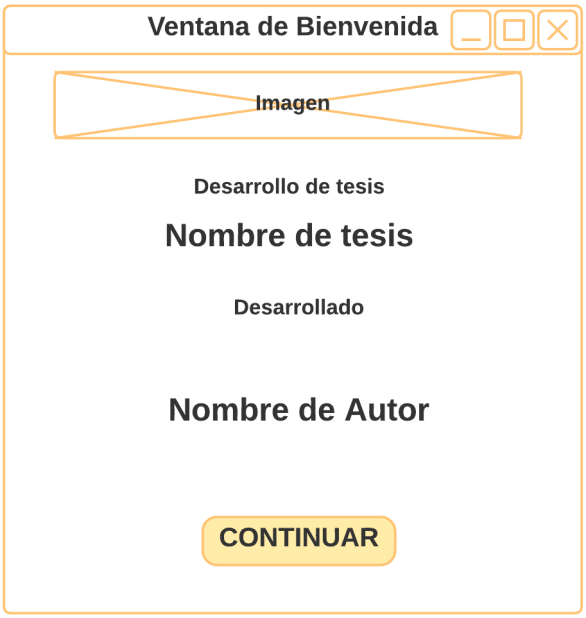


Figura 34: Mockup para la ventana 01
Fuente: elaboración propia

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Tabla 13: Descripción de caso de uso para la ventana de bienvenida

NOMBRE:	Ventana 01: ventana de bienvenida
AUTOR:	Fray Luis Becerra Suarez
FECHA:	22/05/2018
DESCRIPCION: Se muestra información, que describe el nombre y autor del proyecto.	
ACTORES: Usuario a quien se ha solicitado realizar test de prueba.	
PRECONDICION: Para ejecutar la aplicación, es necesario que se tenga instalado y configurado los programas y librerías para la ejecución de la aplicación.	
FLUJO NORMAL: 1. Al hacer click en el botón CONTINUAR, se accede a la siguiente ventana de la aplicación.	

Figura 35: Prototipo desarrollado para Ventana 01
Fuente: Elaboración propia

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Ventana 2: La presente ventana muestra los frames capturados por la cámara y los patrones de conducta facial implementados.

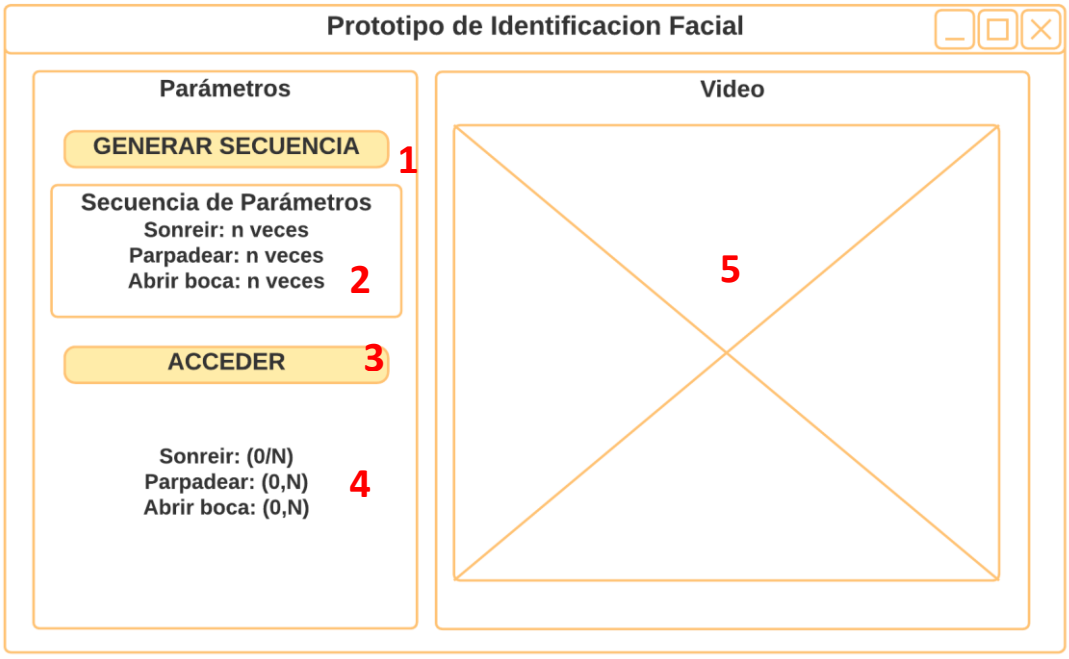


Figura 36: Mockup para la ventana de prototipo de identificación facial
Fuente: elaboración propia

Tabla 14: Descripción de caso de uso de la ventana de prototipo de identificación facial.

NOMBRE:	Ventana de Prototipo de identificación facial
AUTOR:	Fray Luis Becerra Suarez
FECHA:	22/05/2018
DESCRIPCION: Esta es la ventana principal de nuestra aplicación, siendo el resultado de la implementación de los algoritmos y métodos que se han descrito anteriormente.	
ACTORES: Usuario a quien se ha solicitado realizar test de prueba.	
PRECONDICION: Para ejecutar la aplicación, es necesario que se tenga instalado y configurado los programas y librerías para la ejecución de la aplicación.	
FLUJO NORMAL:	

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

1. Al hacer click en el botón GENERAR SECUENCIA, se genera un patrón y un valor aleatorio, que se actualiza en el punto 2 cada vez que se pulsa dicho botón. Esto se realiza con la finalidad de que no se tenga el mismo patrón al momento de hacer la validación.
2. En el botón ACCEDER, se valida el patrón que se ha generado anteriormente. Y en la etiqueta 4 se muestra el estado de la validación del patrón teniendo en cuenta la entrada de video que se muestra en la etiqueta 5.

