



FACULTAD DE INGENIERÍA Y ARQUITECTURA
ESCUELA PROFESIONAL DE INGENIERÍA ELECTRÓNICA

**DESARROLLO DE UN SISTEMA BIOMÉTRICO ORIENTADO AL
RECONOCIMIENTO DE PERSONAS A PARTIR DE LOS
PLIEGUES PALMARES UTILIZANDO PROCESAMIENTO
DIGITAL DE IMÁGENES Y REDES NEURONALES
CONVOLUCIONALES**

**PRESENTADA POR
JOSÉ FERNANDO DUEÑAS TIRADO
JOHN LEONARD LOPEZ SALAS**

ASESOR

**GUILLERMO LEOPOLDO KEMPER VÁSQUEZ
FERNANDO MAURICIO JIMENES MOTTE**

TESIS

PARA OPTAR EL TÍTULO PROFESIONAL DE INGENIERO ELECTRÓNICO

LIMA – PERÚ

2019



CC BY-NC-SA

Reconocimiento – No comercial – Compartir igual

El autor permite transformar (traducir, adaptar o compilar) a partir de esta obra con fines no comerciales, siempre y cuando se reconozca la autoría y las nuevas creaciones estén bajo una licencia con los mismos términos.

<http://creativecommons.org/licenses/by-nc-sa/4.0/>



USMP
UNIVERSIDAD DE
SAN MARTÍN DE PORRES

**FACULTAD DE
INGENIERÍA Y ARQUITECTURA**

ESCUELA PROFESIONAL DE INGENIERÍA ELECTRÓNICA

**DESARROLLO DE UN SISTEMA BIOMÉTRICO ORIENTADO
AL RECONOCIMIENTO DE PERSONAS A PARTIR DE LOS
PLIEGUES PALMARES UTILIZANDO PROCESAMIENTO
DIGITAL DE IMÁGENES Y REDES NEURONALES
CONVOLUCIONALES**

TESIS

**PARA OPTAR EL TÍTULO PROFESIONAL DE INGENIERO
ELECTRÓNICO**

PRESENTADA POR

DUEÑAS TIRADO, JOSÉ FERNANDO

LOPEZ SALAS, JOHN LEONARD

LIMA-PERÚ

2019

Dedico a Dios que me otorgó fuerzas para culminar mi carrera, a mi familia que me brindó su apoyo incondicional, a mis amigos que me dieron compañerismo y a mis maestros que me impartieron sus conocimientos.

José Dueñas T.

Dedico la presente tesis a las personas más maravillosas que Dios pudo poner en mi camino, mis padrinos Ron y Kathy Cline, quienes me brindaron la oportunidad de poder tener una carrera universitaria en una universidad prestigiosa.

John Lopez S.

ÍNDICE

	Página
RESUMEN	xviii
ABSTRACT	xix
INTRODUCCIÓN	xx
CAPÍTULO I. PLANTEAMIENTO DEL PROBLEMA	
1.1 Situación problemática	1
1.2 Definición del problema	3
1.3 Objetivos	3
1.4 Justificación	4
1.5 Limitaciones	6
1.6 Viabilidad	6
CAPÍTULO II. MARCO TEÓRICO	
2.1 Antecedentes de la Investigación	8
2.2 Estado del Arte	11
2.3 Bases Teóricas	24
CAPÍTULO III. METODOLOGÍA	
3.1 Materiales	101
3.2 Método	106
CAPÍTULO IV. DESARROLLO	
4.1 Diagrama de bloques	109
4.2 Diseño del recinto de adquisición de imágenes	110

4.3 Desarrollo de la etapa de preprocesamiento	116
4.4 Diseño de la arquitectura neuronal	125
4.5 Diseño de la interfaz gráfica (GUI)	149
CAPÍTULO V PRUEBAS Y RESULTADOS	
5.1 Pruebas	154
5.2 Cálculo de las tasas de verdaderos reconocimientos y falsos rechazos para cada usuario	182
5.3 Cálculo de la tasa de verdaderos reconocimientos y falsos rechazos globales para la red	199
5.4 Costos en inversión del producto final	202
CONCLUSIONES	203
RECOMENDACIONES	205
GLOSARIO	206
FUENTES DE INFORMACIÓN	210

ÍNDICE DE FIGURAS

	Página
Figura N°1. Sistema biométrico dactilar de la Sede Santa Anita de la USMP	1
Figura N°2. Proceso de generalización de la información en las capas del Neocognitrón de Fukushima para una entrada visual	9
Figura N°3. Arquitectura jerárquica de LeNet-5, propuesta por Yann LeCun	10
Figura N°4. Representación de un dermatoglifo localizado en la región dactilar	25
Figura N°5. Pliegues de la región palmar de una mano humana	26
Figura N°6. Representación de los intervalos de longitud de onda del espectro electromagnético	27
Figura N°7. Fotones que representan a la luz como Partícula	28
Figura N°8. Propagación de la luz como una onda en un espacio tridimensional	28
Figura N°9. Cámara utilizada para la adquisición de las imágenes de las palmas del personal	29
Figura N°10. Proceso de obtención de una imagen digital	30
Figura N°11. Representación de una imagen capturada, la amplitud de dicha imagen a la largo de un segmento AB, el muestreo de la imagen y la cuantificación correspondiente	32
Figura N°12. Modelo generalizado no-lineal de una neurona	42

Figura N°13. Modelo generalizado no-lineal de una neurona con el <i>bias</i> incorporado como un peso sináptico más	43
Figura N°14. Apreciación gráfica de la función de activación ReLU	45
Figura N°15. Arquitectura <i>feed forward</i> de una red neuronal artificial	48
Figura N°16. Representación gráfica de cómo el algoritmo del descenso de gradiente se empela para seguir a una función cuesta abajo hacia un mínimo	56
Figura N°17. Entrenamiento de una red neural multicapa con distintos optimizadores	61
Figura N°18. Arquitectura de una red neuronal convolucional	62
Figura N°19. Entradas de una RNC arregladas en forma de Matriz	64
Figura N°20. Campo local receptivo de una neurona en la capa oculta	65
Figura N°21. Desplazamiento del <i>Kernel</i> o filtro con <i>stride</i> = 1	66
Figura N°22. Imagen RGB de entrada que produce varios mapas de salida como consecuencia de su convolución con distintos <i>kernels</i>	70
Figura N°23. Capa de convolución seguida de una capa de reducción o <i>pooling</i>	70
Figura N°24. La unidad (2,2) de la capa de <i>pooling</i> (derecha) resume un área 2x2 del mapa de características (izquierda)	71
Figura N°25. Función <i>Max-Pool</i> que toma la activación más grande en un área de 2x2	72
Figura N°26. Un arreglo matricial es convertido a un arreglo de una sola columna gracias a <i>Flattening</i>	73
Figura N°27. Arquitectura de una RNC simple	74
Figura N°28. Región palmar iluminada por el arreglo de LED's posicionados alrededor de la cámara Logitech C-922	104
Figura N°29. Tarjeta gráfica Nvidia modelo GTX 1060	105
Figura N°30. Tiempo que le toma a la tarjeta gráfica completar una iteración de una RNC con 5760 muestras de entrenamiento	106
Figura N°31. Diagrama de bloques del presente trabajo	110

Figura N°32. Plantilla palmar donde se colocan las manos derechas de los usuarios	111
Figura N°33. Promedios de la longitud y ancho de las manos de hombres chilenos cuyas edades oscilan entre los 18 y 64 años	112
Figura N°34. Vista superior de la plataforma palmar que contiene el molde o plantilla palmar	114
Figura N°35. Visa tridimensional en AutoCAD del recinto de adquisición de muestras palmares con la plataforma palmar ya acoplada	115
Figura N°36. Vista tridimensional en AutoCAD del modelo del recinto de adquisición de imágenes con la puerta y la plataforma palmar ya acopladas	116
Figura N°37. Imagen HSV de una muestra palmar con su región de interés intacta y la versión HSV de la misma imagen con su región de interés anulada	117
Figura N°38. Diagrama de flujo que sigue el proceso de para anular la región central de la palma de una mano	118
Figura N°39. Diagrama de flujo para el cambio del valor de los píxeles de mascara.jpg	119
Figura N°40. Comparación entre mascara.jpg y la versión binarizada de la misma imagen tras aplicarse el algoritmo de cambio de píxeles	120
Figura N°41. Multiplicación entre la imagen de la izquierda y la máscara para obtener la nueva muestra palmar solo con la región de interés	121
Figura N°42. Imagen de una muestra palmar de dimensiones 960x720, en donde se distingue la información relevante de la irrelevante	122
Figura N°43. Representación del recorte de una muestra palmar trazando líneas verticales y horizontales en las coordenadas de color rojo	123
Figura N°44. Diagrama de flujo para el recorte de las muestras palmares de 960x720	123

Figura N°45. Resultado del recorte de la muestra de 960x720	124
Figura N°46. Secuencia para reescalar y normalizar las muestras de 500x570	125
Figura N°47. Factor de aprendizaje alto genera divergencia, mientras que un factor de aprendizaje bajo permite la convergencia de la función de coste hacia el punto mínimo global	127
Figura N°48. Curvas de precisión y curvas de la función de pérdida correspondientes al Modelo 1	128
Figura N°49. Resumen de la arquitectura del modelo neural 1	129
Figura N°50. Curvas de precisión y curvas de la función de pérdida correspondientes al Modelo 2	130
Figura N°51. Resumen de la arquitectura del modelo neural 2	131
Figura N°52. Curvas de precisión y curvas de la función de pérdida correspondientes al Modelo 3	132
Figura N°53. Resumen de la arquitectura del modelo neural 3	133
Figura N°54. Curvas de precisión y curvas de la función de pérdida correspondientes al Modelo 4	134
Figura N°55. Resumen de la arquitectura del modelo neural 4	135
Figura N°56. Curvas de precisión y curvas de la función de pérdida correspondientes al Modelo 5	136
Figura N°57. Curvas de precisión y curvas de la función de pérdida correspondientes al Modelo 6	137
Figura N°58. Curvas de precisión y curvas de la función de pérdida correspondientes al Modelo 7	138
Figura N°59. Curvas de precisión y curvas de la función de pérdida correspondientes al Modelo 8	139
Figura N°60. Curvas de precisión y curvas de la función de pérdida correspondientes al Modelo 9	141
Figura N°61. Curvas de precisión y curvas de la función de pérdida correspondiente al Modelo 10	142

Figura N°62. Curvas de precisión y curvas de la función de pérdida correspondientes al Modelo 11	143
Figura N°63. Curvas de precisión y curvas de la función de pérdida correspondientes al Modelo 12	144
Figura N°64. Resumen de la arquitectura del modelo neural 12	145
Figura N°65. Tiempo que le toma a la arquitectura 12 para completar una iteración	145
Figura N°66. Curvas de precisión y curvas de la función de pérdida correspondiente al Modelo 13	146
Figura N°67. Curvas de precisión y curvas de la función de pérdida correspondiente al Modelo 14	148
Figura N°68. Tiempo que le toma a la arquitectura 14 para completar una iteración	148
Figura N°69. Número de parámetros para la arquitectura del modelo 14	149
Figura N°70. Diagrama de flujo de la GUI del presente trabajo	151
Figura N°71. Representación gráfica de la capa de seguridad implementada en la GUI para validar el registro de asistencia de un usuario	152
Figura N°72. Representación gráfica de la GUI y su predicción, siendo utilizada por un usuario	153
Figura N°73. Gráfico de barras para los verdaderos reconocimientos en 83 usuarios	200
Figura N°74. Gráfico de barras para los falsos rechazos en 83 usuarios	200
Figura N°75. Gráfico de barras para el TAR de cada usuario	201
Figura N°76. Gráfico de barras para el FRR de cada usuario	201

ÍNDICE DE TABLAS

	Página
Tabla N°1. Tabla de parámetros empleados para el preprocesado	98
Tabla N°2. Tabla de parámetros empleados para el procesado	99
Tabla N°3. Tabla de parámetros empleados para el procesado (continuación)	100
Tabla N°4. Especificaciones técnicas de la cámara digital Logitech C-922	102
Tabla N°5. Especificaciones técnicas de los LED empleados en el presente trabajo	103
Tabla N°6. Algunas especificaciones técnicas de la tarjeta gráfica Nvidia GTX 1060	106
Tabla N°7. Resultados de las pruebas para el usuario José Dueñas	154
Tabla N°8. Resultados de las pruebas para el usuario John Lopez	155
Tabla N°9. Resultados de las pruebas para el usuario Abigail Álvarez	155
Tabla N°10. Resultados de las pruebas para el usuario Alejandro Ríos	155
Tabla N°11. Resultados de las pruebas para el usuario Alex Moreno	156
Tabla N°12. Resultados de las pruebas para el usuario Angie Villalba	156

Tabla N°13. Resultados de las pruebas para el usuario Anthony Salas	156
Tabla N°14. Resultados de las pruebas para el usuario Christian Mendoza	157
Tabla N°15. Resultados de las pruebas para el usuario Clarisa Penedo	157
Tabla N°16. Resultados de las pruebas para el usuario Danita Alcántara	157
Tabla N°17. Resultados de las pruebas para el usuario Carmen Sánchez	158
Tabla N°18. Resultados de las pruebas para el usuario Emanuel Salas	158
Tabla N°19. Resultados de las pruebas para el usuario Gaston Navas	158
Tabla N°20. Resultados de las pruebas para el usuario Ines Mori	159
Tabla N°21. Resultados de las pruebas para el usuario José Carlos Román	159
Tabla N°22. Resultados de las pruebas para el usuario Juan Zapata	159
Tabla N°23. Resultados de las pruebas para el usuario Luis Cornejo	160
Tabla N°24. Resultados de las pruebas para el usuario María Bandera	160
Tabla N°25. Resultados de las pruebas para el usuario Milagrito Bandera	160
Tabla N°26. Resultados de las pruebas para el usuario Patricia Villar	161
Tabla N°27. Resultados de las pruebas para el usuario Pierina Sánchez	161

Tabla N°28. Resultados de las pruebas para el usuario Ricardo Tapullima	161
Tabla N°29. Resultados de las pruebas para el usuario Rosario Quiñonez	162
Tabla N°30. Resultados de las pruebas para el usuario Santiago Quiñonez	162
Tabla N°31. Resultados de las pruebas para el usuario Luis Laurente	162
Tabla N°32. Resultados de las pruebas para el usuario Wilman Fernández	163
Tabla N°33. Resultados de las pruebas para el usuario Yeng Inga Bereche	163
Tabla N°34. Resultados de las pruebas para el usuario Yocce Jupa	163
Tabla N°35. Resultados de las pruebas para el usuario Aracelli Pariona	164
Tabla N°36. Resultados de las pruebas para el usuario Aron Bustamante	164
Tabla N°37. Resultados de las pruebas para el usuario Benjamina Ruiz	164
Tabla N°38. Resultados de las pruebas para el usuario Brandon Norabuena	165
Tabla N°39. Resultados de las pruebas para el usuario Celia Curi	165
Tabla N°40. Resultados de las pruebas para el usuario Charley Lovatón	165
Tabla N°41. Resultados de las pruebas para el usuario Claudia Carbajal	166
Tabla N°42. Resultados de las pruebas para el usuario Daniel Colquichahua	166

Tabla N°43. Resultados de las pruebas para el usuario Deimer Ortiz	166
Tabla N°44. Resultados de las pruebas para el usuario Deysi Mozombite	167
Tabla N°45. Resultados de las pruebas para el usuario Dr. Ballona	167
Tabla N°46. Resultados de las pruebas para el usuario Rocía Gonzáles	167
Tabla N°47. Resultados de las pruebas para el usuario Edson Avellaneda	168
Tabla N°48. Resultados de las pruebas para el usuario Edwin Ramírez	168
Tabla N°49. Resultados de las pruebas para el usuario Enrique Alva	168
Tabla N°50. Resultados de las pruebas para el usuario Enrique Belleza	169
Tabla N°51. Resultados de las pruebas para el usuario Frank Salas	169
Tabla N°52. Resultados de las pruebas para el usuario Gian Carlos Lupo	169
Tabla N°53. Resultados de las pruebas para el usuario Gian Carlos Moya	170
Tabla N°54. Resultados de las pruebas para el usuario Gian Carlos Mori	170
Tabla N°55. Resultados de las pruebas para el usuario Gerson Calle	170
Tabla N°56. Resultados de las pruebas para el usuario Gustavo Heredia	171
Tabla N°57. Resultados de las pruebas para el usuario Heather García	171

Tabla N°58. Resultados de las pruebas para el usuario Hilda Cáceres	171
Tabla N°59. Resultados de las pruebas para el usuario Jacky Poma	172
Tabla N°60. Resultados de las pruebas para el usuario Jaime Lozada	172
Tabla N°61. Resultados de las pruebas para el usuario Jessica Girano	172
Tabla N°62. Resultados de las pruebas para el usuario Jorge García	173
Tabla N°63. Resultados de las pruebas para el usuario Juan Diego Miljanovich	173
Tabla N°64. Resultados de las pruebas para el usuario Luis Bautista	173
Tabla N°65. Resultados de las pruebas para el usuario Luis Fernando Coral	174
Tabla N°66. Resultados de las pruebas para el usuario Luis Flores	174
Tabla N°67. Resultados de las pruebas para el usuario Manuel Ortiz	174
Tabla N°68. Resultados de las pruebas para el usuario Marco Julcapoma	175
Tabla N°69. Resultados de las pruebas para el usuario Marco Mendoza	175
Tabla N°70. Resultados de las pruebas para el usuario Miguel Carhuapuma	175
Tabla N°71. Resultados de las pruebas para el usuario Nat Palace	176
Tabla N°72. Resultados de las pruebas para el usuario Natividad Aguirre	176

Tabla N°73. Resultados de las pruebas para el usuario Nelly Terán	176
Tabla N°74. Resultados de las pruebas para el usuario Nicolás Vega	177
Tabla N°75. Resultados de las pruebas para el usuario Perci Tineo	177
Tabla N°76. Resultados de las pruebas para el usuario Javier Perleche	177
Tabla N°77. Resultados de las pruebas para el usuario Polo Paredes	178
Tabla N°78. Resultados de las pruebas para el usuario Guillermo Kemper	178
Tabla N°79. Resultados de las pruebas para el usuario Renato Li Toso	178
Tabla N°80. Resultados de las pruebas para el usuario Roger Cuzcano	179
Tabla N°81. Resultados de las pruebas para el usuario Rosario Sialer	179
Tabla N°82. Resultados de las pruebas para el usuario Rosita Martínez	179
Tabla N°83. Resultados de las pruebas para el usuario Sr. Loaiza	180
Tabla N°84. Resultados de las pruebas para el usuario Juan Lorgio	180
Tabla N°85. Resultados de las pruebas para el usuario Alejandro Reaño	180
Tabla N°86. Resultados de las pruebas para el usuario Renzo Tello	181
Tabla N°87. Resultados de las pruebas para el usuario Vicky Quispe	181

Tabla N°88. Resultados de las pruebas para el usuario Yolanda Sánchez	182
Tabla N°89. Resultados de las pruebas para el usuario Stefany Onofre	182
Tabla N°90. Tabla del costo final del trabajo	202

RESUMEN

La presente tesis desarrolló un sistema biométrico cuyo objetivo que fuese capaz de identificar a un universo de 80 personas por medio de sus pliegues o líneas palmares, utilizando procesamiento digital de imágenes y redes neuronales convolucionales. Este sistema biométrico ha sido orientado a ser una alternativa viable a los sistemas biométricos dactilares que son muy populares hoy en día, pero que presentan fallas frente a la deformación y distorsión de los patrones dactilares. Para ello se empleó una metodología que consistía en el diseño de un recinto de adquisición de imágenes hecho de melamina al cual se le acopló una estructura posicionadora de manos; que alberga una cámara digital y un arreglo de *Light Emitting Diodes* (LEDs) de luz blanca.

Las imágenes *Red Green Blue* (RGB) adquiridas con la cámara fueron preprocesadas en cuatro pasos, los cuales fueron: aplicación de máscara, delimitación de la región de interés (ROI), reescalamiento y normalización. Las imágenes RGB preprocesadas fueron luego ingresadas a una red neuronal convolucional diseñada en lenguaje *Python* por los propios autores. Finalmente, se diseñó una interfaz gráfica en *Python* que permitió visualizar la predicción hecha por la red neuronal convolucional y que facilitó el registro de los usuarios.

Palabras clave: Biometría, redes neuronales convolucionales, procesamiento digital de imágenes, Python.

ABSTRACT

The following thesis developed a biometric system that would be capable of identifying 80 individuals through their hand palm lines using digital image processing and convolutional neural networks. The system has been oriented at being a viable alternative to the fingerprint biometric systems which are so popular nowadays, but that present faults in the face of distortion of the fingerprint patterns. For this, a methodology was used that consisted of the design of an image acquisition enclosure made of melamine to which a hand positioning structure was attached; it houses a digital camera and an array of white Light Emitting Diodes (LEDs).

The Red Green Blue (RGB) acquired images were then preprocessed in four steps which were: mask application, Region of Interest (ROI) delimitation, rescaling and normalization. The RGB preprocessed images were then entered into a convolutional neural network which was designed in Python by the authors of the thesis. Finally a graphic interface was designed in Python to allow visualization of the network's predictions and to enable user's registration.

Keywords: *Biometry, convolutional neural networks, digital image processing, Python.*

INTRODUCCIÓN

Hoy en día se ha extendido el uso de sistemas biométricos para la verificación de la identidad de una persona alrededor del mundo, tanto en entidades privadas como públicas. En algunas de ellas incluyen: bancos, universidades, hospitales y clínicas, notarías, empresas de telefonía, bases militares, centros de educación e investigación, etc.

Existen diversos tipos de sistemas biométricos conocidos en el mercado como los sistemas dactilares, los vasculares, los de la geometría de la mano, los de voz, etc. El sistema biométrico más utilizado es el de verificación de huellas dactilares debido a que no se trata de un método de verificación de identidad invasivo; además, de que es un sistema accesible económicamente y de fácil implementación y uso.

