

SISTEMA DE RECONOCIMIENTO FACIAL CON REDES NEURONALES PARA LA
TOMA DE ASISTENCIA EN AULAS DE CLASE

MIGUEL EUGENIO JURADO GARCIA

ANDRÉS FELIPE PADILLA PORRAS

UNIVERSIDAD AUTONOMA DE BUCARAMANGA

FALCULTAD DE INGENIRIA

PROGRAMA DE INGENIERIA DE SISTEMAS

BUCARAMANGA 2018

SISTEMA DE RECONOCIMIENTO FACIAL CON REDES NEURONALES PARA LA
TOMA DE ASISTENCIA EN AULAS DE CLASE

MIGUEL EUGENIO JURADO GARCIA

ANDRÉS FELIPE PADILLA PORRAS

TESIS

Director: René Lobo

UNIVERSIDAD AUTONOMA DE BUCARAMANGA

FALCULTAD DE INGENIRIAS

PROGRAMA DE INGENIERIA DE SISTEMAS

BUCARAMANGA 2018

DEDICATORIA

A nuestros padres, aquellos que son nuestra ayuda incondicional para nuestros sueños y metas.

A nuestro tutor René por guiarnos por el camino correcto y apoyar nuestro proyecto.

CONTENIDO

1.	INTRODUCCIÓN	4
2.	PLANTEAMIENTO DEL PROBLEMA	5
3.	PLANTEAMIENTO DE LA SOLUCIÓN	6
4.	OBJETIVOS	8
	4.1. OBJETIVO GENERAL.....	8
	4.2. OBJETIVOS ESPECIFICOS.....	8
5.	RESULTADOS ESPERADOS	8
	5.1. Objetivo específico 1	8
	5.2. Objetivo específico 2	8
	5.3. Objetivo específico 3	8
	5.4. Objetivo específico 4	9
	5.5. Objetivo específico 5	9
6.	ESTADO DEL ARTE	10
7.	MARCO TEORICO	22
	7.1. Framework	22
	7.2. Red neuronal.....	22
	7.3. CNN	22
	7.4. Darknet.....	23
	7.5. Código QR.....	23
	7.6. Zigbee	24
	7.7. Minucia	24
	7.8. Haar Features	24
	7.9. Viola Jones.....	26
	7.10. PCA (Principal Component Analysis).....	26
	7.11. LDA (Linear Discriminant Analysis).....	27
	7.12. DCT (Discrete Cosine Transform) por bloques.....	27
	7.13. Raspberry	28
8.	METODOLOGÍA	29
9.	RESULTADOS OBTENIDOS	31
	9.1. Objetivo específico 1	31

9.2.	Objetivo específico 2	36
9.3.	Objetivo específico 3	39
9.4.	Objetivo específico 4	44
9.5.	Objetivo específico 5	45
10.	Conclusiones	53
11.	REFERENCIAS	56
12.	Anexos	59

LISTA DE TABLAS

Tabla 1: Revisión de la literatura sobre proyectos en los que se aborda el tema de toma de asistencia con diferentes tecnologías	13
Tabla 2: Características para nuestro caso de estudio	33
Tabla 3: Posibles modelos y sus características	34
Tabla 4: Métricas de evaluación del modelo	47
Tabla 5: Valores finales de las métricas.....	50

LISTA DE FIGURAS

Figura 1: Resultado de la pregunta 3 encuesta realizada a docentes	5
Figura 2: Arquitectura propuesta para el sistema.....	6
Figura 3: Arquitectura de un sistema que utiliza la tecnología Zigbee.....	10
Figura 4: Hardware del sistema de reconocimiento por huellas dactilares	10
Figura 5: Aplicación móvil y código QR.....	11
Figura 6: Detección y recorte de caras	11
Figura 7: Arquitectura del sistema que utiliza dos cámaras	12
Figura 8: Método de clasificación de piel	12
Figura 9: Arquitectura de un SRF	14
Figura 10: Eigenfaces en una cara	15
Figura 11: Fisherfaces	16
Figura 12: Arquitectura de Darknet Real Time Grasp Detection Networks.....	18
Figura 13: Puntos de referencia.....	20
Figura 14: Comparación de servicios.....	21
Figura 15: Arquitectura de una Red Neuronal Convolutiva.....	23
Figura 16: Códigos QR	24
Figura 17: Ejemplo de la “ <i>Two-rectangle feature</i> ”	25
Figura 18: Ejemplo de la “ <i>Three-rectangle feature</i> ”	25
Figura 19: Ejemplo de la “ <i>four-rectangle feature</i> ”	26
Figura 20: Detección de características. Región de los ojos (Izquierda) y puente de la nariz (Derecha)	26
Figura 21: Ejemplo del proceso de reducción dimensional por PCA	27
Figura 22: Comparación de la proyección de 2 clases en PCA y LDA	27
Figura 23: Autocaras calculadas sin alinear (superior) y alineadas (inferior).....	28
Figura 24: Raspberry	28
Figura 25: Gráfico de la metodología de investigación.....	29
Figura 26: Cuadro de pros y contras de métodos holísticos de reconocimiento facial.....	32
Figura 27: Cuadro de pros y contras de los métodos híbridos de reconocimiento facial ..	32

Figura 28: Cuadro de pros y contras de métodos basados en “Features” para reconocimiento facial	33
Figura 29: Gráfico de pesos del modelo Yolo.9000 hasta la 4ta semana	35
Figura 30: Página web para la captura del video	37
Figura 31: Solicitud de acceso a la cámara.....	37
Figura 32: Página web con acceso a la cámara.....	37
Figura 33: Diagrama de secuencia de la toma del video	38
Figura 34: Diagrama de secuencia script “organizador.pl”	39
Figura 35: Procesos de nuestra metodología basado en CRISP-DM.....	39
Figura 36: Estructura de la matriz en el script”caritas.pl”	40
Figura 37: Archivo train.txt	41
Figura 38: Archivo de configuración del modelo	42
Figura 39: Archivo obj.names	43
Figura 40: Archivo obj.data	43
Figura 41: Página donde se muestra el reporte de asistencia.....	44
Figura 42: Reporte de asistencia en formato Excel.....	44
Figura 43: reporte de asistencia en formato pdf.....	45
Figura 44: Imagen con orientación vertical.....	46
Figura 45: Imagen con orientación horizontal	46
Figura 46: Posibles clasificaciones de las imágenes.....	47
Figura 47: Respuesta del comando de evaluación.....	49
Figura 48: Profesor siendo detectado por la red neuronal.....	51
Figura 49: Detección realizada por la red (primera foto tomada).....	52
Figura 50: Detección realizada por la red (segunda foto tomada)	52
Figura 51: Ubicación óptima del raspberry en un salón de clase	53

Lista de anexos

Anexo 1: Resultados de la encuesta realizada a estudiantes **60**
Anexo 2: Resultados de la encuesta realizada a docentes **61**

Resumen: Debido al aumento de estudiantes en la universidad y el gran tamaño de los cursos, en especial los de cátedra de la facultad de medicina, se evidencia la necesidad de agilizar el proceso de toma de asistencia de los estudiantes y docentes. En este trabajo se especifican los requerimientos de un sistema de reconocimiento facial para la toma de asistencia automatizada en aulas de clase basado en redes neuronales convolucionales y se muestran resultados del desempeño del sistema en un aula de clase de la Universidad Autónoma de Bucaramanga.

Palabras clave: Inteligencia artificial, redes neuronales, automatización, visión artificial.

1. INTRODUCCIÓN

La toma de asistencia de manera manual es un proceso que puede llegar a consumir tiempo valioso de clase si la cantidad de estudiantes que hay en el curso es demasiado grande. Según una encuesta realizada a estudiantes el 46,8% considera que se pierde tiempo con la toma de asistencia (ver anexo 1).

Uno de los factores relacionados con la toma de asistencia que generan incomodidad entre los docentes (en el caso de la UNAB) es la plataforma SIGA implementado por la universidad y que según una encuesta realizada a docentes al 45,7% de los encuestados no le gusta dicha plataforma (ver anexo 2) y entre las principales razones se encuentra que dicha plataforma no es práctica lo que vuelve el proceso de la toma de lista bastante tedioso.

Otro de los grandes problemas es el proceso de auditoría que se le realiza a los docentes, el cual no es de gran acogida; según la encuesta realizada a docentes, el 42% considera que el proceso de auditoría no se realiza de manera apropiada (ver anexo 2). Algunos docentes manifiestan que los auditores no tienen en cuenta los permisos académicos y situaciones como salir al baño o contestar una llamada afuera del salón como factores por los que el docente no esté en el salón designado ya sea por toda la duración de una clase o solo por unos minutos y que aun así generan reportes de inasistencia.

La solución que se propone en este documento contempla la posibilidad ser implementada en la Universidad Autónoma de Bucaramanga, automatizando la toma de asistencia y así evitar errores relacionados con dicho proceso además de disminuir los gastos de la universidad en materia de auditorías, dándole la capacidad de que se realicen como un proceso interno y no por parte de una empresa ajena a la UNAB.

2. PLANTEAMIENTO DEL PROBLEMA

La toma de asistencia en los salones de clase se hace por lo general de manera manual, si el salón tiene muchos estudiantes esto puede llegar a ser un problema ya que la cantidad de tiempo consumida para realizar este proceso se puede aprovechar en dictar la respectiva clase.

Con el aumento de estudiantes de primer ingreso en la Unab, los cursos han aumentado de tamaño llegando a tener hasta 30 estudiantes y aún más en Cursos de Medicina o Enfermería llegando a haber cursos de 54 estudiantes por lo que la toma de lista en estos salones ahora consume mucho más tiempo. En India, según un proyecto realizado para solucionar esta problemática, los cursos pueden llegar a ser de 70 estudiantes (Talaviya, Ramteke & Shete, 2013).

Otro de los problemas a solucionar es el proceso para reportar la asistencia de docentes y estudiantes a sus aulas de clase, este proceso tiene ciertos problemas en la logística para corroborar dicha asistencia docente por parte de un personal (Auditores) designado a esta tarea. Esto ha ocasionado problemas por las notificaciones de reportes “erróneos” descontando horas laborales, generando descontento para la población docente de la UNAB.

En una encuesta realizada a docentes de la UNAB se pudo observar que las auditorías que realiza la universidad son un proceso que tiene muchas falencias pero que se podría mejorar. Muchos de los docentes (un 45,7%) respondieron que sí tuvieron reportes falsos de asistencia.

Ha tenido reportes de inasistencia falsos por parte de auditoría? O problemas en general con el proceso?

81 respuestas

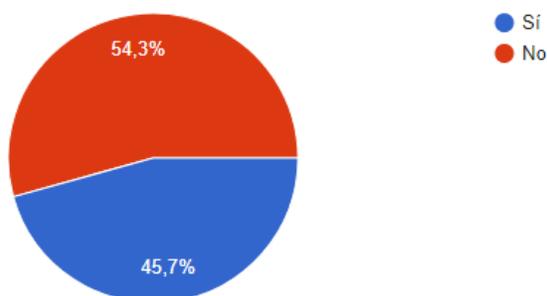


Figura 1: Resultado de la pregunta 3 encuesta realizada a docentes

Por esto se requiere una forma (automática) de validar y generar reportes precisos de la asistencia, tanto de los estudiantes como de los docentes sin que ninguno de ellos intervenga en la ejecución de ese proceso. Así mismo se busca disminuir la intervención de los auditores como recolectores de información para los reportes y así disminuir reportes erróneos que perjudiquen a los profesores.

3. PLANTEAMIENTO DE LA SOLUCIÓN

La recolección de datos, debe tener el medio correcto para poder identificar a las personas sin fallos, en general se hace uso de la biometría para este proceso, pero si vamos a obtener información con el fin de ahorrar tiempo no es pertinente usar tecnologías que requieren intervención de la persona, así se opta por el reconocimiento facial; un método de reconocimiento que solo requiere una cámara que capturará fotos a determinados momentos de la clase para analizarlos y generar los reportes de asistencia.

Posterior a la captura de la foto se hará un tratamiento de imagen con Darknet, este es un framework para uso de redes neuronales el cual nos permite identificar una o más caras en una imagen, se generan unas determinadas coordenadas en la imagen y pueden usarse para recortar cada cara para su posterior análisis.

Su análisis será llevado a cabo por una Red Neuronal Convolutiva (CNN), este modelo de red neuronal nos permite usar los principios de “*deep learning*” donde nuestro sistema llevará un determinado aprendizaje con imágenes de muestra de estudiantes y docentes para hallar patrones diferenciadores o minucias con el objetivo de que el sistema “aprenda” a diferenciar las caras en una imagen. Según sea el porcentaje de precisión podrá hacerse el chequeo de que esa persona está en clase.

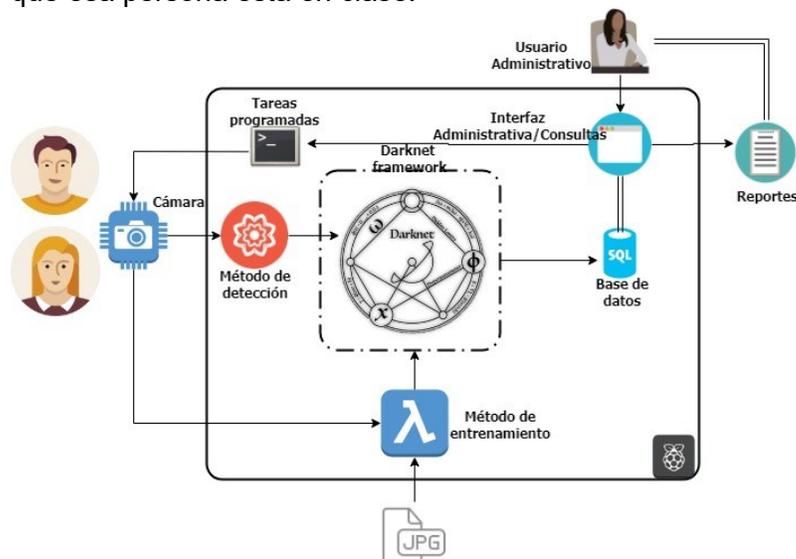


Figura 2: Arquitectura propuesta para el sistema
Fuente: Propia

Relevancia social:

Un cambio significativo en la cotidianidad del estudiante y del profesor permitiendo un mejor aprovechamiento del tiempo de clase.

Implicaciones prácticas:

La toma de asistencia de clases tanto del estudiante como del profesor con el fin de evitar errores en la verificación de la asistencia docente y malentendidos a la hora de realizar el control de horas trabajadas.

¿Con la investigación se llenará algún hueco de conocimiento?

Dará a conocer nuevas aplicaciones de las redes neuronales como método de solución.

Preguntas de investigación

¿Hasta qué momento una red neuronal para reconocimiento facial es funcional con los limitados recursos computacionales de un Raspberry PI?

¿Hasta qué punto llega a ser requerida la computación distribuida para el buen funcionamiento de un sistema de reconocimiento facial?

¿Cuáles son las características que debe tener un sistema de reconocimiento facial para eliminar por completo la toma de asistencia de forma manual en una clase?