   Luego de concluir con la validación, la aplicación genera un mensaje de alerta en caso de ser correcto la validación. De lo contrario, la aplicación se reiniciará en el punto 1.

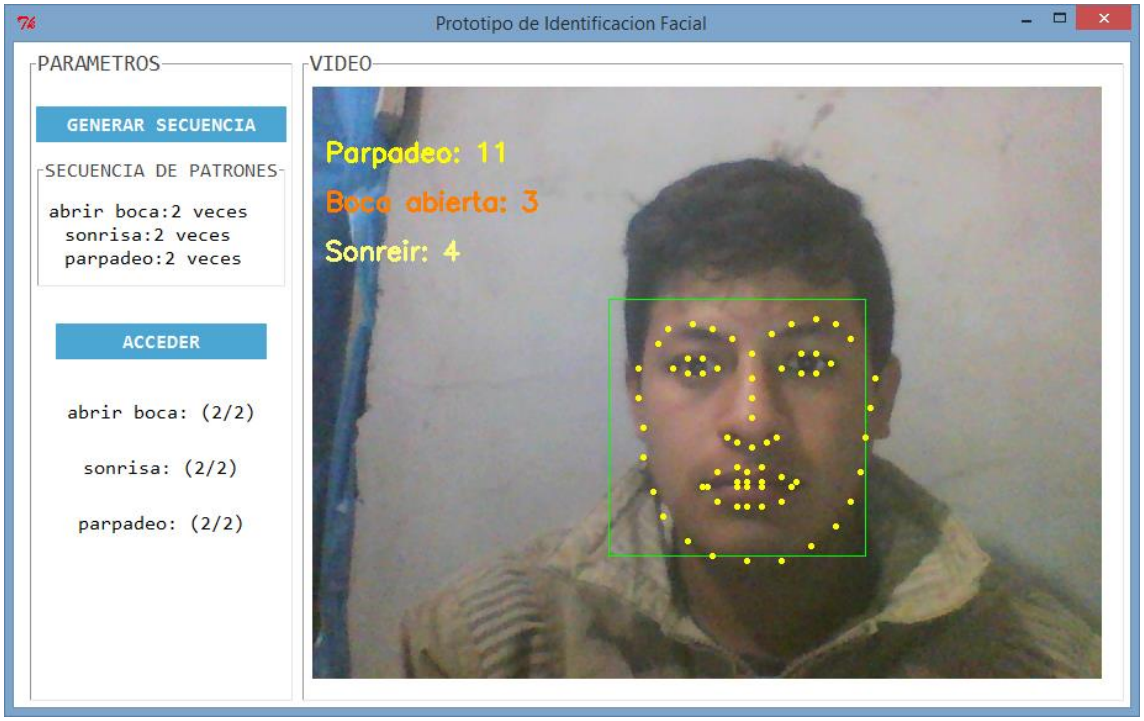


Figura 37: Prototipo desarrollado para Ventana 02
Fuente: Elaboración propia

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Capítulo VI: Conclusiones y Recomendaciones

6.1 Conclusiones

a) Para determinar los puntos de referencia facial, primeramente, se analizó la literatura de diferentes artículos científicos. métodos de detección de rostros, llegando a concluir que el algoritmo de Viola-Jones y HOG, son los que presentan mejores resultados.

   Para la extracción de puntos faciales se implementado el método propuesto por (Kazemi & Sullivan, 2014). Este método es un detector de punto faciales, entrenados en ambientes no controlados, presentan buenos resultados ante los cambios de iluminación y orientación del rostro. Por esta razón, se ha utilizado dicho método para llevar a cabo la detección de puntos faciales.

b) Se ha extraído un total de 68 puntos de referencia facial, de los cuales 10 puntos se extraen de ambas cejas, 12 puntos de ambos ojos, 9 puntos para la nariz, 19 puntos para la boca y 16 puntos para el contorno de la cara.

c) Son diferentes los patrones de conducta facial del complejo facial. Por ello, se ha optado por presentar atención 3 patrones conductuales: sonrisa, parpadeo de los ojos, abrir y cerrar de boca.

d) Se ha utilizado el lenguaje de programación Python 2.7 para el desarrollo del software. Así como también, librerías de procesamiento digital como Opencv 3.0, Dlib, PIL, para el tratamiento de imágenes en secuencia de

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

video. Y la arquitectura en el cual se han desarrollado las pruebas es x64 con un procesador AMD A8-6410.

e) Diferentes puntos se explican a continuación.

- La precisión para el algoritmo Hog + svm es de 1 frente al algoritmo de Viola Jones de 0.65. Así mismo, cuando se evalúa exactitud, el algoritmo Hog +svm es de 0.99 frente al algoritmo de Viola Jones que presenta una tasa de 0.65. Llegando a concluir el mejor algoritmo para detectar rostros es Hog +svm.