Finalmente, la tesis presenta una estructura en cinco capítulos. En el Capítulo I: Planteamiento del problema, se hace referencia al inconveniente que presenta el sistema biométrico existente para reconocer al personal que labora en el laboratorio de Odontología de la Universidad de San Martín de Porres (USMP), por lo que se ideó un sistema biométrico alternativo que reconociera a los individuos por medio de sus pliegues palmares. En el Capítulo II: Marco teórico se evidencia el aporte científico de todas las fuentes recopiladas por los autores. En el Capítulo III: Metodología, se diseñó de un recinto de adquisición de huellas palmares y se aplicaron técnicas de procesamiento digital de imágenes y de clasificación de patrones con redes neuronales convolucionales con ayuda del lenguaje de programación Python de manera que el sistema pudiera cumplir con su objetivo. En el Capítulo IV: Desarrollo, se detallan los pasos seguidos por los autores para cumplir con lo

descrito en el método. En el Capítulo V: Pruebas y resultados, se probó el sistema diseñado por los autores en individuos reales para así determinar su confiabilidad. Por último, las conclusiones, recomendaciones, glosario y fuentes de información.

CAPÍTULO I

PLANTEAMIENTO DEL PROBLEMA

1.1 Situación problemática

La ciudad universitaria de la Universidad de San Martín de Porres en Santa Anita no es ajena al uso de sistemas biométricos dactilares, y cuenta con tres de estos sistemas ubicados en la Puerta 1, Puerta 2 y en los laboratorios de Odontología para la verificación de la identidad del personal que labora ahí y así poder registrar sus asistencias (Ver Figura 1).



Figura (1). Sistema biométrico dactilar de la Sede Santa Anita de la USMP
Tomado de: Instalaciones de la USMP-Santa Anita

Es en este último espacio (laboratorios de Odontología), donde se registran los docentes y sus colaboradores respectivos de la Facultad de Odontología que suman un número de alrededor de 80 personas, en el que se han reportado múltiples fallas a la hora de registrar sus huellas dactilares.

Estas fallas se deben a la naturaleza propia de las huellas dactilares de un ser humano, que sufren desgaste por distintas causas entre las cuales se resaltan:

- La edad
- Escribir mucho
- Accidentes que dañan la superficie dactilar como cortes y quemaduras
- Falta de higiene
- Enfermedades como diabetes y dermatitis
- Trabajar con reactivos químicos

Es justamente por estas causas, que se han desgastado las huellas dactilares de muchos de los docentes de Odontología, a tal punto que el sistema biométrico dactilar ya no es capaz de reconocerlas, a pesar de que no presenta desperfectos técnicos; todo esto conlleva a problemas de índole administrativo y disciplinario, pues al no registrar las huellas el sistema interpreta que el personal no asistió a sus horas respectivas de trabajo.

Finalmente, la universidad opta por una penalización en forma de descuento del sueldo del personal, lo cual a su vez origina reclamos por parte de los sancionados, que tienen que recurrir a métodos más arcaicos para registrar la asistencia; en este caso, tienen que depender del personal de vigilancia para que anote sus horas de llegada y salida, para así evitar futuras sanciones.

Es por ello, que surge la necesidad de proporcionar un método alternativo de verificación de características morfológicas no invasivo que sea capaz de verificar la identidad de las personas que laboran en laboratorio de Odontología de la USMP, sin perder la automaticidad y confiabilidad que proporciona un sistema biométrico.

1.2 Definición del problema

El problema principal radica en la ineficiencia que tiene el sistema biométrico del Laboratorio de Odontología de la USMP para reconocer la superficie dactilar del personal que se ha visto afectado debido a causas de diversa índole especificadas en la situación problemática.

Tal y como se explicó líneas arriba, el sistema biométrico dactilar fue instalado con el propósito de registrar la asistencia del personal que labora en la universidad de forma automática y así simplificar un procedimiento que de otra forma sería engorroso dado el número de personas que laboran en la USMP sede Santa Anita.

Si el sistema biométrico del laboratorio de Odontología no es capaz de cumplir con dicha tarea, los usuarios nunca llegan a registrarse y se generan así sanciones para ellos por parte de la universidad.

Para evitar este tipo de sanciones, los afectados deben dejar constancia escrita de su ingreso y salida de la universidad al personal de vigilancia y cuando llega fin de mes, se presentan estos registros escritos como evidencia de la asistencia al trabajo del personal afectado. De esto se observa que los problemas que se ramifican de la dificultad para registrar las huellas dactilares por parte del sistema biométrico dactilar son: el riesgo de descuento de sueldo a fin de cada mes por presunta inasistencia al centro de labor y la pérdida de automaticidad en el registro de la asistencia del personal.

1.3 Objetivos

1.3.1 Objetivo General

Desarrollar un sistema biométrico no invasivo, con tasa de acierto mínima de 90%, orientado a la identificación de personas a partir de patrones de líneas palmares, considerando un universo de 80 personas.

1.3.2 Objetivos específicos

- Modelar el sistema de procesamiento biométrico para identificación de personas a partir de los pliegues palmares.
- Analizar los parámetros para la identificación biométrica de personas a partir del patrón de líneas palmares de la mano.
- Determinar criterios de selección del sensor para la adquisición de imágenes.
- Diseñar una etapa de preprocesamiento que evidencie de manera adecuada la región palmar de interés para que así se facilite el trabajo de clasificación de la red neuronal convolucional.
- Diseñar la arquitectura de red neuronal convolucional adecuada en *Python*, de manera que se clasifique de manera correcta a un individuo mediante sus líneas o pliegues palmares.
- Diseñar y construir una estructura o recinto que garantice el posicionamiento adecuado de la mano que evidencie adecuadamente los pliegues palmares para una correcta identificación del usuario.
- Automatizar el proceso registro de asistencia para el personal afectado.
- Validar el sistema propuesto a través de los parámetros estándares de medición de desempeño como son: verdaderos reconocimientos, falsos rechazos.

1.4 Justificación

El desarrollo e implementación del presente trabajo adquiere importancia, ya que permitirá que el personal afectado sea capaz de registrar su asistencia al trabajo de manera automática y no invasiva, independientemente del estado de su superficie dactilar; ya que a diferencia

de los patrones dermatoglíficos dactilares, las líneas palmares son más robustas y menos propensas a deformación.

El trabajo además adquiere novedad, porque implementaría una solución que no es muy conocida en el medio debido a la gran popularidad de la que gozan los sistemas biométricos dactilares; además dentro de la solución se utilizará lenguaje *Python* que es un lenguaje de programación libre lo que significa que no requerirá de licencia a la hora de implementar el sistema. En dicho lenguaje se implementarán los algoritmos de diseño de la etapa de preprocesado para resaltar las líneas palmares y el algoritmo de clasificación por redes neurales convolucionales. Además, se diseñará la estructura que permita el correcto posicionamiento de la mano del usuario y que evidencie los patrones palmares para una correcta identificación.

También, existe justificación económica, ya que actualmente el laboratorio de Odontología de la USMP sede Santa Anita cuenta con un sistema biométrico dactilar que utiliza prismas ópticos esenciales para el correcto funcionamiento de dicho sistema; sin embargo, estos prismas deben ser cambiados periódicamente cada 6 meses debido al desgaste que sufre la silicona del prisma. Cada prisma cuesta alrededor de S/. 200 y tiene un costo de instalación de S/. 100; esto quiere decir que aparte de poseer un sistema biométrico que no garantiza el correcto registro de las huellas dactilares afectadas, la USMP tiene que incurrir en gastos de S/. 600 al año de mantenimiento e instalación para un solo módulo biométrico dactilar.

Si se tiene en cuenta eso, los autores estiman de manera preliminar que el coste del presente trabajo de investigación rondaría los S/. 500, siendo solo los LED los elementos que tendrían que cambiarse cada cierto tiempo (el sistema de iluminación utilizado solo cuesta S/. 5). De obtenerse los resultados deseados, el trabajo contribuirá a la USMP en el área de biometría para la correcta identificación de su personal que labora en áreas de acceso restringido, como los laboratorios.

1.5 Limitaciones

La presente tesis se limitará a captar el mapa de líneas palmares de la mano de una persona por tener patrones más robustos y visibles que los patrones dermatoglíficos que se encuentran en los dedos, tampoco se analizarán otros aspectos físicos como el iris, la retina, la cara, etc. Además, hablando en términos de espacio y localización, la aplicación de la solución se verá limitada a las instalaciones de la Facultad de Odontología de la USMP en Santa Anita.

El sistema actualmente está diseñado para reconocer las impresiones palmares de la mano derecha y no de la izquierda; asimismo, solo puede reconocer una mano a la vez. Otra limitación del sistema es que es propenso a presentar fallas a la hora de reconocer manos que presenten suciedad o elementos ajenos a la epidermis, en la región palmar que no hayan sido identificados previamente por la red en el entrenamiento de la misma.

Es pertinente mencionar que las personas que sufran de patologías genéticas que afecten a los dermatoglifos palmares, como el Síndrome de Down, adermatoglifia, etc. no formarán parte del estudio; asimismo, aquellas personas cuyas líneas palmares se vean absorbidas por la tonalidad muy clara o muy oscura de la piel de la palma de su mano; tendrán muchas dificultades para ser reconocidas por el sistema. Tampoco podrán utilizar este sistema, aquellas personas que no cuenten con alguna de sus extremidades superiores, ni tampoco lo podrán usar aquellas personas que teniendo su extremidad superior derecha hayan sufrido algún tipo de lesión o accidente que distorsione la superficie palmar temporal o permanentemente.

1.7 Viabilidad

- **Viabilidad Técnica:** La presente tesis se vuelve viable desde el punto de vista técnico, ya que los autores recibieron capacitación sobre las redes neuronales, gracias al curso electivo llamado Redes Neuronales impartido durante pregrado. Asimismo, los

autores se encuentran familiarizados con el lenguaje de programación *Python* desde los cursos de pregrado de Proyecto de Ingeniería 1 y Proyecto de Ingeniería 2.

Los componentes requeridos para la realización del sistema propuesto (cámara digital, sistema de iluminación, material para la estructura de soporte, la *Graphic Processing Unit* o GPU, la *Central Processing Unit* o CPU) se encuentran disponibles en el mercado peruano, facilitando así su adquisición.

Dado que la tesis se realiza para las instalaciones de la USMP, la adquisición de muestras de líneas palmares es viable; ya que hay disponibilidad y afinidad por parte del personal que trabaja en el laboratorio de Odontología, para brindar sus patrones palmares en pos de ayudar a encontrar una solución que satisfaga sus necesidades.

- **Viabilidad Económica:** Se cuenta con la capacidad económica para adquirir todos los materiales antes descritos (se mencionó un monto que ronda los S/. 500 para los materiales y dispositivos). Es pertinente aludir que uno de los autores ya contaba previamente con la GPU para entrenar a la red neuronal, lo que facilita mucho más la capacidad adquisitiva del equipo ingenieril.
- **Viabilidad Social:** El presente trabajo es viable socialmente porque no es invasivo a la persona, ni pone en riesgo su vida; ya que solo implica tomar fotos de las palmas de sus manos.
- **Viabilidad Operativa:** El sistema no requiere condiciones especiales para poder operar y es perdurable en el tiempo; el único repuesto que se prevé que cambie cada cierto tiempo son los LED para la iluminación interna de la estructura que brinda soporte a la mano del usuario.

CAPÍTULO II

MARCO TEÓRICO

2.1 Antecedentes de la investigación

La primera máquina de aprendizaje fue inventada en el Laboratorio Aeronáutico de Cornell en 1957 por Frank Rosenblatt y se denominó Perceptrón, el cual se comportaba como un clasificador lineal sobre un extractor de características simple (Yann Le Cunn y Marc Aurelio Ranzato, 2013). Rosenblatt diseñó el Perceptrón con dos niveles en donde parámetros que ahora se conocen como pesos se ajustaban de manera proporcional al error generado por la red y al error que se deseaba en realidad. El mismo Rosenblatt intentó extender su modelo del Perceptrón a tres niveles, pero sin mucho éxito (Isasi y Galván, 2004).

Dos años después, Bernard Widrow diseñó una red muy similar al Perceptrón de Rosenblatt a la cual denominó *Adaptive Linear Element* o simplemente Adaline; esta red también poseía dos niveles y ajustaba los pesos en función al error esperado y el error real de la red, aunque tenía aplicaciones distintas a las del Perceptrón de Rosenblatt. Tanto el Perceptrón como Adaline eran modelos lineales que no eran capaces de resolver problemas más complejos que una compuerta XOR, a esto se le denominó el problema de la separabilidad lineal (Isasi et al., 2004). Debido a esto, Minsky y Papert condujeron una investigación en 1969 que dejó como conclusión que las redes neuronales eran inútiles a la hora de resolver problemas complejos (como los problemas del campo de la visión); por lo que el campo sufrió un estancamiento de varios años (Krzysztof J. Cios, 2017).

En 1962, Hubel y Wiesel (que eran neuro-fisiólogos) descubrieron que el cerebro procesaba la información de manera jerárquica (basaron sus hallazgos observando el sistema visual de un gato). Estos investigadores encontraron que en cada nivel de procesamiento el cerebro percibía características más generales (gracias a células complejas, las cuales añadían información extraída en los niveles anteriores), eso quiere decir que en los niveles iniciales se percibían patrones básicos como como elementos horizontales o verticales para luego agrega dicha información en los niveles más avanzados y así poder generalizar estímulos más complejos como imágenes (Krzysztof J. Cios, 2017).

No fue hasta 1980 que los descubrimientos de Hubel y Wiesel fueron aplicados al campo de las redes neuronales, fue Kunihiko Fukushima quien diseñó una red inspirada en la estructura jerárquica del cerebro a la que denominó Neocognitrón (Krzysztof J. Cios, 2017), una red neuronal artificial multicapa diseñada para abordar problemas complejos como son los del campo de la visión tal y como se puede apreciar en la figura 2.

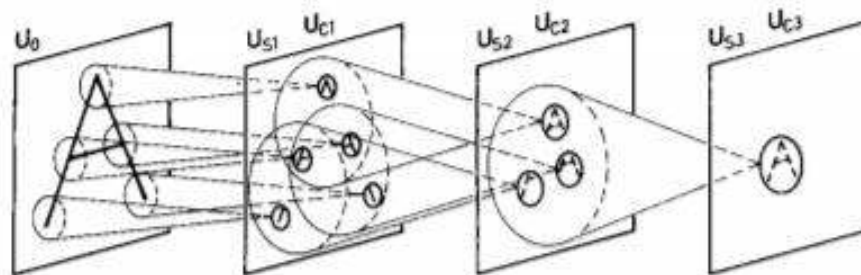


Figura (2). Proceso de generalización de la información en las capas del Neocognitrón de Fukushima para una entrada visual.

Fuente: *Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position*. Kunihiko Fukushima. 1980. p.6

El Neocognitrón sirvió como base para las redes neuronales convolucionales ya que en 1998 Yann LeCun presenta un paper titulado “*Object Recognition with Gradient-Based Learning*” donde introduce el concepto de redes neuronales convolucionales por primera vez las cuales

también están estructuradas de forma jerárquica en donde los niveles iniciales detectan patrones simples y los niveles avanzados detectan patrones complejos; sin embargo, a diferencia del Neocognitron de Fukushima, la red de LeCun (denominada LeNet-5, su estructura se aprecia en la figura 3) incorpora el concepto de aprendizaje supervisado a través del algoritmo de Backpropagation (Kyle Ambert,2017) otra diferencia con el Neocognitron es que LeNet-5 incorpora un Perceptrón Multicapa al final de su arquitectura para realizar la tarea de clasificación.

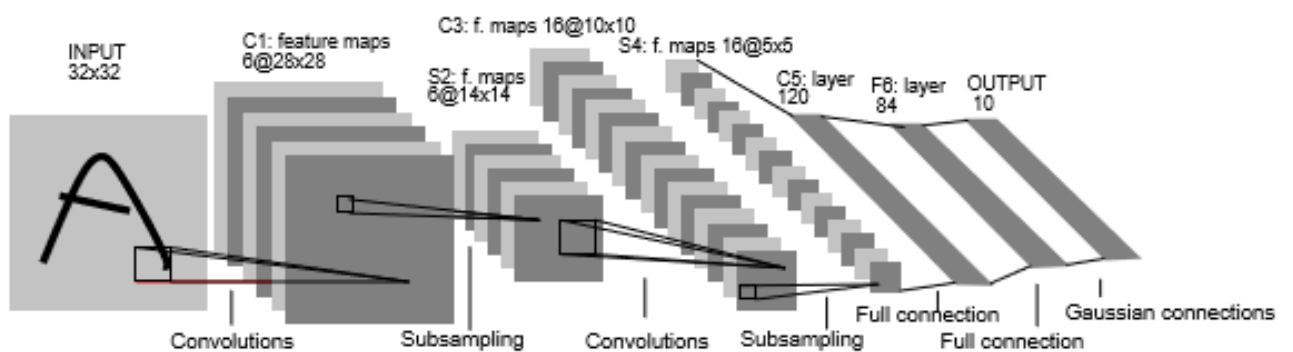


Figura (3). Arquitectura jerárquica de LeNet-5, propuesta por Yann LeCun.

Fuente: *Object Recognition with Gradient-Based Learning*. Yann LeCun, Patrick Haffner, Léon Bottou y Yoshua Bengio. 1998. p.4

No fue hasta el año 2012, sin embargo, que las redes neuronales convolucionales se vuelven populares. En ese año Alex Krizhevsky, Ilya Sutskever y Geoffrey E. Hinton publican un trabajo llamado: *“ImageNet Classification with Deep Convolutional Neural Networks”* en el cual presentan una red neuronal convolucional llamada AlexNet, esta red tomaba la LeNet-5 de LeCun y le cambió las funciones de activación en sus capas escondidas al mismo tiempo que le añadió más capas. AlexNet participó y ganó el *ImageNet Large Scale Visual Recognition Challenge* del 2012 batiendo a su competidor más cercano por más de 10 puntos porcentuales en la tasa de error, con lo que se elevó a las redes neuronales convolucionales al nivel de estado del arte.

Hoy en día las redes neuronales convolucionales se han convertido en una parte esencial de la caja de herramientas de la ciencia de datos (Kyle Ambert, 2017) y son ampliamente empleadas en tareas relacionadas al campo de la visión.

2.2 Estado del Arte

Se procede a mencionar aquellas las publicaciones científicas que abordan el problema que los autores de este trabajo consideraron más relevantes. A continuación, se describen brevemente las publicaciones científicas que emplean redes neuronales convolutivas para la identificación de individuos a través de sus líneas o pliegues palmares.

Título: *Deformation Invariant and Contactless Palmprint Recognition Using Convolutional Neural Network.*

Autores: Amin Jalali, Rammohan Mallipeddi y Minho Lee

Lugar: Universidad Nacional de Kyungpook, Corea del Sur. **Año:** 2015.

Jalali, Mallipeddi y Lee (2015) proponen utilizar una red neuronal convolucional que consiste de cinco capas entre las que se incluyen la capa de entrada, dos pares de capas convolutivas y de reducción, y dos capas totalmente conectadas. Así mismo los autores proponen utilizar la función sigmoide para la etapa de clasificación.

La red es entrenada con la optimización de descenso de gradiente con un factor de aprendizaje fijado en 0.005. Las entradas de esta red son imágenes tomadas de una base de datos de la Universidad Politécnica de Hong Kong (PolyU) de 400 voluntarios con una distribución de edad que va desde los 20 años hasta los 60 años, de cada individuo se extrajeron 20 muestras con un tamaño de 28x28 píxeles. Los autores utilizaron el 80% de sus muestras para el entrenamiento mientras que el 20% restante fue utilizado para realizar las pruebas de validación. Los resultados que los autores evidencian son de una tasa de 99.97% acierto a la hora de identificar un individuo de la base de datos PolyU.

Adicionalmente, los autores proponen su propio método de adquisición de imágenes con la ayuda de una cámara Logitech C-930e de alta resolución, se tomaron 20 muestras de 10 individuos (cada uno) empleando esta cámara y dichas muestras corresponden a imágenes de las palmas de las manos en distintas posiciones y ángulos. En esta segunda etapa, los autores dividieron su data en 85% para data de entrenamiento y 15% para la validación, todo el data set fue probado en la misma arquitectura de red que se empleó para las imágenes de PolyU obteniéndose un 93.4% de tasa de aciertos.

Fortalezas:

- La red diseñada por Jalali et al. (2015) alcanza un desempeño de más del 90% a la hora de reconocer individuos con un número razonablemente reducido de capas y tamaño de las imágenes de entrada.
- Los autores no solo dependen de la base de datos de PolyU sino que también cuentan con su propio método de adquisición de imágenes tal y como se evidencia en su segundo experimento.
- Los autores emplean una cámara de muy buena resolución para la adquisición de imágenes en su segundo experimento lo que garantiza una buena calidad de las muestras a entrenar.

Debilidades:

- El *paper* no especifica el tipo de función de activación que se emplea en las capas de convolución.
- El *paper* no especifica la función de error que se debe minimizar con el descenso de gradiente.
- Los autores emplean el descenso de gradiente como método de optimización lo cual puede generar un mayor tiempo de convergencia a cero por parte de la función de error.
- Los autores emplean una función sigmoide en la etapa de clasificación la cual ya no es considerada como computacionalmente eficiente, lo cual puede tener un efecto negativo a la hora de una rápida convergencia a cero por parte de la función de error.

- El método de adquisición de imágenes propuesto no cuenta con un sistema posicionador de manos para una adquisición de muestras más uniforme.

Título: *Contactless Palmprint Recognition Based On Convolutional Neural Network.*

Autores: Dian, Liu, Dongmei y Sun.

Lugar: Universidad de Beijing Jiaotong, China. **Año:** 2016.

Dian, Liu, Dongmei y Sun (2016) proponen lógica difusa para el preprocesamiento de las imágenes y redes neuronales convolucionales para la extracción de caracteres de la palma de la mano de los individuos en combinación con el método de Hausdorff para la comparación de las muestras.

Las imágenes fueron adquiridas de 3 bases de datos públicas: PolyU II, CASIA y IITD. A continuación, se utilizó segmentación de imagen para delimitar la ROI de las palmas de las manos, tras ello los autores proponen emplear lógica *fuzzy* (o lógica difusa) para mejorar las texturas de dichas imágenes segmentadas ya que, de acuerdo con ellos, mejora la extracción de caracteres por parte de la red neuronal convolucional y también mejora la comparación de muestras.

Los autores proponen utilizar la red convolucional AlexNet (una red creada por Alex Krizhevsky en 2012) para la extracción de los patrones palmares, dicha red consta de 8 capas (5 capas convolutivas y 3 capas totalmente conectadas).

Los autores extraen las salidas de la capa número 7 de AlexNet para realizar la comparación de muestras empleando el método de distancia de Hausdorff el cual representa la máxima distancia de un conjunto hacia el punto más cercano de otro conjunto.