4. OBJETIVOS

4.1. OBJETIVO GENERAL

Desarrollar un prototipo que utilice software de reconocimiento facial, redes neuronales y hardware de bajo costo y consumo para la toma de asistencia en aulas de clase. Caso de estudio Universidad Autónoma de Bucaramanga

4.2. OBJETIVOS ESPECIFICOS

- Definir tecnologías y métodos apropiados para el diseño del sistema
- Configurar el módulo encargado de localizar a los individuos en el salón de clase empleando hardware de bajo costo y consumo (Raspberry)
- Entrenar la red neuronal convolucional que se encargará de reconocer con precisión estudiantes y docentes
- Integrar los componentes en un sistema capaz de generar reportes de asistencia
- Realizar pruebas del prototipo en un ambiente real

5. RESULTADOS ESPERADOS

5.1. Objetivo específico 1

Estudio detallado de los diferentes métodos y técnicas que se utilizan para el reconocimiento facial con sus respectivos pros y contras. Además de identificar las dificultades que se pueden presentar con respecto a la identificación y los tiempos de entrenamiento.

5.2. Objetivo específico 2

Página web accesible utilizando cualquier tipo de dispositivo (móvil, computador, tablet) desde la cual los estudiantes y maestros puedan grabar y subir a la base de datos el video necesario para el entrenamiento junto con su ID.

Por otro lado, el script encargado de hacer el recorte de los frames de los videos que se encuentran en la base de datos y los guarda como imágenes individuales.

5.3. Objetivo específico 3

Scripts que se encarguen de crear los archivos necesarios para el entrenamiento del modelo. Más específicamente el archivo de configuración del modelo, el archivo que referencie las imágenes necesarias para entrenar y las clases (personas) que el modelo va a aprender a identificar.

El script que envía todos los archivos a la red neuronal y que inicia el entrenamiento.

5.4. Objetivo específico 4

Script que permita crear un informe de las personas que fueron identificadas en una imagen general de un salón de clases y el respectivo curso que se estaba viendo en el momento en que se tomó la foto

5.5. Objetivo específico 5

Realizar las respectivas pruebas del sistema en un ambiente real (un salón de clase), haciendo el entrenamiento y la posterior detección de los estudiantes y el profesor presentes en el salón de clase.

Reporte de las pruebas realizadas en el que se especifique qué dificultades se presentaron al momento de realizar las mismas y como se solucionaron dichas dificultades. El reporte también contendrá los resultados del entrenamiento del modelo con las imágenes tomadas en el salón de clase y se especificarán qué errores hubo en el proceso de entrenamiento y como se solucionaron.

6. ESTADO DEL ARTE

A través de los años se han podido ver diferentes aproximaciones al problema planteado, siendo la más común la identificación de la persona por medio de huellas dactilares como se puede apreciar en (Talaviya, Ramteke & Shete, 2013) donde se implementa un sistema que identifica a los estudiantes utilizando sus huellas dactilares almacenadas en una base de datos, y después marca la asistencia en una base distinta, a la cual se conecta utilizando la tecnología Zigbee. Otro trabajo similar (Nawaz, Pervaiz, Korrani & Ud-Din, 2009) realizado en Pakistán utiliza el mismo método de identificación pero solo utiliza una base de datos, contrario a las dos que utiliza el proyecto mencionado anteriormente. Se puede decir que este método tiene el mayor porcentaje de precisión en cuanto a la identificación de las personas (ambas arrojaron más del 98%).

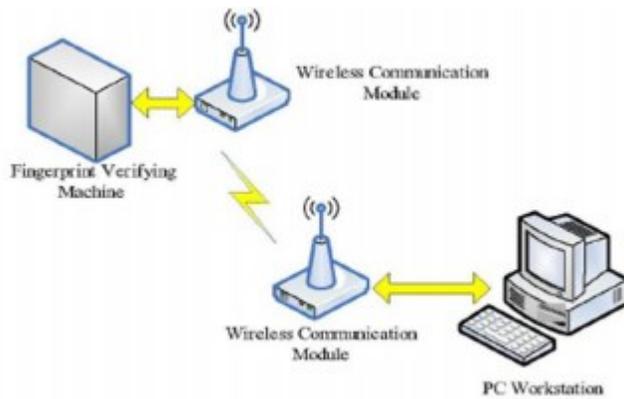


Figura 3: Arquitectura de un sistema que utiliza la tecnología Zigbee

Fuente: (Talaviya, Ramteke & Shete, 2013)



Figura 4: Hardware del sistema de reconocimiento por huellas dactilares

Fuente: (Nawaz, Pervaiz, Korrani & Ud-Din, 2009)

En otro proyecto (Masalha & Hirzallah, 2014) se combinan diferentes técnicas para tomar asistencia, en este caso se utiliza el reconocimiento facial para que el estudiante se registre en el sistema a través de una aplicación móvil, una vez realizado este paso el estudiante escanea un código QR que se le da al docente y una vez completado este paso el sistema marca la asistencia en la base de datos.



Figura 5: Aplicación móvil y código QR
Fuente: (Masalha & Hirzallah, 2014)

En ("A Conceptual Model for Automated Attendance Marking System Using Facial Recognition", 2014) se explica un modelo conceptual que utiliza reconocimiento facial. El sistema toma una foto del aula en la que se vean todos los estudiantes para luego calcular las minucias de cada uno por medio de filtros de gabor y compararlas con las que fueron guardadas anteriormente en una base de datos. Otro método de reconocimiento facial explicado en (Methi, Chauhan & Gupta, 2017) consiste en la implementación del algoritmo "Viola Jones" junto con las llamadas "*Haar Features*" para diferenciar las propiedades faciales de cada persona.

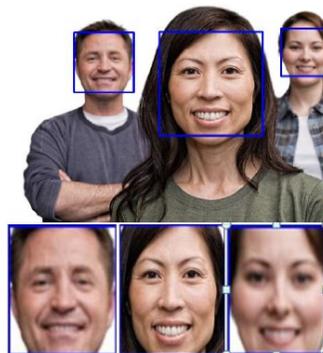


Figura 6: Detección y recorte de caras
Fuente: (Methi, Chauhan & Gupta, 2017)

Uno de los proyectos en los que se muestran resultados es (Kawaguchi, Yohei & Shoji, Tetsuo, 2005), en el cual el sistema utiliza 2 cámaras en lugar de una para el proceso de captura de las caras de los estudiantes; una de las cámaras (ubicada en el techo) guía a la otra (ubicada en la parte frontal del salón) para que esta fotografíe las caras de los estudiantes una por una para luego identificarlas. Uno de los problemas con los que se encontraron los autores es la precisión de la detección de las caras, la cual era muy baja para ser aceptable (37,5%), la solución con la que dieron los autores fue la de hacer ciclos

de observación continua en los que las cámaras hacen varias tomas de los estudiantes en un periodo de tiempo determinado para aumentar el porcentaje de detección de las caras (en el artículo se menciona que este método aumentó la detección de las caras hasta un 80%).

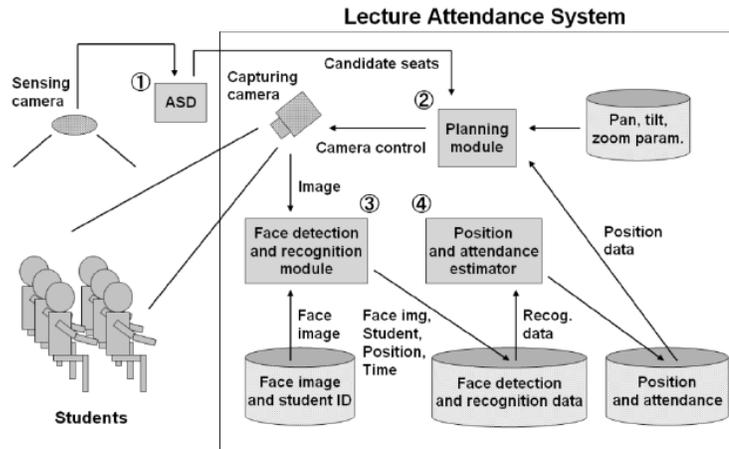


Figura 7: Arquitectura del sistema que utiliza dos cámaras
Fuente: (Kawaguchi, Yohei & Shoji, Tetsuo, 2005)

En (Balcoh, Yousaf, Ahmad & Baig, 2012) se propone un método llamado “Clasificación de piel” para solucionar el problema de la precisión en la detección de caras. Este funciona oscureciendo los píxeles en los que no hay piel (los vuelve negros) y en los que si hay piel los aclara (los vuelve blancos).

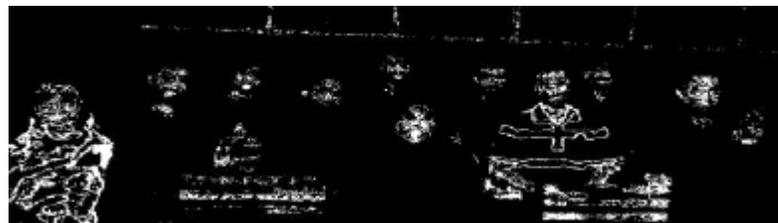


Figura 8: Método de clasificación de piel
Fuente: (Balcoh, Yousaf, Ahmad & Baig, 2012)

La tabla 1 presenta la revisión de la literatura sobre proyectos en los que se aborda el tema de toma de asistencia con diferentes tecnologías

Tabla 1: Revisión de la literatura sobre proyectos en los que se aborda el tema de toma de asistencia con diferentes tecnologías

Autor	Año	País	Tema				Tipo de documento
			1	2	3	4	
(Talaviya, Ramteke & Shete, 2013)	2013	India	X			X	Artículo de revista
(Nawaz, Pervaiz, Korrani & Ud-Din, 2009)	2009	Pakistan	X			X	Artículo de revista
(Masalha & Hirzallah, 2014)	2014	Baréin	X	X			Artículo de revista
(Sajid, Hussain, & Usman, 2014)	2014	Tailandia	X	X			Acta de conferencia
(Methi, Chauhan & Gupta, 2017)	2017	India	X	X			Artículo de revista
(Kawaguchi, Yohei & Shoji, Tetsuo, 2005)	2005	Japon	X	X	X		Artículo científico
(Balcoh, Yousaf, Ahmad & Baig, 2012)	2012	Pakistan	X	X	X		Artículo de revista
Tema 1: Toma de asistencia Tema 2: Reconocimiento facial Tema 3: Métodos para mejorar la detección de caras Tema 4: Identificación por huellas dactilares							

A continuación, se presenta el estándar para la construcción de un sistema de reconocimiento facial y los diferentes métodos que se pueden utilizar para la fase de reconocimiento de la persona.

Estándar de un Sistema de reconocimiento facial (SRF).

El diseño de la arquitectura de un SRF tiene un estándar o un diseño genérico a seguir para poder comprometer una precisión alta.

Su diseño genérico se compone de tres fases: detección, extracción y reconocimiento. La fase de detección se encarga de detección de caras, ubicación de la cara según patrones aprendidos gracias a la fase de extracción; esto concluye que la fase de detección y extracción trabajan simultáneamente.

La fase de extracción es la encargada de extraer características de una imagen donde se convierte en una matriz de covarianza creada a partir de los autovectores por cada uno de los píxeles de la imagen, la matriz podrá ser usada para dos casos, o para entrenar el sistema a reconocer esa imagen según la matriz, o para extraer las características para hacer la comparación de una matriz de covarianza ya generada.

Ejemplo: Clase balón-> Entran solo balones y se genera matriz de covarianza para aprender reconocer balones.

Si se hace uso de la clase balón será para obtener la matriz de balón en la fase de reconocimiento.

La fase de reconocimiento es el paso final donde se hace la comparación por los distintos métodos de reconocimiento, ya sean holísticos, extracción geométrica o híbrida de la matriz de covarianza de la clase generada en la fase de extracción, como producto generará un porcentaje de precisión entre la imagen a reconocer con la clase que ha hecho matching.



Figura 9: Arquitectura de un SRF
Fuente: Propia

Definición de tecnologías y métodos apropiados para el diseño de un SRF en la fase de reconocimiento.

Los métodos más representativos para reconocimiento facial y su revisión para poder notar cuáles serían los indicados para nuestro caso de estudio, donde nuestra clasificación no está basada en encontrar los algoritmos más robustos, si no aquellos que demuestren una calificación respecto al tiempo de cómputo y a la eficiencia de sus resultados.

El entorno de nuestro caso de estudio es controlado, aunque se debe tener en cuenta que estos requerimientos son generados en nuestro entorno del caso de estudio:

Deben ser ejecutados entre 12-100 identificaciones cada 20 minutos, durante un promedio de 3 horas de duración de cada clase, y durante toda una jornada de estudio que abarca entre las 6 am-9 pm.

1. Métodos Holísticos basados en PCA/LDA

- **Eigenfaces**

Este método consiste en el análisis de vectores (Autovectores o vectores propios) de una imagen, donde por cada pixel de la matriz de la imagen habrá un autovector.

Ejemplo: Si se tiene una imagen de 100x100 Pixeles, se tendrían 10000 autovectores.

El primer objetivo de este método es que al finalizar el entrenamiento los autovectores se derivan de una matriz de covarianza, la cuál es la representación de las características principales o valores diferenciales entre una imagen y otra (Caso PCA).

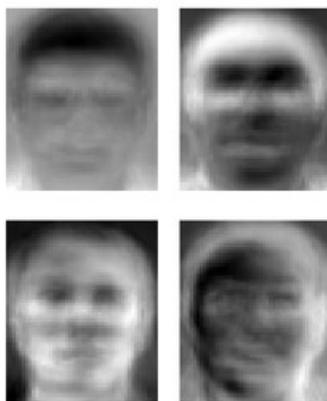


Figura 10: Eigenfaces en una cara
Fuente: ("The Database of Faces", 2002)

- **Probabilistic Eigenfaces (PE)**

Este método es la variación de Eigenfaces en donde su fase de extracción y reconocimiento de cada una de las clases son trabajados con métodos bayesianos como factor de decisión, debido a que "hay similitudes de caras" al aplicar PCA en su Biometría; tales como cicatrices, asimetría facial, barba, vitiligo, lunares, etc. Donde estos llegan a convertirse en un problema estadístico para la máquina para poder reconocer arrojando resultados no tan fiables, por eso este método ve las características que tienen en común y cuáles son las diferenciales de cada clase por métodos bayesianos.

Este algoritmo por lo general se maneja de tal forma, que la extracción de características no se hace por parámetros o un control de extracción, como los sistemas de reconocimiento facial que reconocen ciertos puntos característicos de una cara humana (Lunares, nariz, boca y ojos cada uno por separado), PE

implica en la extracción estadística o caracterización estadística de las muestras de entrenamiento y hacer diferencias estadísticas o pesos estadísticos por métodos bayesianos.

Su rendimiento ante cambios de posición de cara, o cambios de iluminación para una mayor dispersión de la clase entre clases se requiere un tamaño considerable de imágenes para su entrenamiento, por lo tanto, debido a su tamaño, la búsqueda de su dispersión de este método requiere un poco más de procesamiento dado a que los cálculos serán hechos en una CPU y no en una GPU.

- **Fisherfaces**

Este método de reconocimiento es una modalidad de Eigenfaces (PCA) con la utilización del método Discriminante Lineal de Fisher (FLD o LDA).

La iluminación de una imagen y la posición de la cara son márgenes de errores reducidos con Fisherfaces. Su único obstáculo es que la precisión es reducida significativamente si se manejan en una sola base de datos tomas de imágenes normales, cambios de posición e iluminación, esto afecta la fase de reconocimiento. Para evitar esto se crean clases para una sola clase, es decir una clase de imágenes con posición de cara e iluminación normal, una clase con cambios de posición de cara y una clase cambios de iluminación.

Fisherfaces no es más que otra alternativa para reconocimiento, se ha demostrado que en entornos controlados no tiene una precisión significativa ante los métodos tradicionales de reconocimiento facial, pero ante el momento de clasificación o extracción de características tiende a ser mejor puesto a que Fisherfaces basado el FLD consigue más dispersión entre clases.

