

Reconocimiento de hojas del árbol *Pithecellobium Samán* usando visión artificial

Víctor Solórzano Gracia / Pablo Jácome Ruiz

RESUMEN

Existen algoritmos que ayudan en el aprendizaje de los agentes artificiales, de los cuales se aplicará en el presente trabajo los conocidos como, clasificadores en cascada. Los objetivos del presente estudio son: implementar y evaluar el rendimiento de ocho clasificadores en cascada usando dos tipos de extractores de características en el reconocimiento de hojas del árbol *Pithecellobium Samán* también conocido como árbol de Samán. Para la implementación de los clasificadores en cascada, se usa la librería EmguCV junto con Visual Studio y a cada clasificador, se le suministran imágenes del objeto a reconocer e imágenes que no lo sean. Los clasificadores se evaluarán usando las siguientes curvas: ROC (*Receiver Operating Characteristic*) para calcular el área bajo la curva, DET (*Detection Error Tradeoff*) para evaluar las muestras positivas mal clasificadas y PR (*Precision and Recall*) para medir la precisión. Como resultado, el extractor más eficiente es el Local Binary Pattern (LPB).

PALABRAS CLAVE

Aprendizaje automático, clasificadores en cascada, Emgu CV, Haar, LBP, visión artificial, inteligencia artificial.

ABSTRACT

There are algorithms that assist in the learning of artificial agents, which will be applied in the present work known as cascade classifiers. The objectives of this study are: implementing and evaluating performance eight cascaded classifiers using two kinds of feature extractors to recognize *Pithecellobium* leaf tree also known as Saman Saman tree. For the implementation of cascade classifiers, EmguCV library and Visual Studio are used, for each classifier is supplied with images of object to be recognized and images that are not. The classifiers are evaluated using the following curves: ROC (*Receiver Operating Characteristic*) to calculate the area under the curve, DET (*Detection Error Tradeoff*) to assess positive samples misclassified and PR (*Precision and Recall*) to measure accuracy. As a result, the most efficient extractor Local Binary Pattern is (LPB).

KEYWORDS

Machine learning, classifiers cascade Emgu CV, Harr, LBP, computer vision, artificial intelligence.

Introducción

La tecnología moderna ha permitido crear y desarrollar robots, capaces de reemplazar las actividades llevadas a cabo por el humano, ya sean por limitaciones propias de los seres humanos o por la necesidad de tener mayor precisión en procesos internos de la industrialización de las empresa o de las actividades propias del hogar. Para que dichos robots puedan ser capaces de realizar actividades, deben tener una inteligencia artificial (Pruthi, 2012).

Dado el caso, los robots o también llamados agentes artificiales deben “aprender” el medio que los rodea por lo cual se han creado técnicas que permiten llevar a cabo, eficazmente, el aprendizaje automático (Pruthi, 2012).

En este estudio, se usará un tipo de algoritmo de aprendizaje automático llamado clasificador en cascada, junto a dos tipos de extractores de características, que permitirán obtener detalles importantes del objeto a reconocer, reduciendo es costo computacional.

Se implementarán 8 tipos de clasificadores que se les suministrarán imágenes de hojas del árbol de Samán e imágenes que no lo sean, con la finalidad de que el algoritmo pueda “etiquetar” lo que es una hoja de Samán con lo que no lo sea.

Para evaluar el rendimiento de los clasificadores en cascada, se hará un contraste de los resultados obtenidos, haciendo uso de las curvas: ROC, DET y PR, que permitirán la medición de sus rendimientos, con la finalidad de seleccionar que clasificador y tipo de extractor de características se puede alcanzar

una precisión mayor al 80% en el reconocimiento de hojas de Samán.

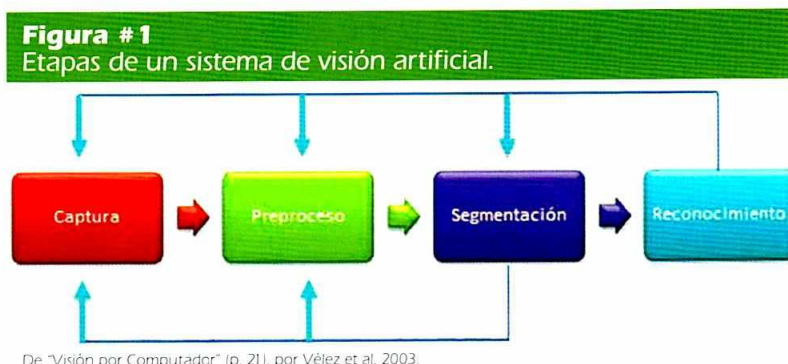
Marco Referencial

Inteligencia Artificial

La Inteligencia Artificial (IA) es el campo que se encarga de construir sistemas que muestren inteligencia, en el momento de realizar alguna actividad (Rossel & Norving, 2004).

Visión Artificial

La visión artificial tiene como finalidad la extracción de información del mundo real, para luego ser procesada por un computador (Vélez, Díaz, & Calle, 2003).



Interacción computador-humano

La interacción computador-humano, es el estudio que permite el diseño, la implementación y la evaluación de los llamados agentes artificiales, para que puedan interactuar con el humano (Hewett et al., 1992).

Los llamados robots sociales son uno de los tipos de robots que interactúan con el humano en el aprendizaje de las cosas y de su entorno. Estos robots tienen una extensa base de datos, que no se puede obtener de otro ser humano, debido a sus limitaciones de conocimiento, lo que permite que el ser humano, se sienta seguro con los robots sociales (Brown, 2013).

Por citar otro ejemplo, se ha desarrollado un robot que puede aprender las poses de

las personas, las cuales son captadas por la cámara de Microsoft Kinect. Se hicieron las pruebas con 24 personas para el aprendizaje del robot y los resultados arrojaron que con pocos ejemplos, el robot tenía una alta precisión y robustez en el reconocimiento de las poses. Además cuenta con un sistema de voz que le permite escuchar y responder a las personas (González-Pacheco, Malfaz, Fernández, & Salichs, 2013).

Aprendizaje automático

Consiste en las técnicas que se usan para que un agente artificial pueda entender el ambiente que lo rodea, a partir de ejemplos (de la Caridad Rivero-Hernández, Salgueiro-Sicilia, & Domínguez-López, 2012). Entre las técnicas del aprendizaje automático, podemos destacar las siguientes:

- **Supervisado:** Es una técnica, en la que se le suministran ejemplos, con su correspondiente etiquetación; es decir, a la clase a la que pertenecen (Okori & Obua, 2011). En este grupo se encuentra el clasificador en cascada, mismo que será usado en este estudio.
- **No supervisado:** Es una técnica, que tiene como entrada ejemplos sin etiquetar dividido en grupos, usando para el efecto, el análisis de conglomerado (Bowd et al., 2014).
- **Semi-supervisado:** Es una técnica, en la que utiliza en su entrenamiento, una porción de datos etiquetados y una gran cantidad de datos no etiquetados (Chun, Junan, Jiyang, Dongsheng, & Aixia, 2014).

Extracción de características

La extracción de las características de una imagen, tiene como objetivo encontrar patrones que representen a esa imagen, reduciendo su dimensión y eliminando infor-

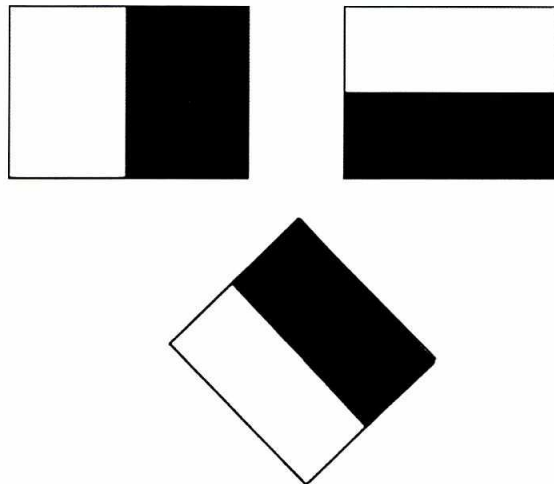
mación innecesaria, además del ruido, con la finalidad de permitir diferenciar el objeto que va a ser reconocido del entorno que lo rodea (Phinyomark, Phukpattaranont, & Limsakul, 2012).

En este trabajo se usaron dos tipos de extractores de características:

Características Haar

Estas características, se basan en los niveles de intensidad que tienen los píxeles de una imagen. La imagen es sometida a varias funciones, que son conocidas como las características de Haar (Essannouni & Aboutajdine, 2011). Sin embargo, esto supone un costo computacional que debe ser considerado al evaluar cada píxel de la imagen, por lo cual se introdujo un nuevo concepto llamado imagen integral, usado para el cálculo de estos rectángulos (Viola & Jones, 2004).