Se tomaron un total de 7752 imágenes en escala de gris correspondientes a 193 individuos para la base de datos PolyU, 5502 imágenes de 312 sujetos

para la base de datos CASIA y 2601 imágenes de 230 usuarios para la base de datos de IITD. El EER obtenido a la hora de la verificación para la primera base de datos fue de 0.0443%, para CASIA el EER fue de 0.0803% y finalmente para IITD el EER fue de 0.1113%.

Fortalezas:

- El uso de lógica fuzzy para el preprocesamiento de las entradas permite que la calidad de las imágenes adquiridas de las tres bases de datos sea mejor.
- La combinación de la AlexNet para la extracción de caracteres palmares y el método de distancia de Hausdorff para comparación de muestras consigue un EER bastante bueno (menos de 0.12% en el peor de los casos).

Debilidades:

- Los autores no diseñan la red neuronal convolucional de este paper, sino que dependen de una red ya configurada como es AlexNet para realizar la extracción de caracteres.
- No se evidencia ningún método de adquisición de imágenes propio (se depende de tres bases de datos públicas), por lo que realizar pruebas de verificación con personas en tiempo real es imposible.

Título: *Palmpoint recognition via discriminative index learning.*

Autores: Jan Svoboda, Jonathan Masci y Michael M. Bronstein.

Lugar: XXIII Conferencia Internacional sobre Reconocimiento de Patrones. **Año:** 2016

Svoboda, Masci y Bronstein (2016) diseñan su propia red neural convolucional para la identificación biométrica con pliegues palmares. Las imágenes que los autores emplean en su trabajo son obtenidas de dos bases de datos públicas que son IITD y CASIA; de la primera base de datos se extrajeron muestras de 230 sujetos (5 muestras de la mano derecha por sujeto) mientras que de la segunda base de datos se tomaron muestras de 283 individuos (5 muestras por persona).

La red que propone Svoboda et al. (2016) tiene la siguiente arquitectura: 2 capas convolucionales (ambas afectadas por la función de activación ReLU) seguidas de una capa de *pooling* y dos capas convolucionales adicionales (afectadas por la función de activación tangente hiperbólica), seguidas de dos capas totalmente conectadas (los autores no especifican qué función de activación afecta a la capa de salida así que se asume que no utilizan alguna función); la salida de esta red arroja un vector de características de 32 valores.

La función de error propuesta por los autores es una llamada pérdida *d-prime* la cual también se llama *discriminative index*, que de acuerdo a los mismos autores, permite lograr una mejor distribución en la separación de las puntuaciones genuinas/impostoras.

La red fue optimizada siguiendo el método AdaDelta con un factor de aprendizaje inicial de 0.01 el cual fue reduciéndose cada 100 épocas (en total los autores reportan haber empleado 400 épocas, no se especifica el *hardware* donde se entrenó la red). Los resultados arrojan que para las muestras de la base de datos de IITD se obtuvo un EER de 1.64% y para CASIA se obtuvo un EER de 1.86%.

Fortalezas:

- Los autores acoplan una función de error que normalmente no se usa en redes neuronales de manera exitosa obteniendo un EER aceptable (menor a 2%).
- Se emplea una optimización adecuada (del estado del arte) como es Adagrad para que la función converja a cero.
- Para conseguir invarianza a la traslación, los autores alteran las muestras de las imágenes durante el proceso de aprendizaje; desplazándolas unos pocos píxeles o realizando una escala ligera (de hasta 8 píxeles).

Debilidades:

- No se evidencia ningún método de adquisición de imágenes propio (se depende de dos bases de datos públicas), por lo que realizar pruebas de verificación con personas en tiempo real es imposible.

- Al entrenar una red durante 400 épocas se puede incurrir en riesgo de *overfitting*.

Título: *Palmpoint Recognition with Deep Convolutional Features*.

Autores: Qiule Sun, Jianxin Zhang, Aoqi Yang y Qiang Zhang.

Lugar: Laboratorio de Diseño Avanzado y Computación Inteligente de la Universidad de Dalian, China. **Año:** 2017.

Sun, Zhang, Yang y Zhang (2017) proponen utilizar la arquitectura de la CNN-F, una red neural convolucional propuesta por Ken Chatfield, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman en su paper titulado *Return of the Devil in the Details: Delving Deep into Convolutional Nets* en el año 2014. Esta red está inspirada en la AlexNet ya que también cuenta con cinco capas de convolución (con su respectiva capa de max-pool cada una) y tres capas totalmente conectadas (con la función de activación Softmax aplicada a la capa final para realizar las predicciones de dicha arquitectura de red).

Esta red toma como entradas imágenes a color de 224x224 de acuerdo con Chatfield et al. (2014), sin embargo, Sun et al. (2017) reporta que sus imágenes (de tamaño 128x128 y en escala de gris) no cumplen con los requisitos de CNN-F, por lo que aplican técnicas de PDI para convertir dichas imágenes al formato óptimo.

Las imágenes de las palmas de manos (de distintos individuos) que emplean Sun et al. (2017) fueron adquiridas de la base de datos pública PolyU, en total los autores reportan haber extraído un total de 3855 muestras de un universo de 386 individuos. La red fue implementada en Matlab R2015b, y se entrenó utilizando una CPU de 3.30 GHz con 4G de memoria RAM. Los resultados obtenidos reportan un EER de 0.25%

Fortalezas:

- La CNN-F aplicada al reconocimiento de personas por medio de sus pliegues palmares reporta un EER aceptable (0.25%) aunque no mejor que el método de Dian et al. (2016).

Debilidades:

- Los autores no diseñan la red neuronal convolucional de este *paper*, sino que dependen de una red ya configurada como es CNN-F para realizar sus experimentos.
- La red fue implementada en Matlab, por lo que su uso en algún producto comercial está virtualmente descartado al menos que cuente con la licencia apropiada.
- El uso de una CPU para el entrenamiento de la red no es óptimo ya que puede requerir demasiado tiempo para este fin (el *paper* no especifica qué tiempo de entrenamiento se necesitó).
- No se evidencia ningún método de adquisición de imágenes propio (se depende de una base de datos pública), por lo que realizar pruebas de verificación con personas en tiempo real es imposible.

Título: *Supervised Hashing with Deep Convolutional Features for Palmprint Recognition.*

Autores: Jingdong Cheng, Qiule Sun, Jianxin Zhang y Qiang Zhang.

Lugar: Laboratorio de Diseño Avanzado y Computación Inteligente de la Universidad de Dalian, China. Año: 2017.

Cheng, Zhang, Sun y Zhang (2017) proponen integrar las redes neuronales convolucionales con Hash supervisado. En primer lugar, los autores obtienen las muestras de la base de datos de PolyU (no se especifica el número exacto de muestras ni el número de individuos en el experimento, así que se asume que se emplearon las 7720 muestras en escala de gris de los 386 individuos registrados en dicha base de datos pública).

Una vez hecho esto se ingresan las imágenes a la red neuronal convolucional CNN-F (la arquitectura de esta red ya se explicó líneas arriba) para capturar los caracteres palmares. Los autores deciden extraer los caracteres a la salida de la 5ta capa de convolución de la CNN-F para convertirlos en un vector que ellos denominan X, al cual se le aplica PCA para reducirle la dimensionalidad.

A continuación, se aplica el método COSDISH para codificar dichos caracteres convolutivos de baja dimensionalidad en tramas de 256 bits, 128

bits y 64 bits (se hace esto para comparar en cuál trama se obtienen mejores resultados).

Para la comparación los autores emplean la distancia de Hamming para calcular la similitud entre dos muestras de palmas de las manos; los programas fueron desarrollados en Matlab R2015b, utilizando una CPU de 3.30 GHz con 4G de memoria RAM. En los resultados los autores muestran que se obtuvo un EER de 0% con la trama de 128 bits y de 256 bits mientras que para la trama de 64 bits se obtuvo un EER de 1.11%.

Fortalezas:

- Este paper evidencia la obtención de una representación de pliegues palmares basada en código a través de un método aprendible.
- La combinación de Hash supervisado con redes neuronales convolucionales para la identificación de personas mediante sus pliegues palmares permite la obtención de un EER de 0% lo cual es notorio.

Debilidades:

- Los autores no diseñan la red neuronal convolucional de este paper, sino que dependen de una red ya configurada como es CNN-F para realizar sus experimentos.
- No se evidencia ningún método de adquisición de imágenes propio (se depende de una base de datos pública), por lo que realizar pruebas de verificación con personas en tiempo real es imposible.

Título: *Palmpoint Recognition Based on Deep Convolutional Neural Networks.*

Autores: Xueqiu Dong, Liye Mei y Junhua Zhang

Lugar: II Conferencia Internacional de Ciencia de la Computación y Comunicación Inteligente. **Año:** 2018.

Dong, Mei y Zhang utilizan la arquitectura de MobileNet para la identificación de personas por medio de sus líneas palmares, MobileNet es un modelo de red neural convolucional de código abierto creado por Google en junio de 2017 e implementado en Tensorflow diseñado específicamente para maximizar

precisión y puede ser empleada en tareas de clasificación, detección, segmentación, etc.

Los autores adquieren las muestras de palmas de las manos de la base de datos pública PolyU. Se extrajeron 6000 imágenes en escala de gris con un tamaño de 128x128 correspondientes a 500 palmas diferentes, de las cuales 5400 fueron usadas para entrenamiento y 600 para pruebas, adicionalmente los autores agregaron ligera distorsión a las muestras en la fase experimental para evitar el *overfitting*.

La MobileNet acepta imágenes de 224x224 en color, así que los autores tuvieron que aplicar técnicas de PDI para ajustar sus entradas a los requerimientos de MobileNet. Esta red cuenta con 28 capas, las cuales emplean función de activación ReLU salvo la capa de salida que emplea la función de activación Softmax. La red es entrenada en Keras, en tanto que la función objetivo definida por los autores es la *softmax loss function* que en realidad significa que se usa la función de pérdida de entropía cruzada una vez que softmax ha sido aplicada en la salida. El factor de aprendizaje es fijado en 0.01 mientras que el método de optimización escogido por Dong et al. (2018) es el descenso de gradiente estocástico con el cual se obtiene una convergencia a cero por parte de la función de coste a las 100 iteraciones. Los resultados muestran que se obtuvo una precisión del 99.95% a la hora de reconocer las muestras.

Fortalezas:

- Se obtiene una precisión excelente al utilizar MobileNet para el reconocimiento biométrico de líneas palmares.

Debilidades:

- Los autores no diseñan la red neuronal convolucional de este paper, sino que dependen de una red creada por Google.
- No se evidencia ningún método de adquisición de imágenes propio (se depende de una base de datos pública), por lo que realizar pruebas de verificación con personas en tiempo real es imposible.

Título: *PalmNet: Gabor-PCA Convolutional Networks for Touchless Palmprint Recognition.*

Autores: Angelo Genovese, Vincenzo Piuri, Konstantinos N. Plataniotis, y Fabio Scotti.

Lugar: Paper IEEE. **Año:** 2019.

Genovese, Piuri, Plataniotis y Scotti (2019) proponen su propia red neural convolucional llamada Palmnet para el reconocimiento de personas a través de sus pliegues palmares. Esta red es entrenada utilizando un procedimiento sin supervisión para el ajuste de los filtros basado en respuestas de Gabor y análisis de componentes principales sin requerir de etiquetas de clase (es decir la identidad de los usuarios no se necesita salvo a la hora de evaluar la precisión del método que proponen los autores).

Esta red es capaz de adaptarse a diferentes bases de datos de acuerdo con Genovese et al. (2109), siendo así mismo capaz de extraer caracteres palmares altamente confiables. Cuando dicha red es aplicada a cualquier a cualquier muestra segmentada de una palma, arroja una plantilla biométrica representada como un vector. Los autores emplean muestras de palmas de manos de cuatro bases de datos diferentes las cuales son: CASIA (5455 muestras de 624 individuos), IITD (2669 muestras de 467 individuos), REST (1937 muestras de 358 individuos) y Tongji (5182 muestras de 600 individuos).

Las imágenes son preprocesadas en tres pasos: segmentación de la mano, extracción de puntos de valle y cálculo de la región de interés. Estas imágenes son ingresadas a la Palmnet la cual tiene tres capas (L0 que es la capa de entradas que recibe las imágenes de las cuatro bases de datos preprocesadas, L1 primer capa de convolución, L2 segunda capa de convolución y L3 que es la capa de binarización). Los filtros de las capas L1 y L2 se pueden configurar para tener distintos tamaños utilizando procedimientos sin supervisión alguna, los autores consideraron dos tipos de filtros adaptivos: Filtros de Gabor y filtros basados en PCA. Teniendo en cuenta esto, los autores generan dos posibles topologías para Palmnet: Una

utilizando filtros de Gabor únicamente en L1 y L2; y la otra empleando filtros de gabor en L1 y filtros PCA en L2.

Una vez que las imágenes son ingresadas a la Palmnet (bajo cualquiera de sus dos topologías) para que dicha red extraiga los patrones palmares pertinentes; al final de la etapa de extracción se obtiene un vector de características palmares que los autores denominan H.

En la etapa de clasificación, los autores emplean el clasificador k-NN basado en distancia euclídeana (los autores usan $k=1$), es decir, dos vectores de características como H1 y H2 son comparados para el cálculo de la distancia euclídeana. Los resultados de Palmnet utilizando filtros Gabor y filtros PCA muestran que se obtuvieron las siguientes performances a la hora de comparación: Para la base de datos CASIA se obtuvo una tasa de acierto a la hora de reconocer exitosamente una muestra palmar, para IITD se obtuvo una tasa de acierto de 99.37%, para REST se obtuvo una tasa de acierto de 97.17% y finalmente para Tongji la tasa de acierto a la hora de clasificar correctamente los patrones palmares fue de 99.83%.

Fortalezas:

- Los autores integran de manera exitosa los conceptos de respuestas de Gabor y análisis de componentes principales con redes neuronales convolutivas en un diseño novedoso que muestra un desempeño de estado del arte al obtener tasas de reconocimiento exitoso de al menos 97.17% para cuatro tipos distintos de base de datos públicas.

Debilidades:

- No se evidencia ningún método de adquisición de imágenes propio (se depende de cuatro bases de datos públicas), por lo que realizar pruebas de verificación con personas en tiempo real es imposible.

Título: *Centralized Large Margin Cosine Loss for Open-set Deep Palmprint Recognition.*

Autores: Dexing Zhong y Jinsong Zhu.

Lugar: IEEE Transactions on Circuits and Systems for video Technology. Año: 2019.

Dexing Zhong y Jinsong Zhu (2019), proponen utilizar una función de pérdida distinta a la “softmax-loss” (que en realidad es la función softmax aplicada a una capa de salida seguida de la función de pérdida de entropía cruzada) denominada C-LMCL (Centralized Large Margin Cosine Loss). Tal y como en otros papers del estado del arte que abordan el mismo problema, los autores obtienen las muestras de las palmas de distintas manos de dos bases de datos públicas (PolyU y Tongji en este caso). Se tomaron 9600 imágenes de Tongji correspondientes a 480 palmas (diez muestras por cada mano) para el entrenamiento y 2400 imágenes de 120 palmas (10 muestras por cada mano) para las pruebas de verificación; para el caso de PolyU los autores emplearon 4800 imágenes correspondientes a 400 palmas (seis muestras por mano) para el entrenamiento mientras que dejaron 1200 imágenes de 100 palmas (seis muestras por mano) para las correspondientes pruebas.

Los autores emplean un filtro gaussiano para remover cualquier ruido que pueda haber en las muestras para luego delimitar una ROI cuadrada que es la que contiene los patrones palmares a clasificar. Una vez hecho esto la ROI extraída es reescalada a las dimensiones de 224x224. Finalmente, los autores le aplican incremento de datos (*data augmentation*) a sus muestras de manera que la red tenga mejor capacidad de generalización, *data augmentation* incluyó lo siguiente en este trabajo: Rotación, recorte de tamaño, ajuste de contraste y brillo, suavizado y cambio de color.

Estas muestras preprocesadas son ingresadas a una red neural ya diseñada llamada ResNet-20 la cual tiene seis capas (una capa de entradas, cuatro capas convolucionales y una capa de salida totalmente conectada), la salida de esta red arroja un vector de características de tamaño 128. Zhong et al.

(2019) indica que se emplea la C-LMCL como reemplazo a *softmax-loss* debido a que tiene un mayor poder discriminativo que dicha función y es más adecuada para ser empleada en *data sets* que contengan bastante ruido.

El entrenamiento de la red se realizó con ayuda de una tarjeta gráfica NVIDIA GTX 1080, utilizándose la optimización de descenso de gradiente estocástico con una tasa de aprendizaje de 0.01. En la validación y verificación se empleó similitud de coseno entre dos vectores de características (recordar que la red arroja vectores de tamaño 128 como sus salidas) para efectuar la respectiva comparación. Los resultados obtenidos muestran que se obtuvo una precisión a la hora de identificar personas de un 99.93% y con un EER de 0.26 (para una tasa de aprendizaje de 0.01).

Fortalezas:

- Los autores emplean con éxito la función de error C-LMCL en una ResNet para obtener un porcentaje de precisión casi perfecta a la hora de identificar individuos.

Debilidades:

- Los autores no diseñan la red neuronal convolucional de este *paper*, sino que dependen de una red creada por alguien más.
- No se evidencia ningún método de adquisición de imágenes propio (se depende de dos bases de datos públicas), por lo que realizar pruebas de verificación con personas en tiempo real es imposible.

2.3 Bases Teóricas

La base teórica del presente trabajo se dividirá en 4 partes principales, las cuales se mencionan a continuación:

- Base Teórica biológica
- Base Teórica de la adquisición de imágenes
- Base Teórica de la Etapa de preprocesamiento
- Base Teórica de la etapa de procesamiento

2.3.1 Base Teórica Biológica

Aquí se expone lo concerniente a los pliegues palmares que son el objeto de estudio del presente trabajo.

2.3.1.1 Dermatoglifos y Pliegues Palmares

La palabra dermatoglifo tiene su origen del griego *derma* que significa piel y de *glyphe* que significa dibujo, los cuales hacen referencia a las líneas dermopapilares de las manos y los pies; el concepto fue propuesto por Harold Cummins y Charles Midlo en su trabajo *Finger Prints, Palms and Soles: An Introduction to Dermatoglyphics* en el año 1926. Tanto los dermatoglifos como los pliegues palmares se encuentran en la capa superior de la piel, es decir, en la epidermis.

Se dice que alrededor de los tres meses de gestación se pueden apreciar tanto los dedos como las palmas de las manos, en la parte digital e interdigital se perciben las almohadillas sobre las cuales se conforman las figuras que corresponden a las líneas dermopapilares evidenciadas entre el cuarto y sexto mes de gestación. Avanzando hacia el séptimo mes ya no se generan más cambios en la morfología de los dermatoglifos u otras configuraciones (Martínez, Oviedo, Portilla y Rojas, 2016). Ver figura 4.



Figura (4). Representación de un dermatoglifo localizado en la región dactilar

Fuente: <https://francis.naukas.com/2009/04/10/por-que-las-huellas-dactilares-son-unicas-y-permiten-identificarnos/>

Al respecto de los pliegues palmares, Martínez et al. (2016) refiere lo siguiente:

Aparecen 3 tipos de pliegues; el primero que se ubica en la región distal, este va desde el borde cubital, el primero que se ubica en la región distal, este va desde el borde cubital hasta el segundo espacio interdigital; el segundo corresponde al de la flexión proximal, el cual se extiende desde el borde radial hasta llegar al borde cubital, el tercer pliegue es aquel que es longitudinal es decir, envuelve la eminencia tenar. La disposición y configuración de estos pliegues se verán afectadas por factores genéticos y ambientales. (p.14)

Para tener una idea más concreta de los pliegues palmares, se puede acudir a la Figura 5.

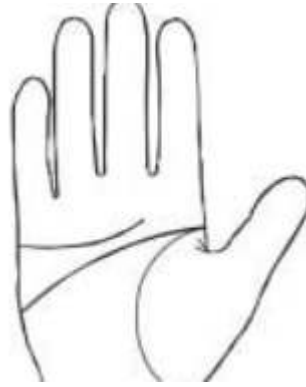


Figura (5). Pliegues de la región palmar de una mano humana

Fuente: https://www.researchgate.net/figure/Figura-1-Pliegues-palmares-esperados-LT-longitudinal-tenar-PP-palmar-proximal-PD_fig1_228500604

Estos pliegues son de gran interés para el presente trabajo, ya que serán los que deban ser captados por un sensor (cámara digital C-922 de Logitech) para luego ser preprocesados mediante programación Python y posteriormente ser procesados y clasificados mediante RNC (Redes Neuronales Convolucionales) con la finalidad de identificar a los usuarios del laboratorio de Odontología de la USMP- Santa Anita.

La presente tesis, entonces, adoptará la postura de Martínez et al. (2016) y se limitará a denominar los patrones palmares como líneas o pliegues mas no dermatoglifos.

2.3.2 Base Teórica de adquisición de imágenes

Aquí se menciona brevemente el espectro electromagnético como la fuente de iluminación; también se explica la forma de adquisición de la imagen por parte del sensor (cámara digital) y finalmente, se habla de forma concisa de la imagen en formato digital.

2.3.2.1 Espectro electromagnético y la luz

La principal fuente de energía para muchas imágenes hoy en día es el espectro electromagnético, siendo este tipo de imágenes las más familiares (Gonzalez y Woods, 2007). Los rangos de longitud de onda que abarca el EM se pueden apreciar en la Figura 6.