6.2 Recomendaciones

a) Implementar otros métodos de detección de rostros, por ejemplo, redes neuronales convoluciones multitask. También, si se quiere mejorar la detección de rostros cuando hay exceso de iluminación implementar método de patrones binarios locales.

b) Implementar otros patrones faciales como rotación del rostro y otros estados de ánimo aparte de sonrisa, para discernir entre rostro real y una fotografía.

c) Se debe analizar otras medidas para el patrón de conducta de la sonrisa, teniendo en cuenta los diferentes puntos faciales del rostro para disminuir la tasa de error. Esto se debe, a que, la triangulación CHN que se ha propuesto solo evalúa el Angulo entre los puntos de chalion y nassion.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

d) En caso de que se quiera utilizar el código fuente de esta aplicación, se debe optimizar para mejorar el refresco de cada frame en la aplicación. Y optar por trabajar en una arquitectura de procesamiento diferente a AMD.

e) Los sistemas de reconocimiento facial, se están volviendo muy común en nuestro día a día. Por ello, se recomienda hacer la implementación en la nube, para ver cual es el comportamiento que presenta en estos sistemas.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Referencias

Alshamsi, H., Kepuska, V., & Meng, H. (2017). *Automated Facial Expression Recognition App Development on Smart Phones using Cloud Computing .*

Calderon Castro, P. A., Tejada Castro, M. L., Yerovi Ricaurte, E. J., Ortega Ponce, L. X., & Espinoza Mina, M. A. (2016). *Reconocimiento facial: estudio bibliometrico de 20 años.*

Cama Castillo, Y. A. (2015). *Prototipo computacional para la detección y clasificación de expresiones faciales mediante la extracción de patrones binarios locales.* Lima.

Castillo Hilario, M. B. (2015). *Capacidad de codificación y decodificación facial deemociones en estudiantes universitarios de ingeniería.* Lima-Peru. Recuperado el 10 de abril de 2018, de http://cybertesis.unmsm.edu.pe/bitstream/cybertesis/4104/1/Castillo_hm.pdf.

Cheverria, M. (11 de 09 de 2016). *MindMeister*. Recuperado el 22 de 05 de 2018, de https://www.mindmeister.com/es/790554676/modelo-basado-en-prototipos

Coursera. (2018). *Deteccion de objetos.* Obtenido de www.coursera.org/learn/deteccion-objetos

Dalal, N., & Triggs, B. (2005). *Histograms of Oriented Gradients for Human Detection.* Recuperado el 2 de Abril de 2018, de http://lear.inrialpes.fr/people/triggs/pubs/Dalal-cvpr05.pdf

Eset. (2017). *Eset Security Report Latinoamérica.*

Garcia del Pardo, N., Gonzalez Castro, V., Alegre, E., & Fidalgo Fernandez, E. (2017). *Comporación de métodos de detección de rostros en imágenes digitales.* Recuperado el 4 de Abril de 2018, de http://pitia.unileon.es/varp/sites/default/files/JA2017_paper_21.pdf

Gee Sern, H., Hsiao Chia, P., & Kai Hsiang, C. (2014). *Landmark Based Facial Component Reconstruction for Recognition Across Pose.*

Juhong, A., & Pintavirooj, C. (2017). *Face recognition based on facial landmark detection.*

Kazemi, V., & Sullivan, J. (2014). *One Millisecond Face Alignment with an Ensemble of Regression Trees.*

Lopez Romero, W. L. (2016). *Sistema de Control del Estado de Somnolencia en Conductores de Vehículos.* Ecuador.

Pineda Cortés, L. A. (2017). *La Computación en México por especialidades acádemicas.* México: Acádemia Mexicana de Computación A.C.

Quintero Forero, A. X. (2016). *Extracción de características morfológicas de rostro humano a partir de imágenes.* Bogota D.C.

Rosebrock , A. (10 de Noviembre de 2014). *www.pyimagesearch.com*. Recuperado el 11 de Junio de 2018, de https://www.pyimagesearch.com/2014/11/10/histogram-oriented-gradients-object-detection/

Rosebrok, A. (s.f.). *Histogram of Oriented Gradients (and car logo recognition)*. Recuperado el 20 de 4 de 2018, de https://gurus.pyimagesearch.com/lesson-sample-histogram-of-oriented-gradients-and-car-logo-recognition/#

Sachin Sudhakar, F., Mohammand, S., & Li-Jia, L. (2015). *Multi-view Face Detection Using Deep Convolutional Neural Networks.*

Soukupova, T., & Cech, J. (2016). *Real Time Eye Blink Detection using Facial Landmarks.* Slovenia.

University, C. M. (2018). *OpenFace Documentation.*

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

ANEXO I
Algoritmos extracción de punto de referencia facial

AÑO	AUTOR	ALGORITMO
2014	Zhang, Z., Zhang, W., Liu, J., Tang, X.	Multi View
2014	Kazemi & Sullivan	Ensemble of Regression Trees
2013	Xiang Yu; Junzhou Huang; Shaoting Zhang; Wang Yan; Dimitris N. Metaxas	Cascade Deformable Shape model
2013	Tadas Baltrusaitis; Peter Robinson; Louis-Philippe Morency	Constrained Local Neural Field model (CLNF)
2013	Yi Sun, Xiaogang Wang, Xiaoou Tang	Deep Convolutional Network Cascade
2013	Belhumeur, P.N., Jacobs, D.W., Kriegman, D.J., Kumar, N.	Bayesian Approach
2013	Xuehan Xiong , de la Torre, F.	Supervised Descent Method
2012	Xudong Cao, Yichen Wei, Fang Wen, Jian Sun	Explicit Shape Regression
2012	Y Tong, X Liu, FW Wheelerb, PH Tub,	Semi-Supervised learning
2012	Dantone, M., Gall, J., Fanelli, G., Van Gool, L	Conditional Regression Forests
2012	(Sukno, et al, 2012	Combinatorial Search and Shape Regression
2012	Xiangxin Zhu, Ramanan, D	Tree-structured models
2011	Rapp, V., Senechal, T., Bailly, K., Prevost, L	Multiple Kernel Learning
2008	David Cristinacce, Tim Cootes	Constrained local models (CLM)
2008	De la Torre, F., Minh Hoai Nguyen	Parameterized Kernel Principal Component Analysis (PKPCA
2007	Yuchi Huang, Quinshang Liu, DimitrisMetaxas	Component based Deformable Mode