Figura #2
Tres tipos de características de Haar.



De "Robust Real-Time Face Detection", por Viola, P., & Jones, M. J., 2004, *International Journal Of Computer Vision*, 57(2), 137-154.

Como ejemplo, tenemos que se aplican las características de Haar, para la detección de caras. Se planteó un método para reducir la dimensión de las características de Haar para reducir el costo computacional y mejorar el tiempo de detección de caras (Hong, Xi-Jun, & Peng, 2013).

Por citar otro ejemplo, en el cual se usaron las características de Haar para la implementación de AAM (Active Appearance Model) que permite el seguimiento de caras, en tiempo real, siendo implementado y probado en un celular Nokia modelo N900 (Tresadern, Ionita, & Cootes, 2012).

Características LBP (Local Binary Pattern)

Local Binary Pattern o LBP es un tipo de extractor de textura de imágenes. El trabajo original se desarrollaba con 8 pixeles vecinos de un pixel, utilizando el pixel central como umbral. El algoritmo, básicamente trabaja de la siguiente manera: divide la imagen en pequeñas celdas y escoge el pixel del centro de cada celda, entonces compara la intensidad de color gris del pixel central, con la intensidad de color gris de sus pixeles vecinos. La comparación comienza con los pixeles que van de acuerdo a las manecillas del reloj, si el valor de intensidad de gris del pixel central es mayor o igual al valor de intensidad de gris del pixel que se está evaluando, se coloca un “1” en dicha celda; caso contrario, se coloca un “0”. (Suruliandi, Meena, & Reena Rose, 2012).

También se puede apreciar el uso de las características LPB en la clasificación de especies de bosques, de las cuales se extrajeron las características LBP de 2.240 micro imágenes, para obtener las características de texturas y luego poder ser clasificadas en dos clases. (*Hardwood y Softwood*) (Martins, Oliveira, Nisgoski, & Sabourin, 2013).

Clasificador en cascada

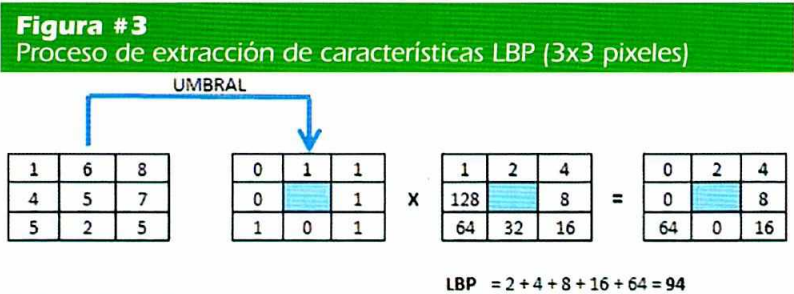
Está formado por los clasificadores débiles que se unen linealmente para formar un clasificador fuerte. Entonces se unen los clasificadores fuertes en forma de cascada (la salida de un clasificador fuerte es la entrada del siguiente clasificador fuerte y así sucesivamente), formando consecuentemente, un clasificador en cascada (Viola & Jones, 2004).

Después se reemplazó la combinación lineal de clasificadores débiles, de los que estaba conformado cada clasificador fuerte, por una combinación de árboles de decisiones débiles. Un clasificador en cascada, está dividido en etapas, que son los clasificadores fuertes. La imagen de entrada, es analizada en pequeñas sub-ventanas, por los árboles de decisiones débiles. Los clasificadores fuertes entrenan a los árboles de decisiones débiles, hasta alcanzar una mínima tasa de éxito deseada ,entre otros parámetros, para luego avanzar a la siguiente etapa del clasificador, así sucesivamente

hasta que las sub-ventanas son analizadas por todas las etapas del clasificador (Lienhart & Maydt, 2002).

Rendimiento de los clasificadores
Matriz de confusión

La matriz de confusión, es el resultado de comparar la clasificación sobre unas muestras realizadas por el clasificador, con las cla-



Fuente: Elaboración propia.

El uso de LBP se hizo sobre la clasificación de la morfología del cáncer de mama. De las imágenes obtenidas, se extrajeron las características LBP, con el fin de obtener y extraer la textura de la forma de dicho cáncer, usando posteriormente un clasificador en el que se entrenó para diferenciar tres clases de morfología de cáncer de mama (Ojansivu et al., 2013).

ses a las que pertenece cada muestra (Mendieta, 2013).

Tomando en cuenta que el clasificador de este estudio es binario (dos clases), los resultados del clasificador serán: positivos (T) o negativos (F). Pero los resultados del clasificador no siempre coinciden con la clase a la que pertenecen en la realidad, por lo que el clasificador, tiende a equivocarse (Mendieta, 2013).

En la realidad según Mendieta (2013), el resultado obtenido de, un clasificador binario, puede ser:

- Verdadero positivo (TP)
- Verdadero negativo (TN)
- Falso positivo (FP)
- Falso negativo (FN)

Figura #4
Matriz de confusión

		Realidad	
Predicción		TP	FP
		FN	TN

Fuente: Elaboración propia.

La matriz de confusión no es suficiente para poder medir el rendimiento del clasificador, por lo cual, se usan sus datos para calcular ciertos indicadores, que ayudarán a graficar las curvas y medir el rendimiento del clasificador (Mendieta, 2013).

Según Mendieta (2013), los indicadores son los siguientes:

- Tasa de verdaderos positivos (True Positive Rate) (TPR): También llamado sensibilidad. Este indicador mide el rendimiento del clasificador, en elegir las muestras positivas correctamente.
- Tasa de verdaderos negativos (True

Negative Rate) (TNR): También se lo conoce como especificidad. Este indicador, mide el rendimiento del clasificador, en elegir las muestras negativas correctamente.

- Tasa de falsos positivos (False Positive Rate) (FPR): También se lo conoce como fall-out. Determina el rendimiento del clasificador en elegir incorrectamente las muestras positivas.
- Valor predictivo positivo (Positive Predictive Value) (PPV): También llamado precisión. Indica la capacidad del clasificador, en elegir las muestras positivas, cuando en la realidad lo son.

Curva ROC

Es una gráfica para un clasificador binario que consiste en dibujar la tasa de falsos positivos con la tasa de verdaderos positivos, según vaya variando el umbral de decisión. El umbral de decisión va tomando de forma incremental las muestras, hasta llegar al total de muestras, logrando obtener todos los puntos de las curvas. Una vez dibujada la curva se la analiza trazando una diagonal en la gráfica, cuanto más elevada esté la curva sobre la diagonal, mejor será su rendimiento. Esto se obtiene cuando predomina la tasa de verdaderos positivos (Mendieta, 2013).

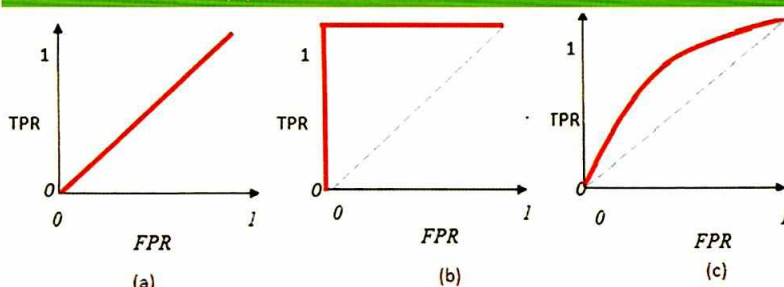
Una forma numérica de representar la curva ROC es calculando el área bajo la curva AUC (Area Under Curve), que es la probabilidad de que el clasificador etiquete mejor las muestras positivas como verdaderos positivos (Fogarty, Baker, & Hudson, 2005).

Otra forma numérica, es la Tasa de Error Igual (EER) (Equal Error Rate) y representa la tasa de error que corresponde a tener la misma probabilidad de una mala clasificación tanto positiva como negativa. Se calcula trazando una diagonal invertida sobre la curva ROC, la intersección entre la diagonal inver-

tida y la curva ROC es el valor de la tasa de error igual. Mientras menor sea este valor, mejor será el rendimiento del clasificador (Mendieta, 2013).