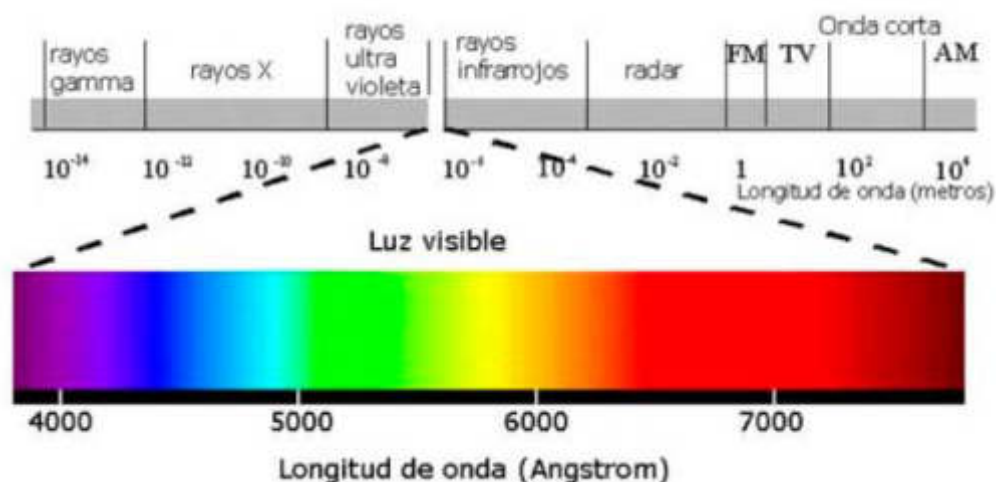


Figura (6). Representación de los intervalos de longitud de onda del espectro electromagnético

Fuente: <https://sobrecuriosidades.com/2014/03/10/que-es-el-espectro-electromagnetico/>

Se dice que la luz posee la particularidad de comportarse en forma dual (partícula y onda), como partícula adquiere el nombre de fotones (ver Figura 7); mientras que como onda tiene características tanto eléctricas como magnéticas, generando una perturbación electromagnética a medida que se propaga por el medio (Mondragón, 2017), tal y como se aprecia en la Figura 8:

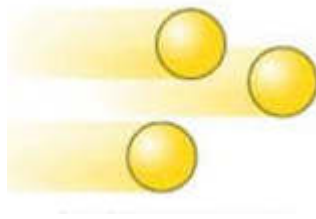


Figura (7). Fotones que representan a la luz como partícula

Fuente: <http://www.educando.edu.do/portal/origen-la-luz/>



Figura (8). Propagación de la luz como una onda

Fuente: <http://www.educando.edu.do/portal/origen-la-luz/>

De no ser por una fuente luminosa, la adquisición de imágenes sería virtualmente imposible para este o cualquier trabajo que aborde problemas del campo de la visión y por ello era pertinente mencionarlo en el marco teórico.

2.3.2.2 Adquisición de imágenes

Las imágenes son comúnmente generadas por una combinación de una determinada fuente de iluminación y de un objeto que sea capaz de absorber o reflejar dicha energía. Para este propósito existen tres tipos de arreglos de sensores que transforman la energía luminosa proveniente de la fuente de iluminación en imágenes digitales, dicha energía es tomada y transformada en voltaje gracias a una combinación de la energía eléctrica de entrada y el material del sensor que responderá a una particular clase de energía a ser detectada. La forma de onda de voltaje se convierte

entonces en la salida del sensor, la cual es posteriormente digitalizada para obtener una cantidad digital (Gonzalez y Woods, 2007).

Las tres formas principales de adquisición de imágenes que Gonzalez y Woods mencionan en su libro son:

- Adquisición con sensor único
- Adquisición con tiras de sensores (usado en escáneres)
- Adquisición con matrices de sensores (usado por las cámaras digitales)

La adquisición de imágenes de las palmas de los usuarios en el presente trabajo se lleva a cabo con una cámara Logitech C-922 de 1080p a 30 tramas por segundo (ver figura 9); dado que la C-922 es una cámara digital su adquisición de imágenes se lleva a cabo con la 3era forma mencionada líneas arriba.



Figura (9). Cámara utilizada para la adquisición de las imágenes de las palmas del personal

Fuente: <https://www.logitech.com/es-es/product/c922-pro-stream-webcam>

Al respecto de este tipo de sensores Gonzalez et al. (2007) refiere que:

La respuesta de cada sensor es proporcional a la integral de la energía luminosa proyectada en la superficie sensorial, una propiedad utilizada en astronomía y otras aplicaciones que requieren de imágenes con poco ruido. La reducción de ruido se obtiene al dejar que el sensor integre la señal de entrada de luz por minutos o inclusive horas. Dado que la matriz de sensado es de dos dimensiones, tiene la ventaja de que una imagen completa puede obtenerse al concentrar el patrón de energía sobre la superficie de la matriz. (p. 73)

La forma en que las cámaras digitales captan la energía luminosa y la transforman en imágenes digitales se aprecia en la figura 10.

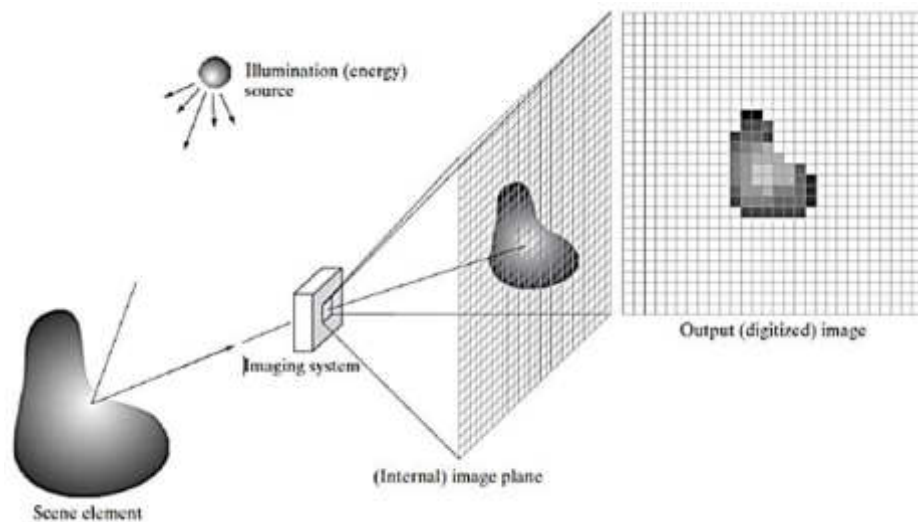


Figura (10). Proceso de obtención de una imagen digital
Fuente: Digital Image Processing. Rafael Gonzalez y Richard Woods.
Tercera Edición. Prentice Hall. 2007. p.73

2.3.2.3 La Imagen digital

Las imágenes pueden ser denotadas como funciones bidimensionales de la forma $f_s(x_s, y_s)$ el valor o amplitud de la función f_s en las coordenadas espaciales (x_s, y_s) viene a ser una cantidad positiva y escalar, cuyo significado físico será determinado por la fuente de la misma imagen. Si la imagen es generada por un proceso físico, sus valores serán proporcionales a la energía irradiada desde una fuente física, por ello la función $f_s(x_s, y_s)$ debe ser no nula y finita (Gonzalez et al., 2007).

De acuerdo con Gonzalez et al. (2007), la función $f_s(x_s, y_s)$ está compuesta por dos componentes:

1. La iluminación, que se refiere a la cantidad de luz que procede de una fuente y que incide sobre la escena.
2. Reflectancia, que se refiere a la cantidad de luz reflejada por el objeto.

De tal manera que $f_s(x_s, y_s)$ queda definida de la forma siguiente:

$$f_s(x_s, y_s) = i_s(x_s, y_s)r_s(x_s, y_s) \quad (1.0)$$

Donde:

i_s : Iluminación.

r_s : Reflectancia.

Además:

$$0 < i_s(x_s, y_s) < \infty$$

$$0 < r_s(x_s, y_s) < 1$$

Existe una gran variedad de maneras para obtener imágenes, aunque el objetivo permanece inmutable, el cual consiste en

generar una imagen digital a partir de información sensada lo cual implica, justamente, convertir los datos detectados por el sensor a forma digital a través de dos procesos: cuantificación y muestreo (Gonzalez et al., 2007).

Una imagen puede ser continua tanto en sus coordenadas x-y como en su amplitud, el proceso para digitalizar los valores de coordenadas se denomina muestreo, mientras que el proceso para la digitalización de la amplitud es llamado cuantificación. En la práctica el método de muestreo es determinado por el arreglo del sensor (o sensores) usado para generar dicha imagen; y tanto la cuantificación como el muestreo tienen un impacto determinante en la calidad de la imagen digital (Gonzalez et al., 2007). Para apreciar de mejor manera ambos procesos, se puede recurrir a la Figura 11.

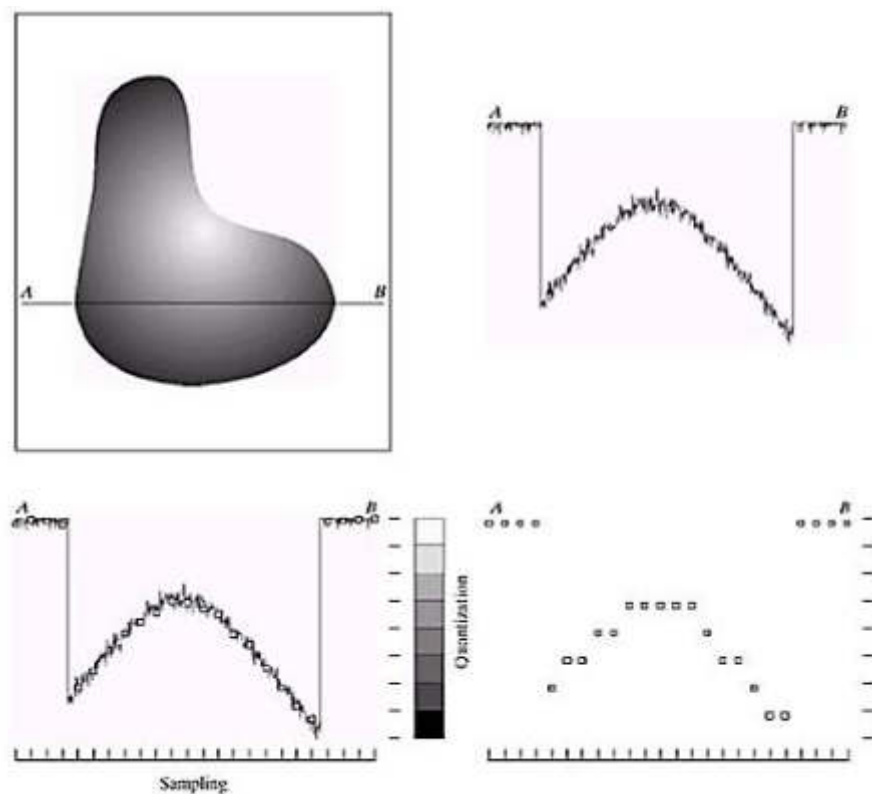


Figura (11). Representación de una imagen capturada (superior izquierda), la amplitud de dicha imagen a la largo de un segmento AB (superior derecha), el muestreo de la imagen (inferior izquierda) y la cuantificación correspondiente (inferior derecha)

Fuente: Digital Image Processing. Rafael Gonzalez y Richard Woods.
Tercera Edición. Prentice Hall. 2007. p.53

Como resultado del muestreo y la cuantificación se obtiene una matriz de números reales de tamaño $M \times N$ (en este caso se presenta la matriz de una imagen en escala de grises):

$$f_s(x_s, y_s) = \begin{bmatrix} f_s(0,0) & \cdots & f_s(0, N-1) \\ \vdots & \ddots & \vdots \\ f_s(M-1,0) & \cdots & f_s(M-1, N-1) \end{bmatrix}$$

Sin embargo, en el presente trabajo se utilizan imágenes digitales en modelo RGB el cual también se presenta con una resolución espacial de $M \times N$, con la diferencia que ahora se cuentan con tres canales o componentes (correspondientes a rojo, verde y azul), en donde cada componente tiene 8 bits y 256 tonalidades. La representación matricial de una imagen RGB se muestra a continuación:

R:

$$f_s(x_s, y_s, 1) = \begin{bmatrix} f_s(0,0,1) & \cdots & f_s(0, N-1,1) \\ \vdots & \ddots & \vdots \\ f_s(M-1,0,1) & \cdots & f_s(M-1, N-1,1) \end{bmatrix}$$

G:

$$f_s(x_s, y_s, 2) = \begin{bmatrix} f_s(0,0,2) & \cdots & f_s(0, N-1,2) \\ \vdots & \ddots & \vdots \\ f_s(M-1,0,2) & \cdots & f_s(M-1, N-1,2) \end{bmatrix}$$

B:

$$f_s(x_s, y_s, 3) = \begin{bmatrix} f_s(0,0,3) & \cdots & f_s(0, N-1,3) \\ \vdots & \ddots & \vdots \\ f_s(M-1,0,3) & \cdots & f_s(M-1, N-1,3) \end{bmatrix}$$

Aparte de M y N , que no tienen restricción alguna salvo ser enteros positivos, aparece el concepto de niveles de intensidad representados por L ; que resulta útil a la hora de considerar el almacenamiento de información. Para ello, el número de niveles de intensidad se entiende como un número entero potencia de 2, se asume que los niveles discretos son (Gonzalez et al., 2007). Entonces:

$$L = 2^{\rho} \quad (2.0)$$

Donde:

L : Niveles de intensidad

ρ : Número entero positivo

2.3.3 Base Teórica de la Etapa de Pre-Procesamiento

En este apartado se habla de la interpolación bilineal para el re-escalamiento de las imágenes utilizando lenguaje de programación Python.

2.3.3.1 Re-Escalamiento de imágenes

Las imágenes adquiridas con la cámara C-922 tienen un tamaño de 1920x1080 píxeles, este tamaño generaría demasiada carga computacional si se ingresara directamente a la capa de entradas de la red neuronal convolucional, para ello la imagen que contiene los patrones de las líneas palmares de los usuarios debe ser re-escalada antes de ingresar a la RNC.

Para reducir (o agrandar) las dimensiones de una imagen se utiliza la interpolación de imagen, la cual usa la data existente para valores en locaciones desconocidas (Gonzalez et al., 2007). La función para cambiar el tamaño de una imagen en Python (denominada *resize*) usa por defecto la interpolación bilineal la cual (de acuerdo con Gonzalez y Woods) toma los cuatro vecinos más cercanos de un píxel para estimar la intensidad en un punto dado. Sea la siguiente ecuación:

$$V(x_p, y_p) = a_p x_p + b_p y_p + c_p (x_p y_p) + d_p \quad (3.0)$$

Donde:

V : Intensidad del píxel en las coordenadas (x_p, y_p)

a_p, b_p, c_p, d_p : Coeficientes

Dichos coeficientes se determinan por álgebra lineal por medio de cuatro ecuaciones en cuatro pares de coordenadas distintas. Sean entonces los cuatro puntos definidos de la siguiente manera:

$$(x_{p1}, y_{p1}) = Q_{11}$$

$$(x_{p2}, y_{p2}) = Q_{12}$$

$$(x_{p3}, y_{p3}) = Q_{21}$$

$$(x_{p4}, y_{p4}) = Q_{22}$$

Donde:

$x_{p(1...4)}, y_{p(1...4)}$: Pares de coordenadas de los 4 vecinos más cercanos al par (x_p, y_p)

$Q_{11}, Q_{12}, Q_{21}, Q_{22}$: Puntos asociados a las coordenadas $(x_{p1}, y_{p1}), (x_{p2}, y_{p2}), (x_{p3}, y_{p3})$ y (x_{p4}, y_{p4}) respectivamente

$f(Q_{11}), f(Q_{12}), f(Q_{21}), f(Q_{22})$: Intensidad de píxel en los puntos Q_{11}, Q_{12}, Q_{21} y Q_{22} respectivamente

El sistema lineal que define a los coeficientes de la ecuación 3.0 se describe de la siguiente manera:

$$\begin{bmatrix} 1 & x_{p1} & y_{p1} & x_{p1}y_{p1} \\ 1 & x_{p1} & y_{p2} & x_{p1}y_{p2} \\ 1 & x_{p2} & y_{p1} & x_{p2}y_{p1} \\ 1 & x_{p2} & y_{p2} & x_{p2}y_{p2} \end{bmatrix} \begin{bmatrix} d_p \\ a_p \\ b_p \\ c_p \end{bmatrix} = \begin{bmatrix} f(Q_{11}) \\ f(Q_{12}) \\ f(Q_{21}) \\ f(Q_{22}) \end{bmatrix}$$

Los coeficientes entonces quedan como:

$$d_p = \frac{f(Q_{11})x_{p2}y_{p2}}{(x_{p1}-x_{p2})(y_{p1}-y_{p2})} + \frac{f(Q_{12})x_{p2}y_{p1}}{(x_{p1}-x_{p2})(y_{p2}-y_{p1})} + \frac{f(Q_{21})x_{p1}y_{p2}}{(x_{p1}-x_{p2})(y_{p2}-y_{p1})} + \frac{f(Q_{22})x_{p1}y_{p1}}{(x_{p1}-x_{p2})(y_{p1}-y_{p2})}$$

$$a_p = \frac{f(Q_{11})y_{p2}}{(x_{p1}-x_{p2})(y_{p2}-y_{p1})} + \frac{f(Q_{12})y_{p1}}{(x_{p1}-x_{p2})(y_{p1}-y_{p2})} + \frac{f(Q_{21})y_{p2}}{(x_{p1}-x_{p2})(y_{p2}-y_{p1})} + \frac{f(Q_{22})y_{p1}}{(x_{p1}-x_{p2})(y_{p2}-y_{p1})}$$

$$b_p = \frac{f(Q_{11})x_{p2}}{(x_{p1}-x_{p2})(y_{p2}-y_{p1})} + \frac{f(Q_{12})x_{p2}}{(x_{p1}-x_{p2})(y_{p1}-y_{p2})} + \frac{f(Q_{21})x_{p1}}{(x_{p1}-x_{p2})(y_{p1}-y_{p2})} + \frac{f(Q_{22})x_{p1}}{(x_{p1}-x_{p2})(y_{p2}-y_{p1})}$$

$$c_p = \frac{f(Q_{11})}{(x_{p1}-x_{p2})(y_{p2}-y_{p1})} + \frac{f(Q_{12})}{(x_{p1}-x_{p2})(y_{p2}-y_{p1})} + \frac{f(Q_{21})}{(x_{p1}-x_{p2})(y_{p2}-y_{p1})} + \frac{f(Q_{22})}{(x_{p1}-x_{p2})(y_{p1}-y_{p2})}$$

Para tener una idea de cómo se aplica lo anteriormente descrito en una imagen supóngase que se quiere reescalar una imagen de 3x3 a una imagen de 2x2, para ello considérese 1 de los 3 canales del formato RGB:

A_0	B_0	C_0
D_0	E_0	F_0
G_0	H_0	I_0

Se tiene lo siguiente:

$$\text{Factor} = \frac{\text{Tamaño de la matriz Original}}{\text{Tamaño de la matriz nueva}} = \frac{3}{2}$$

Para el siguiente ejemplo los factores de crecimiento y de submuestreo se definen como:

$$\text{Factor de crecimiento} = \text{Tamaño de la matriz nueva} = 2$$

$$\text{Factor de submuestreo} = \text{Tamaño de la matriz Original} - 1 = 3 - 1 = 2$$

La matriz crece entonces 2 veces (el doble) por interpolación bilineal y luego se ejecuta el submuestreo:

	A_0	Z_{12}	B_0	Z_{14}	C_0	Z_{16}
	Z_{21}	Z_{22}	Z_{23}	Z_{24}	Z_{25}	Z_{26}
y_{p2}	D_0	Z_{32}	E_0	Z_{34}	F_0	Z_{36}
	Z_{41}	Z_{42}	Z_{43}	Z_{44}	Z_{45}	Z_{46}
y_{p1}	G_0	Z_{52}	H_0	Z_{54}	I_0	Z_{56}
	Z_{61}	Z_{62}	Z_{63}	Z_{64}	Z_{65}	Z_{66}

x_{p1}

x_{p2}

Considérese además que:

- $A_0 = 121$
- $B_0 = 159$
- $C_0 = 50$
- $D_0 = 118$
- $E_0 = 80$
- $F_0 = 100$
- $G_0 = 90$
- $H_0 = 110$
- $I_0 = 120$

Si se quisiese saber la intensidad del píxel Z_{44} se debe seguir el procedimiento de la ecuación 3.0, por lo que:

$$V(x_p, y_p) = a_p(x_p) + b_p(y_p) + c_p(x_p)(y_p) + d_p$$

$$V(x_p, y_p) = Z_{44} = a_p(4) + b_p(4) + c_p(4)(4) + d_p$$

Hay que hallar los coeficientes a_p , b_p , c_p y d_p utilizando los 4 vecinos más cercanos a Z_{44} que son: E_0 , F_0 , H_0 y I_0 . Para ello se tiene la matriz:

$$\begin{bmatrix} 1 & x_{p1} & y_{p1} & x_{p1}y_{p1} \\ 1 & x_{p1} & y_{p2} & x_{p1}y_{p2} \\ 1 & x_{p2} & y_{p1} & x_{p2}y_{p1} \\ 1 & x_{p2} & y_{p2} & x_{p2}y_{p2} \end{bmatrix} \begin{bmatrix} d_p \\ a_p \\ b_p \\ c_p \end{bmatrix} = \begin{bmatrix} f(Q_{11}) \\ f(Q_{12}) \\ f(Q_{21}) \\ f(Q_{22}) \end{bmatrix} = \begin{bmatrix} H_0 \\ E_0 \\ I_0 \\ F_0 \end{bmatrix}$$

Hallando d_p :

$$d_p = \frac{H_0 x_{p2} y_{p2}}{(x_{p1} - x_{p2})(y_{p1} - y_{p2})} + \frac{E_0 x_{p2} y_{p1}}{(x_{p1} - x_{p2})(y_{p2} - y_{p1})} + \frac{I_0 x_{p1} y_{p2}}{(x_{p1} - x_{p2})(y_{p2} - y_{p1})} + \frac{F_0 x_{p1} y_{p1}}{(x_{p1} - x_{p2})(y_{p1} - y_{p2})}$$

$$d_p = \frac{110(5)(3)}{(-2)(2)} + \frac{80(5)(5)}{(-2)(-2)} + \frac{120(3)(3)}{(-2)(-2)} + \frac{100(3)(5)}{(-2)(2)}$$

$$d_p = -412.5 + 500 + 270 - 375 = -17.5$$

Hallando a_p :

$$a_p = \frac{H_0 y_{p2}}{(x_{p1} - x_{p2})(y_{p2} - y_{p1})} + \frac{E_0 y_{p1}}{(x_{p1} - x_{p2})(y_{p1} - y_{p2})} + \frac{I_0 y_{p2}}{(x_{p1} - x_{p2})(y_{p1} - y_{p2})} + \frac{F_0 y_{p1}}{(x_{p1} - x_{p2})(y_{p2} - y_{p1})}$$

$$a_p = \frac{110(3)}{(-2)(-2)} + \frac{80(5)}{(-2)(2)} + \frac{120(3)}{(-2)(2)} + \frac{100(5)}{(-2)(-2)}$$

$$a_p = 82.5 - 100 - 90 + 125 = 17.5$$

Hallando b_p :

$$b_p = \frac{H_0 x_{p2}}{(x_{p1} - x_{p2})(y_{p2} - y_{p1})} + \frac{E_0 x_{p2}}{(x_{p1} - x_{p2})(y_{p1} - y_{p2})} + \frac{I_0 x_{p1}}{(x_{p1} - x_{p2})(y_{p1} - y_{p2})} + \frac{F_0 x_{p1}}{(x_{p1} - x_{p2})(y_{p2} - y_{p1})}$$

$$b_p = \frac{110(5)}{(-2)(-2)} + \frac{80(5)}{(-2)(2)} + \frac{120(3)}{(-2)(2)} + \frac{100(3)}{(-2)(-2)}$$

$$b_p = 137.5 - 100 - 90 + 75 = 22.5$$

Hallando c_p :

$$c_p = \frac{H_0}{(x_{p1} - x_{p2})(y_{p1} - y_{p2})} + \frac{E_0}{(x_{p1} - x_{p2})(y_{p2} - y_{p1})} + \frac{I_0}{(x_{p1} - x_{p2})(y_{p2} - y_{p1})} + \frac{F_0}{(x_{p1} - x_{p2})(y_{p1} - y_{p2})}$$

$$c_p = \frac{110}{(-2)(2)} + \frac{80}{(-2)(-2)} + \frac{120}{(-2)(-2)} + \frac{100}{(-2)(2)}$$

$$c_p = -27.5 + 20 + 30 - 25 = -2.5$$

Ahora se utilizan los coeficientes hallados para encontrar la intensidad de Z_{44} :

$$Z_{44} = 17.5(4) + 22.5(4) - 2.5(4)(4) - 17.5 = 102.5$$

El mismo procedimiento se debe seguir para los demás píxeles. Para hacer el submuestreo en el ejemplo, se calculó un factor de 2, de manera que por cada columna que se toma se eliminan las dos que le suceden, lo mismo ocurre para las filas:

	✓	✗	✗	✓	✗	✗
✓	A_0	Z_{12}	B_0	Z_{14}	C_0	Z_{16}
✗	Z_{21}	Z_{22}	Z_{23}	Z_{24}	Z_{25}	Z_{26}
✗	D_0	Z_{32}	E_0	Z_{34}	F_0	Z_{36}
✓	Z_{41}	Z_{42}	Z_{43}	Z_{44}	Z_{45}	Z_{46}
✗	G_0	Z_{52}	H_0	Z_{54}	I_0	Z_{56}
✗	Z_{61}	Z_{62}	Z_{63}	Z_{64}	Z_{65}	Z_{66}

La matriz resultante de 2x2 es:

A_0	Z_{14}
Z_{41}	Z_{44}

2.3.4 Base Teórica de la Etapa de Procesamiento

Aquí se menciona todo lo concerniente a las redes neurales convolutivas aplicadas a las imágenes re-escaladas. Dichas imágenes RGB reescaladas se convierten en la data de entrada a la red; sin embargo, antes de pasar detallar las distintas etapas que transcurren dentro de una RNC se debe precisar qué cosa es una red neuronal.