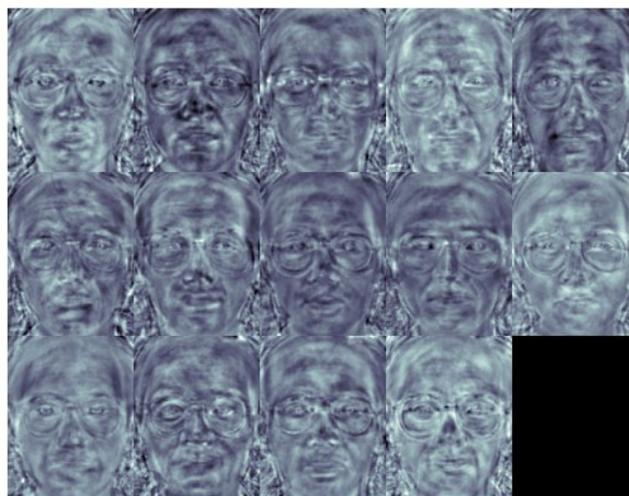


Figura 11: Fisherfaces

Fuente: ("The Database of Faces", 2002)

- **Support Vector Machines (SVM)**

Este método es más procedimental para poder lidiar con muchas clases o individuos con un SVM Binario.

Un SVM Binario, es un método que consiste en usar dos parámetros y cada uno será una clase; estos parámetros en el contexto de reconocimiento facial son:

- Las diferencias entre imágenes varias personas
- Las diferencias entre las imágenes de un solo individuo

La finalidad de este método es la generación de una superficie de decisión. El proceso de decisión es hacer la diferencia de una imagen A entre todas las personas, la mayor probabilidad de las personas entre la imagen, pasarán a la diferencia entre cada clase individuo de la persona con mayor probabilidad en el anterior matching y la imagen A. Finalmente aquel resultado que tenga la mayor probabilidad, por descarte, es el individuo que intentamos identificar.

- **Evolution Pursuit (EP)**

Evolution Pursuit es un modelo adaptativo y evolutivo de reconocimiento facial. Su estructura se basa en el uso de algoritmos evolutivos donde se analizan los cromosomas (Individuos) con una búsqueda de base óptima de la imagen a identificar.

La identificación de la base óptima generalmente se trabaja manteniendo un tamaño constante de población de soluciones candidatas conocidas como cromosomas.

La eficacia del algoritmo es su recursividad, no hay estándar para EP, debido a que dependiendo del caso dependerá es del poder de exploración del algoritmo genético.

Las contras de EP radican en que requiere gran trabajo de observación y de cierto aprendizaje asistido al algoritmo genético, teniendo en cuenta que entre más cromosomas mayores es la complejidad.

2. Feature-based Methods

- **Convolutional Neural Network(CNN)**

Convolutional Neural Network o Redes Neuronales Convolucionales es la técnica con mayor margen en clasificación y detección frente a otras.

Las CNN evitan y reducen los costos computacionales de ejecutar numerosas veces un pequeño clasificador en varios espacios de la imagen, esto hace y convierte a las CNN como un método que no se debe ignorar.

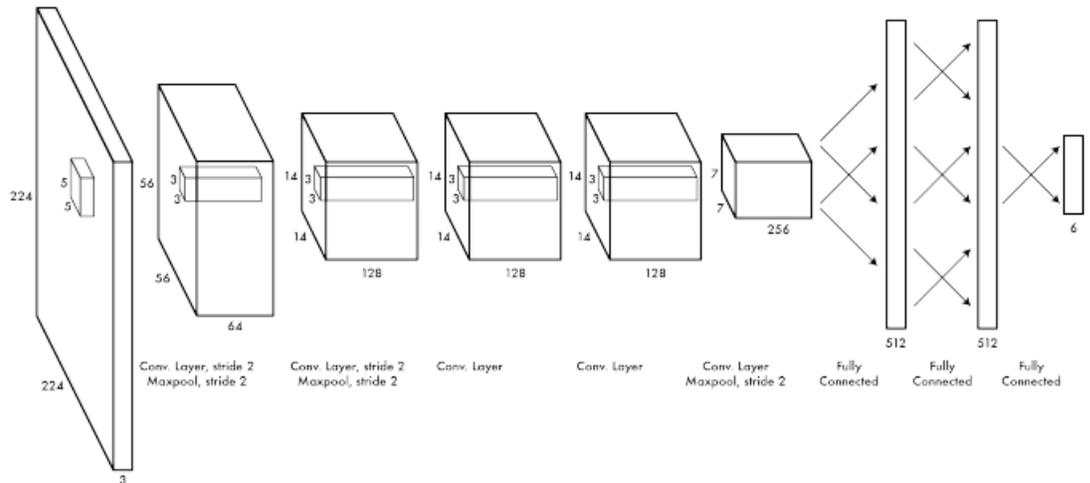


Figura 12: Arquitectura de Darknet Real Time Grasp Detection Networks
Fuente: (“Redmon, J., & Angelova, A”,2015)

Algunas implementaciones de CNN para reconocimiento y clasificación de imágenes:

Darknet: Open Source Neuronal Networks in C:

Darknet es un framework con distintivos métodos de solución hacia identificación, tratamiento, manejo de imágenes.

Unas de sus variantes son el uso de las redes neuronales con CIFAR-10 o entre otras para la identificación de las imágenes usando también softmax, PCA, y modelos. Esto permite aplicarlo hacia la detección de caras e identificación de la misma.

Su arquitectura es basada en capas como cualquier CNN, cada capa tendrá la función de reducir las dimensiones de una imagen en sectores más pequeños de imagen en donde se trabajará el algoritmo softmax para generar “weights” o pesos de la totalidad de la imagen.

Los weights o pesos serán nuestro equivalente a matriz de covarianza cuando hablábamos de métodos basados en PCA, estos tienen las características de la imagen.

Algunos puntos a tener en cuenta sobre Darknet:

Implementación de filtros de imagen para reducir el ruido de las imágenes y su tiempo de cómputo reducido por ser basado en CNN.

Capacidad de configurar el sistema como reconocimiento y clasificación por métodos híbridos de CNN con métodos bayesianos, modelos en 2d, graphs, entre otros.

El entrenamiento de la red neuronal no está limitado a solo imágenes, si no que el entrenamiento y detección también puede hacerse por medio de videos.

Algunos contras de Darknet es el número de imágenes requeridas por entrenamiento por clase; esto deja de ser un contra, si se usa un video no mayor de 15 segundos del individuo mostrando su cara.

Tensorflow: Es una librería de software open source tiene la capacidad de crear una CNN para clasificación e identificación de imágenes, puede ser basada en una arquitectura multicapa, estas capas son seguidas por capas conectadas que conducen a un clasificador softmax y por las regresiones lineales se puede identificar la imagen.

3. Hybrid methods

Los métodos híbridos son fácilmente ligados y llegados a este puesto con la perfecta combinación de muchas técnicas con fines de complementarse en cualquiera las fases de un SRF. Aunque hay muchas combinaciones estas son las más relevantes.

- **Modular Eigenfaces**

Esta técnica es de enfoque rápido, simple y válido para procesar el rostro humano es bastante eficaz. En el procesamiento del rostro humano, normalmente Eigenfaces extrae información útil sin tomar en cuenta el cambio de la iluminación.

Una característica importante es que la imagen es cortada en trozos donde se encuentran las características por Eigenfaces, acto seguido cada trozo será analizado por varios métodos a elegir por el usuario para el diseño de su sistema, ya sea para su entrenamiento, clasificación y reconocimiento del mismo.

- **Component-based**

El análisis basado en componentes puede ser soportado por casi todos los anteriores métodos, su característica importante es la estricta extracción de datos, ya sea en modelos 2D o 3D de una cara.

Un SRF basado en componentes es sin duda uno de los métodos más robustos y precisos.

Hay distintas modalidades o combinaciones híbridas que lo convierten en una apuesta como método de preferencia, uno de los más precisos es el Modelo SVM- 3D Morphing- Geometric Methods. Este método permite crear un modelo que permite la inserción manual de las características sobre un modelo 3D (puntos de referencia) para su entrenamiento, clasificación y reconocimiento en base a este.

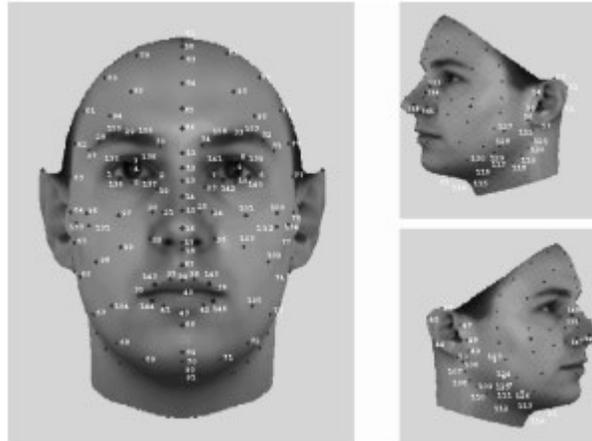


Figura 13: Puntos de referencia

Fuente: ("Heiselet, B., Serre, T., Pontil, M., & Poggio, T", 2001)

4. Servicios

Kairos, Amazon y Microsoft son las únicas empresas que aparentemente son trascendentales y proveen servicios de tratamiento de imágenes o de visión artificial que permitan reconocimiento facial. Su acceso está basado por API's, las limitantes se ven reflejadas en la capacidad para identificar o detectar muchas caras por imagen (Necesidad ligada a nuestro caso de estudio) a excepción de Kairos.

Kairos puede detectar todas las caras y sus ventajas son que se puede obtener la extracción de características, una de sus desventajas es su plan de precios:

- Posible consumo en fase de pruebas: Serían 500 dólares por mes
- Posible consumo en nuestro caso de estudio: Serían 3000 dólares por mes

Amazon nos provee un máximo de 15 detecciones por imagen mostrando sus parámetros, imposibilitando escalar nuestra solución y aunque con precios más bajos no nos permite dar solución a nuestro problema.

Microsoft permite hasta 64 caras por imagen pero sus precios también varían por consumo.

	Kairos	Amazon	Google	Microsoft	IBM	Affectiva	OpenCV
Face Detection	✓	✓	✓	✓	✓	✓	✓
Face Recognition (Image)	✓	✓	✗	✓	✗	✗	✗
Face Recognition (Video)	✓	✗	✗	✗	✗	✗	✓
Emotional Depth (%)	✓	✗	✗	✓	✗	✓	✗
Emotions Present (Y/N)	✓	✓	✓	✓	✗	✓	✗
Age & Gender	✓	✓	✗	✓	✓	✓	✗
Multi-face Tracking	✓	✓	✓	✓	✓	✓	✓
SDK	✓	✗	✗	✗	✗	✓	✓
API	✓	✓	✓	✓	✓	✓	✗
Ethnicity	✓	✗	✗	✗	✗	✗	✗

Figura 14: Comparación de servicios
Fuente: (Virdee-Chapman, 2017)

7. MARCO TEORICO

7.1. Framework

Es una estructura de niveles que indica que tipo de programas pueden o deben ser contruidos y cómo se interrelacionan. Algunos frameworks incluyen programas, interfaces de programación, u ofrecen herramientas de programación para su uso (Rouse, 2015). Un framework puede ser un kit de funciones de un sistema y cómo se interrelacionan; las capas de un sistema operativo; las capas de una aplicación subsistema; como la comunicación puede ser estandarizada en cierto nivel de una red; entre otros (Rouse, 2015).

Entre los muchos ejemplos de frameworks se encuentran:

- *Resource Description Framework*: Reglas del “*World Wide Web Consortium*” para como describir cualquier recurso de internet como una página web y su contenido (Rouse, 2015).
- *Sender Policy Framework*: Un acercamiento definido y programación para hacer que el *e-mail* sea más seguro (Rouse, 2015).

7.2. Red neuronal

Sistema adaptativo complejo conformado por varias capas que puede cambiar su estructura interna basado en la información que entra al mismo (Shiffman, 2013). Las redes neuronales tienen la capacidad de “aprender” (de ahí la característica adaptativa) de diferentes maneras que son:

- **Aprendizaje supervisado**: Este tipo de aprendizaje consiste en que (en el caso de reconocimiento facial) un “profesor” le muestra al sistema una serie de caras (el profesor ya conoce los nombres asociados con las caras), el sistema hace su predicción sobre a quién pertenecen esas caras y compara dichas predicciones con las respuestas correctas y hace ajustes según sus errores (Shiffman, 2013).
- **Aprendizaje no supervisado**: Es requerida cuando no hay ejemplos de respuestas correctas como en el anterior método. Un ejemplo de este tipo de aprendizaje es el de dividir elementos en grupos de acuerdo a patrones desconocidos (Shiffman, 2013).
- **Aprendizaje de refuerzo**: Es una estrategia construida en base a la observación. Si la red neuronal toma una decisión y el resultado es de alguna manera negativo, el sistema realiza el mismo proceso pero tomando una decisión distinta y evaluando el resultado de la misma y comparándolo con la anterior y así sucesivamente (Shiffman, 2013).

7.3. CNN

Una Red Neuronal Convolutiva (CNN) es una red neuronal cuyas redes se encuentran con una jerarquía evolutiva en términos de procesos y estas son alternadas por capas o layers donde comunican los anteriores resultados de una capa inferior a una capa posterior.

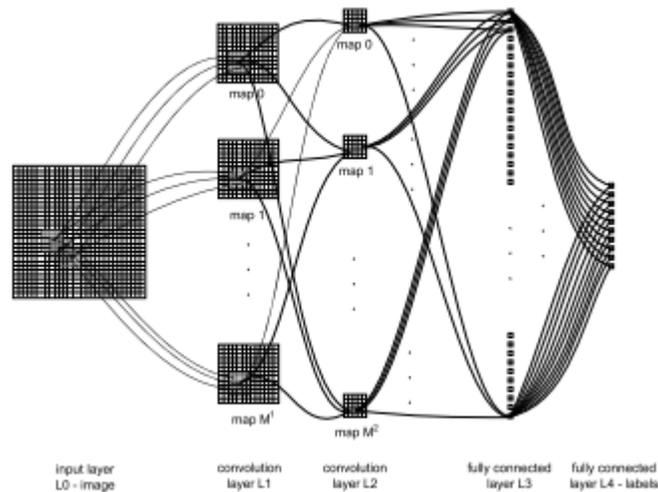


Figura 15: Arquitectura de una Red Neuronal Convolutiva
Fuente: (Cireşan et al., 2011)

7.4. Darknet

Es un framework de código abierto para redes neuronales escrito en el lenguaje de programación C y CUDA que soporta computación de CPU y GPU. Tiene diferentes funciones como:

- **YOLO (*You Only Look Once*)**: Es un sistema de detección de objetos en tiempo real (45 frames por segundo). Este sistema aplica una red neural a la imagen; esta red divide la imagen en regiones y hace probabilidades a cada región (que hay en la región).

7.5. Código QR

También llamado “Código de respuesta rápida”, es un tipo de código de barras bidimensional, el cual guarda su información en una matriz de puntos y permite guardar más información alfanumérica que los códigos de barras convencionales. Estos códigos contienen unas marcas cuadradas que cumplen la función de darle al lector una información posicional que indica la existencia de un código para ser leído, lo que a su vez aumenta la velocidad a la que se lee el mismo (Qrcode.com, 2017).