En la figura 5 se visualizan tres posibles escenarios de la curva ROC, siendo el mejor escenario el literal (c)

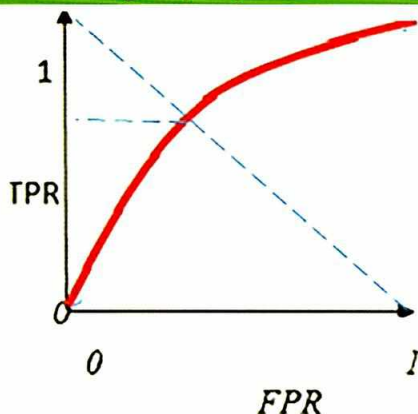
Figura #5
Tres escenarios de las curvas ROC (a) Mal clasificador (b) Clasificador ideal (c) Buen Clasificador



Adaptada de "Detección y reconocimiento de semáforos por visión artificial" (p. 39), por Mendieta, V., 2013.

En la figura 6 se observa como se traza la diagonal invertida y del punto que intersecta con la curva ROC, se calcula el valor en el eje de las "y".

Figura #6
Tasa de error igual (EER)



Adaptada de "Detección y reconocimiento de semáforos por visión artificial" (p. 40), por Mendieta, V., 2013.

Curva DET

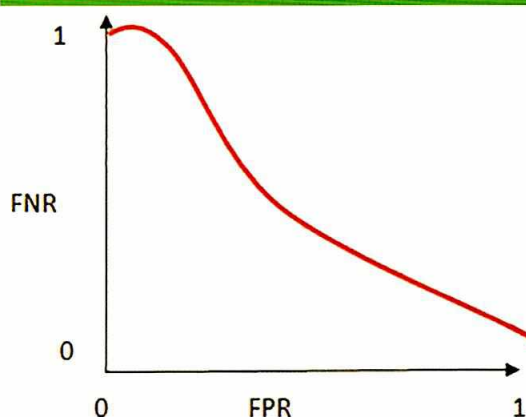
Es una gráfica para clasificadores binarios, en la que se dibuja la tasa de falsos negativos con la tasa de falsos positivos mientras se va variando el umbral de decisión. En esta gráfica se pueden evaluar las muestras positivas mal clasificadas por el clasificador. Este grafico

se puede interpretar como el coste de la mala clasificación de las muestras positivas y las muestras negativas. Mientras más cerca pase la curva por el punto (0,0) mejor será el rendimiento del clasificador (Mendieta, 2013).

Curva PR

Es otra forma de medir el rendimiento del clasificador binario. Se dibuja la tasa de verdaderos positivos, llamada en esta gráfica como *recall*, con el valor predictivo positivo llamado en esta gráfica, como precisión. El resultado de ésta gráfica permite saber el rendimiento del clasificador, al conservar la preci-

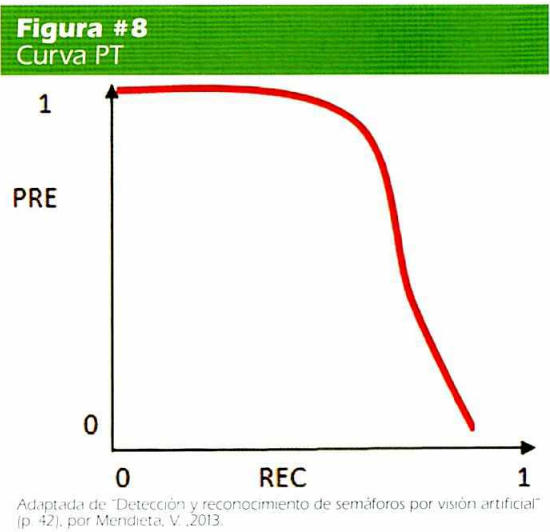
Figura #7
Curva DET



Adaptada de "Detección y reconocimiento de semáforos por visión artificial" (p. 41), por Mendieta, V., 2013.

sión en el momento de clasificar las muestras positivas, mientras va variando el umbral de decisión. Mientras más elevada y horizontal esté la curva, más preciso será el clasificador. También se la puede obtener de forma numérica, calculando la Precisión Media Interpolada de 11 puntos (API11) (Average Precision Interpolated in 11 points). Mientras más alto sea el valor de API11 con respecto a 0,5 mejor será la precisión del clasificador (Mendieta, 2013).

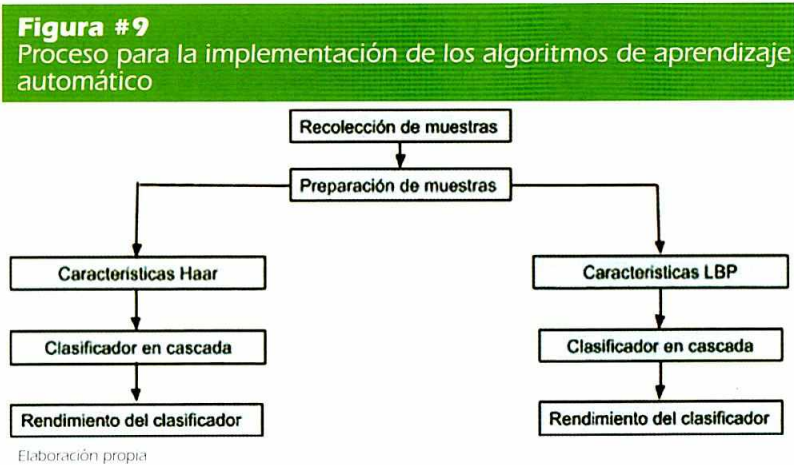
En la figura 8 se puede observar la curva PR y se puede visualizar que la curva se mantiene elevada y horizontalmente, por lo que se puede decir que el clasificador tiene una buena precisión.



namiento y 700 para las pruebas de rendimiento) de las hojas en diferente vista, y se han recolectado 6000 muestras negativas que contienen fondos en donde pueden estar las hojas de Samán (suelos, calles, veredas, etc.). Se quiere que este fondo no sea reconocido como muestras positivas. Las muestras recogidas tienen una resolución de 640x480 y 716x536 píxeles.



Metodología



Recolección de las muestras

Las muestras serán divididas en dos categorías: muestras positivas y muestras negativas. Debido a que las hojas de Samán tienen diferentes colores (verde, amarillo y café cuando se están marchitando) se eligió las hojas de color verde.

Las hojas de Samán tienen diferentes tamaños y formas, por tal motivo se han usado 4000 muestras positivas (3300 para el entre-



Preparación de muestras

Conversión RGB a grises

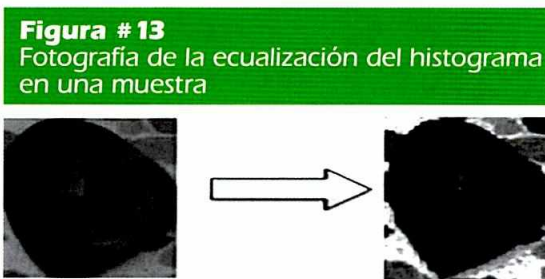
Usando la librería Emgu CV, primeramente, se convierten todas las muestras a escalas grises como se observa en la figura 12. Las muestras en escala grises tienen valores de píxeles entre 0 y 255. Trabajando en escala de grises se hace más fácil el proceso de extracción de características.



Fuente: Elaboración propia

Ecuilización del histograma

La ecualización del histograma consiste en aumentar o reducir las partes de la imagen donde las intensidades de los píxeles sean muy claras u oscuras, mejorando el contraste de la imagen.



Fuente: Elaboración propia

Entrenamiento del clasificador

Se va a reducir el tamaño de las muestras a 20x20 píxeles, debido a que sería un gran costo computacional trabajar con los tamaños originales de las muestras positivas. Luego se comienza a crear el clasificador en cascada. Se describirán solo los parámetros que se han usado para posteriormente, a través de la evaluación del algoritmo, saber cuáles son los parámetros óptimos para el reconocimiento de hojas de Saman. Los parámetros usados son los siguientes:

- Número de etapas que tendrá el clasificador
- Tipo de extractor de características: LPB, HAAR
- Mínima tasa de éxito que se desea alcanzar en cada etapa del clasificador

Rendimiento del clasificador

Para conocer el rendimiento del clasificador se realizarán los siguientes pasos:

1) Se usará una cantidad de muestras positivas (no deben ser las mismas muestras usadas para entrenar al algoritmo). Se usarán en total 700 muestras positivas. Del total de muestras se elegirá cierta cantidad de muestras a evaluar.

2) Se creará la matriz de confusión con los aciertos y fallidos que tuvo el clasificador para etiquetar las muestras a la clase a la que pertenecen.

3) Se calcularán los siguientes indicadores :

- Tasa de verdaderos positivos
- Tasa de verdaderos negativos
- Tasa de falsos positivos
- Valor predictivo positivo
- Valor predictivo negativo

4) Con los indicadores, se crean las tres curvas (ROC, RET, PR) para medir el rendimiento del algoritmo.