ANEXO II
Algoritmos De Detección De Rostro

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

AÑO	AUTOR	METODO O ALGORITMO
2017	Natalia Garcıa del Prado, Vıctor Gonzalez-Castro, Enrique Alegre, Eduardo Fidalgo Fernandez	Viola & Jones
2017	Natalia Garcıa del Prado, Vıctor Gonzalez-Castro, Enrique Alegre, Eduardo Fidalgo Fernandez	HOG +SVM en DLIB
2017	Natalia Garcıa del Prado, Vıctor Gonzalez-Castro, Enrique Alegre, Eduardo Fidalgo Fernandez	MTCNN
2003	Ueno, 2003	Segmentación por color de piel
S.F	Scott, Chun, Boonping, s.f)	Template matching + segmentación por color de piel

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

ANEXO III

Detección de rostro según el método de Viola & Jones y HOG +SVM

R: rostros detectados correctamente.

NR: rostros detectados incorrectamente.

RD: rostros que no han sido detectados.

	ALGORITMOS					
	Viola & Jones			HOG +SVM		
	R	NR	RD	R	NR	RD
img_1	1	0	0	1	0	0
img_2	1	0	0	1	0	0
img_3	1	0	0	1	0	0
img_4	1	0	0	1	0	0
img_5	0	1	1	1	0	0
img_6	1	0	0	1	0	0
img_7	1	2	0	1	0	0
img_8	1	0	0	1	0	0
img_9	1	0	0	1	0	0
img_10	1	0	0	1	0	0
img_11	1	0	0	1	0	0
img_12	0	2	1	1	0	0
img_13	1	0	0	1	0	0
img_14	1	3	0	1	0	0
img_15	1	0	0	1	0	0
img_16	1	0	0	1	0	0
img_17	1	0	0	1	0	0
img_18	1	0	0	1	0	0
img_19	1	0	0	1	0	0
img_20	1	0	0	1	0	0
img_21	1	2	0	1	0	0
img_22	1	0	0	1	0	0
img_23	1	0	0	1	0	0
img_24	0	0	1	1	0	0
img_25	1	0	0	1	0	0
img_26	1	2	0	1	0	0
img_27	1	0	0	1	0	0
img_28	0	0	1	1	0	0
img_29	1	0	0	1	0	0
img_30	1	0	0	1	0	0
img_31	1	1	0	1	0	0
img_32	1	0	0	1	0	0
img_33	1	0	0	1	0	0
img_34	1	0	0	1	0	0
img_35	1	0	0	1	0	0
img_36	0	0	1	0	0	1
img_37	1	0	0	1	0	0
img_38	1	0	0	1	0	0
img_39	1	0	0	1	0	0
img_40	1	0	0	1	0	0
img_41	0	0	1	1	0	0
img_42	1	0	0	1	0	0
img_43	1	0	0	1	0	0
img_44	1	0	0	1	0	0
img_45	1	0	0	1	0	0
img_46	1	0	0	1	0	0
img_47	0	4	1	1	0	0
img_48	1	0	0	1	0	0
img_49	1	3	0	1	0	0
img_50	1	0	0	1	0	0
img_51	1	0	0	1	0	0
img_52	0	0	1	1	0	0
img_53	1	0	0	1	0	0
img_54	1	0	0	1	0	0
img_55	1	0	0	1	0	0
img_56	1	0	0	1	0	0
img_57	1	0	0	1	0	0
img_58	1	0	0	1	0	0
img_59	1	0	0	1	0	0
img_60	1	3	0	1	0	0
img_61	1	1	0	1	0	0
img_62	1	0	0	1	0	0
img_63	1	0	0	1	0	0
img_64	1	1	0	1	0	0
img_65	1	0	0	1	0	0
img_66	1	0	0	1	0	0
img_67	1	2	0	1	0	0
img_68	1	0	0	1	0	0
img_69	1	0	0	1	0	0

img_70	1	0	0	1	0	0
img_71	0	1	1	1	0	0
img_72	1	0	0	1	0	0
img_73	1	0	0	1	0	0
img_74	1	0	0	1	0	0
img_75	1	1	0	1	0	0
img_76	1	0	0	1	0	0
img_77	1	0	0	1	0	0
img_78	1	0	0	1	0	0
img_79	1	2	0	1	0	0
img_80	1	0	0	1	0	0
img_81	1	0	0	1	0	0
img_82	1	0	0	1	0	0
img_83	1	0	0	1	0	0
img_84	1	0	0	1	0	0
img_85	1	0	0	1	0	0
img_86	1	0	0	1	0	0
img_87	1	0	0	1	0	0
img_88	1	0	0	1	0	0
img_89	1	0	0	1	0	0
img_90	1	0	0	1	0	0
img_91	1	0	0	1	0	0
img_92	1	0	0	1	0	0
img_93	1	0	0	1	0	0
img_94	1	0	0	1	0	0
img_95	1	0	0	1	0	0
img_96	1	0	0	1	0	0
img_97	1	0	0	1	0	0
img_98	1	0	0	1	0	0
img_99	1	0	0	1	0	0
img_100	1	0	0	1	0	0
img_101	1	0	0	1	0	0
img_102	1	0	0	1	0	0
img_103	1	1	0	1	0	0
img_104	1	0	0	1	0	0
img_105	1	1	0	1	0	0
img_106	1	0	0	1	0	0
img_107	1	2	0	1	0	0
img_108	1	0	0	1	0	0
img_109	1	1	0	1	0	0
img_110	1	1	0	1	0	0
img_111	1	0	0	1	0	0
img_112	1	0	0	1	0	0