2.3.4.1 Redes Neuronales

Simon Haykin (2008) define a las redes neurales de la forma siguiente:

Una red neuronal es un procesador masivamente distribuido en paralelo conformado por unidades de procesamiento simples, que tiene una propensión natural a almacenar conocimiento experimental y hacerlo disponible para su uso. Se asemeja al cerebro en dos aspectos:

1. El conocimiento es adquirido por la red desde su entorno a través de un proceso de aprendizaje.
2. Fuerza de conexión interneuronal, conocida como pesos sinápticos, son empleados para almacenar el conocimiento adquirido. (p.2)

Estas redes son de naturaleza artificial, pero aun así tratan de emular las funciones de un cerebro humano hasta cierto nivel. Las unidades básicas de una red neural artificial son las neuronas artificiales (Haykin, 2008), y a continuación se procederá a explicar cómo están modeladas matemáticamente estas unidades.

Como se puede apreciar en la figura 12, un modelo neural (de acuerdo con Haykin, 2008) está conformado por tres elementos básicos:

1. Un conjunto de sinapsis o enlaces de conexión entre neuronas, si se considerase una señal OL_j , esta señal de entrada (*input signal*) tiene una sinapsis j que la conecta a una neurona k ; dicha sinapsis tiene un peso sináptico (también llamados

synaptic weights) denotado como W_{kj} (este peso puede adoptar tanto valores positivos como valores negativos).

2. Un sumador (*summing junction*) para sumar las señales de entrada multiplicadas por los pesos sinápticos.
3. Una función de activación (o *activation function*) que limita la amplitud de la salida (*output*) de una neurona.

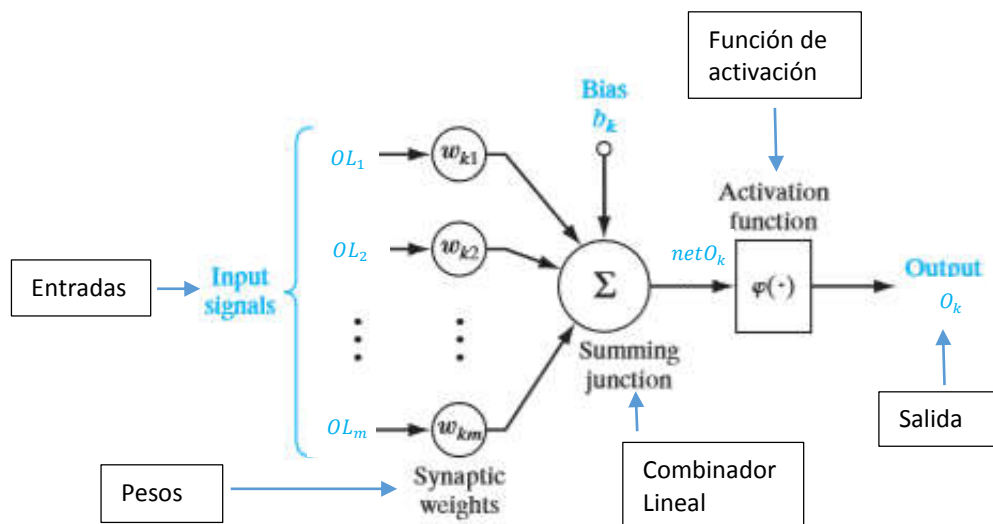


Figura (12). Modelo generalizado no-lineal de una neurona
 Fuente: *Neural Networks and Learning Machines*. Simon Haykin. Tercera Edición. Prentice Hall. 2008. p.11. (Nota: La figura fue editada por los autores para evitar colisión de parámetros)

Matemáticamente, lo descrito líneas arriba se entiende como:

$$U_k = \sum_{j=1}^m OL_j W_{kj} \quad (4.0)$$

Siendo U_k la salida del combinador lineal, adicionalmente a estos 3 elementos básicos de un modelo neural, existe

también un elemento externo a la neurona llamado umbral o sesgo b_k (también llamado *bias*), cuya función es la de ajustar la salida junto con la suma ponderada de las entradas a la neurona (Haykin, 2008); entonces la ecuación 4.0 no resulta suficiente para denotar el rol del *bias* en el modelo neural por lo que se escribe la variable $netO_k$:

$$netO_k = \sum_{j=1}^m OL_j W_{kj} + b_k \quad (4.1)$$

El *bias* b_k puede ser acoplado al modelo neuronal de manera más elegante si se le considera como el peso sináptico de una conexión que une a la neurona k con una entrada fijada en 1 (tal y como se aprecia en la figura 13):

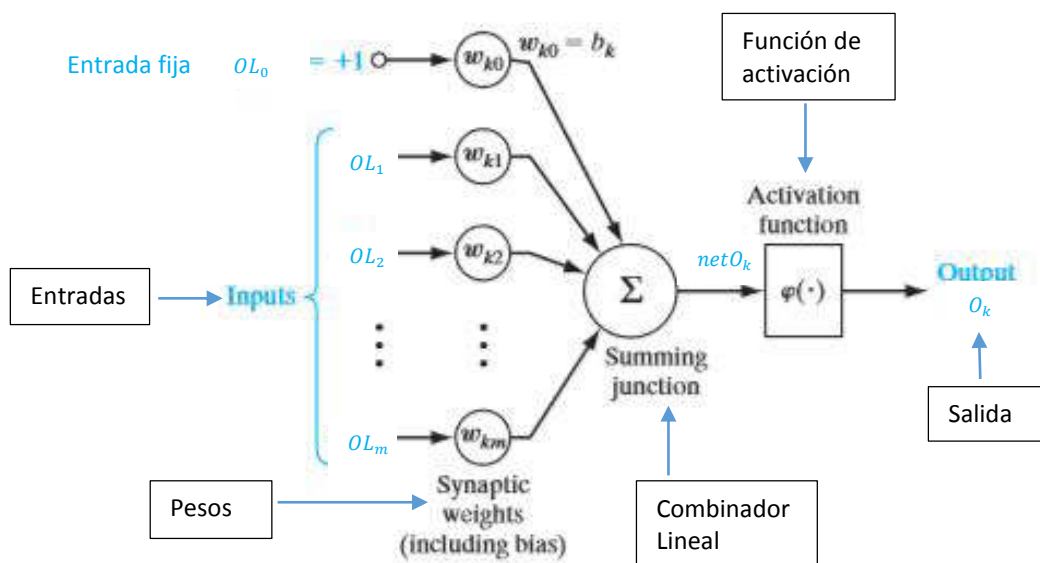


Figura (13). Modelo generalizado no-lineal de una neurona con el *bias* incorporado como un peso sináptico más

Fuente: *Neural Networks and Learning Machines*. Simon Haykin. Tercera Edición. Prentice Hall. 2008. p.12 (Nota: La figura fue editada por los autores para evitar colisión de parámetros)

Finalmente V_k pasa por una función de activación φ para dar así con la salida O_k de la neurona k, ello queda denotado como:

$$O_k = \varphi(V_k) \quad (4.2)$$

Existen muchas funciones de activación que se han utilizado para distintos fines con el tiempo, algunas de ellas son:

- Función identidad
- Función escalón unitario binario
- Función escalón unitario bipolar
- Función sigmoide binaria
- Función tangente hiperbólica
- Etc.

Para el presente trabajo se emplearán dos funciones de activación, las cuales son: ReLU (Rectified Linear Unit) y Softmax (también llamada función exponencial normalizada o softargmax).

ReLU es una función de activación que fue empleada por la red Alexnet, una de las primeras redes neurales convolutivas; creada por Alex Krizhevsky, Ilya Sutskever y Geoffrey E. Hinton en su trabajo llamado: *"ImageNet Classification with Deep Convolutional Neural Networks"* en el año 2012, este paper presenta a ReLU como una alternativa más eficiente que las opciones de función de activación tradicionales como la tangente hiperbólica y la función sigmoide, por ello es utilizada actualmente en aplicaciones que involucran a redes neurales profundas para resolver problemas relacionados a la visión computarizada (como el presente trabajo). Esta función se puede entender más claramente con su fórmula matemática:

$$y_{ReLU} = \max(0, x_{ReLU}) \quad (5.0)$$

Donde:

x_{ReLU} : Entrada a la función de activación ReLU.

La salida y_{ReLU} es definida entonces como la parte positiva de su argumento x_{ReLU} , arrojando 0 para todo valor negativo del mismo. Esto se puede apreciar gráficamente en la figura 14:

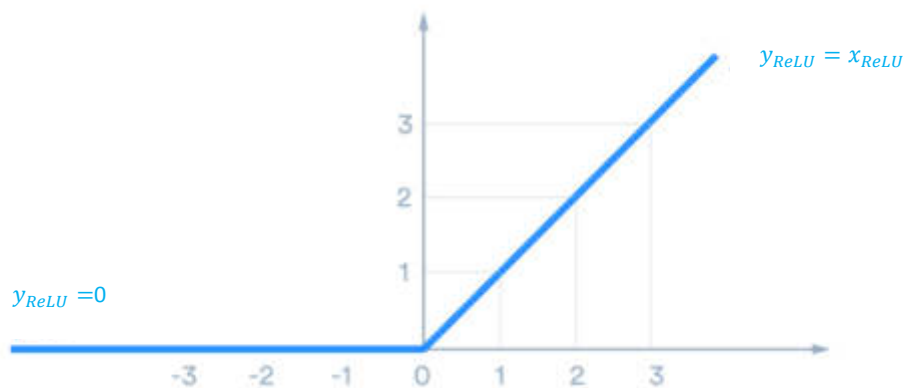


Figura (14). Apreciación gráfica de la función de activación ReLU

Fuente: <https://medium.com/tiny-mind/a-practical-guide-to-relu-b83ca804f1f7>.

(Nota: Los nombres de los 2 ejes cartesianos de la figura fueron editados para evitar colisión de parámetros)

Esta identidad lineal para valores positivos (y cero para los negativos) tiene los siguientes efectos:

- Facilidad de cómputo, ya que el cálculo de la gradiente en la etapa de aprendizaje es bastante sencillo para ReLU (1 si x_{ReLU} es positivo, 0 para el caso contrario)
- Convergencia rápida, es consecuencia de la facilidad de cómputo; todas aquellas redes neurales convolucionales que operan con ReLU se entrenan muchas veces más rápido que sus

equivalentes que emplean otro tipo de función de activación tradicional (Krizhevsky, Sutskever y Hinton, 2012)

Softmax, por otro lado, se emplea para definir a la capa de salida de un modelo neural artificial (Michael Nielsen, 2015) y es frecuentemente usado en aplicaciones que involucran clasificación (como es el caso del presente trabajo).

Para entender la función Softmax, se imagina el vector \overrightarrow{netO} denotado como:

$$\overrightarrow{netO} = [netO_1, netO_2, netO_3, netO_4, netO_5]$$

Donde el número de elementos de \overrightarrow{netO} coincide con el número de K clases distintas en la capa de salida de una red neuronal que debe realizar un proceso de clasificación (en este caso K=5 para simplificar el ejemplo). Ahora, si se tratase a cada de elemento de \overrightarrow{netO} como una potencia del número e, el nuevo vector quedaría como:

$$\vec{p} = [e^{netO_1}, e^{netO_2}, e^{netO_3}, e^{netO_4}, e^{netO_5}]$$

Donde:

e: Base de logaritmo natural.

La sumatoria de todos los elementos de \vec{p} se describe como:

$$\sum_k e^{a_k} = e^{netO_1} + e^{netO_2} + e^{netO_3} + e^{netO_4} + e^{netO_5}$$

La función Softmax queda entonces definida como:

$$o_i = \varphi_i(\overline{netO}) = \frac{e^{netO_i}}{\sum_k e^{netO_k}} \quad (6.0)$$

Donde cada elemento de f es un número entre 0 y 1, y la sumatoria de todos los elementos de f es igual a la unidad (k es índice de tiempo discreto). La ecuación 6.0 implica que las activaciones de salida siempre serán positivas, lo que combinado al hecho de que la sumatoria de todas las activaciones es siempre 1 deja como conclusión que la salida de una capa Softmax es en realidad una distribución de probabilidad (Nielsen, 2015); lo cual justifica su aplicación en tareas de clasificación de imágenes, patrones, caracteres, símbolos, etc.

Hasta el momento se ha visto a las neuronas como elementos independientes que tienen una salida que resulta de la suma ponderada de los pesos o sinapsis que unen a la neurona a distintas entradas, también se vio como se aplican las funciones de activación a la salida del combinador lineal. Sin embargo, las redes neurales artificiales están compuestas por varias de estas neuronas artificiales, que están distribuidas por la misma red de tal manera que conforman distintas arquitecturas de red, una de las más empleadas es la red neural *feed forward* (es la arquitectura que se empleará para el presente trabajo), la cual se aprecia en la figura 15:

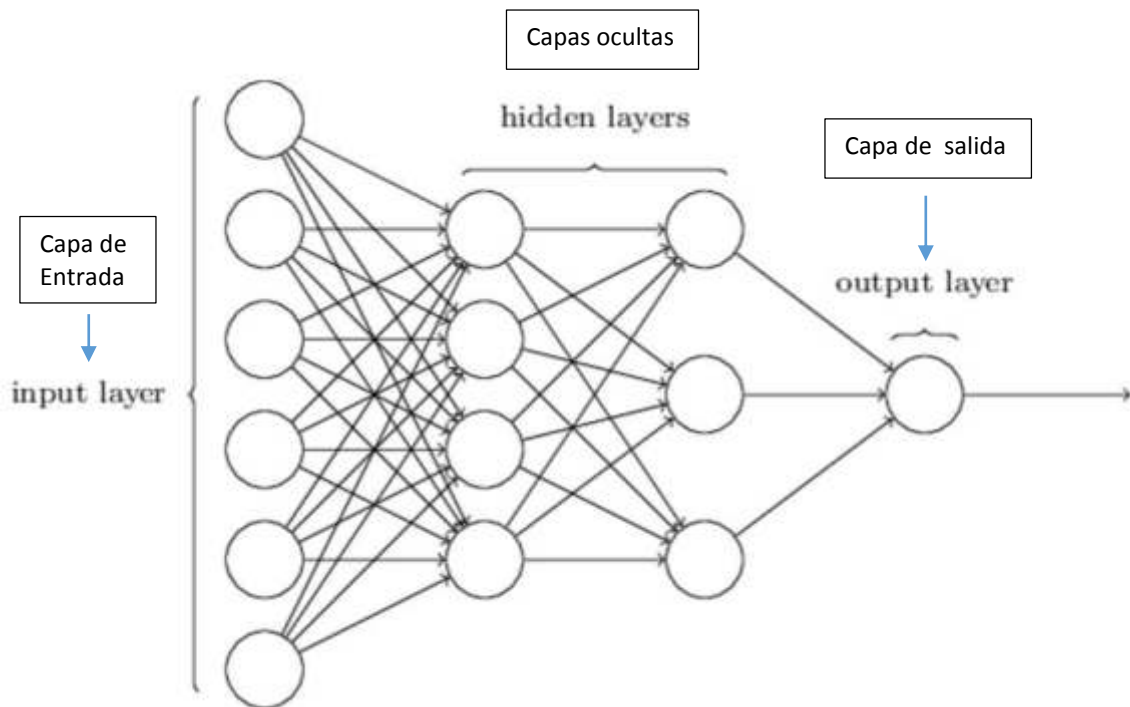


Figura (15). Arquitectura feed forward de una red neuronal artificial
 Fuente: Neural Networks and Deep Learning. Michael Nielsen. *Determination Press*. 2015. p.11

Las neuronas son distribuidas en columnas, estas columnas reciben el nombre de capas; la capa del extremo izquierdo recibe el nombre de capa de entrada (o *input layer*) y la capa del extremo derecho se denomina capa de salida (o también llamada *output layer*), las capas que se encuentran entre la capa de entrada y la capa de salida reciben el nombre de capas escondidas (o *hidden layers*). Para una red neuronal *feed forward* de N capas lo siguiente siempre se cumple:

- Hay 1 capa de entrada
- Hay 1 capa de salida
- Hay N-2 capas escondidas

2.3.4.2 Aprendizaje en Redes Neuronales

Se debe encontrar un algoritmo que permita encontrar pesos (*weights*) y sesgos (*biases*) tal que la salida de la red se aproxime a la salida deseada; y para cuantificar que tan bien se consigue lo expuesto se define una función de costo o pérdida (Nielsen, 2015).

Existen distintas funciones de pérdida, una de las más utilizadas en redes neurales es la función del error cuadrático medio (o MSE por sus siglas en inglés) definida por:

$$E = \frac{1}{2} \sum_i (y_{Ti} - O_i)^2 \quad (7.0)$$

Donde:

y_{Ti} : Salida deseada para el nodo de salida i

O_i : Salida del nodo de salida i

Normalmente se emplearía la ecuación 7.0 para tratar de calcular el error global que produce una red neural, pero el presente trabajo lidia con la clasificación de patrones y ya se mencionó líneas arriba que la capa de salida de la red será una capa Softmax, es decir, una capa cuyas salidas representan la probabilidad que tiene una imagen (o patrón) de pertenecer a una de las K clases (o usuarios a clasificar). Para ello se requiere de una función que cuantifique la diferencia entre dos distribuciones de probabilidad, y para ello en el presente trabajo se empleara la función de entropía cruzada definida como:

$$E = - \sum_x y_{Ti} \ln(O_i) \quad (7.1)$$

Donde:

E : Función de pérdida

y_{Ti} : Salida o probabilidad deseada para el nodo de salida i

O_i : Salida o predicción de la red para el nodo de salida i

Para entenderla, considérese el ejemplo con el que se obtuvo la ecuación 6.0, donde cada elemento de $f(\vec{a})$ para $K=5$ es:

$$\varphi(\overline{netO}) = \left[\begin{array}{c} \frac{e^{netO_1}}{e^{netO_1} + e^{netO_2} + e^{netO_3} + e^{netO_4} + e^{netO_5}}, \frac{e^{netO_2}}{e^{netO_1} + e^{netO_2} + e^{netO_3} + e^{netO_4} + e^{netO_5}}, \frac{e^{netO_3}}{e^{netO_1} + e^{netO_2} + e^{netO_3} + e^{netO_4} + e^{netO_5}}, \\ \frac{e^{netO_4}}{e^{netO_1} + e^{netO_2} + e^{netO_3} + e^{netO_4} + e^{netO_5}}, \frac{e^{netO_5}}{e^{netO_1} + e^{netO_2} + e^{netO_3} + e^{netO_4} + e^{netO_5}} \end{array} \right]$$

Donde cada elemento de $\varphi(\overline{netO})$ representa la probabilidad que tiene una entrada a la red de pertenecer a cualquiera de las $K=5$ clases (recordar que cada elemento es un número real entre 0 y 1). Ahora considérese que se tiene un vector con la distribución de probabilidad deseada, para este ejemplo suponer que la entrada corresponde a la segunda clase, de manera que dicho vector queda definido como:

$$\vec{Y}_T = [0,1,0,0,0]$$

Ahora se requiere ver qué tan cercana es la distribución predicha por la red (cada elemento $\varphi(\overline{netO})$) de con respecto a la distribución verdadera (cada elemento de \vec{Y}_T). Para ello se aplica la ecuación 8.1:

$$\begin{aligned} E = & -(0) \ln \left(\frac{e^{netO_1}}{e^{netO_1} + e^{netO_2} + e^{netO_3} + e^{netO_4} + e^{netO_5}} \right) \\ & - (1) \ln \left(\frac{e^{netO_2}}{e^{netO_1} + e^{netO_2} + e^{netO_3} + e^{netO_4} + e^{netO_5}} \right) \\ & - (0) \ln \left(\frac{e^{netO_3}}{e^{netO_1} + e^{netO_2} + e^{netO_3} + e^{netO_4} + e^{netO_5}} \right) \\ & - (0) \ln \left(\frac{e^{netO_4}}{e^{netO_1} + e^{netO_2} + e^{netO_3} + e^{netO_4} + e^{netO_5}} \right) \\ & - (0) \ln \left(\frac{e^{netO_5}}{e^{netO_1} + e^{netO_2} + e^{netO_3} + e^{netO_4} + e^{netO_5}} \right) \end{aligned}$$

Quedando:

$$E = -(1) \ln \left(\frac{e^{netO_2}}{e^{netO_1} + e^{netO_2} + e^{netO_3} + e^{netO_4} + e^{netO_5}} \right)$$

Se observa que todas las posiciones de \vec{Y}_T son cero salvo la posición que contiene la probabilidad deseada (es decir el 100%, puesto que una entrada solo puede pertenecer a una sola clase), de manera que las contribuciones que realizan al error total de la red son literalmente nulas; entonces la ecuación 8.1 se puede reescribir para dar con una ecuación más simplificada en función del ejemplo que se acaba de utilizar:

$$E_i = -\ln \left(\frac{e^{netO_i}}{\sum_k e^{netO_k}} \right) \quad (7.2)$$

Pero recordar que $\varphi(\vec{netO}) = \frac{e^{netO_i}}{\sum_k e^{netO_k}}$, entonces reemplazando 6.0 en 7.2:

$$E_i = -\ln(\varphi(\vec{netO})) \quad (7.3)$$

La ecuación 7.3 entonces indica el error de la red al comparar qué tan cercana o alejada esta la predicción hecha por la red con respecto a la probabilidad real o deseada. Normalmente cuando se diseña una red neural, los pesos (y también los umbrales) son inicializados de manera aleatoria, ya se ha visto cómo los scores o net inputs (entiéndase las salidas del combinador lineal) de la capa de salida dependen de las activaciones de las neuronas de la capa anterior y de los pesos que unen a los nodos de salida con dichas neuronas), previo a ser ingresados a la función de activación Softmax.