5) Se incrementa el número de muestras a evaluarse y se repiten todos los pasos anteriores, hasta que la cantidad de muestras evaluadas sean igual al total de muestras. Así se consiguen todos los puntos de las curvas.

Obtenidas las curvas se procede a analizarlas de acuerdo a la tabla #1 en la que se describe el tipo de prueba y el resultado esperado.

Tabla # 1

Pruebas con sus respectivos resultados esperados

Tipo de Prueba	Resultado Esperado
Cálculo de AUC en la curva ROC	El valor de AUC mayor a 0,5
Obtener coordenadas de FNR y FPR cuando FNR llega a ser menor 0,2	Mayor rango de FPR cuando FNR es menor a 0,2
Cálculo de APIII en la curva PR	El valor de APIII mayor a 0,5

Fuente: Elaboración propia

Se va a evaluar el tiempo de entrenamiento de los 8 clasificadores en cascadas divididos en dos grupos:

- Tiempo de entrenamiento de los 4 clasificadores en cascada usando LBP.
- Tiempo de entrenamiento de los 4 clasificadores en cascada usando Haar.

Resultados

Los clasificadores en cascada, que se generaron al variar los parámetros de entrenamiento de cada uno de ellos tales como: número de etapas del clasificador (16 y 18 etapas) y mínima tasa de éxito (0,95 y 0999) usando el extractor LBP y luego HAAR dando un total de 8 clasificadores en cascada. En la tabla 2, se pueden visualizar los diferentes tipos de clasificadores implementados para la ejecución de las pruebas.

En las figuras 14 a 21 se pueden observar los resultados de las curvas ROC, DET, PR de los 8 clasificadores en cascadas y en las figuras 22 y 23 se presentan los tiempos de entrenamiento de cada uno de los clasificadores en cascada.

Tabla # 2

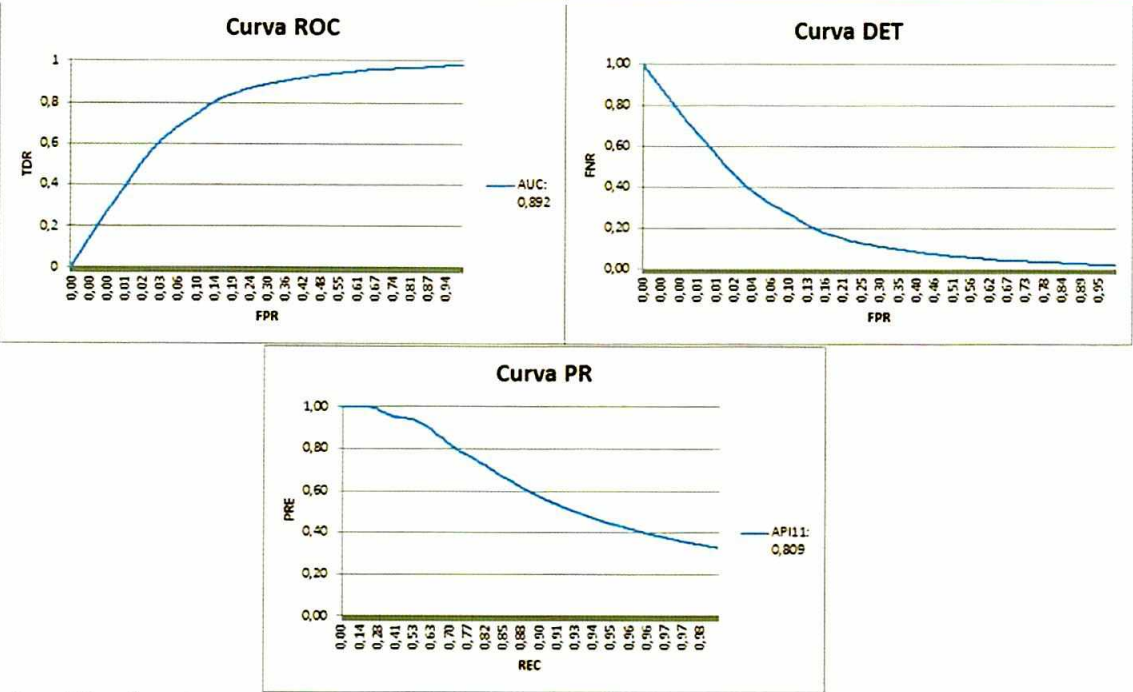
Los ocho clasificadores en cascada usando diferentes parámetros

Clasificador	Parámetros de entrenamiento
Clasificador 1	Clasificador LBP: • Parámetro mínima tasa de éxito: 0.999 • Parámetro número de etapas: 18
Clasificador 2	Clasificador Haar: • Parámetro mínima tasa de éxito:0.999 • Parámetro número de etapas: 18
Clasificador 3	Clasificador LBP: • Parámetro mínima tasa de éxito:0.95 • Parámetro número de etapas: 18
Clasificador 4	Clasificador Haar: • Parámetro mínima tasa de éxito:0.95 • Parámetro número de etapas: 18
Clasificador 5	Clasificador LBP: • Parámetro mínima tasa de éxito:0.999 • Parámetro número de etapas: 16
Clasificador 6	Clasificador Haar: • Parámetro mínima tasa de éxito:0.999 • Parámetro número de etapas: 16
Clasificador 7	Clasificador LBP: • Parámetro mínima tasa de éxito:0.95 • Parámetro número de etapas: 16
Clasificador 8	Clasificador Haar: • Parámetro mínima tasa de éxito:0.95 • Parámetro número de etapas: 16

Fuente: Elaboración propia

Resultado de clasificador 1 (LBP, tasa de éxito 0.999, 18 etapas).

Figura # 14
Curvas para evaluar el rendimiento del clasificador 1



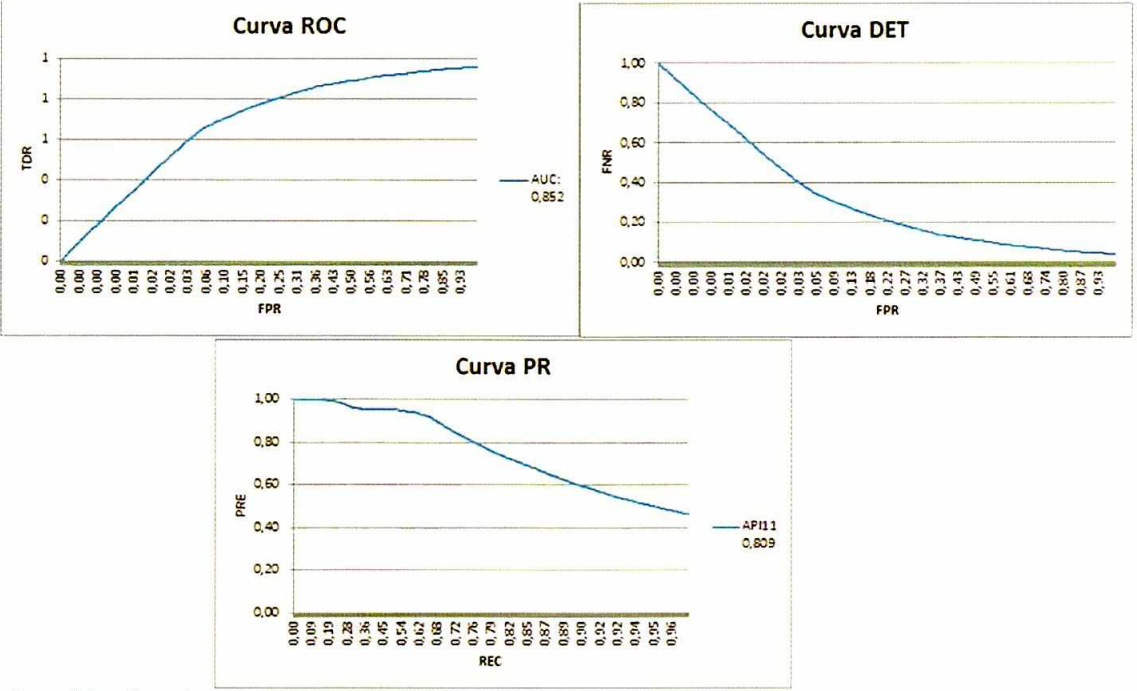
Fuente: Elaboración propia

Tabla #3	
Pruebas y resultados obtenidos del clasificador 1	
Tipo de Prueba	Resultado Esperado
Cálculo de AUC en la curva ROC	0,89
Obtener coordenadas de FNR y FPR cuando FNR llega a ser menor 0,2	0,84
Calculo de API11 en la curva PR	0,80

Fuente: Elaboración propia

Resultado de clasificador 2 (Haar, tasa de éxito 0.999, 18 etapas)