img_113	1	0	0	1	0	0
img_114	1	0	0	1	0	0
img_115	0	1	1	1	0	0
img_116	1	0	0	1	0	0
img_117	1	0	0	1	0	0
img_118	1	0	0	1	0	0
img_119	1	0	0	1	0	0
img_120	1	0	0	1	0	0
img_121	1	0	0	1	0	0
img_122	0	0	1	1	0	0
img_123	1	1	0	1	0	0
img_124	1	0	0	1	0	0
img_125	1	0	0	1	0	0
img_126	1	0	0	1	0	0
img_127	1	1	0	1	0	0
img_128	1	0	0	1	0	0
img_129	1	0	0	1	0	0
img_130	1	0	0	1	0	0
img_131	1	1	0	1	0	0
img_132	1	0	0	1	0	0
img_133	1	0	0	1	0	0
img_134	1	0	0	1	0	0
img_135	0	0	1	1	0	0
img_136	1	0	0	1	0	0
img_137	1	0	0	1	0	0
img_138	1	0	0	1	0	0
img_139	1	0	0	1	0	0
img_140	1	0	0	1	0	0
img_141	1	0	0	1	0	0
img_142	1	0	0	1	0	0
img_143	1	1	0	1	0	0
img_144	1	0	0	1	0	0
img_145	0	0	1	1	0	0
img_146	1	0	0	1	0	0
img_147	1	2	0	1	0	0
img_148	1	0	0	1	0	0
img_149	1	0	0	1	0	0
img_150	1	0	0	1	0	0
img_151	1	0	0	1	0	0
img_152	0	0	1	1	0	0
img_153	1	1	0	1	0	0
img_154	1	0	0	1	0	0
img_155	1	2	0	1	0	0

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

img_156	0	0	1	1	0	0
img_157	1	2	0	1	0	0
img_158	1	0	0	1	0	0
img_159	1	0	0	1	0	0
img_160	1	1	0	1	0	0
img_161	1	0	0	1	0	0
img_162	1	0	0	1	0	0
img_163	1	1	0	1	0	0
img_164	1	0	0	1	0	0
img_165	0	1	1	1	0	0
img_166	1	0	0	1	0	0
img_167	1	1	0	1	0	0
img_168	1	1	0	1	0	0
img_169	1	0	0	1	0	0
img_170	1	2	0	1	0	0
img_171	1	1	0	1	0	0
img_172	0	1	1	1	0	0
img_173	0	1	1	1	0	0
img_174	1	0	0	1	0	0
img_175	1	1	0	1	0	0
img_176	1	0	0	1	0	0
img_177	1	0	0	1	0	0
img_178	1	0	0	1	0	0

img_179	1	2	0	1	0	0
img_180	0	0	1	1	0	0
img_181	1	0	0	1	0	0
img_182	1	0	0	1	0	0
img_183	1	0	0	1	0	0
img_184	1	0	0	1	0	0
img_185	0	1	1	1	0	0
img_186	1	1	0	1	0	0
img_187	1	0	0	1	0	0
img_188	1	3	0	1	0	0
img_189	1	0	0	1	0	0
img_190	1	0	0	1	0	0
img_191	1	2	0	1	0	0
img_192	1	0	0	1	0	0
img_193	1	1	0	1	0	0
img_194	1	0	0	1	0	0
img_195	0	0	1	1	0	0
img_196	1	2	0	1	0	0
img_197	1	1	0	1	0	0
img_198	1	0	0	1	0	0
img_199	1	1	0	1	0	0
img_200	1	0	0	1	0	0