No es descabellado entonces suponer que una inicialización aleatoria de los pesos produzca un error considerable en la salida, para ello se debe disponer de algún algoritmo que permita la

modificación o actualización de los pesos de manera que se minimice dicho error para garantizar una correcta clasificación de patrones o líneas palmares como es el caso del presente trabajo.

Dicho algoritmo en cuestión es el algoritmo de retro-propagación (mundialmente conocido como *Backpropagation*) el cual consiste en saber cómo es que un cambio en cada peso de la red afecta al error total producido en la ecuación 7.3 para luego proceder a actualizarlo de manera que dicho error converja a cero; para ello se emplea la Regla Delta Generalizada en donde se halla la derivada parcial con la regla de la cadena la cual se muestra a continuación:

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{O_j} \frac{\partial O_j}{\partial netO_j} \frac{\partial netO_j}{\partial w_{ij}} \quad (8.0)$$

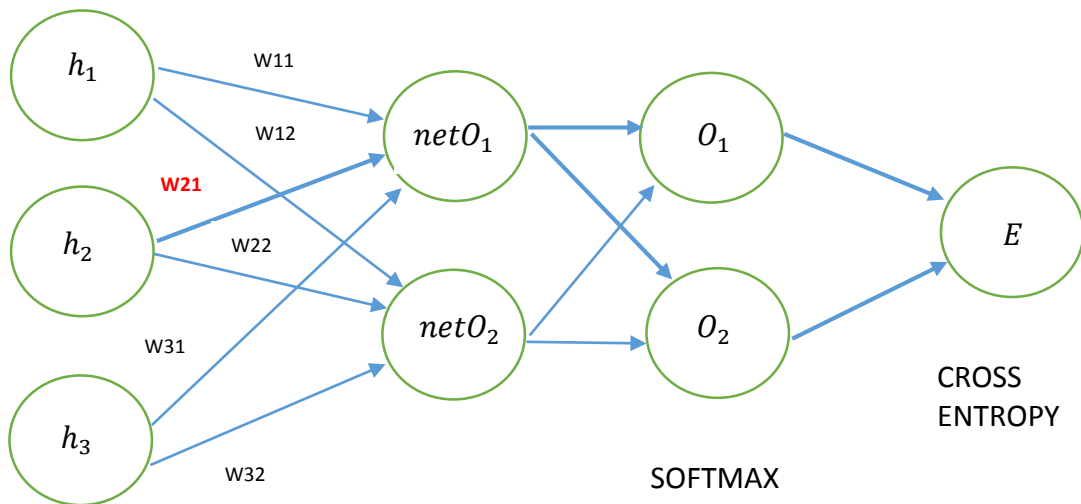
Donde:

$\frac{\partial E}{\partial O_j}$: Derivada de la función de pérdida con respecto a la función de activación.

$\frac{\partial O_j}{\partial netO_j}$: Derivada de la función de activación con respecto a la salida del combinador lineal.

$\frac{\partial netO_j}{\partial w_{ij}}$: Derivada de la salida del combinador lineal con respecto al peso w_{ij} .

Para tener un poco más de claridad sobre cómo se emplea la regla de la cadena, considérese el siguiente modelo:



Supóngase además que se desea calcular la derivada de la función de pérdida (entropía cruzada) con respecto al peso w_{21} (nótese los caminos resaltados que conducen desde w_{21} hasta el nodo E). Se comienza por definir cada nodo:

- $E = -y_{T1} \log(O_1) - y_{T2} \log(O_2)$
- $O_1 = \frac{e^{netO_1}}{e^{netO_1} + e^{netO_2}}$
- $O_2 = \frac{e^{netO_2}}{e^{netO_1} + e^{netO_2}}$
- $netO_1 = h_1 w_{11} + h_2 w_{21} + h_3 w_{31}$
- $netO_2 = h_1 w_{12} + h_2 w_{22} + h_3 w_{32}$

Ahora se procede a calcular las derivadas parciales siguiendo el camino resaltado en el modelo en cuestión:

- $\frac{\partial E}{\partial O_1} = \frac{-y_{T1}}{O_1}$
- $\frac{\partial E}{\partial O_2} = \frac{-y_{T2}}{O_2}$

- $\frac{\partial O_1}{\partial netO_1} = \frac{e^{netO_1}}{e^{netO_1} + e^{netO_2}} + \left(\frac{e^{netO_1}}{e^{netO_1} + e^{netO_2}}\right)^2 = O_1(1 - O_1)$
- $\frac{\partial O_2}{\partial netO_1} = \frac{-e^{netO_2}e^{netO_1}}{(e^{netO_1} + e^{netO_2})^2} = -O_2O_1$
- $\frac{\partial netO_1}{\partial w_{21}} = h_2$

Finalmente se procede a aplicar regla de la cadena descrita en 8.0:

$$\frac{\partial E}{\partial w_{21}} = \frac{\partial E}{\partial O_1} \frac{\partial O_1}{\partial netO_1} \frac{\partial netO_1}{\partial w_{21}} + \frac{\partial E}{\partial O_2} \frac{\partial O_2}{\partial netO_1} \frac{\partial netO_1}{\partial w_{21}}$$

Nótese que hay una suma de dos reglas de cadena, esto se explica porque existen dos caminos que pasan por Y_1 que conducen a w_{21} y se deben sumar las derivadas que “pasan” por esos caminos. Para concluir, $\frac{dE}{dw_{21}}$ queda definido como:

- $\frac{\partial E}{\partial w_{21}} = \frac{-y_{T1}}{O_1} (O_1(1 - O_1))h_2 + \frac{-y_{T2}}{O_2} (-O_2O_1)h_2$
- $\frac{\partial E}{\partial w_{21}} = h_2(y_{T2}O_1 - y_{T1} + y_{T1}O_1)$
- $\frac{\partial E}{\partial w_{21}} = h_2(O_1(y_{T1} + y_{T2}) - y_{T1})$

La regla de la cadena permite evaluar qué tanto influye un peso w_{ij} en el error global E , pero no permite actualizar dicho peso (y los otros pesos de la red) de manera tal que la función de pérdida se minimice lo más posible.

El proceso por el cual una función $f(x)$ se minimiza (o maximiza dependiendo de la aplicación) al alterar un parámetro x se denomina optimización; en donde la función a minimizar es conocida como función de pérdida o función de coste (Goodfellow, Bengio y Courville, 2017). Para el presente trabajo la función de pérdida ya fue explicada líneas arriba y

definida en la ecuación 7.3, pero todavía no se define el tipo de optimización que permite que esa función decrezca lo más posible en función de los pesos.

La optimización más común (la más simple dicho sea de paso) es la optimización por descenso de gradiente que se define de la manera siguiente:

$$r_{new} = r_{old} - \eta \frac{\partial f(r)}{\partial r} \quad (9.0)$$

Donde:

η : *Learning Rate* (Factor de aprendizaje).

r : Parámetro a actualizar de manera que $f(r)$ se minimice.

La derivada de $f(r)$ proporciona entonces la pendiente de la función en el punto r , es decir brinda información sobre cómo debe cambiar la entrada x de manera que haya una ligera mejora en f , para una función que posea múltiples entradas se debe emplear el concepto de derivadas parciales y la regla de la cadena (esto se explicó líneas arriba) para determinar cómo afecta cada elemento de la función a la misma (Goodfellow et al., 2017).

Ver la figura 16 para apreciar cómo la derivada de una función se emplea para ayudar a dicha función a alcanzar su mínimo (puede ser local o global).

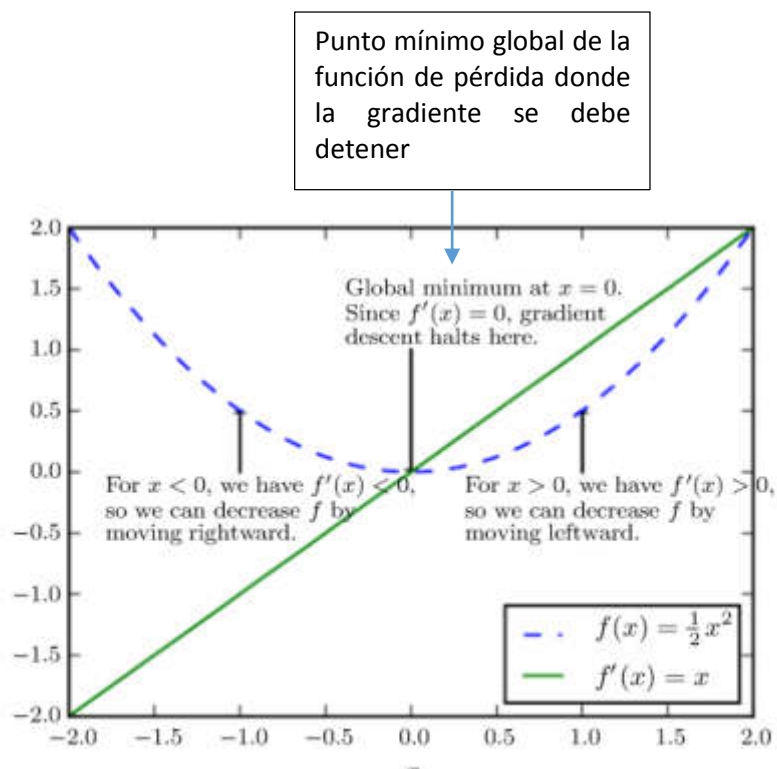


Figura (16). Representación gráfica de cómo el algoritmo del descenso de gradiente se empela para seguir a una función cuesta abajo hacia un mínimo
Fuente: *Deep Learning*. Ian Goodfellow, Yoshua Bengio y Aaron Courville.

The MIT Press. 2017. p.83

Para una red neural convencional la ecuación 9.0 quedaría expresada de la siguiente manera:

$$w_{ij}(new) = w_{ij}(old) - \eta \frac{\partial E}{\partial w_{ij}} \tag{9.1}$$

Lamentablemente la optimización por descenso de gradiente tiene una tasa de convergencia a cero demasiado lenta, si es que acaso η es muy pequeño; y si por otro lado el factor de aprendizaje es demasiado grande la función se puede “saltar” el verdadero punto mínimo.

Por ello con el tiempo se han propuesto variantes a este tipo de optimización como el descenso de gradiente estocástico y el descenso de gradiente por *mini-batch*; también se han propuesto otros métodos alternativos de optimización como:

- Adagrad
- Adadelta
- RMSProp
- Gradiente acelerada de Nesterov
- Etc.

En el año 2015, sin embargo, Diederik P. Kingma y Jimmy Lei Ba publicaron un paper titulado “Adam: A Method for Stochastic Optimization”. Este trabajo presenta al mundo un algoritmo para la optimización basada en gradientes de primer orden de funciones objetivas estocásticas y que se basa en estimaciones adaptivas de momentos de orden inferior (Kingma y Lei, 2015).

Entre las ventajas que Kingma y Lei mencionan sobre la optimización Adam en su *paper*, se resalta:

- Es un método de implementación directa
- El método es computacionalmente eficiente
- El método tiene pocos requerimientos de memoria
- Es adecuado para problemas que abarcan grandes cantidades de data y/o parámetros

Adam se constituye como un método de tasa de aprendizaje adaptiva, esto quiere decir que se calcula tasas de aprendizaje individuales para distintos parámetros (resolviendo el problema del descenso de gradiente que tiene un η estático y de naturaleza global), otra peculiaridad de dicho método es que emplea estimaciones del primer y segundo momento de la gradiente para adaptar dicho factor de aprendizaje para cada peso de la red neural. Para determinar estos momentos Adam se apoya en medias móviles calculadas con el gradiente evaluado en el *batch* (o en su defecto el *mini batch*) actual (Kingma et al, 2015). Dichos momentos según Kingma y Lei se denotan como:

$$m_{new} = \beta_1 m_{old} + (1 - \beta_1) g_t \quad (9.2)$$

$$v_{new} = \beta_2 m_{old} + (1 - \beta_2) g_t^2 \quad (9.3)$$

Donde:

m, v : Medias móviles o estimaciones del primer y segundo momento respectivamente

g_t : Gradiente en la iteración t

β_1 : Hiper-parámetro con valor por defecto de 0.9

β_2 : Hiper-parámetro con valor por defecto de 0.999

De acuerdo con Kingma et al, 2015; los vectores que contienen a las medias móviles se deben inicializar con cero para la primera iteración. Entonces se tiene lo siguiente para m (para v se sigue un proceso análogo/similar):

El momento se inicializa en 0:

$$m_0 = 0$$

Para la primera iteración:

$$m_1 = \beta_1 m_0 + (1 - \beta_1) g_1 = (1 - \beta_1) g_1$$

Para la segunda iteración:

$$m_2 = \beta_1(1 - \beta_1) g_1 + (1 - \beta_1) g_2$$

Para la tercera iteración:

$$m_3 = \beta_1^2(1 - \beta_1) g_1 + \beta_1(1 - \beta_1) g_2 + (1 - \beta_1) g_3$$

Para la iteración t , m_t queda definido (en forma general) como:

$$m_t = (1 - \beta_1) \sum_{i=0}^t \beta_1^{t-1} g_i \quad 9.4$$

Dado que m y v son estimaciones del primer y segundo momento respectivamente, se debería cumplir lo siguiente (idealmente):

$$E[m_t] = E[g_t] \quad 9.5$$

$$E[v_t] = E[g_t^2] \quad 9.6$$

Donde los valores esperados de los estimadores deberían ser iguales al parámetro que se desea estimar (en este caso la gradiente), de cumplirse esta propiedad se estaría hablando de estimadores imparciales. Lamentablemente esta propiedad no se cumple para las medias móviles, ya que al inicializarlas en cero los estimadores se vuelven parciales (o sesgados) hacia 0 en sus primeros momentos, para resolver este dilema se aplica corrección de sesgo, o también llamado *Bias Correction* (no confundir con el *bias* de las capas de una red neural artificial) (Kingma et al, 2015).

Corrigiendo los estimadores, se tiene lo siguiente:

$$\widehat{m}_t = \frac{m_t}{1 - \beta_1^t} \quad 9.7$$

$$\widehat{v}_t = \frac{v_t}{1 - \beta_2^t} \quad 9.8$$

Donde:

t : Iteración actual

Teniendo m_t y v_t ya corregidos, solo queda especificar el algoritmo por el cual se actualizan los pesos de la red neural artificial utilizando el optimizador Adam:

$$w_{ij}(new) = w_{ij}(old) - \eta \frac{\widehat{m}_t}{\sqrt{\widehat{v}_t + \epsilon}} \quad 9.9$$

Donde:

ϵ : Hiper-parámetro suavizante que previene que el denominador se convierta en cero.

Nótese la similitud de la ecuación 9.9 con la ecuación 9.1. Ambas tienen por objetivo primordial la actualización de los pesos de manera que la función de pérdida sea mínima; la diferencia es que Adam lo consigue de forma más eficiente al tener un factor de aprendizaje adaptivo. Para ver el desempeño de Adam con respecto a otros optimizadores se puede recurrir a la figura 17:

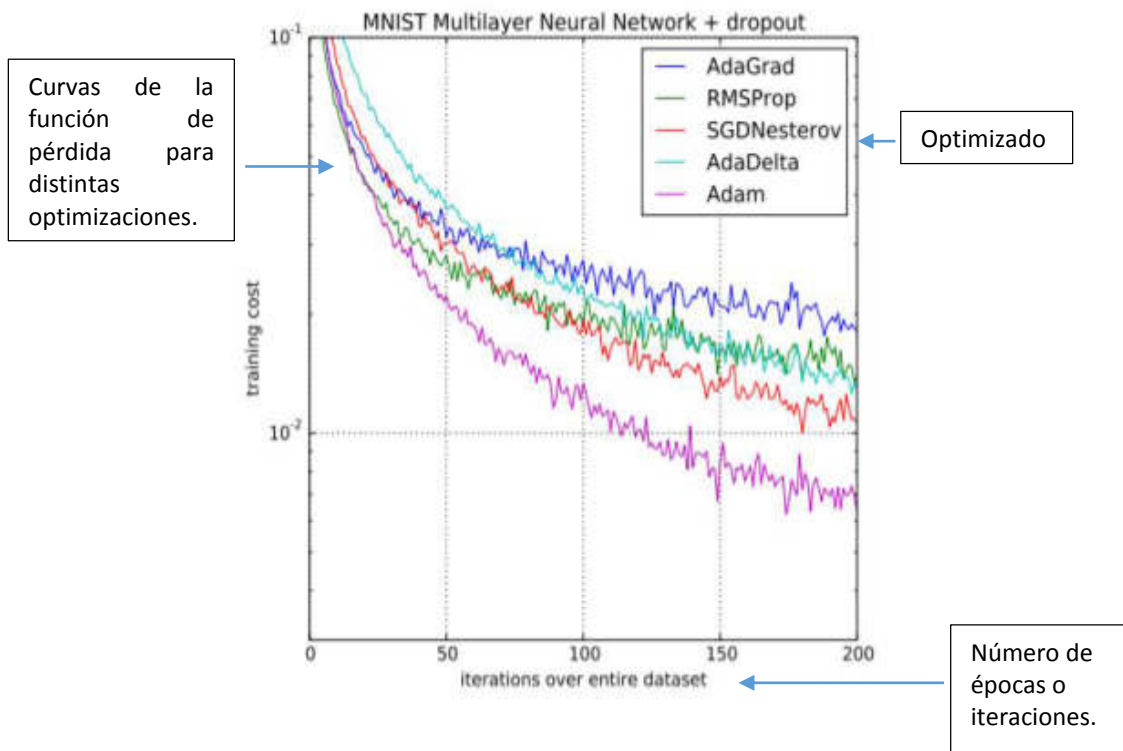


Figura (17). Entrenamiento de una red neural multicapa con distintos optimizadores en donde se aprecia a la función de pérdida convergiendo a cero y en donde se evidencia también el desempeño de Adam (morado) con respecto a los demás métodos de optimización

Fuente: *Adam: A Method for Stochastic Optimization*. Diederik P. Kingma y Jimmy Lei Ba. 2015. p.7

2.3.4.3 Redes Neuronales Convolucionales

Las redes neuronales convolucionales (RNC para abreviar) son una variante de las redes neurales artificiales convencionales, este tipo de red neural utiliza una arquitectura especialmente adaptada para la clasificación de imágenes y son en la actualidad utilizadas en la mayoría de las aplicaciones que responden a problemas del campo de la visión (Nielsen, 2015). La arquitectura de una red convolucional se puede apreciar en la figura 18.

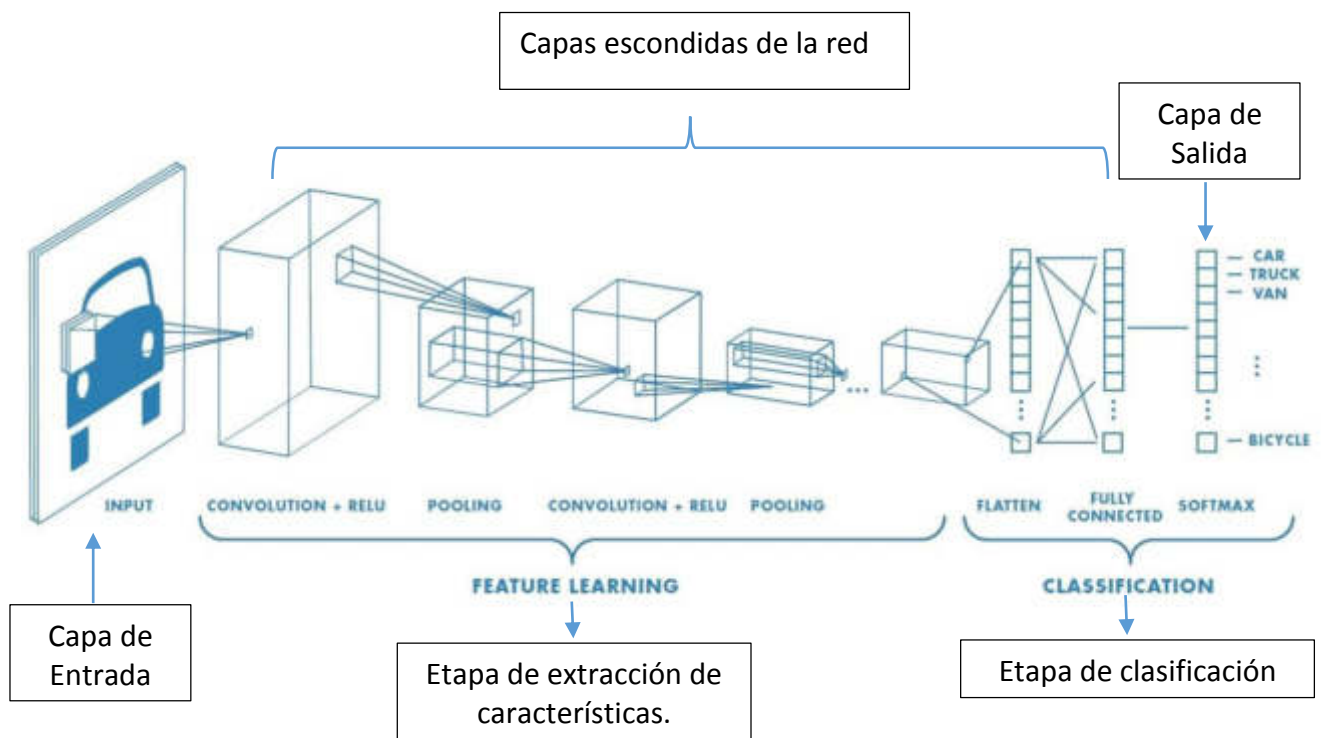


Figura (18). Arquitectura de una red neuronal convolucional (RNC)

Fuente: <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>

Gracias a la figura 18 se puede observar que la arquitectura de una RNC; si bien diferente, no dista demasiado de la arquitectura de una red neural convencional puesto que ambas poseen capas de entrada, capas escondidas y capa de salida; ambas también usan funciones de activación y ambas poseen una función que denota el error generado por la red.