Figura #15
Curvas para evaluar el rendimiento del clasificador 2



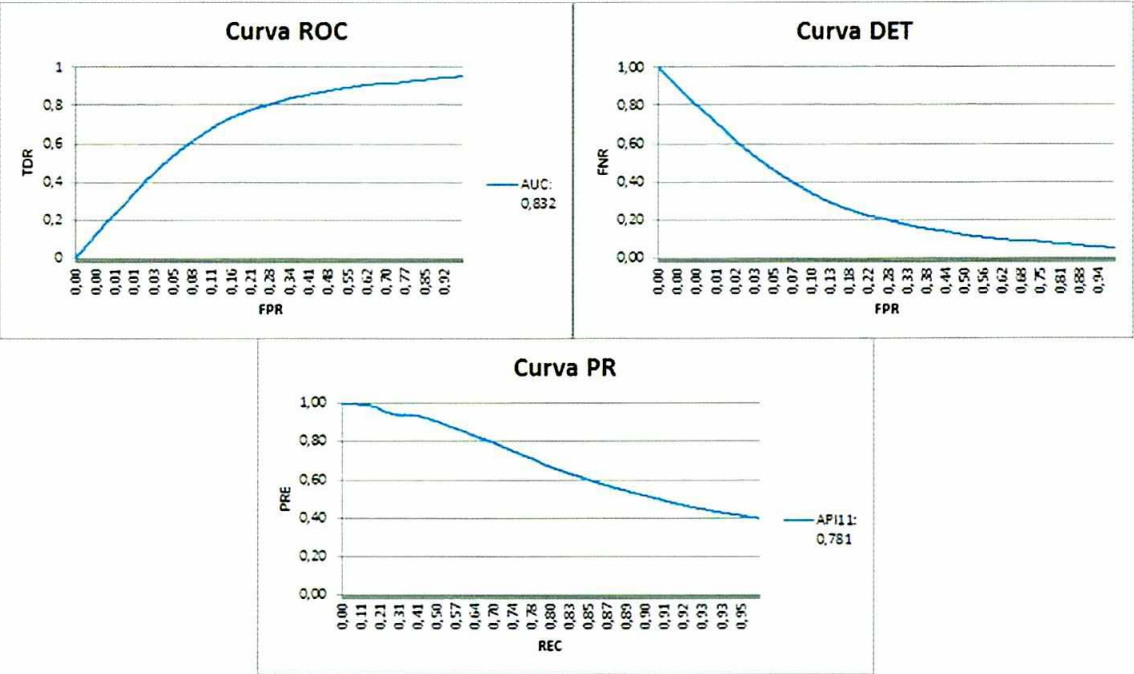
Fuente: Elaboración propia

Tabla #4	
Pruebas y resultados obtenidos del clasificador 2	
Tipo de Prueba	Resultado Esperado
Cálculo de AUC en la curva ROC	0,852
Obtener coordenadas de FNR y FPR cuando FNR llega a ser menor 0,2	0,73
Calculo de API11 en la curva PR	0,80

Fuente: Elaboración propia

Resultado de clasificador 3 (LBP, tasa de éxito 0.95, 18 etapas)

Figura #16
 Curvas para evaluar el rendimiento del clasificador 3



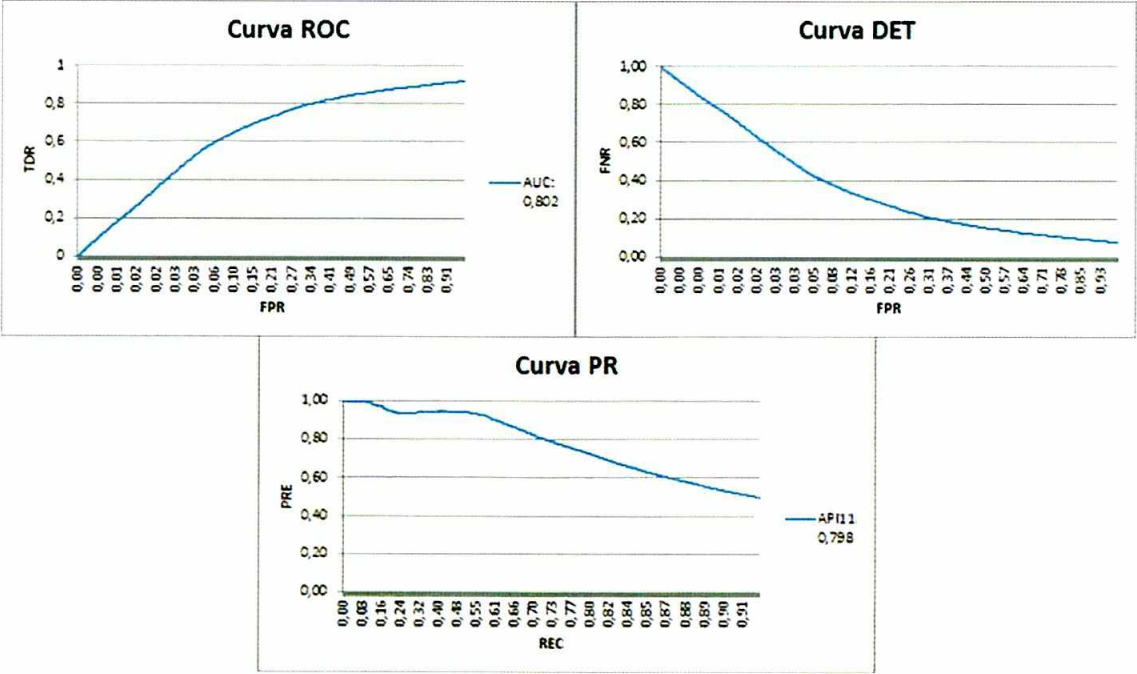
Fuente: Elaboración propia

Tabla #5	
Pruebas y resultados obtenidos del clasificador 3	
Tipo de Prueba	Resultado Esperado
Cálculo de AUC en la curva ROC	0,832
Obtener coordenadas de FNR y FPR cuando FNR llega a ser menor 0,2	0,78
Calculo de APIII en la curva PR	0,781

Fuente: Elaboración propia

Resultado de clasificador 4 (Haar,
tasa de éxito 0.95, 18 etapas)

Figura # 17
Curvas para evaluar el rendimiento del clasificador 4



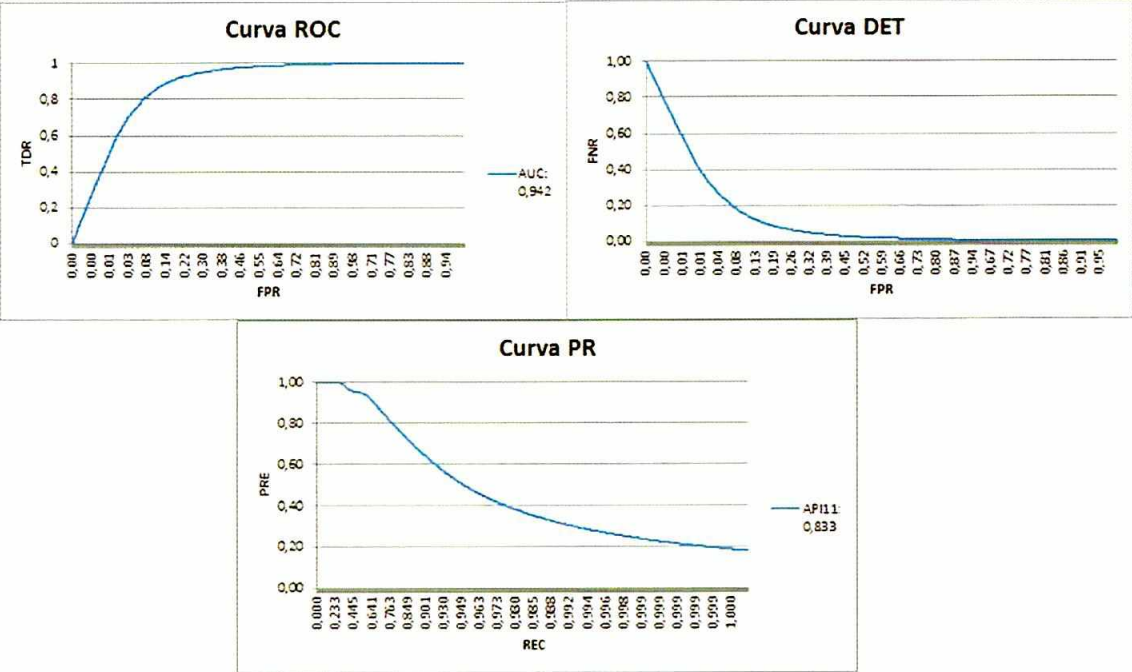
Fuente: Elaboración propia

Tabla #6	
Pruebas y resultados obtenidos del clasificador 4	
Tipo de Prueba	Resultado Esperado
Cálculo de AUC en la curva ROC	0,802
Obtener coordenadas de FNR y FPR cuando FNR llega a ser menor 0,2	0,69
Calculo de API11 en la curva PR	0,798

Fuente: Elaboración propia

Resultado de clasificador 5 (LBP, tasa de éxito 0.999, 16 etapas).