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

ANEXO IV

Resultados obtenidos mediante EAR, para la detección de parpadeo

FPS	EAR	FPS	EAR	FPS	EAR
1	0.22222222	40	0.30516855	80	0.33333333
2	0.27709819	41	0.30555556	81	0.27709819
3	0.27827004	42	0.32638889	82	0.27709819
4	0.29617438	43	0.27391855	83	0.30470607
5	0.35867438	44	0.35923784	84	0.30470607
6	0.2907014	45	0.33784105	85	0.25498827
7	0.3036867	46	0.35416667	86	0.30555556
8	0.31006327	47	0.33784105	87	0.27709819
9	0.22154264	48	0.30555556	88	0.22086305
10	0.26150561	49	0.31006327	89	0.3036867
11	0.31006327	50	0.35867438	90	0.24847093
12	0.27777778	51	0.35416667	91	0.27777778
13	0.25	52	0.27405865	92	0.30470607
14	0.25655744	53	0.2623952	93	0.30921379
15	0.27475149	54	0.33333333	94	0.22222222
16	0.31006327	55	0.33029379	95	0.25583775
17	0.22810008	56	0.31006327	96	0.2769283
18	0.25655744	57	0.27777778	97	0.27777778
19	0.25583775	58	0.35867438	98	0.24932041
20	0.26311489	59	0.33784105	99	0.25
21	0.28228549	60	0.33784105	100	0.22086305
22	0.26736111	61	0.27709819	101	0.24932041
23	0.29513889	62	0.2409836	102	0.24847093
24	0.22810008	63	0.24932041	103	0.27709819
25	0.29513889	64	0.35416667	104	0.25
26	0.27777778	65	0.1385491	105	0.18701737
27	0.25655744	66	0.13888889	106	0.1758497
28	0.29513889	67	0.1758497	107	0.17635938
29	0.28228549	68	0.19977251	108	0.21802605
30	0.27391855	69	0.20413716	109	0.18951782
31	0.32291667	70	0.21802605	110	0.15552605
32	0.28228549	71	0.14484101	111	0.11771576
33	0.27777778	72	0.13007735	112	0.15504342
34	0.28228549	73	0.19444444	113	0.16108603
35	0.33333333	74	0.25	114	0.14544633
36	0.33784105	75	0.28228549	115	0.22222222
37	0.33333333	76	0.24932041	116	0.22154264
38	0.33784105	77	0.30470607	117	0.22154264
39	0.35923784	78	0.27777778	118	0.28228549
		79	0.30555556	119	0.27777778

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

120	0.27777778	162	0.12403473	204	0.27777778
121	0.27709819	163	0.16788764	205	0.28228549
122	0.30470607	164	0.14532364	206	0.24847093
123	0.24915052	165	0.14972317	207	0.11757292
124	0.30555556	166	0.12403473	208	0.19311273
125	0.24915052	167	0.22222222	209	0.20486111
126	0.27709819	168	0.25	210	0.25
127	0.30470607	169	0.22222222	211	0.27319886
128	0.22154264	170	0.30470607	212	0.249226
129	0.30470607	171	0.27607882	213	0.23514585
130	0.30555556	172	0.27709819	214	0.23543153
131	0.25	173	0.27709819	215	0.24266855
132	0.27777778	174	0.30470607	216	0.22805997
133	0.27777778	175	0.24847093	217	0.25498827
134	0.27709819	176	0.27709819	218	0.22877967
135	0.27777778	177	0.30470607	219	0.22738039
136	0.22222222	178	0.27709819	220	0.25655744
137	0.27777778	179	0.30470607	221	0.24266855
138	0.22154264	180	0.30470607	222	0.22222222
139	0.25	181	0.27709819	223	0.24847093
140	0.27709819	182	0.30453618	224	0.27709819
141	0.27709819	183	0.30921379	225	0.22086305
142	0.30470607	184	0.27607882	226	0.22222222
143	0.28228549	185	0.30470607	227	0.22086305
144	0.22154264	186	0.27709819	228	0.2409836
145	0.22154264	187	0.30555556	229	0.22222222
146	0.24847093	188	0.30470607	230	0.28228549
147	0.22222222	189	0.28160591	231	0.28160591
148	0.25	190	0.30470607	232	0.24194886
149	0.22222222	191	0.27709819	233	0.22877967
150	0.30921379	192	0.28160591	234	0.25498827
151	0.28055896	193	0.30470607	235	0.24847093
152	0.22877967	194	0.30816684	236	0.22738039
153	0.22222222	195	0.28228549	237	0.25655744
154	0.27607882	196	0.28228549	238	0.27777778
155	0.22222222	197	0.27777778	239	0.22877967
156	0.27709819	198	0.30453618	240	0.22222222
157	0.22154264	199	0.28228549	241	0.22222222
158	0.24932041	200	0.28160591	242	0.27777778
159	0.12451737	201	0.27607882	243	0.26388889
160	0.16788764	202	0.25587786	244	0.22222222
161	0.21875	203	0.28055896	245	0.27777778

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

246	0.22222222	248	0.28228549	250	0.28611362
247	0.22222222	249	0.23611111		

ANEXO V
Resultados obtenidos para la detección de boca abierta y cerrada

FPS	MAR				
1	0.027729678	37	0.041666667	75	0.041585524
2	0.020833333	38	0.047337521	76	0.020792762
3	0.020833333	39	0.0625	77	0.019574007
4	0.020833333	40	0.050296116	78	0.020833333
5	0.020833333	41	0.0625	79	0.020833333
6	0	42	0.020833333	80	0.041666667
7	0.0625	43	0.041666667	81	0.07099093
8	0.041666667	44	0.050296116	82	0.020833333
9	0.058823529	45	0.0625	83	0.050296116
10	0.022222222	46	0.050296116	84	0.041666667
11	0	47	0.041666667	85	0.041666667
12	0	48	0.058925565	86	0.071129449
13	0.058823529	49	0.041666667	87	0.041666667
14	0.039215686	50	0.041666667	88	0.041585524
15	0.020833333	51	0.075871412	89	0.071129449
16	0.0625	52	0.050296116	90	0.050296116
17	0.041666667	53	0.066666667	91	0.050296116
18	0.041666667	54	0.0625	92	0.041666667
19	0.020833333	55	0.039148015	93	0.039148015
20	0	56	0.0625	94	0.020833333
21	0.041666667	57	0.039215686	95	0.041666667
22	0.041666667	58	0.044444444	96	0
23	0.020833333	59	0.066666667	97	0.027729678
24	0.020833333	60	0.041666667	98	0.019574007
25	0.020833333	61	0.050198168	99	0.027681827
26	0.020833333	62	0.066666667	100	0.044346007
27	0.020833333	63	0.041585524	101	0.020792762
28	0.041666667	64	0.062378286	102	0.039215686
29	0.041666667	65	0.071129449	103	0.019574007
30	0.041666667	66	0.0625	104	0.019574007
31	0.029462783	67	0.020792762	105	0
32	0.050296116	68	0.041585524	106	0.019574007
33	0.020833333	69	0.041666667	107	0.039215686
34	0.050296116	70	0.020833333	108	0.044346007
35	0.041666667	71	0.075703369	109	0.044346007
36	0.020833333	72	0.041585524	110	0.066519011
		73	0.041666667	111	0.2
		74	0.044444444	112	0.288249046