Figura 16: Códigos QR
Fuente: (Qrcode.com, 2017)

7.6. Zigbee

Es un lenguaje inalámbrico utilizado para conectar o comunicar dispositivos. Otorga diferentes ventajas entre las que se encuentran la interoperabilidad, la cual logra mediante la estandarización de aspectos como la comunicación básica hasta cómo operan los productos. Además, es de bajo consumo, permitiendo que un dispositivo funcione por años con una sola batería ("What is Zigbee?", 2014).

Este lenguaje es ampliamente utilizado en IoT y está orientado a la domótica en aspectos como control de temperatura, de luces, alerta de intrusos, detección de Co2 o fugas de agua.

7.7. Minucia

Rasgo o detalle (en este caso de la cara de una persona) como puede ser la distancia entre los ojos o el puente de la nariz (son las que más se utilizan para la detección de caras).

Filtros de Gabor: Filtro de dos dimensiones utilizado para reducir el ruido de una imagen sin afectar la dirección de la misma.

7.8. Haar Features

Son rasgos visuales simples utilizados para el entrenamiento de sistemas de reconocimiento de objetos. El ejemplo más conocido en que se utilizan es el algoritmo de Viola-Jones, en el cual se utilizan para la detección de zonas de interés (por ejemplo, en que parte de una imagen está la cara de una persona) en una imagen por medio de diferentes funciones.

Entre sus funciones básicas se encuentran:

- *Two-rectangle feature*: Diferencia entre la suma de los pixeles

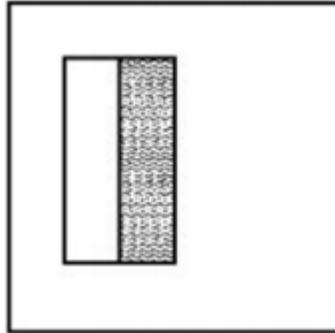


Figura 17: Ejemplo de la “*Two-rectangle feature*”
Fuente: (Viola & Jones, 2004)

- *Three-rectangle feature*: Suma entre dos triángulos externos, sustraída de la suma en un rectángulo central

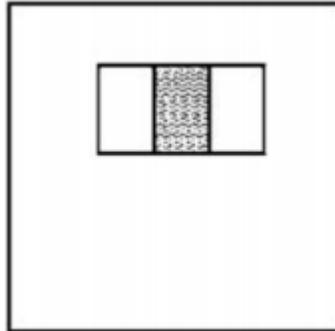


Figura 18: Ejemplo de la “*Three-rectangle feature*”
Fuente: (Viola & Jones, 2004)

- *four-rectangle feature*: Diferencia entre pares de rectángulos diagonales

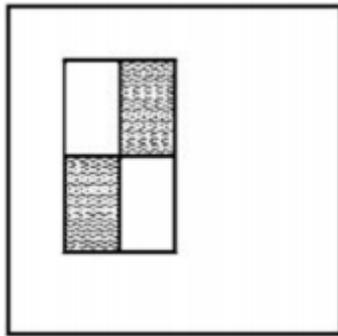


Figura 19: Ejemplo de la “four-rectangle feature”
Fuente: (Viola & Jones, 2004)

7.9. Viola Jones

Es un algoritmo utilizado para la detección de objetos en tiempo real que es ampliamente utilizado ya que realiza el proceso de detección de manera rápida. Este algoritmo utiliza *Haar Features* para detectar ciertas características en los rostros de las personas como por ejemplo que la región de los ojos generalmente es más oscura que la nariz y las mejillas.



Figura 20: Detección de características. Región de los ojos (Izquierda) y puente de la nariz (Derecha)
Fuente: (Viola & Jones, 2004)

7.10.PCA (Principal Component Analysis)

Es un método que transforma una determinada cantidad de variables correladas a un pequeño número de variables incorreladas llamadas componentes principales; estos componentes en resumen son una mejor representación de un grupo de imágenes por medio de vectores en un sistema o modelo de coordenadas permitiendo una reducción dimensional y tiempo de análisis de esta misma.

Este proceso requiere un entrenamiento previo del sistema con otras imágenes referenciadas con puntos de performance o peso decisivo (Porcentualidad).

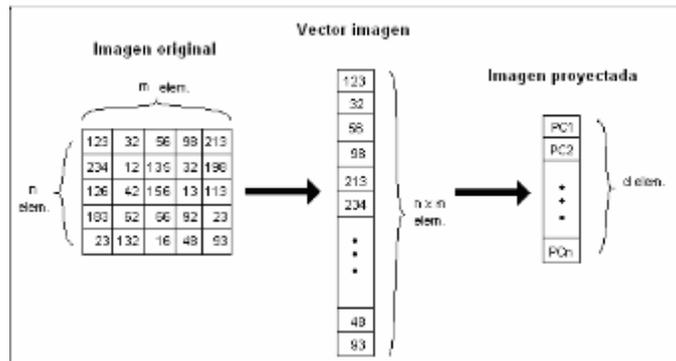


Figura 21: Ejemplo del proceso de reducción dimensional por PCA
Fuente: (Gimeno ,2010)

7.11.LDA (Linear Discriminant Analysis)

Es un método que permite reducir un problema de identificación o ubicación de alta dimensión mediante la proyección de datos en un espacio vectorial más pequeño.

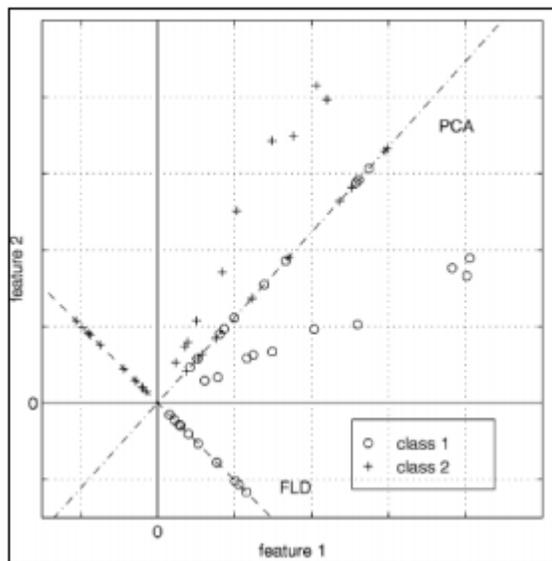


Figura 22: Comparación de la proyección de 2 clases en PCA y LDA
Fuente: (Gimeno ,2010)

7.12.DCT (Discrete Cosine Transform) por bloques

DCT es una transformación que representa una secuencia finita de datos como en la suma de una serie de funciones cosenoidales oscilando diferentes frecuencias.

Esta técnica es muy utilizada en aplicaciones de procesamiento; una modalidad de esta técnica es el análisis por bloques, que permite dividir los datos en bloques y analizar características determinadas. Este sistema hace uso de sistemas basados en análisis de modelos obligando que el sistema no podría funcionar correctamente si no se encuentra alineadas las imágenes. (Abdallah, A. S, 2007)



Figura 23: Autocaras calculadas sin alinear (superior) y alineadas (inferior)
Fuente: (Gimeno, 2010)

7.13.Raspberry

Es una computadora de bajo costo y de tamaño de una tarjeta de crédito, que permite una portabilidad y usabilidad al usuario.

Posee puertos 4XUSB, LAN, HDMI, Audio Input/Output, Camera SSID y acceso por WIFI y BLUETOOTH.



Figura 24: Raspberry
Fuente: (Raspberry Pi ,2017)

8. METODOLOGÍA

- Definir tecnologías y métodos apropiados para el diseño del sistema
 - Buscar en bases de datos académicas trabajos relacionados con el tema o la solución
 - Buscar en repositorios de código aplicaciones de las técnicas y métodos
 - Clasificar los resultados encontrados definiendo parámetros para su comparación

- Configurar el módulo encargado de localizar a los individuos en el salón de clase empleando hardware de bajo costo y consumo (Raspberry)
 - Definir el proceso y secuencia en que interactúa el hardware de captura de datos con el software de reconocimiento
 - Crear el módulo que se encargue de capturar los resultados y recortar las imágenes en las coordenadas correspondientes a los individuos
 - Integrar lo anterior en un flujo de información cuyo resultado es un conjunto de imágenes recortadas.

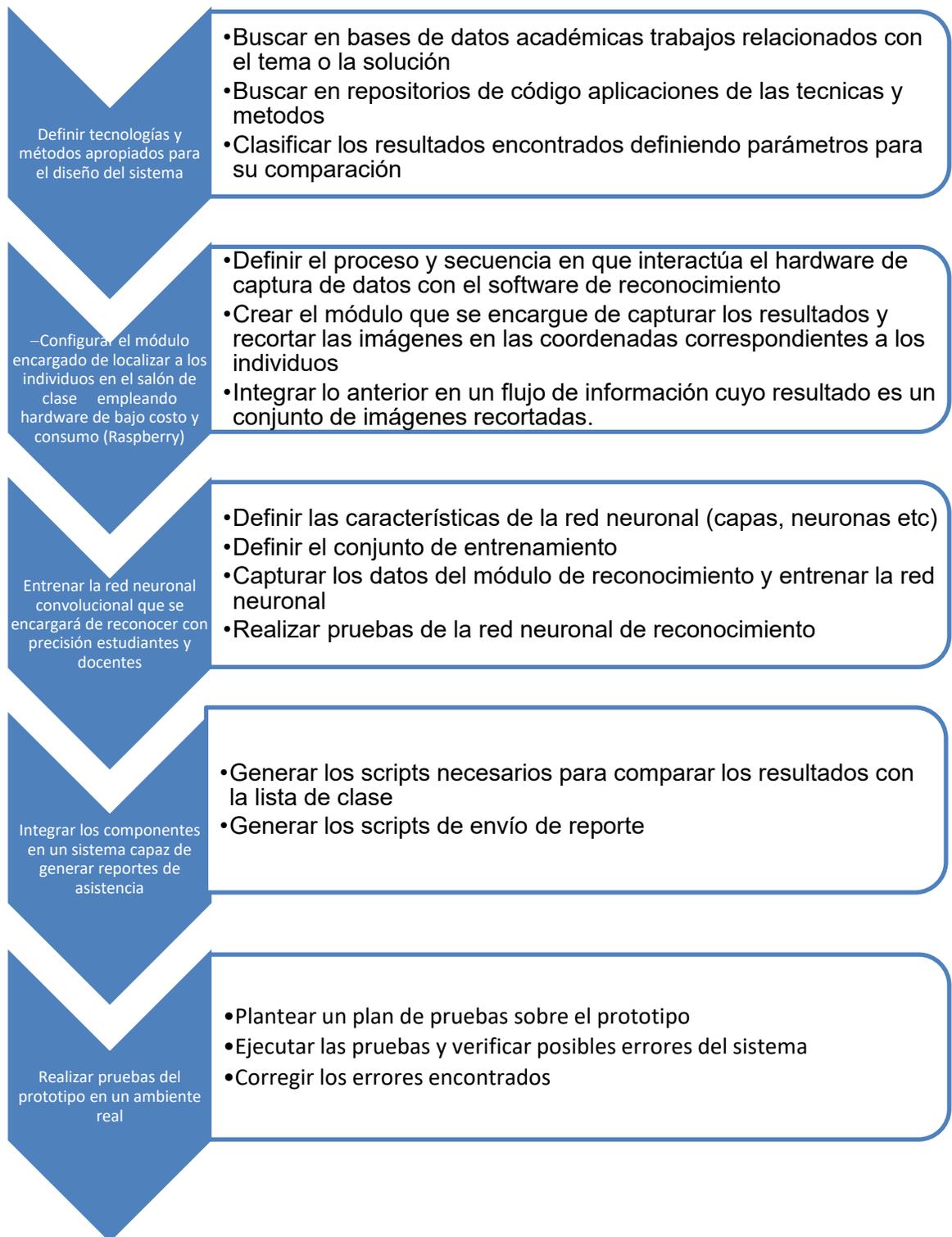
- Entrenar la red neuronal convolucional que se encargará de reconocer con precisión estudiantes y docentes
 - Definir las características de la red neuronal (capas, neuronas etc)
 - Definir el conjunto de entrenamiento
 - Capturar los datos del módulo de reconocimiento y entrenar la red neuronal
 - Realizar pruebas de la red neuronal de reconocimiento

- Integrar los componentes en un sistema capaz de generar reportes de asistencia
 - Generar los scripts necesarios para comparar los resultados con la lista de clase
 - Generar los scripts de envío de reporte

- Realizar pruebas del prototipo en un ambiente real
 - Plantear un plan de pruebas sobre el prototipo
 - Ejecutar las pruebas y verificar posibles errores del sistema
 - Corregir los errores encontrados

Figura 25: Gráfico de la metodología de investigación

Fuente: Propia



9. RESULTADOS OBTENIDOS

9.1. Objetivo específico 1

Se consultaron diferentes métodos y técnicas utilizadas para el reconocimiento de imágenes, se compararon métodos holísticos, métodos híbridos y métodos basados en “feautres”, estas comparaciones se pueden observar en las figuras 26, 27 y 28 para cada tipo de método.

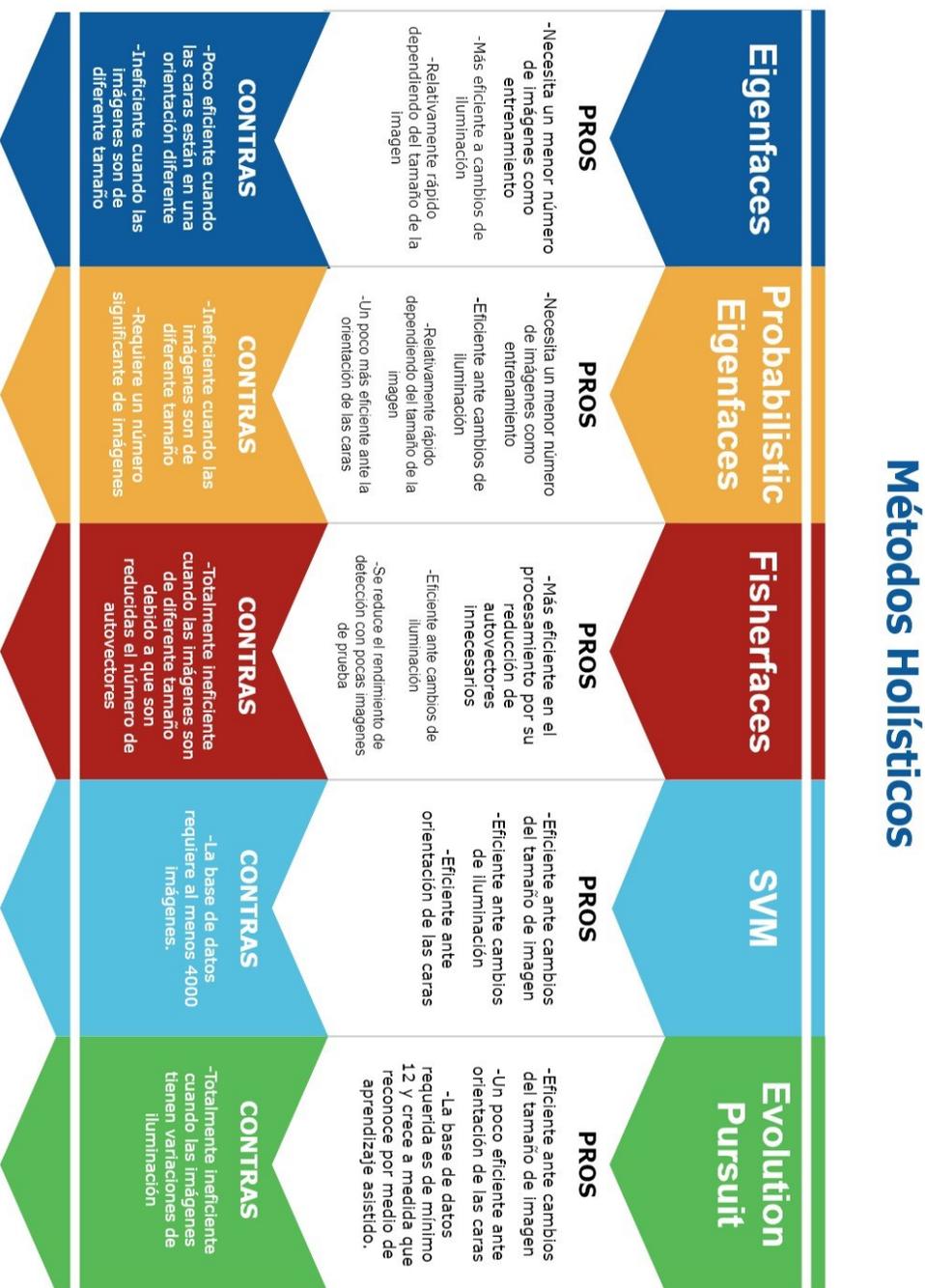


Figura 26: Cuadro de pros y contras de métodos holísticos de reconocimiento facial
Fuente: Propia