Figura #18
Curvas para evaluar el rendimiento del clasificador 5



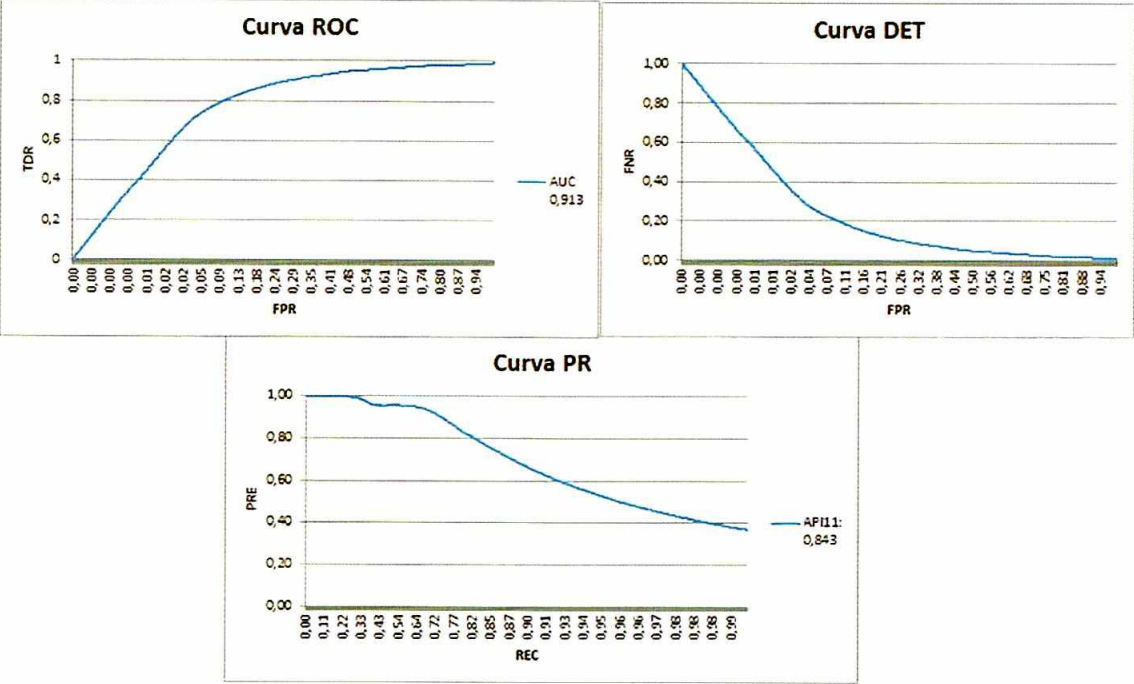
Fuente: Elaboración propia

Tabla #7	
Pruebas y resultados obtenidos del clasificador 5	
Tipo de Prueba	Resultado Esperado
Cálculo de AUC en la curva ROC	0,942
Obtener coordenadas de FNR y FPR cuando FNR llega a ser menor 0,2	0,92
Cálculo de APIII en la curva PR	0,833

Fuente: Elaboración propia

Resultado de clasificador 6 (Haar,
tasa de éxito 0.999, 16 etapas)

Figura # 19
Curvas para evaluar el rendimiento del clasificador 6



Fuente: Elaboración propia

Tabla # 8

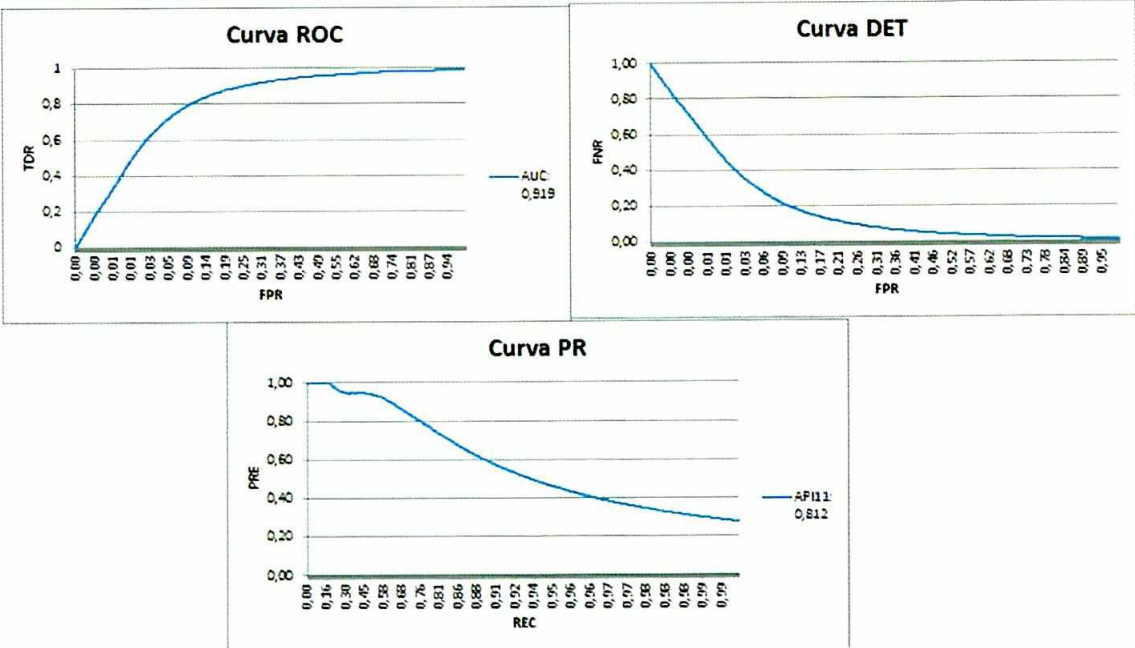
Pruebas y resultados obtenidos del clasificador 6

Tipo de Prueba	Resultado Esperado
Cálculo de AUC en la curva ROC	0,913
Obtener coordenadas de FNR y FPR cuando FNR llega a ser menor 0,2	0,92
Calculo de APIII en la curva PR	0,843

Fuente: Elaboración propia

Resultado de clasificador 7 (LBP, tasa de éxito 0.95, 16 etapas)

Figura #20
Curvas para evaluar el rendimiento del clasificador 7



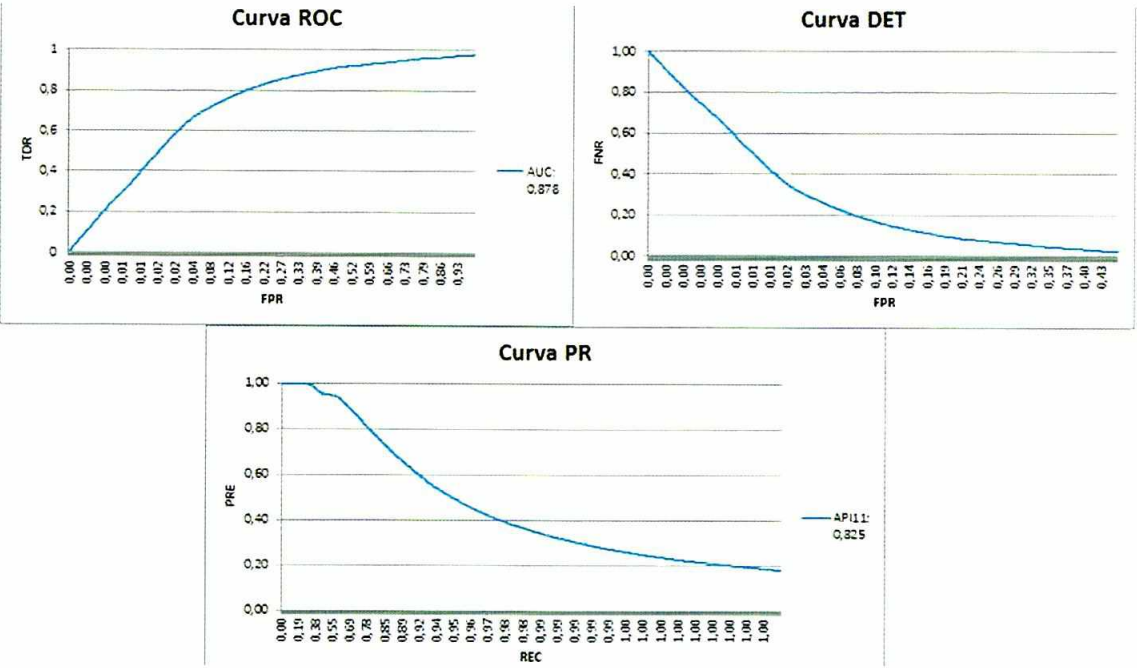
Fuente: Elaboración propia

Tabla #9	
Pruebas y resultados obtenidos del clasificador 7	
Tipo de Prueba	Resultado Esperado
Cálculo de AUC en la curva ROC	0,919
Obtener coordenadas de FNR y FPR cuando FNR llega a ser menor 0,2	0,92
Calculo de APII en la curva PR	0,812