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

113	0.314562907	157	0.019607843	201	0.490468174
114	0.35623525	158	0	202	0.490468174
115	0.35623525	159	0.019574007	203	0.466666667
116	0.406732441	160	0.019607843	204	0.428571429
117	0.406084899	161	0.039148015	205	0.400949157
118	0.405698812	162	0.020792762	206	0.314562907
119	0.430541965	163	0.019607843	207	0.125
120	0.382335883	164	0.019574007	208	0.020833333
121	0.422222222	165	0.019607843	209	0.073960026
122	0.423801507	166	0.019607843	210	0.058722022
123	0.423801507	167	0.027729678	211	0.063226177
124	0.454351489	168	0.019607843	212	0.018518519
125	0.500417141	169	0.039215686	213	0.018518519
126	0.430541965	170	0.019607843	214	0.041666667
127	0.467208861	171	0.047255834	215	0.019607843
128	0.454073043	172	0.058823529	216	0.058823529
129	0.501692091	173	0.071129449	217	0.058823529
130	0.453196845	174	0.083333333	218	0.039148015
131	0.405698812	175	0.145833333	219	0.058722022
132	0.429447829	176	0.291666667	220	0.058722022
133	0.454073043	177	0.291098669	221	0.0625
134	0.456322025	178	0.332595053	222	0.058722022
135	0.452919107	179	0.353476955	223	0.058823529
136	0.382335883	180	0.37694106	224	0.058722022
137	0.476668124	181	0.421287067	225	0.058722022
138	0.409090049	182	0.444444444	226	0.066829841
139	0.221730035	183	0.401839167	227	0.039215686
140	0.190880785	184	0.377777778	228	0.039215686
141	0.083333333	185	0.399114063	229	0.062378286
142	0.020833333	186	0.421287067	230	0.039215686
143	0.020833333	187	0.4	231	0.098039216
144	0.039215686	188	0.422222222	232	0.2
145	0.020833333	189	0.421287067	233	0.377777778
146	0.020833333	190	0.49872935	234	0.359159176
147	0.047337521	191	0.430541965	235	0.401839167
148	0.020833333	192	0.466666667	236	0.401839167
149	0.039215686	193	0.512494617	237	0.399114063
150	0.019607843	194	0.534716839	238	0.422222222
151	0.047337521	195	0.571428571	239	0.5
152	0.039215686	196	0.550793466	240	0.432512501
153	0.020833333	197	0.511111111	241	0.465633074
154	0.020833333	198	0.512494617	242	0.467208861
155	0.039215686	199	0.547619048	243	0.382335883
156	0	200	0.523809524	244	0.454351489

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

245	0.424957254
246	0.401839167
247	0.476190476
248	0.406732441
249	0.474980333
250	0.476945862
251	0.4
252	0.44346007
253	0.428571429
254	0.468245951
255	0.42406139
256	0.429447829
257	0.379616945
258	0.428571429
259	0.428571429
260	0.430541965
261	0.401839167
262	0.416666667
263	0.375
264	0.25
265	0.039215686
266	0.062378286
267	0
268	0.05364919
269	0.019607843
270	0.019607843
271	0.058823529
272	0.039215686
273	0.039148015
274	0
275	0.019607843
276	0.018518519
277	0.019607843
278	0.019607843
279	0.019607843
280	0.019607843
281	0.019607843
282	0.019607843
283	0.019607843
284	0.039215686
285	0.020833333
286	0.020833333
287	0.041666667
288	0.039215686
289	0.058823529
290	0.039148015
291	0
292	0
293	0
294	0.019607843
295	0.019607843
296	0.039215686
297	0
298	0.020833333
299	0.041666667
300	0.041666667
301	0.018490007
302	0.037037037
303	0.020833333
304	0.019574007
305	0.018518519
306	0.039215686
307	0.039215686
308	0.041666667
309	0.039215686
310	0.020792762
311	0.020792762
312	0.020792762
313	0.020792762
314	0.039148015
315	0.039215686
316	0.039215686
317	0.039215686
318	0.019607843
319	0.039215686
320	0.058823529
321	0.019607843
322	0.039215686
323	0.041585524
324	0.041666667
325	0.058722022
326	0.0625
327	0.019607843
328	0.039215686
329	0.039215686
330	0.058823529
331	0.058823529
332	0.039215686
333	0.041666667

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

ANEXO VI

Resultados obtenidos mediante la triangulación CHN para la detección de sonrisa.