La diferencia más llamativa radica en que las entradas para una red neural común eran representadas como un arreglo de una columna vertical de neuronas, mientras que en una RNC las entradas se representan como matrices de neuronas de tamaño $M \times N$, y los valores de dichas neuronas coinciden con los valores de intensidad de pixel de la imagen utilizada como entrada (Nielsen, 2015).

Otras diferencias (no menos importantes) que tienen las redes neurales convolutivas respecto a las redes neurales totalmente conectadas son el uso de la convolución para obtener las unidades de las capas ocultas, el recorrido un filtro alrededor de la imagen o mapa de características denominado *Stride*, el uso de *Padding* que evita la pérdida de información si es que esta se encuentra en el borde la imagen y compartición de parámetros (pesos de los filtros) por parte de las unidades de las capas ocultas (Albawi, Mohammed y Al-zawi, 2017).

Le Cunn, Haffner, Botton y Bengio (1998) describen las ventajas que tiene una red neural convolucional sobre una red neural totalmente conectada, estas son:

- Parámetros compartidos: Ya que todas las unidades de una misma capa dependen de las activaciones del filtro convolucional que se empleó para obtener dicha capa
- Robustez a la distorsión geométrica: Las redes neuronales comunes totalmente conectadas no presentan invariancia incorporada con respecto a cambios de traslación, escala o distorsión de las entradas; las RCN por su parte resuelven este problema al forzar la réplica de la configuración de pesos a través del espacio
- Relevancia de la topología de la entrada: Debido a que las RCN reciben las entradas en forma de matriz (las imágenes son justamente arreglos de dos dimensiones) y fuerzan la extracción de características locales al restringir los campos receptivos de las unidades de las capas adyacentes a ser locales también

Para apreciar cómo se distribuyen las neuronas de entrada en una RNC se puede recurrir a la figura 19.

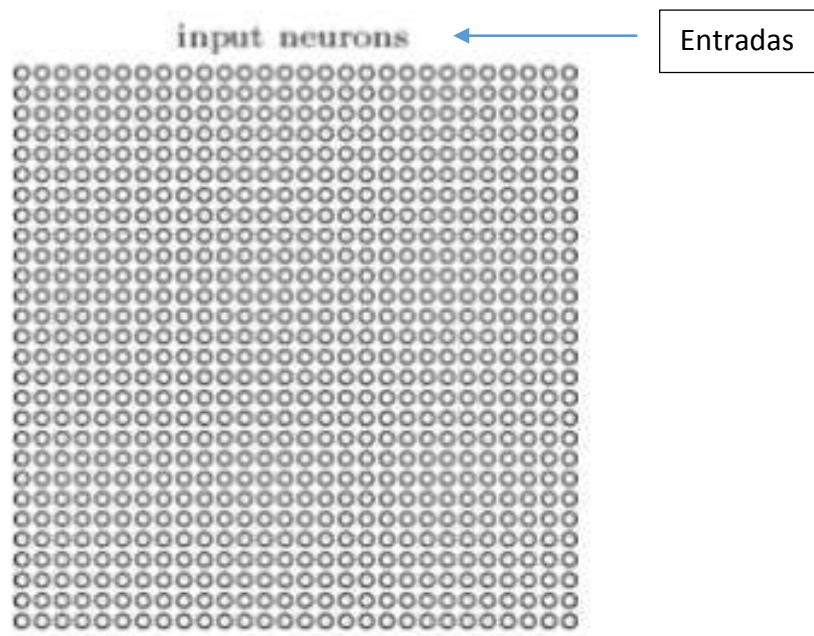


Figura (19) Entradas (*input neurons*) de una RNC arregladas en forma de matriz

Fuente: *Neural Networks and Deep Learning*. Michael Nielsen. *Determination Press*. 2015. p.170

Como es usual, estas neuronas se conectan a las neuronas de las capas escondidas, pero a diferencia de las neuronas de una red neural normal donde todas las neuronas de la capa previa estaban conectadas a las neuronas de la capa actual; en una RNC los píxeles no se conectan a todas las neuronas de la siguiente capa sino que tienen conexiones en áreas pequeñas y localizadas; se forma entonces una región en la imagen que viene a convertirse en el campo local receptor de una neurona de la capa escondida contigua a la capa de entradas (Nielsen, 2015). Este proceso de conexión localizada se aprecia en la figura 20.

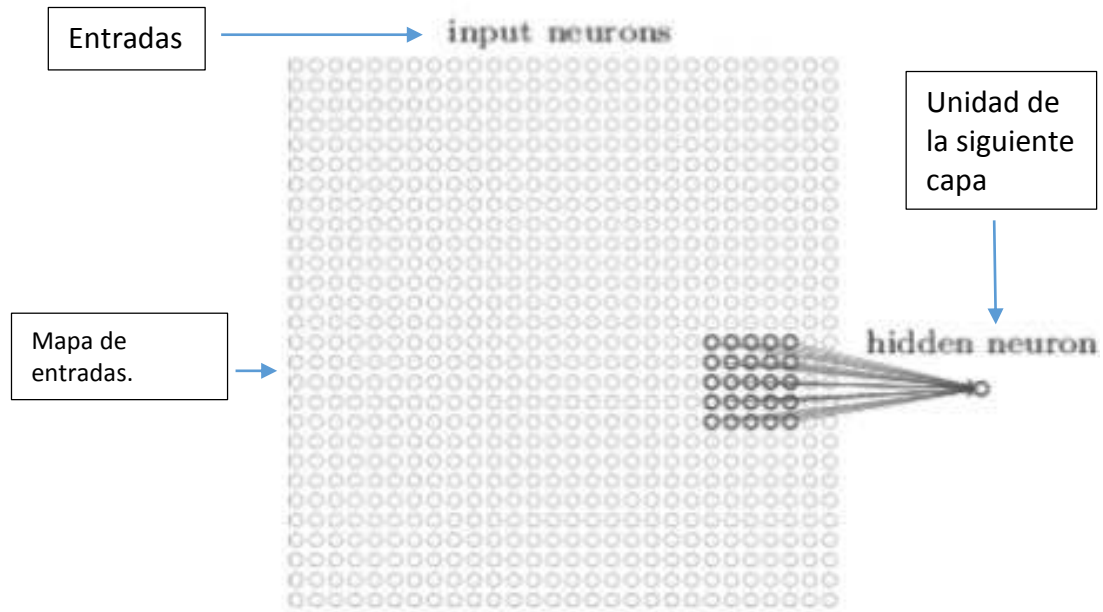


Figura (20). Campo local receptivo de una neurona en la capa oculta (*hidden neuron*)

Fuente: *Neural Networks and Deep Learning*. Michael Nielsen. *Determination Press*. 2015. p.170

Esto quiere decir que cada neurona de la capa escondida aprende una región específica de la imagen en donde que cada conexión aprende un peso y un *bias*; en otras palabras, en una RNC todas las neuronas de una capa escondida aprenden el mismo peso (y también *bias*) debido a que el campo de recepción local está asociado a una ventana móvil conocida como *Kernel* (o filtro) el cual recorre a toda la imagen mediante pasos o *strides* (este es definido por el diseñador de la red, normalmente el *stride* adopta el valor de 1), este *Kernel* también es una matriz (aunque mucho más pequeña que la imagen de entrada) y los valores de cada posición de esta matriz son los pesos compartidos por las neuronas de la siguiente capa (Nielsen, 2015).

Para tener una idea gráfica de este desplazamiento del Kernel se puede recurrir a la figura 21.

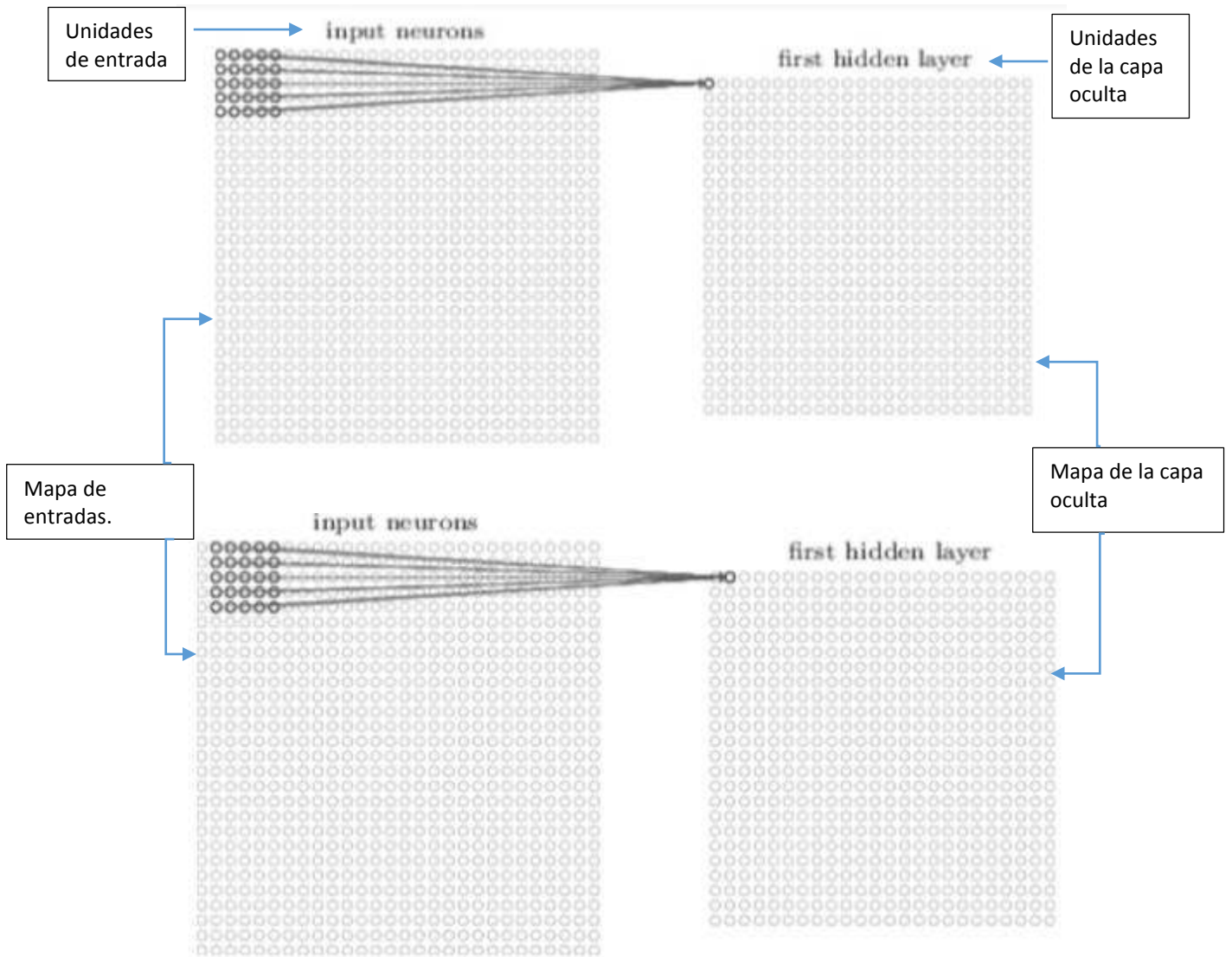


Figura (21). Desplazamiento del *Kernel* o filtro con *stride* = 1 (avanza un píxel a la vez hacia la derecha y luego hacia abajo hasta recorrer toda la imagen)

Fuente: *Neural Networks and Deep Learning*. Michael Nielsen. *Determination Press*. 2015. p.171

La manera en que el *kernel* interactúa con la región que ha de ser detectada por una neurona de la siguiente capa es a través de una convolución discreta (Goodfellow et al., 2017).

En una RNC la convolución discreta se lleva a cabo entre matrices (la matriz correspondiente a la imagen o *feature map* y la matriz correspondiente al *kernel* o *feature detector*), por lo que se tiene lo siguiente:

$$S(i, j) = (X_{2d} * F_{2d})(i, j) = \sum_{m_d} \sum_{n_d} X_{2d}(m_d, n_d) F_{2d}(i - m_d, j - n_d) \quad (10.0)$$

Donde:

S : Función de convolución de X_{2d} y F_{2d}

X_{2d} : Matriz bidimensional que representa a la imagen

F_{2d} : Matriz bidimensional que representa al filtro

i, j : Posiciones

La convolución es conmutativa por lo que 10.0 se reescribe como:

$$S(i, j) = (X_{2d} * F_{2d})(i, j) = \sum_{m_d} \sum_{n_d} X_{2d}(i - m_d, j - n_d) F_{2d}(m_d, n_d) \quad (10.1)$$

La ecuación 10.3 es más sencilla de implementar en una librería de *machine learning*, ello tiene como consecuencia que el *kernel* se “voltee” o rote (Goodfellow et al., 2017), la ecuación 10.3 se puede entender mejor si se considera el siguiente *feature map* (bloque verde) y el *kernel* (bloque azul):

Mapa de Entradas

x_{11}	x_{12}	x_{13}	x_{14}
x_{21}	x_{22}	x_{23}	x_{24}
x_{31}	x_{32}	x_{33}	x_{34}

KERNEL

F_{11}	F_{12}
F_{21}	F_{22}

De acuerdo con la ecuación 10.1, el kernel ha de ser rotado antes de realizar la convolución. Entonces el nuevo bloque azul queda como:

**KERNEL
ROTADO**

F_{22}	F_{21}
F_{12}	F_{11}

Supóngase ahora que este nuevo kernel recorrerá la imagen de entrada de 1 píxel en 1 píxel (*stride=1*), el mapa de salida queda entonces definido como se puede apreciar a continuación:

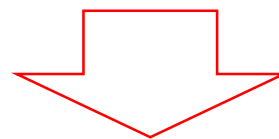
Mapa de Entradas

x_{11}	x_{12}	x_{13}	x_{14}
x_{21}	x_{22}	x_{23}	x_{24}
x_{31}	x_{32}	x_{33}	x_{34}

**KERNEL
ROTADO**

F_{22}	F_{21}
F_{12}	F_{11}

*



φ indica que se está aplicando una función de activación.

$\varphi(x_{11}F_{22} + x_{12}F_{21} + x_{21}F_{12} + x_{22}F_{11})$	$\varphi(x_{12}F_{22} + x_{13}F_{21} + x_{22}F_{12} + x_{23}F_{11})$	$\varphi(x_{13}F_{22} + x_{14}F_{21} + x_{23}F_{12} + x_{24}F_{11})$
$\varphi(x_{21}F_{22} + x_{22}F_{21} + x_{31}F_{12} + x_{32}F_{11})$	$\varphi(x_{22}F_{22} + x_{23}F_{21} + x_{32}F_{12} + x_{33}F_{11})$	$\varphi(x_{23}F_{22} + x_{24}F_{21} + x_{33}F_{12} + x_{34}F_{11})$

MAPA DE SALIDA

Nótese que las dimensiones del mapa de características de salida son menores a las dimensiones del mapa de entrada, esto se debe a que si se tiene una imagen de dimensiones $M \times N$ y un kernel de dimensiones $k_1 \times k_2$, el mapa de salida tiene las siguientes dimensiones:

$$\#Filas = (M - k_1 + 1)/s \quad (11.0)$$

$$\#Columnas = (N - k_2 + 1)/s \quad (11.1)$$

Donde:

s : *Stride* (o pasos) del *kernel* o filtro $k_1 \times k_2$ para recorrer la matriz $M \times N$

Otra cosa que se debe resaltar es que todos los elementos (o neuronas) del mapa de salida comparten los mismos pesos provenientes del *kernel*, esto quiere decir que las neuronas del mapa de salida detectan el mismo patrón o característica solo que en distintas posiciones de la imagen de entrada (Nielsen, 2015); sin embargo, una imagen puede contener múltiples características o patrones complejos a ser detectados y dado que un *kernel* solo ayuda a detectar un tipo de característica de la entrada, han de ser necesarios múltiples *kernel* que a su vez darán origen a múltiples mapas de salida cuando estos se convolucionen con la imagen de entrada.

Estos nuevos mapas de salida de la misma capa de convolución se van apilando uno sobre otro generando así un arreglo multidimensional llamado tensor (Goodfellow et al., 2017), la misma entrada puede ser un tensor si se considera una imagen RGB o HSV que tiene tres canales, y por ello tendría tres matrices apiladas una sobre otra. Esto se aprecia en la figura 22.

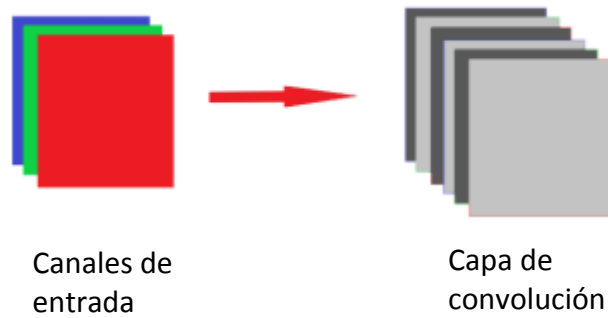


Figura (22). Imagen RGB de entrada que produce varios mapas de salida como consecuencia de su convolución con distintos *kernels*

Fuente: <https://www.kaggle.com/moghazy/guide-to-cnns-with-data-augmentation-keras>

Adicionalmente a una capa de convolución, se suele añadir una capa de reducción o *pooling* (ver figura 23); la función de estas capas es la de simplificar la información en la salida de una capa de convolución lo cual se logra preparando un mapa condensado de características (Nielsen, 2015).

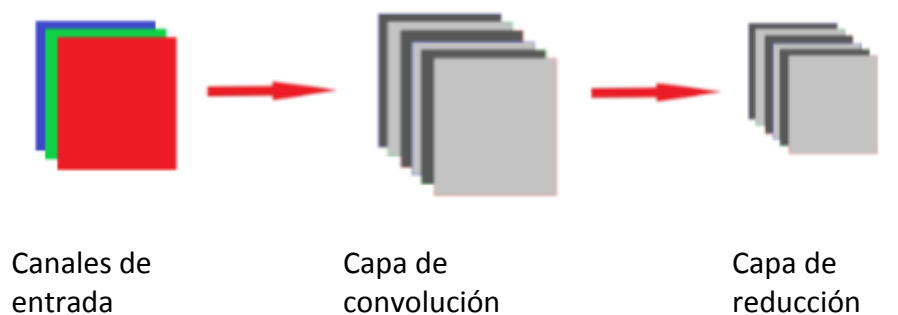


Figura (23). Capa de convolución seguida de una capa de reducción o *pooling*

Tomado de: <https://www.kaggle.com/moghazy/guide-to-cnns-with-data-augmentation-keras> , el 19 de Abril de 2019

Cada unidad de la capa de *pooling* cumple la tarea de resumir un área o región del mapa de la capa de convolución (Nielsen, 2015), dicha región es definida por el diseñador de la red. Esto se puede apreciar en la figura 24.

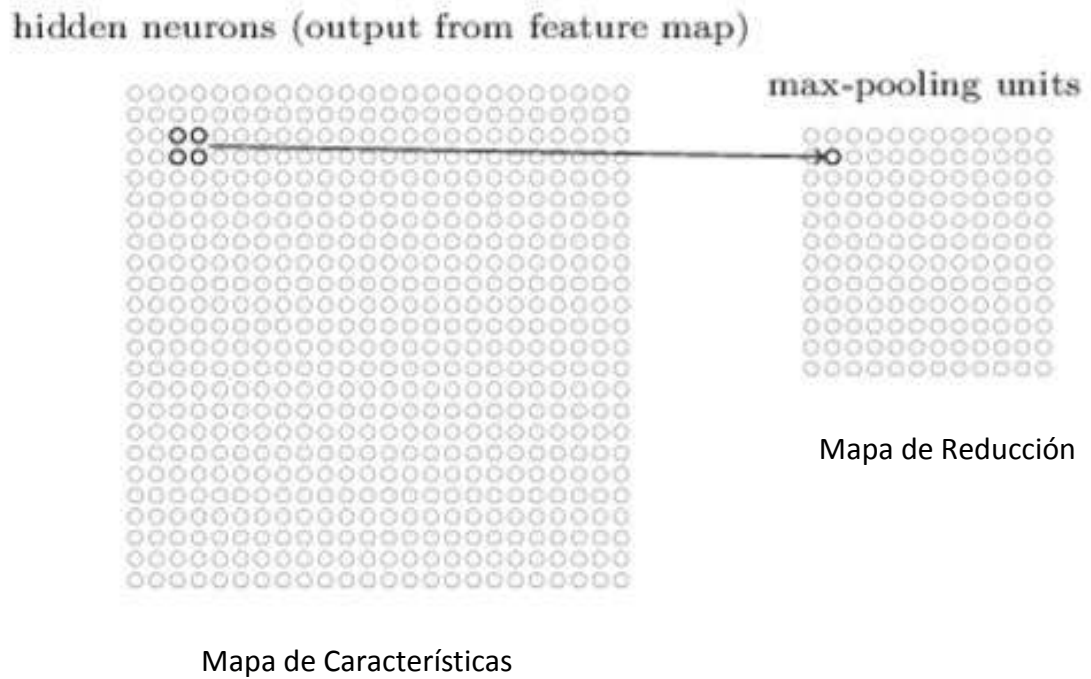


Figura (24). La unidad (2,2) de la capa de *pooling* (derecha) resume un área 2x2 del mapa de características (izquierda).