Figura 27: Cuadro de pros y contras de los métodos híbridos de reconocimiento facial
Fuente: Propia

Feature-based methods



Figura 28: Cuadro de pros y contras de métodos basados en “Features” para reconocimiento facial
Fuente: Propia

Todos los algoritmos, métodos o técnicas permiten desarrollar, implementar y desplegar un Sistema de Reconocimiento Facial; la gran diferencia entre estos métodos es su capacidad de procesamiento. Para nuestro proyecto se ha determinado las características mostradas en la tabla 2:

Tabla 2: Características para nuestro caso de estudio

Característica	Intervalo Mínimo	Intervalo Máximo
Clases/Aula	5 clases/aula	35 clases/aula
Promedio de Entrenamiento	1 hora	36 horas

Posibles problemas

El problema principal y reto; en el contexto de un escenario real donde se entrene el sistema de reconocimiento con todo el campus universitario, es la Carga de Trabajo (Throughput o TrainingTimeRate).

Haciendo un conteo sobre el número de salas, la carga de trabajo solamente del Campus el Jardín sería de un aproximado de 160 aulas con un promedio de 6 cursos que se dictan en el aula y un promedio de 25 estudiantes por curso. Lo anterior abarcaría dos probables cargas de trabajo para generar el modelo.

Antes de ver los posibles escenarios y sus modelos se debe abarcar el problema del tiempo de entrenamiento y la capacidad de computo, en nuestro caso el entrenamiento se realizará con una GPU Nvidia Gtx 960 de 4Gb; por otro lado, los modelos no están reflejados a un contexto de computación distribuida (Que requeriría mucho menos tiempo). En la tabla 3 se muestran los dos posibles modelos con sus respectivas características de instancias a entrenar y como se aplicarían al contexto de nuestro proyecto.

Tabla 3: Posibles modelos y sus características

Modelo	Instancias a entrenar	Contexto del trabajo
Todas las instancias en un solo modelo	<p>(Instancias PromedioxAulasPromedio)=4000</p> <p>Comparación: Yolo-9000 tardó en entrenarse 3 semanas ininterrumpidas y reconoce 9000 objetos diferentes sin mayores similitudes y con alta precisión.</p> <p>El tiempo de entrenamiento de este modelo tomaría más de 12 semanas debido al cluster natural que causa la biometría facial y debido a que solo se pueden analizar 64 imágenes por evolución.</p>	<p>El problema del reconocimiento facial en un modelo que abarque todo el campus es la cantidad de tiempo que le tardará a los distintos modelos separar las características distintivas de los individuos e identificarlos, incluyendo el proceso de captura, preparación de los datos y posterior entrenamiento.</p> <p>El problema natural de entrenar este modelo son las características en común o biometría compartida, que a mayor escala tarda muchas evoluciones y tiempo en poder llegar a un vector de pesos estable o diferenciador de la biometría.</p>
Un modelo x Aula x Materia	<p>AulaxMateria=960 Modelos de 25 Instancias cada uno</p> <p>En las pruebas, 1 modelo (10 Instancias) se ha entrenado en 2,5 horas con una precisión del 93%</p> <p>Entrenar los 960 modelos tardaría 100 días en terminarse con nuestra máquina.</p> <p>Como las cargas de trabajo son independientes a los pesos esta forma sería más fácil de aplicar computación distribuida o por lo menos un servidor para tratamiento de gráficos con 4</p>	<p>La biometría natural y la extracción de características es más rápida y tarda menos tiempo en llegar al modelo con vectores de pesos estables.</p>

Modelo	Instancias a entrenar	Contexto del trabajo
	tarjetas gráficas de la misma característica tardaría 25 días en completarse el entrenamiento y haciendo uso de NVIDIA GPU CLOUD tardaría menos, aunque se requiere una inversión para la contratación de un servicio mayor al de la capa gratuita.	

El problema en la biometría humana para todos los modelos es lo que tarda en diferenciarse de sus facciones humanas a sus individuales, todos los modelos basados en pesos y/o feautres tienden a generar clusters debido a sus características similares (Cara, Forma, Geometría) en el caso de Reconocimiento Facial.

En la figura 29 podemos ver los pesos de las 4 semanas de entrenamiento del modelo Yolo-9000 y cómo se comportan como clusters los pesos de las primeras dos semanas (Cada recuadro es una semana de trabajo)

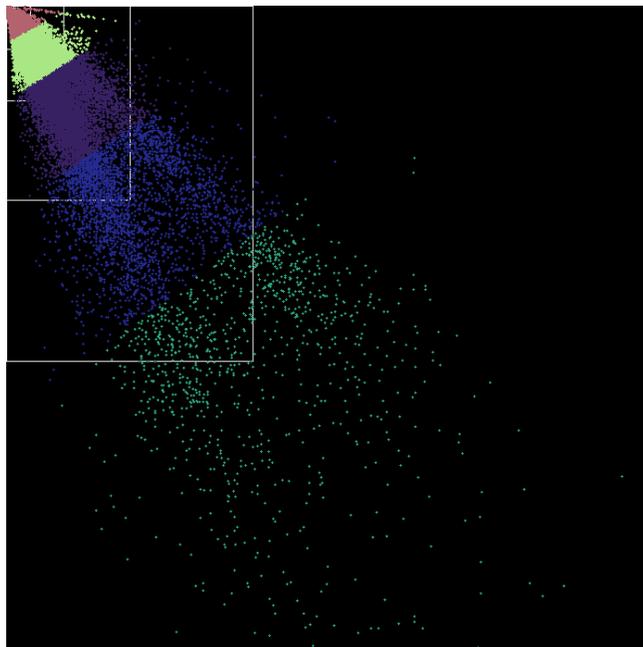


Figura 29: Gráfico de pesos del modelo Yolo.9000 hasta la 4ta semana
Fuente: Propia

Los modelos en general son complejos y para el problema de la biometría compartida hay muchos modelos que no trabajan muy bien y la diferencia de detección entre clases es muy poca:

Ejemplo: Se testea Clase 1 y Detecta Clase 1 30% Clase 2 28% Clase 3 26%

Los feautre based models son los únicos que tienen formas de lidiar con este problema de clustering; Redes Neuronales tiende a salir más rápido de la fase de pesos en cluster con semillas aleatorias.

En el caso de redes neuronales se ha estudiado en específico la librería de Redes Neuronales Darknet y se ha elegido finalmente por una función en específico que es Model-Kmeans; este modelo quita los pesos que los hace similares (Caso perfecto para diferenciar los que los hace similares y evitar detecciones erróneas).

9.2. Objetivo específico 2

Proceso de captura y preparación del set de entrenamiento

Para la captura del video necesario para generar el conjunto de entrenamiento de la red neuronal, para esto se dispone de una página web (ver figura 30) disponible para dispositivos móviles, computadores portátiles y equipos de escritorio. En esta página el estudiante ingresa su ID de la universidad y luego se graba por un periodo de 40 segundos; como se observa en la figura 31, la persona tiene que dar acceso a su cámara para iniciar la grabación y la deberá detener de manera manual con el botón “para”. En la figura 32 se puede observar una captura de la página cuando se le da acceso a la cámara. Este proceso se resume en el diagrama de secuencia de la figura 33.

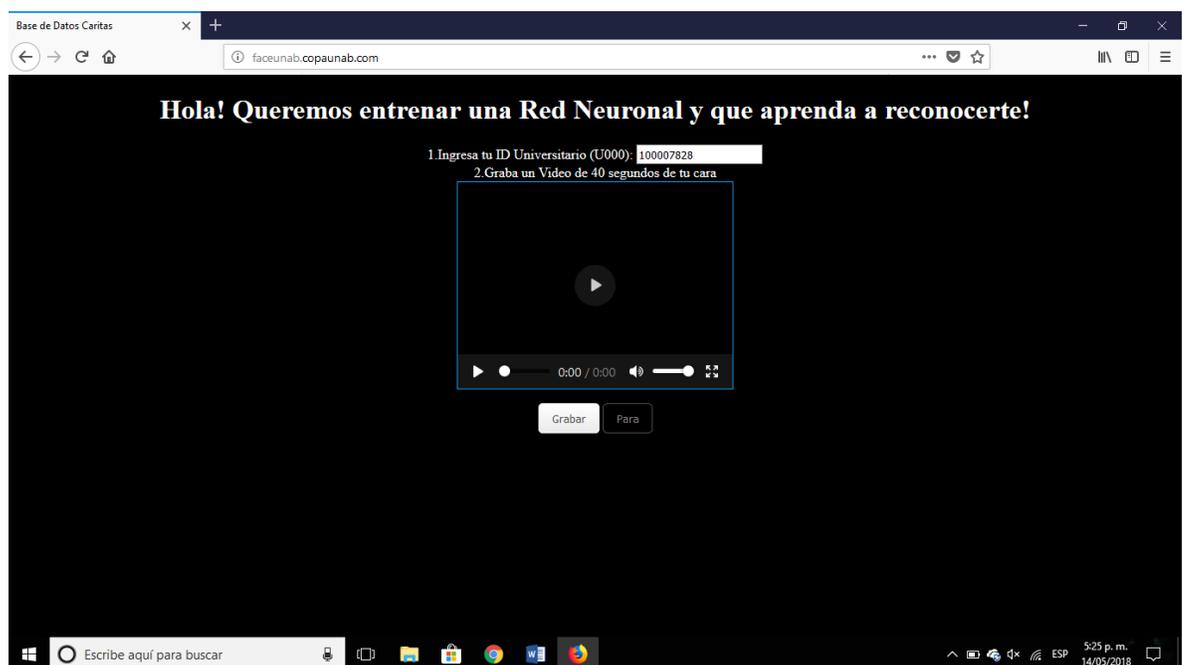


Figura 30: Página web para la captura del video
Fuente: Propia

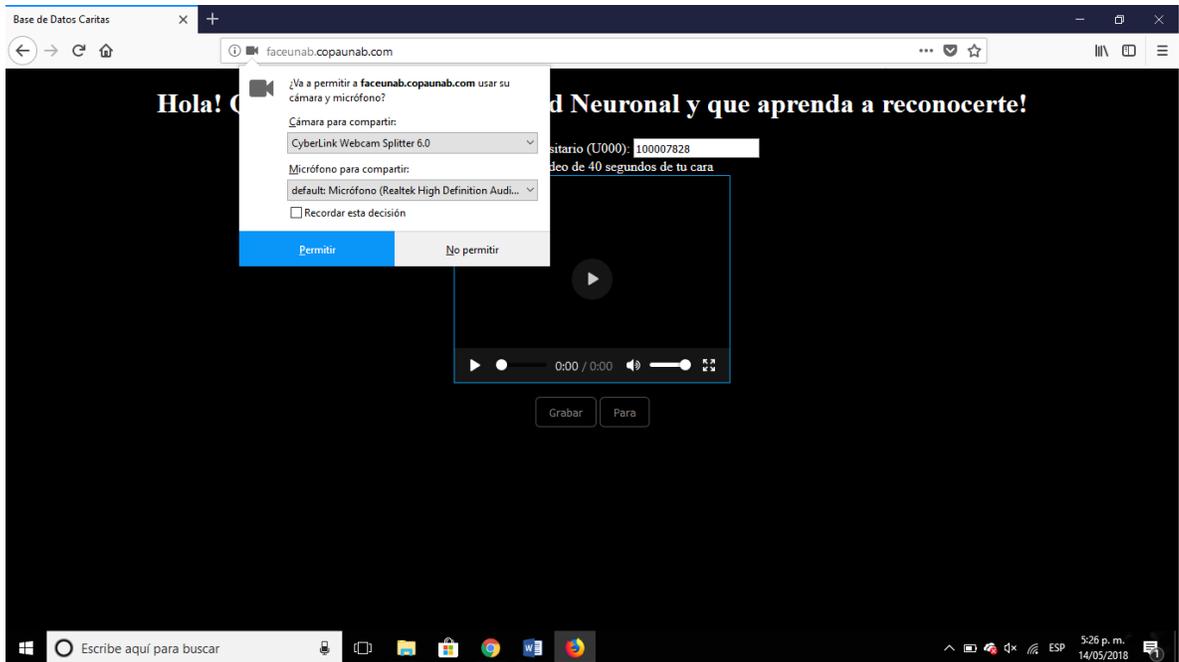


Figura 31: Solicitud de acceso a la cámara
Fuente: Propia



Figura 32: Página web con acceso a la cámara

Fuente: Propia

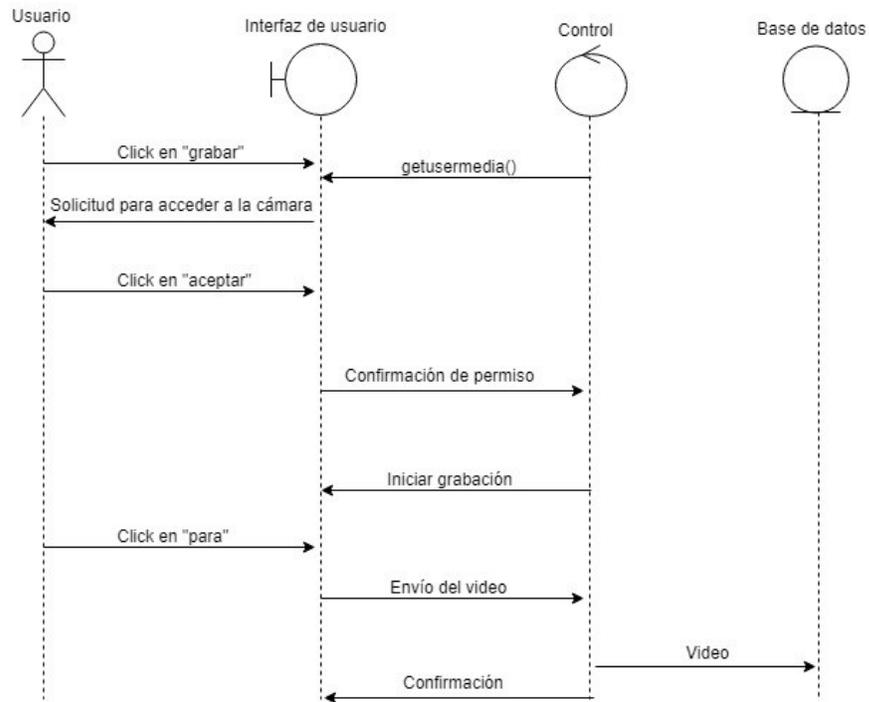


Figura 33: Diagrama de secuencia de la toma del video
Fuente: Propia

Una vez que los videos ya estén cargados en la base de datos, se ejecuta el script de perl llamado "organizador.pl" (este script se ve resumido en la figura 34) cuya función es descargar los videos de la base de datos junto con los ID de las personas a los que están asociados y los ubica en la misma carpeta donde se encuentra el script para luego ejecutar un comando ffmpeg que permite sacar los frames de los videos para convertirlos en imágenes individuales en formato .jpg y los nombra acorde al nombre del video (ID) y el número del frame siguiendo la estructura "nombreDelVideo_numeroDelFrame.jpg". El sistema saca el número de clases para construir y entrenar el modelo por medio de un query "COUNT()" que cuenta la cantidad de IDs que hay en la base de datos.