Fuente: Elaboración propia

Resultado de clasificador 8 (Haar, tasa de éxito 0.95, 16 etapas)

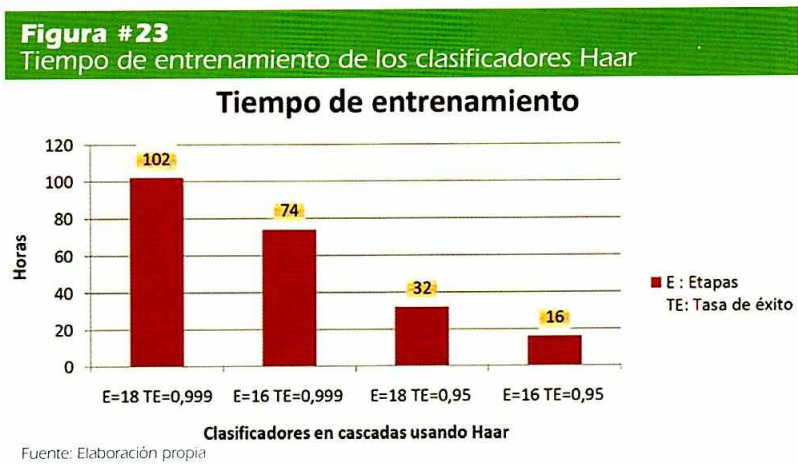
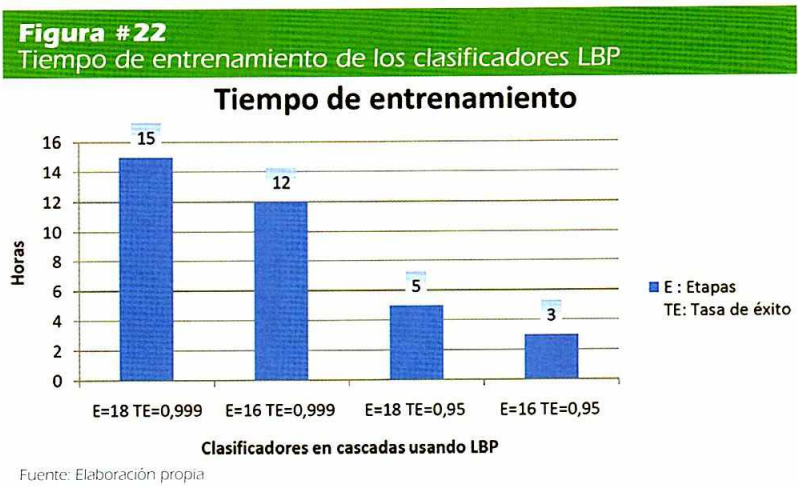
Figura #21
Curvas para evaluar el rendimiento del clasificador 8



Fuente: Elaboración propia

Tabla # 10	
Pruebas y resultados obtenidos del clasificador 8	
Tipo de Prueba	Resultado Esperado
Cálculo de AUC en la curva ROC	0,878
Obtener coordenadas de FNR y FPR cuando FNR llega a ser menor 0,2	0,92
Calculo de APIII en la curva PR	0,825

Fuente: Elaboración propia



Fuente: Elaboración propia



Fuente: Elaboración propia

Análisis de Resultados

Se puede notar, además, que sí se traza una diagonal sobre las curvas ROC, dichas curvas están por encima de la misma. Sí se quiere expresar, numéricamente, la curva ROC, el valor de AUC calculados en dichas curvas, son mayores a 0,5, entendiéndose que la tasa de verdaderos positivos predomina sobre la tasa falsos positivos.

Si se analiza las curvas DET, las figuras 18 a 21 presentan un total de rango de 0,92 y 0,93, lo que significa que el clasificador tiene pocos errores cuando se clasifican las muestras positivas, usando clasificadores con 16 etapas en vez de usar un clasificador con 18 etapas.

La curva PR se la puede analizar numéricamente viendo los valores de API11, demostrando la precisión de los clasificadores. Las figuras 18 y 19 muestran mejor precisión, mayor a 0,8 (igual al 80%). Se puede notar además, que la diferencia en precisión de la figura 18 y 19 es mínima, pero la mejor precisión la tiene la figura 19. Observando la figura 19 se puede notar que el decrecimiento de su curva PR es menor que el de la figura 18, esto significa que mientras más elevada se vea la curva, más preciso es el clasificador, así mismo, mientras más horizontal se mantenga su curva, más constante es el procesamiento de las muestras, lo que nos da como resultado que la figura 19 es la más precisa.

El número de etapas que posea un clasificador en cascada, influye en el rendimiento del mismo. A mayor número de etapas, el rendimiento va decreciendo, como se puede observar en las figuras 14 y 18. Esto se debe, a que mientras más etapas tenga un clasificador, más específico se torna; es decir, si no se evalúa una imagen que tenga las mismas características que las imágenes que se usaron para el entrenamiento, el clasificador en cascada no reconoce la imagen evaluada, en este caso, como una hoja de Samán.

Además, se puede visualizar en la gráfica del tiempo de entrenamiento que son las figuras 22 y 23, que LBP es mucho más rápido que Haar. Eso se debe que LBP trabaja con integrales que lo hacen más rápido frente a Haar que trabaja con números flotantes, lo que implica que LBP es más rápido en su entrenamiento para reconocer objetos.

Las figuras 18 y 19 representan la evaluación de los clasificadores 5 y 6 respectivamente y presentan el mejor rendimiento sus curvas en comparación a las demás por lo cual, ambos clasificadores son eficaces en el reconocimiento de hojas de Samán.

Aunque el clasificador 6 tiene una precisión de 0,84 (84%) (se puede visualizar en el valor de API11 de la figura 19), en contraste con el clasificador 5 (obsérvese el valor de API11 de la figura 18), que tiene una precisión de 0,83 (83%), se ha elegido usar el clasificador 5, para la detección de hojas de Samán, puesto que el tiempo de entrenamiento de LBP es menor a Haar.

El clasificador 5 definido en la tabla 2, que tiene el mejor resultado que se adecúa a futuros trabajos, tiene los siguientes parámetros:

- Extractor de características: LBP
- Número de etapas: 16
- Tasa de éxito: 0,999

Aunque se tiene un buen rendimiento evaluando los clasificadores 5 y 6 de las figuras 18 y 19 respectivamente, se tiene que analizar cuáles son los factores externos que afectaron al rendimiento de los clasificadores en cascada. Existen causas que no tienen relación con el algoritmo en sí, ni de sus extractores de características, ni de sus parámetros.

Las causas externas que afectaron al rendimiento del clasificador fueron las siguientes:

- El número de muestras positivas para el entrenamiento, depende de la complejidad del objeto a reconocer. La hoja del árbol de Samán tiene muchas formas y tamaños.
- Otro de los factores que redujo el porcentaje de detección, fue la iluminación que provoca cambios en la tonalidad de las muestras lo que provoca que el clasificador no pueda identificar bien una hoja de Samán.
- Como último factor tenemos el fondo que tenían las muestras positivas ya que lo ideal era usar fondos homogéneos, que estén libres de cualquier otra figura, puesto que el algoritmo también aprende del fondo de las muestras positivas. Esto se lo puede comprobar en la figura 24.

En la figura 25 se puede apreciar que solamente unas partes de la hoja, son reconocidas como muestras positivas. Para este estudio solo basta que se reconozca parte de la hoja, ya que se puede usar un sistema, que cuando detecte una hoja o parte de la misma, realice una acción, como la de limpieza.