FPS	ANGULO
1	45.82
2	48.06
3	44.95
4	46.3
5	44.98
6	45.22
7	46.26
8	48.06
9	44.98
10	44.95
11	44.98
12	44.98
13	44.98
14	47.34
15	45.84
16	48.35
17	47.34
18	48.64
19	47.09
20	48.64
21	47.09
22	47.09
23	47.09
24	48.64
25	48.64
26	48.06
27	45.65
28	48.06
29	48.06
30	44.95
31	48.06
32	48.06
33	49.7
34	50.71
35	53.68
36	57.97
37	54.76
38	57.97
39	60.56
40	59.44
41	63.47
42	61.41
43	61.41
44	61.41
45	64.02
46	61.41
47	62.48
48	61.41
49	63.63
50	61.41
51	62.48
52	62.53
53	60.71
54	62.53
55	62.53
56	62.48
57	62.48
58	62.44
59	62.48
60	56.86
61	52.19
62	48.09
63	48.06
64	46.77
65	44.95
66	47.46
67	46.8
68	44.95
69	44.95
70	45.65
71	44.95
72	46.77
73	46.77
74	46.77
75	44.95
76	48.06
77	44.95
78	45.69
79	45.65
80	45.69
81	47.43
82	48.09
83	45.69
84	48.06
85	45.69
86	45.65
87	47.46
88	47.46
89	48.09
90	49.73
91	47.46
92	45.69
93	48.09
94	48.09
95	48.09
96	45.69
97	49.7
98	50.45
99	53.68
100	56.86
101	61.59
102	62.5
103	62.5
104	65.36
105	67.04
106	65.95
107	67.21
108	65.04
109	68.33
110	65.09
111	63.63
112	64.02
113	67.09
114	64.02
115	62.48
116	67.09
117	65.58
118	67.21
119	65.43

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

120	64.65
121	63.47
122	66.67
123	59.16
124	52.19
125	48.09
126	48.09
127	49.7
128	49.06
129	49.7
130	44.95
131	45.69
132	48.09
133	45.69
134	48.11
135	48.11
136	45.69
137	45.65
138	47.46
139	44.41
140	45.69
141	48.09
142	48.06
143	49.06
144	48.06
145	48.06
146	48.09
147	48.09
148	48.09
149	48.09
150	48.09
151	49.08
152	49.06
153	51.52
154	53.68
155	52.22
156	55.23
157	57.97
158	58.1
159	60.56
160	64.37
161	63.63
162	64.37
163	64.02
164	62.48
165	62.48
166	64.37
167	63.15
168	63.63
169	63.63
170	64.37
171	64.37
172	62.48
173	61.37
174	62.48
175	64.37
176	62.48
177	63.15
178	56.86
179	53.68
180	50.74
181	48.09
182	49.7
183	46.8
184	46.8
185	49.7
186	48.35
187	46.8
188	46.8
189	48.09
190	49.7
191	48.09
192	48.09
193	48.09
194	49.7
195	48.09
196	48.09
197	48.35
198	48.35
199	48.35
200	48.35
201	48.35
202	49.7
203	45.84
204	48.35
205	48.35
206	48.35
207	48.35
208	48.35
209	48.35
210	48.35
211	48.35
212	47.38
213	47.38
214	46.63
215	46.63
216	47.38
217	47.38
218	46.63
219	48.35
220	46.8
221	46.77
222	44.98
223	49.06
224	45.69
225	45.69
226	47.37
227	48.37
228	48.06
229	48.09
230	49.7
231	48.06
232	48.06
233	50.02
234	47.66
235	50.02
236	47.75
237	47.75
238	47.05
239	48.8
240	49.7
241	49.7
242	48.06
243	48.06
244	48.09
245	49.7
246	47.32
247	48.06
248	49.06

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

249	49.7
250	47.09
251	49.47

252	51.08
253	50.48
254	48.1

255	46.26

ANEXO VII

Resultados obtenidos al evaluar MAR con valor de 0.1 y 0.3 para detectar boca abierta aplicado en 50 imágenes

FRAME	MAR >0.3	MAR >0.1
1	0	1
2	0	1
3	1	1
4	1	1
4	0	0
5	1	1
6	0	0
7	0	1
8	0	1
9	0	1
10	0	1
11	0	1
12	0	1
13	0	1
14	1	1
15	1	1
16	1	1
17	1	1
18	1	1
19	1	1
20	1	1
21	1	1
22	1	1
23	0	1
24	1	1
25	1	1
26	0	1
27	1	1
28	0	1
29	0	1
30	1	1
31	0	0
32	0	1
33	0	1
34	1	1
35	0	1
36	0	1
37	0	1
38	0	1
39	0	1
40	0	1
41	0	1
42	0	1
43	0	0
44	0	0
45	0	1
46	0	1
47	0	1
48	0	1
49	0	1
50	0	1

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

ANEXO VIII

Resultado de detección de sonrisa en imágenes únicamente con sonrisa y en imágenes con diferentes estados de ánimo.

Img	Rostros con diferente estado de animo	Rostros únicamente con sonrisa
1	0	1
2	1	1
3	1	1
4	0	1
4	0	1
5	0	1
6	0	1
7	1	1
8	0	1
9	0	1
10	1	1
11	0	1
12	0	0
13	0	1
14	1	1
15	1	1
16	1	1
17	1	1
18	0	1
19	0	1
20	0	1
21	0	1
22	0	1
23	0	1
24	1	1
25	1	1
26	1	1
27	1	1
28	0	1
29	0	1
30	1	1
31	1	1
32	0	1
33	1	1
34	1	1
35	0	1
36	1	1
37	1	1
38	1	1
39	0	1
40	0	1
41	0	1
42	0	1
43	0	1
44	0	1
45	1	1
46	0	1
47	0	1
48	0	1
49	0	1
50	0	