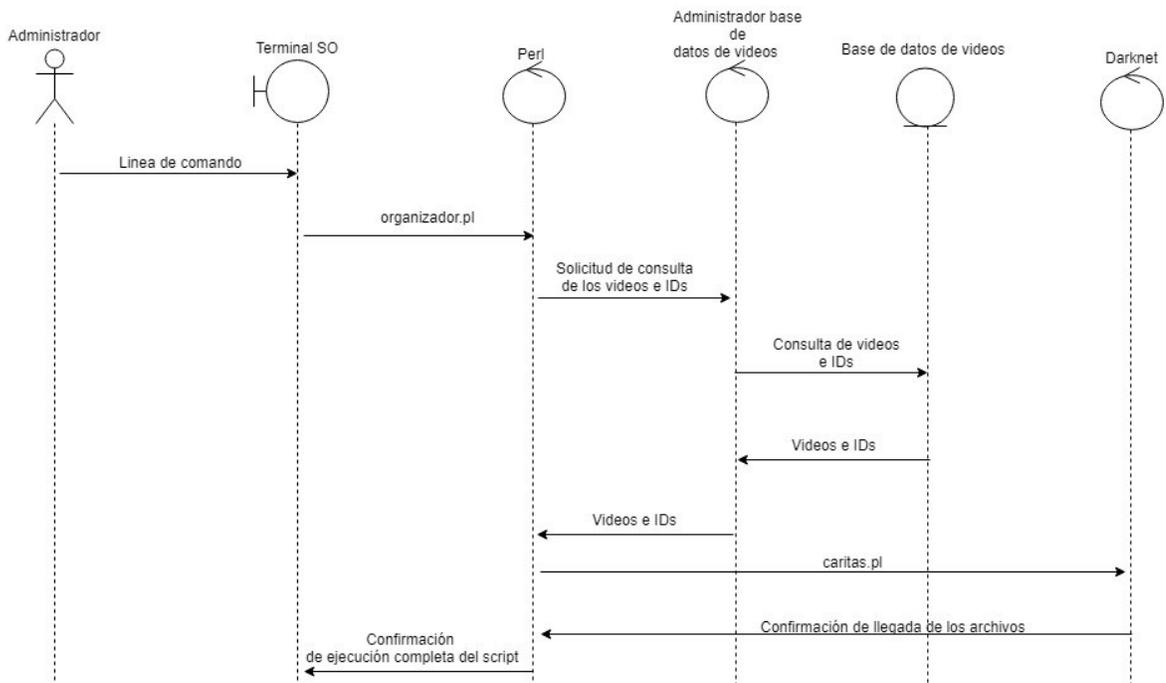


Figura 34: Diagrama de secuencia script “organizador.pl”
Fuente: Propia

9.3. Objetivo específico 3

Para automatizar el entrenamiento de la red neuronal se deben ejecutar 4 fases que son de importancia para la implementación y desarrollo del modelo. En la figura 35 se observan las fases de captura, preparación de datos, configuración del modelo y entrenamiento en su respectivo orden

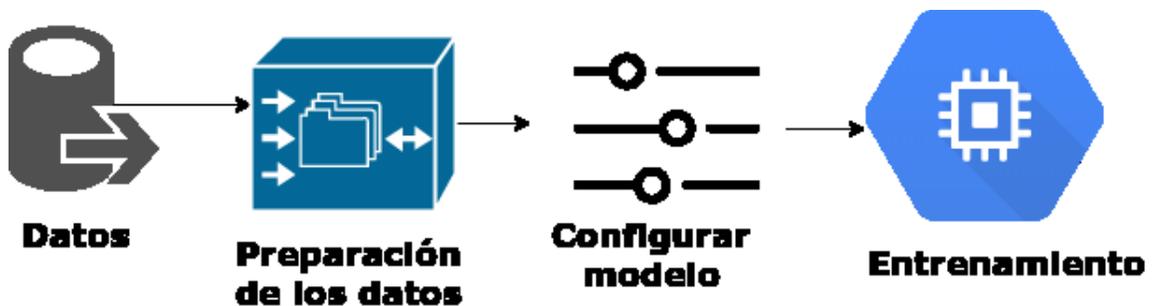


Figura 35: Procesos de nuestra metodología basado en CRISP-DM
Fuente: Propia

Fase de captura de datos: es la fase abordada en el objetivo 2 donde nuestro tipo de datos son videos que son guardados en una base de datos.

Fase de preparación de los datos: también abordada en el objetivo 2, se inicia con el tratamiento de los videos para convertirlos en imágenes, posteriormente sus frames son capturados y guardados en formato .jpg.

Fase de configuración del modelo:

Desde el script “organizador.pl”, se ejecuta el script de perl llamado “caritas.pl” el cual realiza una detección en darknet con un modelo entrenado únicamente para detectar las caras. El modelo mencionado anteriormente se entrenó de manera manual utilizando un dataset de 48000 imágenes a las cuales se les recortaron las caras. Se modificó la configuración de darknet para que al momento de detectar la cara, en lugar de dibujar un cuadro en las coordenadas, el sistema guarde las coordenadas en una matriz junto con el nombre del archivo siguiendo una estructura donde en los dos primeros campos se guardan las coordenadas X,Y y en los dos últimos campos se guarda el ancho y el alto respectivamente de la cara detectada (en la 36 se puede ver la estructura de la matriz en el script). Finalmente, el script elimina cualquier archivo .mp4, .pl o .py para liberar espacio, debido a que los archivos con estas extensiones ya no se necesitan en el siguiente paso del proceso.

```
text.= “ “.$x.” “.$y”.” “.$width.” “.$height
```

Figura 36: Estructura de la matriz en el script “caritas.pl”
Fuente: Propia

Posteriormente se debe crear el archivo “train.txt” por medio de un script llamado “listado.pl” el cual lista los directorios de los archivos de las imágenes que se extrajeron de los videos y los guarda en un solo archivo (train.txt) como se observa en la figura 37, esencialmente creando el conjunto de entrenamiento.

```
/home/migeru/darknet/x64/Release/data/img/Profesor_141.jpg
/home/migeru/darknet/x64/Release/data/img/Profesor_142.jpg
/home/migeru/darknet/x64/Release/data/img/Profesor_143.jpg
/home/migeru/darknet/x64/Release/data/img/Profesor_144.jpg
/home/migeru/darknet/x64/Release/data/img/Profesor_907.jpg
/home/migeru/darknet/x64/Release/data/img/U000110931_305.jpg
/home/migeru/darknet/x64/Release/data/img/U000110931_306.jpg
/home/migeru/darknet/x64/Release/data/img/U000110931_307.jpg
/home/migeru/darknet/x64/Release/data/img/U000110931_308.jpg
/home/migeru/darknet/x64/Release/data/img/U000110931_309.jpg
/home/migeru/darknet/x64/Release/data/img/U000110931_310.jpg
/home/migeru/darknet/x64/Release/data/img/U000110931_311.jpg
/home/migeru/darknet/x64/Release/data/img/U000110931_312.jpg
/home/migeru/darknet/x64/Release/data/img/U000110931_313.jpg
/home/migeru/darknet/x64/Release/data/img/U000110931_314.jpg
/home/migeru/darknet/x64/Release/data/img/U000110931_315.jpg
/home/migeru/darknet/x64/Release/data/img/U000110931_316.jpg
```

Figura 37: Archivo train.txt

Fuente: Propia

Luego se crea el archivo yolov2-tiny-voc.cfg (ver figura 38) el cual es la configuración del modelo, partiendo de una configuración predeterminada de darknet y es cambiada en varios parametros como el número de imágenes a analizar por evolución (Batch), el número de filtros de la última capa convolucional; dada la función $((\#coordenadas+1+\#clases)*numconvoluciones$ ó $(\#clases+5)*5$), el parámetro random=1 es agregado para darle la capacidad al modelo de cambiar los tamaños de las imágenes cada vez que se termine un batch, permitiendo así un margen de distancias probables de una cara (Profundidad con el entorno o perspectiva) sin necesidad de manipular el dataset, y el último parámetro softmax=1 habilita las capas para hacer el pre y post procesamiento con softmax al momento de detectar, lo cual permite generar detecciones más precisas con una demora de 8 segundos más de lo esperado que en este caso no afectan de manera negativa ya que lo que se busca es precisión y no velocidad.

```

[convolutional]
size=1
stride=1
pad=1
filters=125
activation=linear

[region]
anchors = 1.08,1.19, 3.42,4.41, 6.63,11.38, 9.42,5.11,
bias_match=1
classes=20
coords=4
num=5
softmax=1
jitter=.2
rescore=1

object_scale=5
noobject_scale=1
class_scale=1
coord_scale=1

absolute=1
thresh = .6
random=1

```

Figura 38: Archivo de configuración del modelo
Fuente: Propia

El script también crea el archivo obj.names, el cual contiene los nombres con los que se identifica cada clase en la fase de detección. El archivo funciona como un array, donde cada línea equivale a (N-1) al referenciar a clase (ver figura 39). Ejemplo: ID Name Profesor pertenece a la clase 9; por lo cual en el archivo estará posicionada en la línea número $10=(Clase+1)$.

```
1 U00115719
2 U00115398
3 U00115031
4 U00114914
5 U00114684
6 U00114358
7 U00113879
8 U00115000
9 U00110931
10 Profesor
```

Figura 39: Archivo obj.names
Fuente: Propia

Finalmente se crea el archivo obj.data (ver figura 40) que contiene la ubicación o directorio de los archivos que se requieren para ciertos parámetros como el archivo train.txt o el archivo que contiene las diferentes clases (obj.names).

```
classes= 10
train = train.txt
valid = train.txt
names = obj.names
backup = backup
```

Figura 40: Archivo obj.data
Fuente: Propia

Los parámetros train y valid reciben el archivo train.txt que estará ubicado en la carpeta raíz del proyecto, lo mismo el parámetro names con obj.names y la carpeta backup es referenciada y previamente creada.

Como nota aclaratoria respecto a directorios, los scripts siempre se ejecutan como carpeta Raíz en /home/user/darknet, por lo cual al referenciar solamente el nombre del archivo sin el directorio, su estructura da a entender que está dentro de la carpeta directorio en la raíz del árbol del proyecto. (Revisar con el comando en bash “tree”).

Fase de entrenamiento

El siguiente paso es ejecutar el script “train.pl” que se encarga iniciar el entrenamiento del modelo con los archivos (obj.data, obj.names, train.txt) y la configuración explicada anteriormente. El sistema termina el entrenamiento al realizar 1000 iteraciones del proceso de construcción del modelo. Para que el modelo se considere útil, debe tener una precisión

de mínimo el 85% y una pérdida de información por debajo del 50% durante su entrenamiento.

9.4. Objetivo específico 4

Luego de detectar a las personas, el sistema escribe el resultado de la detección en la base de datos. Después se extrae esta información de la base de datos para armar el reporte de asistencia, en el cual se muestra si la persona asistió o no junto con la fecha en la que se realizó la detección. El reporte está disponible en una página web desde la cual se puede descargar en formato Excel y pdf. En la figura 41 se muestra la página con el reporte y en la parte superior izquierda se pueden ver los botones que permiten descarga el reporte (en las figuras 42 y 43 se pueden ver los reportes en formato Excel y pdf respectivamente).



Figura 41: Página donde se muestra el reporte de asistencia
Fuente: Propia

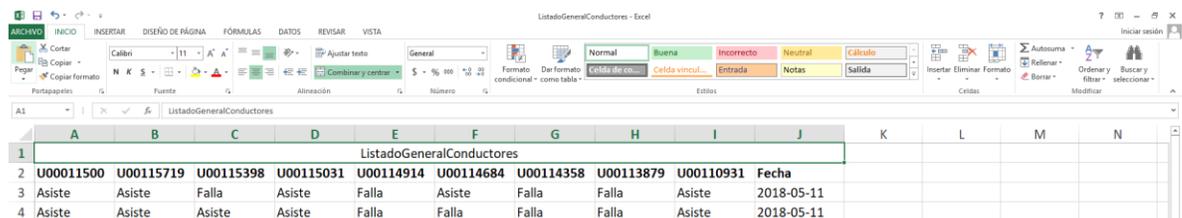


Figura 42: Reporte de asistencia en formato Excel
Fuente: Propia

The screenshot shows a PDF document titled "Listado General de Conductores" displayed in Adobe Acrobat Reader DC. The document contains a table with the following data:

U00011500	U00115719	U00115398	U00115031	U00114914	U00114684	U00114358	U00113879	U00110931	Fecha
Asiste	Asiste	Falla	Asiste	Falla	Asiste	Falla	Falla	Asiste	2018-05-11
Asiste	Asiste	Asiste	Asiste	Falla	Falla	Falla	Falla	Asiste	2018-05-11

Figura 43: reporte de asistencia en formato pdf
Fuente: Propia

9.5. Objetivo específico 5

Para realizar las pruebas del sistema, se dispuso de la participación de 9 estudiantes y un profesor. En la etapa de captura de los videos se tuvieron dos problemas con la red de la universidad; el primero, fue que la red no permitía el envío videos de más de 10 segundos desde la página debido a limitaciones del tamaño de los archivos. El segundo problema fue que, debido a problemas con el caché del servidor de la universidad no era posible enviar más de un video por dispositivo; estos problemas se hicieron evidentes luego de intentar subir videos de más de 10 segundos después de haber enviado uno más corto con el objetivo de comprobar el correcto funcionamiento de la página. Debido a lo mencionado anteriormente, la toma de los videos se hizo con un dispositivo móvil y luego se subieron los videos directamente a la base de datos. Por otra parte, se tomaron fotos de la vista general del salón de clase donde se vieran los estudiantes y el profesor.

En la fase de entrenamiento de la red neuronal se tuvo un inconveniente con la orientación de las imágenes extraídas de los videos, a pesar de que las imágenes tenían una orientación vertical (ver figura 44), al momento de realizar el entrenamiento, la red neuronal la tomó con una orientación horizontal (ver figura 45), lo cual conllevó a que el entrenamiento realizado no fuera de utilidad y por consiguiente se tuvo que hacer el entrenamiento nuevamente luego de corregir este error.