Conclusiones

Los clasificadores 5 y 6 tienen una precisión de 83% y 84% respectivamente, superando el porcentaje de precisión del objetivo planteado que fue 80%, además de presentar los mejores rendimientos a diferencia del resto de clasificadores evaluados. Por lo tanto, ambos clasificadores en cascada son ideales para el reconocimiento de hojas de Samán pero se eligió el clasificador 5 que trabaja con LBP, ya que su entrenamiento es más rápido que el clasificador 6 que trabaja con Haar.

Adicionalmente, se encontró una dificultad al momento de detectar las hojas de Samán, cuando se trabajó con diferentes ambientaciones y condiciones que variaban en iluminación, sombras, colores, a más del

ruido que se introdujo en la captura de la imagen por parte de la cámara, influyendo negativamente en el rendimiento de los clasificadores en cascada. No se puede tener control sobre el ambiente en el que se trabaja pero si se puede mitigar sus efectos. Si bien, se usó el ecualizador de histograma para mitigar los efectos del ambiente, no fue suficiente, porque aún se tenían falsos positivos; por lo tanto, se deberán sumar otros métodos que no se propusieron en este estudio.

Se recomienda, tener cuidado en la aplicación de un clasificador que trabaje con Haar, ya que Haar maneja números flotantes y en un sistema que tenga limitaciones de memoria, sería un riesgo elevado, ya que consumiría muchos de los recursos del sistema en donde se desea ejecutar el clasificador.

En futuros trabajos se piensa usar el clasificador elegido, en un computador para un prototipo de robot aspiradora inteligente, el cual, a través de una cámara, capte una secuencia de imágenes que el clasificador en cascada deberá reconocer, siempre cuando, cada secuencia contenga hojas de Samán logrando así el inicio del proceso de limpieza. Entonces como la aplicación va a funcionar en tiempo real, se deberá realizar pruebas de tiempo, para evaluar cuánto tarda el clasificador en cascada para procesar cada fotograma proveniente de la cámara.

A pesar de que se alcanzó una precisión del 83% con el clasificador elegido, un futuro trabajo sería mejorar los métodos del preprocesamiento (suavizado, normalización, entre otros) en las muestras, para aumentar la robustez del clasificador frente a los cambios en la iluminación y el ruido en la escena. Además, se puede proponer métodos para elegir de una imagen, la zona de interés (cálculo de contornos del objeto a reconocer), delimitando los cálculos de detección.

Este estudio pretende ser la base para futuros trabajos, si se lo quiere orientar en otras áreas por lo que será recomendable, modificarlo para nuevos requerimientos. Por ejemplo, se puede aplicar este clasificador en cascada en la garita de entrada a una urbanización, para que el visitante, al ser observado por el lente de la cámara, el clasificador identifique si se trata de una cara o no lo que está observando. Entonces si el resultado del primer clasificador es una cara, se usa otro tipo de clasificador como SVM (Support Vector Machine), que permita identificar si esa cara o rostro, pertenece a un residente de la ciudadela, con la finalidad, de permitir su ingreso. Con estos dos clasificadores se filtran gran cantidad de falsos positivos. Este sistema o aplicación permitiría automatizar el proceso de ingreso de residentes a la ciudadela donde habitan.

Bibliografía

- Bowd, C., Weinreb, R. N., Balasubramanian, M., Lee, I., Jang, G., Yousefi, S., & Goldbaum, M. H. (2014). Glaucomatous Patterns in Frequency Doubling Technology (FDT) Perimetry Data Identified by Un-supervised Machine Learning Classifiers. *Plos ONE*, 9(1), 1-8. doi:10.1371/journal.pone.0085941.
- Brown, A. (2013). THE ROBOTIC MOMENT. *Mechanical Engineering*, 135(9), 32-37.
- Chun, Z., Junan, Y., Jiyang, Z., Dongsheng, L., & Aixia, Y. (2014). Semi-Supervised Learning by Local Behavioral Searching Strategy. *Applied Mathematics & Information Sciences*, 8(4), 1781-1787. doi:10.12785/amis/080435.
- de la Caridad Rivero-Hernández, D., Salgueiro-Sicilia, Y., & Domínguez-López, R. (2012). Evaluación de varias técnicas de aprendizaje automático en el Software Weka. (Spanish). *Innovación Tecnológica*, 18(3), 1-10.
- Essannouni, L. L., & Aboutajdine, D. D. (2011). Correlation of robust Haar-like feature. *Electronics Letters*, 47(17), 961-962. doi:10.1049/el.2011.1534.
- Fogarty, J., Baker, R.S., and Hudson, S.E. (2005). Case Studies in the use of ROC Curve Analysis for Sensor-Based Estimates in Human Computer Interaction. *Proceedings of Graphics Interface (GI 2005)*, pp. 129-136.
- González-Pacheco, V., Malfaz, M., Fernández, F., & Salichs, M. A. (2013). Teaching Human Poses Interactively to a Social Robot. *Sensors* (14248220), 13(9), 12406-12430. doi:10.3390/s130912406.
- Hewett, T. T., Baecker, R., Card, S., Carey, T., Jean, G., Mantei, M., . . . Verplank, W. (1992). *ACM SIGCHI Curricula for Human-Computer Interaction*. New York: ACM.
- Hong, Z., Xi-Jun, L., & Peng, Y. (2013). Research on Face Recognition Based on Embedded System. *Mathematical Problems In Engineering*, 1-6. doi:10.1155/2013/519074.
- Lienhart, R., & Maydt, J. (2002). An Extended Set of Haar-like Features for Rapid Object Detection. *IEEE ICIP*, 1, 900-903.
- Martins, J. J., Oliveira, L. L., Nisgoski, S. S., & Sabourin, R. R. (2013). A database for automatic classification of forest species. *Machine Vision & Applications*, 24(3), 567-578. doi:10.1007/s00138-012-0417-5.
- Mendieta, V. (2013). Detección y reconocimiento de semáforos por visión artificial. Universidad Carlos III de Madrid, España.

- Ojansivu, V., Linder, N., Rahtu, E., Pietikäinen, M., Lundin, M., Joensuu, H., & Lundin, J. (2013). Automated classification of breast cancer morphology in histopathological images. *Diagnostic Pathology*, 8(Suppl 1), 1-4. doi:10.1186/1746-1596-8-S1-S29.
- Okori, W., & Obua, J. (2011). SUPERVISED LEARNING ALGORITHMS FOR FAMINE PREDICTION. *Applied Artificial Intelligence*, 25(9), 822-835. doi:10.1080/08839514.2011.611930.
- Phinyomark, A., Phukpattaranont, P., & Limsakul, C. (2012). INVESTIGATING LONG-TERM EFFECTS OF FEATURE EXTRACTION METHODS FOR CONTINUOUS EMG PATTERN CLASSIFICATION. *Fluctuation & Noise Letters*, 11(4), -1. doi:10.1142/S0219477512500289.
- Poole, D., Mackworth, A., & Goebel, R. (1998). *Computational Intelligence : A logical Approach*. New York: Oxford University Press.
- Pruthi, S. (2012). Wireless Robotics: A History, an Overview, and the Need for Standardization. *Wireless Personal Communications*, pp 597-609.
- Rossel, S., & Norvig, P. (2004). *Inteligencia Artificial, un enfoque moderno*. Madrid: Prentice Hall.
- Serrano, J. F., Díaz, A. B., & Calle, Á. S. (2003). *Visión por computador*. Madrid: Universidad Rey Juan Carlos.
- Suruliandi, A. A., Meena, K. K., & Reena Rose, R. R. (2012). Local binary pattern and its derivatives for face recognition. *IET Computer Vision*, 6(5), 480-488. doi:10.1049/iet-cvi.2011.0228.
- Teahan, W. J. (2010). *Artificial Intelligence – Agents and Environments*. BookBoon.
- Tresadern, P. P., Ionita, M. M., & Cootes, T. T. (2012). Real-Time Facial Feature Tracking on a Mobile Device. *International Journal Of Computer Vision*, 96 (3), 280-289. doi:10.1007/s11263-011-0464-9.
- Viola, P., & Jones, M. J. (2004). Robust Real-Time Face Detection. *International Journal Of Computer Vision*, 57 (2), , 137-154. .

Victor Solórzano Gracia

Ingeniero en Telecomunicaciones de la Universidad Espíritu Santo - Ecuador

E-mail: victors0991@gmail.com

Pablo Jácome Ruiz

Ingeniero en Electrónica y Telecomunicaciones de la Escuela Superior Politécnica Del Litoral (ESPOL).

Realizando estudios para la obtención de título Magíster en Ciencias de la Ingeniería, mención Ingeniería Eléctrica en la Universidad de Chile.

Universidad Espíritu Santo - Ecuador.

E-mail: psjacome@gmail.com