Figura 44: Imagen con
Fuente: Propia



orientación vertical



Figura 45: Imagen con orientación horizontal
Fuente: Propia

Una de las preocupaciones que se tenían era la resolución de las imágenes sobre las cuales se hace la detección de las personas una vez el modelo ya está entrenado. En un principio se tomaron fotos de 1200x720, lo cual no se creía suficiente para que la red neuronal identificase a las personas en el salón de clase (pero al final si lo fue). Para poder evadir esta posible eventualidad, se optó por tomar 12 fotos con diferentes resoluciones, más específicamente 2 fotos con una resolución de 3600x2160, 2 de 2400x1440, 2 de 1200x720, 2 de 2160x3600, 2 de 1440x2440 y 2 de 720x1200. Por otro lado, tener varias fotos de cada resolución nos permitiría identificar personas que no estuvieran en la primera o que estuvieran mirando en un ángulo desde el cual la red no es capaz de detectar.

Métricas seguidas para la evaluación del modelo

El proceso de evaluación consiste en el uso de métricas que permiten ver la consistencia del modelo.

Las métricas de evaluación son probadas con el dataset de prueba o test que en este caso es el mismo dataset de entrenamiento. El primer paso es determinar cómo fue clasificada cada detección de cada imagen de prueba; acorde a (Han, Kamber & Pei, 2012), estas clasificaciones son:

Verdaderos positivos (TP): Estas son las tuplas positivas que el clasificador marcó correctamente. El valor TP es la cantidad de positivos verdaderos.

Verdaderos negativos (TN): Estas son las tuplas negativas que el clasificador etiquetó correctamente. El valor TN es la cantidad de negativos verdaderos.

Falsos positivos (FP): Estas son las tuplas negativas que incorrectamente fueron etiquetadas como positivas. El valor FP sea la cantidad de falsos positivos.

Falsos negativos (FN): Estas son las tuplas positivas que fueron mal etiquetadas como negativas. El valor FN sea la cantidad de falsos negativos.

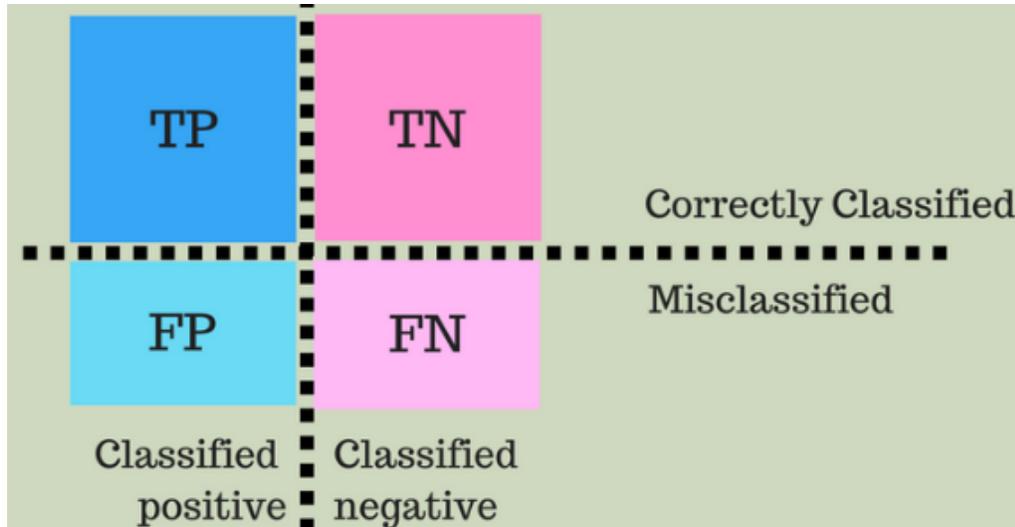
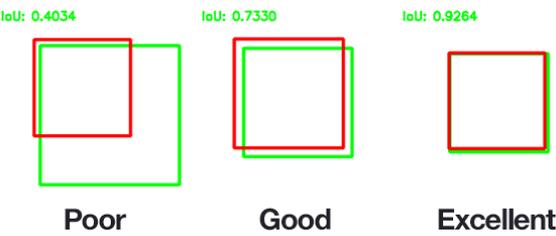


Figura 46: Posibles clasificaciones de las imágenes
Fuente: ("Classification Performance Metrics - NLP-FOR-HACKERS", 2018)

Luego de haber clasificado las imágenes, se calculan las métricas que nos darán la verdadera viabilidad del modelo. Estas métricas se muestran en la tabla 4:

Tabla 4: Métricas de evaluación del modelo

Métricas de Rendimiento	Definición y Cálculo
Precisión	Esta métrica responde a la pregunta: ¿Cuál es la probabilidad de quién es, si predijimos previamente que la muestra es esa persona? Su fórmula sería: $TP/(TP+FP)$
Mean Average Precision (mAP)	Esta métrica es el promedio de precisión de cada una de las clases. Su formulas sería: ΣP_i donde P es la precisión de cada clase
Average IoU(Promedio de Intersección sobre uniones)	Esta métrica no tiene un peso como tal si el modelo es preciso o no, este da un acercamiento sobre el comportamiento de la clasificación comparando los resultados con los recuadros delimitadores del dataset.  <p style="text-align: center;">Fuente: (Han, Kamber & Pei, 2012)</p>
Recall (Recuperación)	Esta métrica responde a la pregunta: ¿Si una muestra es positiva para la clase, cual es la probabilidad de el sistema lo capte? Su fórmula sería: $TP/(TP+FN)$
F1-Score	Esta métrica combina dos métricas previamente mencionadas: Recall + Precision. Esta se define como

Métricas de Rendimiento	Definición y Cálculo
	la media armónica de la precisión y la recuperación, siendo la que más peso en la evaluación del modelo. Su fórmula sería: $(TP+TN)/(TP+TN+FP+FN)$

Evaluación del modelo

Al hacer uso del modelo para detectar a los estudiantes y al profesor en el aula de clase, se obtuvieron resultados positivos, el sistema detectó a los 9 estudiantes y al profesor, todos con una precisión superior al 89% (ver figura 47) en un tiempo de 25 segundos.

```

detections_count = 1819, unique_truth_count = 1411
class_id = 0, name = U00115719, ap = 100.00 %
class_id = 1, name = U00115398, ap = 89.93 %
class_id = 2, name = U00115031, ap = 90.91 %
class_id = 3, name = U00114914, ap = 99.65 %
class_id = 4, name = U00114684, ap = 90.91 %
class_id = 5, name = U00114358, ap = 99.57 %
class_id = 6, name = U00113879, ap = 100.00 %
class_id = 7, name = U00115000, ap = 90.91 %
class_id = 8, name = U00110931, ap = 90.00 %
class_id = 9, name = Profesor, ap = 90.91 %
for thresh = 0.25, precision = 0.99, recall = 0.98, F1-score = 0.99
for thresh = 0.25, TP = 1389, FP = 13, FN = 22, average IoU = 84.86 %

mean average precision (mAP) = 0.942784, or 94.28 %
Total Detection Time: 25.000000 Seconds

```

Figura 47: Respuesta del comando de evaluación
Fuente: Propia

En la tabla 5 se muestran los valores de las métricas del modelo luego de haber recibido la información que se mostró en la anterior figura.

Tabla 5: Valores finales de las métricas

Métricas de Rendimiento	Resultados
Precisión	La precisión por cada clase es superior a 89.93%, contando algunas clases con el 100% de precisión se puede decir que el modelo está identificando INDIVIDUALMENTE de manera correcta y precisa.
Mean Average Precision (mAP)	El promedio de precisión es de 94.28%
Average IoU(Intersección sobre uniones)	La intersección entre el cuadrante delimitador del resultado y del dataset es de un promedio de 84.86% lo que significa casi un 15% del cuadrante resultado está por fuera del cuadrante delimitador del dataset. Esto nos permite ver que no está tan alejado el margen de dibujo de la clasificación.
Recall (Recuperación)	La recuperación del modelo es del 98% en captar la clase.
F1-Score	Esta métrica nos arroja un 99% de probabilidad de que el sistema pueda predecir TP y TN de forma correcta; teniendo solo un 1% de FP y FN

Resultado de la detección en un salón de clase

Una vez se realizó la evaluación del modelo y se comprobó que era viable, se procedió a identificar a las personas de manera individual como se observa en la figura 48 donde la red identifica al profesor con un 87% de seguridad para esta detección en particular.

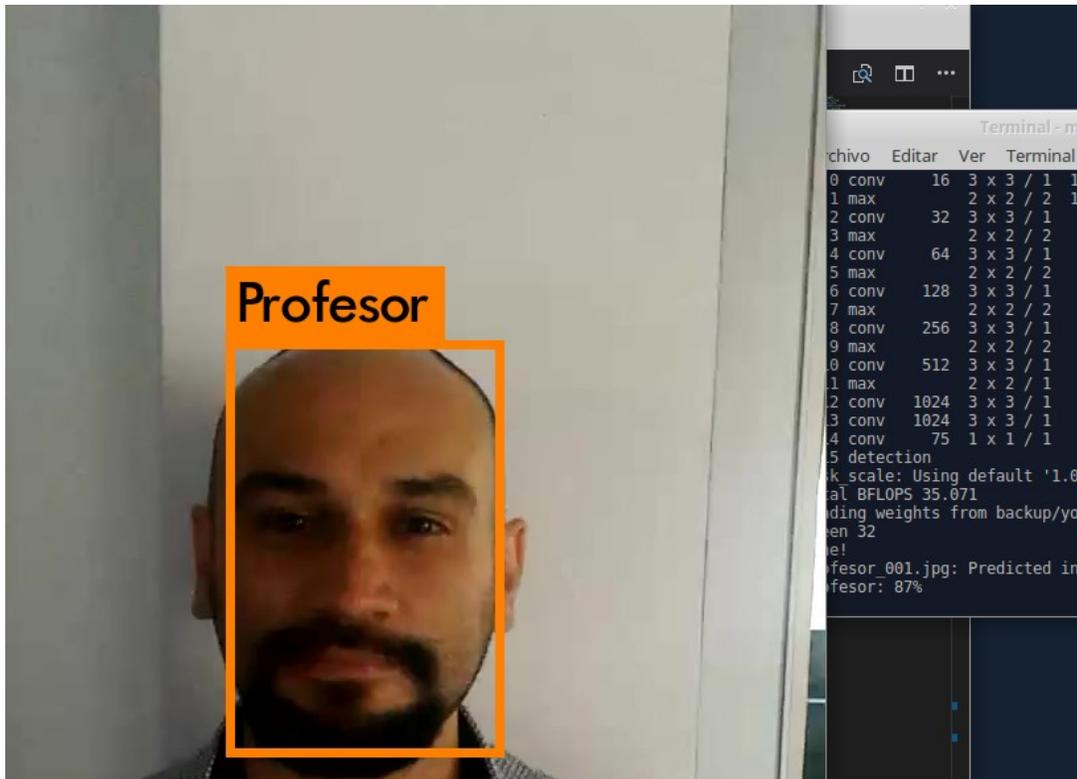


Figura 48: Profesor siendo detectado por la red neuronal
Fuente: Propia

Luego de identificar a todas las personas de manera individual, se pasó a realizar otra detección, pero esta vez en fotos de la vista general del salón. Como se observa en las figuras 49 y 50, la red fue capaz de detectar correctamente a todas las personas presentes en ese momento. Cabe resaltar que no se detectó a todos en una sola imagen, razón por la cual se toman varias imágenes y también se debe mencionar que la persona ubicada en la parte inferior derecha de las imágenes (la de la blusa gris) no se encontraba entre las personas a las que se les tomó video para el entrenamiento.

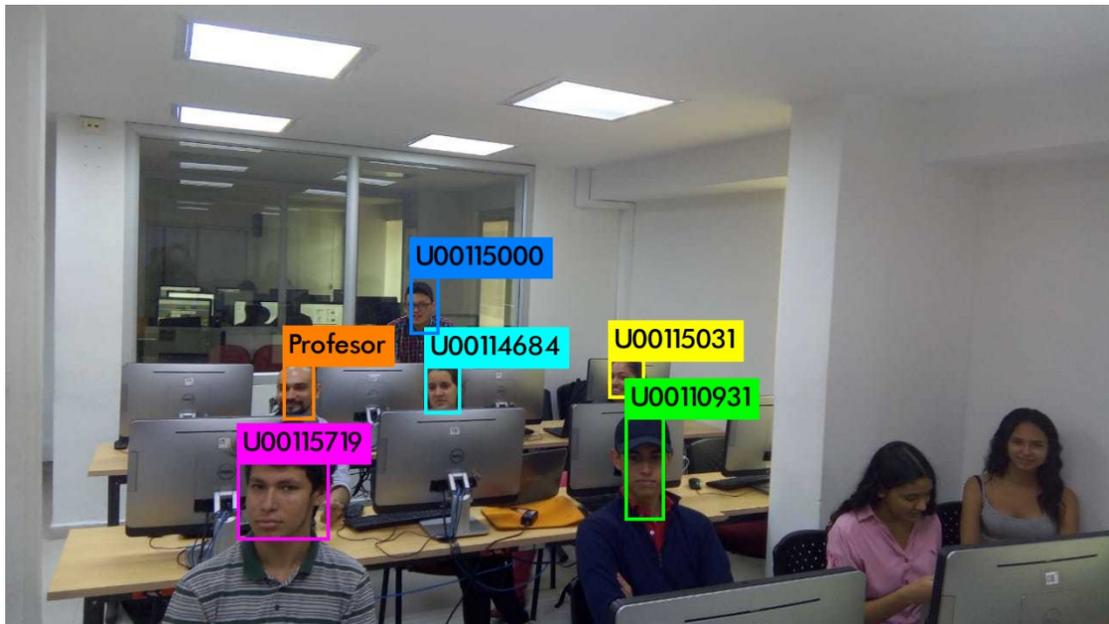


Figura 49: Detección realizada por la red (primera foto tomada)
Fuente: Propia

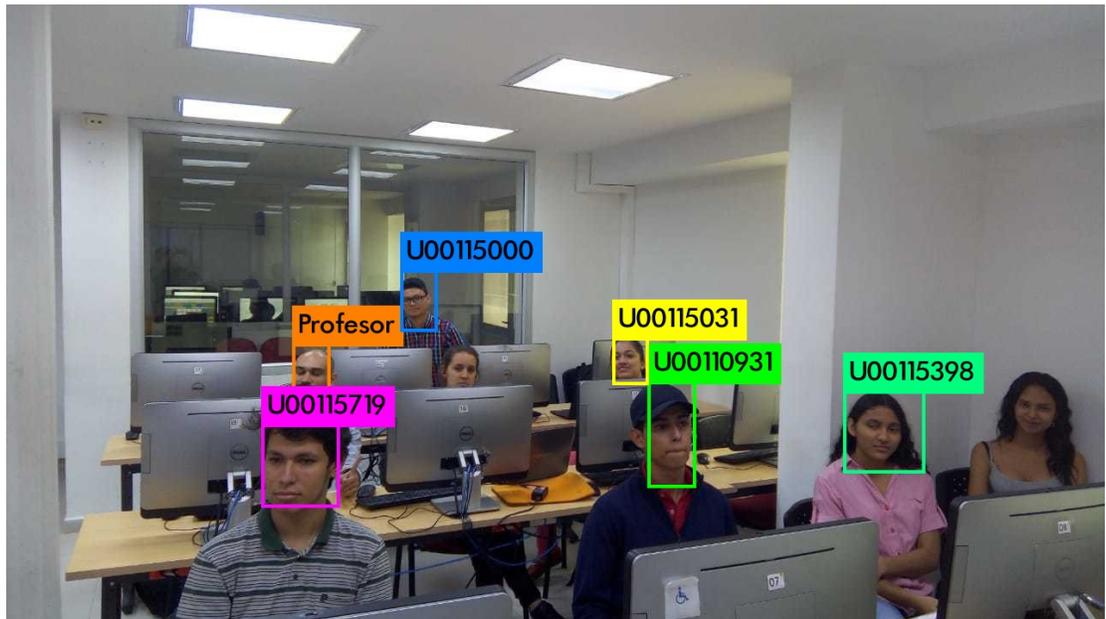


Figura 50: Detección realizada por la red (segunda foto tomada)
Fuente: Propia

En caso de implementación, lo mejor sería que el raspberry estuviera ubicado en la parte frontal del salón justo encima del tablero, como se muestra en la figura 51.

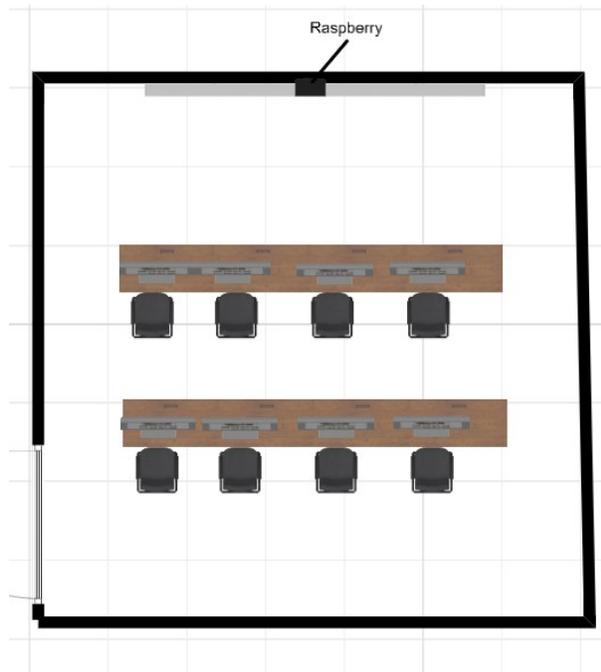


Figura 51: Ubicación óptima del raspberry en un salón de clase
Fuente: Propia

10. Conclusiones

Conclusiones desde las preguntas de investigación

Pregunta de investigación 1: ¿Hasta qué momento una red neuronal para reconocimiento facial es funcional con los recursos computacionales de un raspberry pi?

El tiempo de respuesta de detección del Raspberry es de 37 Segundos en la anterior imagen; por lo cual no hay problemas con su objetivo de detectar dado a que nuestro escenario no estaba evaluado en respuestas de tiempo real si no en respuesta por precisión y capacidad de computo.

Sin embargo, la capacidad del raspberry es excepcional; su limitante hardware nos muestra claramente que tardaría mucho tiempo en entrenar la red neuronal con su hardware y actualmente no hay una librería que permita hacer uso de su unidad de gráficos con darknet, tensorflow u otras plataformas, métodos u algoritmos de clasificación de imágenes.

Pregunta de investigación 2: ¿Hasta qué punto llega a ser requerida la computación distribuida para el buen funcionamiento de un sistema de reconocimiento facial?

Depende del escenario. En los dos escenarios probables, la computación distribuida, específicamente GPU clouds reduciría enormemente el tiempo del entrenamiento del modelo ya sea del escenario 1 o del escenario 2 (explicados en los resultados del objetivo 1).

Más que decir hasta qué punto, es decir que la computación distribuida es necesaria para la aplicación real de este tipo de sistemas ya que, para poder llegar a implementarlo en el espacio de la comunidad académica y administrativa para la captura, preparación de los datos y entrenamiento del modelo, el tiempo es récord.

Pregunta de investigación 3: ¿Cuáles son las características que debe tener un sistema de reconocimiento facial para eliminar por completo la toma de asistencia de forma manual en una clase?

No solo el sistema si no la planeación y logística por parte de los administrativos o profesionales del entre educativo que encabezarían el proceso, ya que la planeación y el cumplimiento de los tiempos es clave para terminar el modelo durante las vacaciones y dejarlo listo para el inicio de clases.

La precisión del sistema y el manejo de los tiempos se deben tener en cuenta para imprevistos como:

- Que el sistema tenga en cuenta ausencias justificadas por parte del profesor
- Que el sistema tenga en cuenta las ausencias justificadas por parte de los estudiantes
- Que el sistema tenga en cuenta las clases canceladas por motivos de actividades universitarias, mal tiempo o desastres naturales
- Que el sistema este enlazado con los cambios de horario de los profesores y aulas cuando hay recuperación de hora de clase fuera del horario y aula normales.

Conclusiones finales

Para una implementación es necesario hacer inversión en términos de hardware, no solo los raspberrys para los salones y sus cámaras, sino también en servidores que puedan soportar la carga de información que se envía al sistema tanto en la fase de

captura de datos como en la fase de detección (las fotos que toman los raspberrys) y equipos que puedan realizar los entrenamientos de las personas de manera rápida.

Por otro lado, se debe invertir una infraestructura robusta en cuanto a la parte del almacenamiento de los datos debido al gran aumento del flujo de información que se daría en caso de que se implementase el sistema. Datos como los videos, los frames que se sacan de estos, los resultados de las detecciones y los reportes de asistencia entrarían a formar parte del flujo de información de la universidad. También sería necesario capacitar al personal que estaría encargado del manejo y monitoreo del sistema.

Como conclusión final, es posible hacer un despliegue del sistema, pero sería un plan a implementar a largo plazo debido a la inversión y los cambios que se deben hacer en la infraestructura de la universidad (servidores, bases de datos, personal).

11. REFERENCIAS

Talaviya, G., Ramteke, R., & Shete, A. (2013). Wireless Fingerprint Based College Attendance System Using Zigbee Technology. *International Journal Of Engineering And Advanced Technology*, (3), 201-203. Retrieved from <https://pdfs.semanticscholar.org/a873/5eb75d3f1411798525fdc65875a8237b0c99.pdf>

Nawaz, T., Pervaiz, S., Korrani, A., & Ud-Din, A. (2009). Development of Academic Attendance Monitoring System Using Fingerprint Identification. *International Journal Of Computer Science And Network Security*, (9), 164-168. Retrieved from https://www.researchgate.net/profile/Tabassam_Nawaz/publication/242098052_Development_of_Academic_Attendance_Monitoring_System_Using_Fingerprint_Identification/links/5576abb008ae7521586c3c2b.pdf

Masalha, F., & Hirzallah, N. (2014). A Students Attendance System Using QR Code. *International Journal Of Advanced Computer Science And Applications*, (3), 75-79. Retrieved from https://thesai.org/Downloads/Volume5No3/Paper_10-A_Students_Attendance_System_Using_QR_Code.pdf

Sajid, M., Hussain, R., & Usman, M. (2014). *A conceptual Model For Automates Attendace Marking System Using Facial Recognition. Ninth International Conference on Digital Information Mangement* (Págs. 7-10). Phitsanulok: IEEE.

Methi, D., Chauhan, A., & Gupta, D. (2017). Attendance System Using Face Recognition. *International Journal Of Advanced Research In Science, Engineering And Technology*, (4), 3897-3902. Retrieved from <https://www.ijarset.com/upload/2017/may/11-IJARSET-DIVYAGUPTA.pdf>

Kawaguchi, Yohei & Shoji, Tetsuo. (2005). Face Recognition-based Lecture Attendance System, Retrieved from https://www.researchgate.net/publication/241608617_Face_Recognition-based_Lecture_Attendance_System

Balcoh, N., Yousaf, H., Ahmad, W., & Baig, I. (2012). Algorithm for Efficient Attendance Management: Face Recognition based approach. *International Journal Of Computer Science Issues*, 9(4), 146-150.

Qrcode.com. (2017). *History of QR Code | Qrcode.com | DENSO WAVE*. [online] Available at: <http://www.qrcode.com/en/history/> [Accessed 21 Sep. 2017].

Shiffman, D. (2013). *The Nature of Code*. 1st ed. Shannon Fry, p.445.

Rouse, M. (2015). *Framework. whatis*. Retrieved 23 September 2017, from <http://whatis.techtarget.com/definition/framework>

What is Zigbee?. (2014). *Zigbee Alliance*. Retrieved 23 September 2017, from <http://www.zigbee.org/what-is-zigbee/>

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

Cireşan, D., Meier, U., Masci, J., Gambardella, L. and Schmidhuber, J. (2011). Flexible, High Performance Convolutional Neural Networks for Image Classification. Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence, [online] 2, pp.1237-1242. Available at: <http://people.idsia.ch/~juergen/ijcai2011.pdf> [Accessed 20 Sep. 2017].

Gimeno Hernández, R. (2010). Estudio de técnicas de Reconocimiento facial.

Raspberry Pi. (2017). *What is a Raspberry Pi?*. [online] Available at: <https://www.raspberrypi.org/help/what-%20is-a-raspberry-pi/> [Accessed 2 Oct. 2017].

Abdallah, A. S., Abbott, A. L., & El-Nasr, M. A. (2007, May). A new face detection technique using 2D DCT and self organizing feature map. In *Proc. of World Academy of Science, Engineering and Technology* (Vol. 21, pp. 15-19)

The Database of Faces. (2002). *The Database of Faces*. [online] Available at: <http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html> [Accessed 5 Sep. 2017].

Belongie, S., Malik, J., & Puzicha, J. (2002). Shape matching and object recognition using shape contexts. *IEEE transactions on pattern analysis and machine intelligence*, 24(4), 509-522

Belhumeur, P. N., Hespanha, J. P., & Kriegman, D. J. (1997). Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on pattern analysis and machine intelligence*, 19(7), 711-720.

Adini, Y., Moses, Y., & Ullman, S. (1997). Face recognition: The problem of compensating for changes in illumination direction. *IEEE Transactions on pattern analysis and machine intelligence*, 19(7), 721-732.

Phillips, P. J. (1999). Support vector machines applied to face recognition. In *Advances in Neural Information Processing Systems*(pp. 803-809).

Liu, C., & Wechsler, H. (2000). Evolutionary pursuit and its application to face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(6), 570-582.

Gordon, G. G. (1991, September). Face recognition based on depth maps and surface curvature. In *Geometric Methods in Computer Vision* (Vol. 1570, pp. 234-248). International Society for Optics and Photonics.

Lawrence, S., Giles, C. L., Tsoi, A. C., & Back, A. D. (1997). Face recognition: A convolutional neural-network approach. *IEEE transactions on neural networks*, 8(1), 98-113.

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Courbariaux, M., Hubara, I., Soudry, D., El-Yaniv, R., & Bengio, Y. (2016). Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1. *arXiv preprint arXiv:1602.02830*.

Shyam, R., & Singh, Y. N. (2014, December). Evaluation of eigenfaces and fisherfaces using bray curtis dissimilarity metric. In *Industrial and Information Systems (ICIIS), 2014 9th International Conference on* (pp. 1-6). IEEE.

Redmon, J., & Angelova, A. (2015, May). Real-time grasp detection using convolutional neural networks. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on* (pp. 1316-1322). IEEE.

Hongxun, Y., Wen, G., Mingbao, L., & Lizhuang, Z. (2000). Eigen features technique and its application. In *Signal Processing Proceedings, 2000. WCCC-ICSP 2000. 5th International Conference on* (Vol. 2, pp. 1153-1158). IEEE.

Bedre, J. S., & Sapkal, S. (2012). Comparative Study of Face Recognition Techniques: A Review. *Emerging Trends in Computer Science and Information Technology–2012 (ETCSIT2012) Proceedings published in International Journal of Computer Applications®(IJCA)*, 12.

Heiselet, B., Serre, T., Pontil, M., & Poggio, T. (2001). Component-based face detection. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on* (Vol. 1, pp. I-I). IEEE.

Virdee-Chapman, B. (2017). *Face Recognition: Kairos vs Microsoft vs Google vs Amazon vs OpenCV. Kairos*. Retrieved 30 October 2017, from <https://www.kairos.com/blog/face-recognition-kairos-vs-microsoft-vs-google-vs-amazon-vs-opencv>

Han, J., Kamber, M., & Pei, J. (2012). *Data Mining Concepts and Techniques* (3rd ed.). Elsevier.

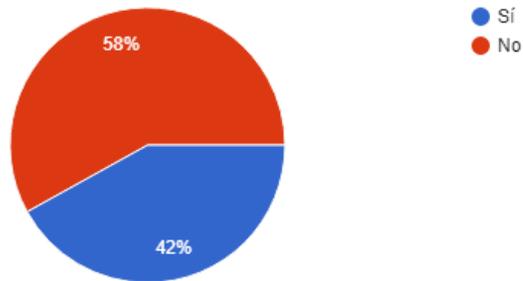
Classification Performance Metrics - NLP-FOR-HACKERS. (2018). Retrieved from <https://nlpforhackers.io/classification-performance-metrics/>

12. Anexos

Anexo 1: Resultados de la encuesta realizada a estudiantes

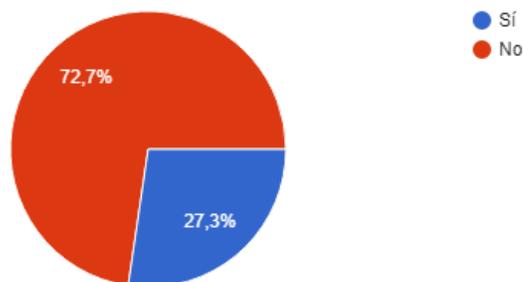
Le genera incomodidad la toma de asistencia?

205 respuestas



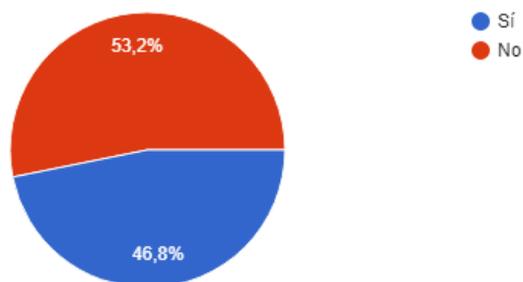
Ha tenido reportes de inasistencia falsos por parte de un docente?

205 respuestas



Considera que se pierde el tiempo de clase en la toma de asistencia?

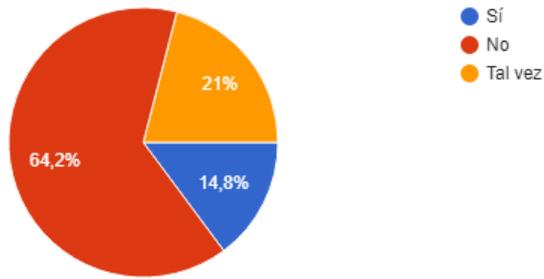
205 respuestas



Anexo 2: Resultados de la encuesta realizada a docentes

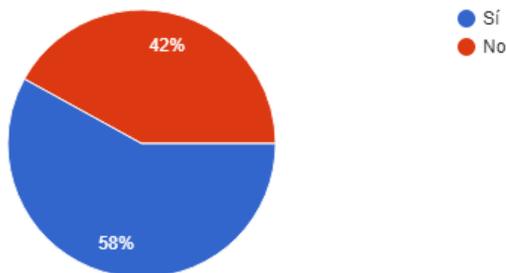
Considera que se pierde tiempo de clase en la toma de asistencia?

81 respuestas



Considera que el proceso de la Auditoría de Control de asistencia docente se maneja apropiadamente

81 respuestas



Ha tenido reportes de inasistencia falsos por parte de auditoría? O problemas en general con el proceso?

81 respuestas