Fuente: Neural Networks and Deep Learning. Michael Nielsen. *Determination Press*. 2015. p.174

La forma en que una unidad de la capa de reducción puede resumir un área determinada del mapa de características de la capa convolutiva puede variar, una de las formas más empleadas es la de *max-pool* (es la que se usará en el presente trabajo), esta función simplemente toma el valor de la activación más fuerte de una región de dimensiones $k_{p1} \times k_{p2}$, ello se puede evidenciar en la figura 25.

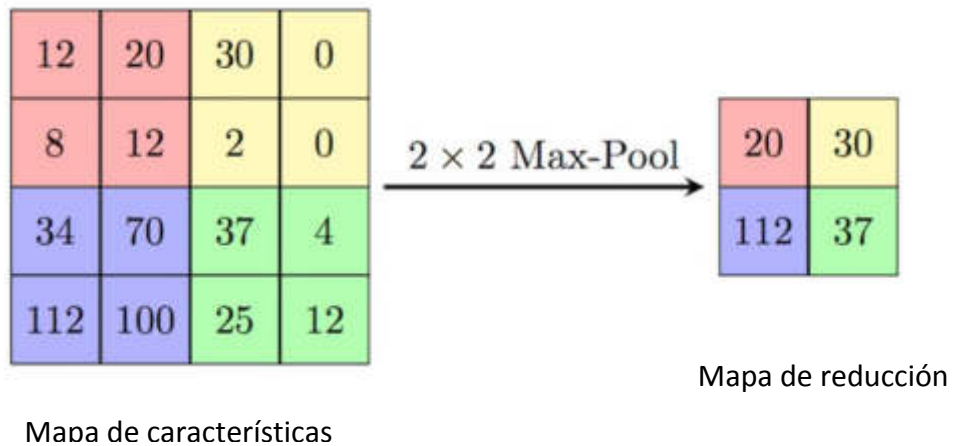


Figura (25). Función *Max-Pool* que toma la activación más grande en un área de 2x2, resumiendo así la matriz de la izquierda en la matriz de la derecha

Fuente: https://computersciencewiki.org/index.php/Max-pooling/_/_Pooling

Dependiendo de la complejidad de la imagen o los patrones a clasificar la arquitectura de una RNC puede incluir varias sucesiones de capas de convolución y de *pooling* de manera que se garantice una correcta extracción de características.

Una vez que se cuenta con los mapas de características agrupados (*pooled feature maps*) y adecuados, se debería ingresar la información a una *fully connected layer* (capa totalmente conectada); que es parte de una red neural convencional. El problema es que la entrada de una red neural común no puede ser un tensor (recordar que los mapas de características son justamente tensores); es por ello, que el tensor pasa por un proceso llamado *flattening* el cual lo “aplana” de manera que se convierta en una arreglo de una sola columna tal y como se aprecia en la figura 26.

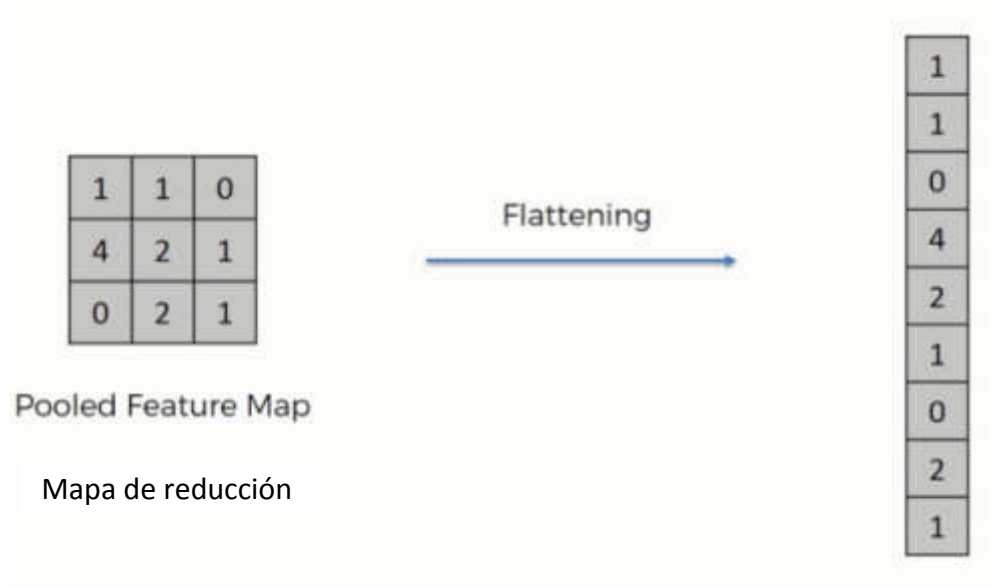


Figura (26). Un arreglo matricial es convertido a un arreglo de una sola columna gracias a *Flattening*

Fuente: <https://www.superdatascience.com/blogs/convolutional-neural-networks-cnn-step-3-flattening>

La salida de *Flatten* es entonces la entrada de una red neural convencional totalmente conectada, a partir de este punto la red se comporta exactamente como la red neural descrita en los apartados 2.3.4.1 y 2.3.4.2.

2.3.4.4 Aprendizaje en una Red Neuronal Convolutiva

Las RNC (Redes Neuronales Convolutivas), de forma similar a una red totalmente conectada, ejecutan el proceso de feed-forward de manera que también generan una función de coste o error (para el presente trabajo recordar que la función de error es cross entropy). Para que esta función de error se reduzca, las RNC emplean el back-propagation tal y como lo hacen las redes neurales clásicas (descrito en el apartado 2.3.4.2) de manera que los pesos de la capa totalmente conectada y los pesos de los filtros cambien para que la función de coste converja a cero, logrando así el aprendizaje de la red.

Las ecuaciones descritas en los apartados 2.3.4.1 y 2.3.4.2 se pueden aplicar entonces a una RNC; para entender cómo es que todas estas ecuaciones se aplican y relacionan en el proceso de *back-propagation* considérese la red neural convolutiva de la figura 27:

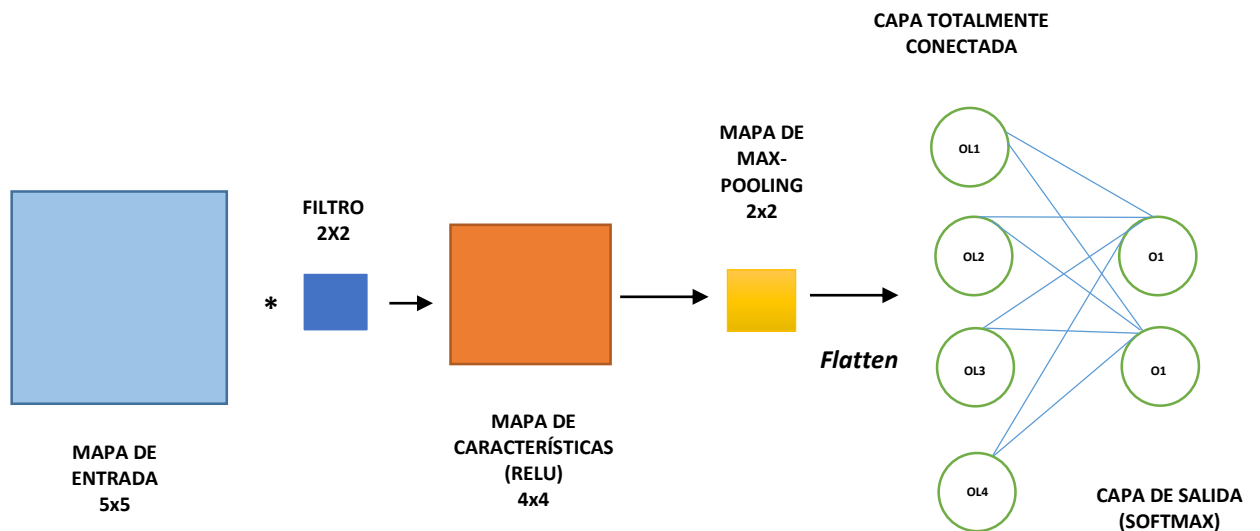


Figura (27). Arquitectura de una RNC simple, compuesta por un mapa de entrada (bloque celeste), un filtro (bloque azul), un *feature map* (bloque naranja), un mapa de reducción (bloque amarillo), 4 nodos de la capa totalmente conectada y dos nodos de salida (circunferencias)

Elaboración: los autores

Obsérvese que el mapa de características está afectado por la función de activación ReLU (ecuación 5.0) y que las unidades de salida están siendo afectadas por la función de activación Softmax (ecuación 6.0). Además, considérese que la red utiliza como función de error a la función de entropía cruzada (ecuación 7.1). También, se considera un *stride* =1. Supóngase que el mapa de entrada de la figura 27 está definido como:

0.3 x_{11}	0.2 x_{12}	0.25 x_{13}	0.35 x_{14}	0.1 x_{15}
0.15 x_{21}	0.5 x_{22}	0.6 x_{23}	0.65 x_{24}	0.7 x_{25}
0.75 x_{31}	0.8 x_{32}	0.85 x_{33}	0.4 x_{34}	0.45 x_{35}
0.8 x_{41}	0.9 x_{42}	0.95 x_{43}	0.3 x_{44}	0.35 x_{45}
0.4 x_{51}	0.45 x_{52}	0.1 x_{53}	0.15 x_{54}	0.7 x_{55}

Donde los valores de los píxeles de esta matriz de 5x5 están normalizados (entre 0 y 1). Los valores del filtro 2x2 que convolucionamos con la matriz de entrada conectada se inicializan de manera aleatoria siguiendo la inicialización descrita por Kaiming He, Xiangyu Zhang, Shaoqing Ren y Jian Sun en su paper titulado: “Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification” del año 2015.

En este *paper*, He et al. (2015) propone un método para inicializar los pesos de una red que utiliza funciones de activación ReLU y PreLU. Esa inicialización se conoce como inicialización He, en donde los valores se toman aleatoriamente dentro de una distribución uniforme de la forma $[-limit, limit]$, entonces:

$$limit = \sqrt{\frac{6}{N_{in}}} \dots\dots\dots (12.0)$$

Donde:

N_{in} : Número de unidades de entradas al tensor de pesos

Para el caso del filtro de 2x2 de la figura 23, la ecuación 12.0 queda como:

$$limit = \sqrt{\frac{6}{25}} = 0.4899$$

De manera que los valores de este filtro pueden estar entre -0.4899 y 0.4899, teniendo ello en cuenta se eligen muestras de esta distribución uniforme de la siguiente manera:

$$F_{11} = 0.15$$

$$F_{12} = 0.2$$

$$F_{21} = 0.25$$

$$F_{22} = 0.3$$

La matriz del filtro 2x2 queda entonces como:

0.15 F_{11}	0.2 F_{12}
0.25 F_{21}	0.3 F_{22}

Teniendo definidas las matrices de entrada y del filtro, se pueden calcular los valores del mapa de características de 4x4; primero se definen las *net-inputs* del mapa de características o *feature map* con la ecuación 10.2. Recordar, además que el filtro recorre a la imagen con un *stride* o paso de 1:

- $$netOC_{11} = x_{11}F_{11} + x_{12}F_{12} + x_{21}F_{21} + x_{22}F_{22} = (0.3)(0.15) + (0.2)(0.2) + (0.15)(0.25) + (0.5)(0.3) = 0.2725$$
- $$netOC_{12} = x_{12}F_{11} + x_{13}F_{12} + x_{22}F_{21} + X_{23}F_{22} = (0.2)(0.15) + (0.25)(0.2) + (0.5)(0.25) + (0.6)(0.3) = 0.385$$

- $netOC_{13} = x_{13}F_{11} + x_{14}F_{12} + x_{23}F_{21} + x_{24}F_{22} = (0.25)(0.15) + (0.35)(0.2) + (0.6)(0.25) + (0.65)(0.3) = 0.4525$
- $netOC_{14} = x_{14}F_{11} + x_{15}F_{12} + x_{24}F_{21} + x_{25}F_{22} = (0.35)(0.15) + (0.1)(0.2) + (0.65)(0.25) + (0.7)(0.3) = 0.445$
- $netOC_{21} = x_{21}F_{11} + x_{22}F_{12} + x_{31}F_{21} + x_{32}F_{22} = (0.15)(0.15) + (0.5)(0.2) + (0.75)(0.25) + (0.8)(0.3) = 0.55$
- $netOC_{22} = x_{22}F_{11} + x_{23}F_{12} + x_{32}F_{21} + x_{33}F_{22} = (0.5)(0.15) + (0.6)(0.2) + (0.8)(0.25) + (0.85)(0.3) = 0.65$
- $netOC_{23} = x_{23}F_{11} + x_{24}F_{12} + x_{33}F_{21} + x_{34}F_{22} = (0.6)(0.15) + (0.65)(0.2) + (0.85)(0.25) + (0.4)(0.3) = 0.5525$
- $netOC_{24} = x_{24}F_{11} + x_{25}F_{12} + x_{34}F_{21} + x_{35}F_{22} = (0.65)(0.15) + (0.7)(0.2) + (0.4)(0.25) + (0.45)(0.3) = 0.4725$
- $netOC_{31} = x_{31}F_{11} + x_{32}F_{12} + x_{41}F_{21} + x_{42}F_{22} = (0.75)(0.15) + (0.8)(0.2) + (0.8)(0.25) + (0.9)(0.3) = 0.7425$
- $netOC_{32} = x_{32}F_{11} + x_{33}F_{12} + x_{42}F_{21} + x_{43}F_{22} = (0.8)(0.15) + (0.85)(0.2) + (0.9)(0.25) + (0.95)(0.3) = 0.8$
- $netOC_{33} = x_{33}F_{11} + x_{34}F_{12} + x_{43}F_{21} + x_{44}F_{22} = (0.85)(0.15) + (0.4)(0.2) + (0.95)(0.25) + (0.3)(0.3) = 0.535$
- $netOC_{34} = x_{34}F_{11} + x_{35}F_{12} + x_{44}F_{21} + x_{45}F_{22} = (0.4)(0.15) + (0.45)(0.2) + (0.3)(0.25) + (0.35)(0.3) = 0.33$
- $netOC_{41} = x_{41}F_{11} + x_{42}F_{12} + x_{51}F_{21} + x_{52}F_{22} = (0.8)(0.15) + (0.9)(0.2) + (0.4)(0.25) + (0.45)(0.3) = 0.535$

- $netOC_{42} = x_{42}F_{11} + x_{43}F_{12} + x_{52}F_{21} + x_{53}F_{22} = (0.9)(0.15) + (0.95)(0.2) + (0.45)(0.25) + (0.1)(0.3) = 0.4675$
- $netOC_{43} = x_{43}F_{11} + x_{44}F_{12} + x_{53}F_{21} + x_{54}F_{22} = (0.95)(0.15) + (0.3)(0.2) + (0.1)(0.25) + (0.15)(0.3) = 0.2725$
- $netOC_{44} = x_{44}F_{11} + x_{45}F_{12} + x_{54}F_{21} + x_{55}F_{22} = (0.3)(0.15) + (0.35)(0.2) + (0.15)(0.25) + (0.7)(0.3) = 0.3625$

Una vez calculadas las net-inputs del mapa de características, se les aplica la función de activación ReLU para obtener así las unidades del mapa de características de 4x4 (bloque naranja de la figura 23), para ello se emplea la ecuación 5.0:

- $OC_{11} = \max(0, netOC_{11}) = \max(0, 0.2725) = 0.2725$
- $OC_{12} = \max(0, netOC_{12}) = \max(0, 0.385) = 0.385$
- $OC_{13} = \max(0, netOC_{13}) = \max(0, 0.4525) = 0.4525$
- $OC_{14} = \max(0, netOC_{14}) = \max(0, 0.445) = 0.445$
- $OC_{21} = \max(0, netOC_{21}) = \max(0, 0.55) = 0.55$
- $OC_{22} = \max(0, netOC_{22}) = \max(0, 0.65) = 0.65$
- $OC_{23} = \max(0, netOC_{23}) = \max(0, 0.5525) = 0.5525$
- $OC_{24} = \max(0, netOC_{24}) = \max(0, 0.4725) = 0.4725$
- $OC_{31} = \max(0, netOC_{31}) = \max(0, 0.7425) = 0.7425$
- $OC_{32} = \max(0, netOC_{32}) = \max(0, 0.8) = 0.8$
- $OC_{33} = \max(0, netOC_{33}) = \max(0, 0.535) = 0.535$
- $OC_{34} = \max(0, netOC_{34}) = \max(0, 0.33) = 0.33$
- $OC_{41} = \max(0, netOC_{41}) = \max(0, 0.535) = 0.535$
- $OC_{42} = \max(0, netOC_{42}) = \max(0, 0.4675) = 0.4675$
- $OC_{43} = \max(0, netOC_{43}) = \max(0, 0.2725) = 0.2725$
- $OC_{44} = \max(0, netOC_{44}) = \max(0, 0.3625) = 0.3625$

La matriz del *feature map* de la figura 23 queda entonces definida como:

0.2725 OC_{11}	0.385 OC_{12}	0.4525 OC_{13}	0.445 OC_{14}
0.55 OC_{21}	0.65 OC_{22}	0.5525 OC_{23}	0.4725 OC_{24}
0.7425 OC_{31}	0.8 OC_{32}	0.535 OC_{33}	0.33 OC_{34}
0.535 OC_{41}	0.4675 OC_{42}	0.2725 OC_{43}	0.3625 OC_{44}

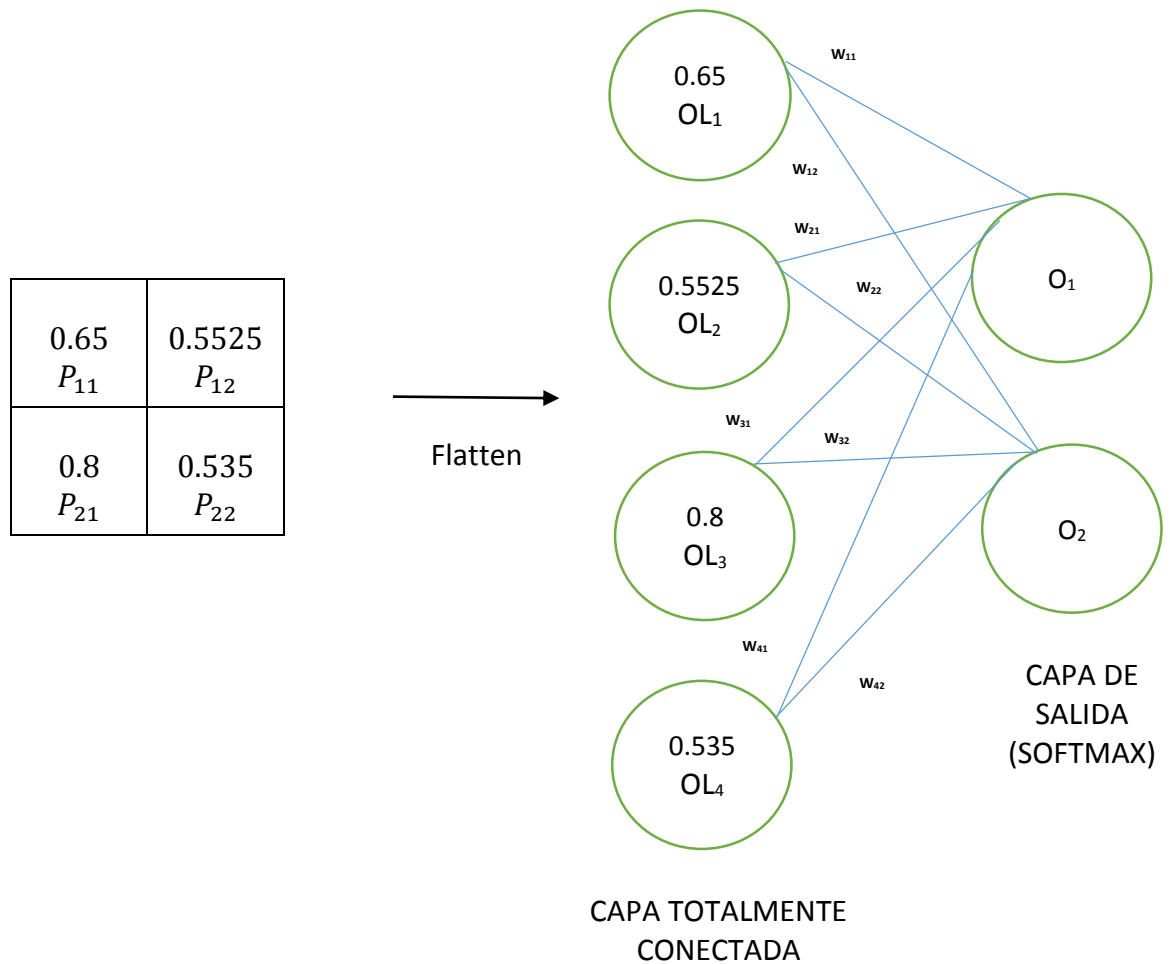
Esta matriz es ahora reducida por la función *Max-Pooling* en una ventana de 2x2, de la misma forma que ocurre en la figura 21, ahora se calculan las unidades del mapa de Max-Pool (bloque amarillo de la figura 23):

- $P_{11} = \max(OC_{11}, OC_{12}, OC_{21}, OC_{22}) = \max(0.2725, 0.385, 0.55, 0.65) = 0.65$
- $P_{12} = \max(OC_{13}, OC_{14}, OC_{23}, OC_{24}) = \max(0.4525, 0.445, 0.5525, 0.4725) = 0.5525$
- $P_{21} = \max(OC_{31}, OC_{32}, OC_{41}, OC_{42}) = \max(0.7425, 0.8, 0.535, 0.4675) = 0.8$
- $P_{22} = \max(OC_{33}, OC_{34}, OC_{43}, OC_{44}) = \max(0.535, 0.33, 0.2725, 0.3625) = 0.535$

El mapa de reducción de 2x2 de la figura 23 queda definido como:

0.65 P_{11}	0.5525 P_{12}
0.8 P_{21}	0.535 P_{22}

Este mapa de reducción es convertido ahora en un arreglo de una columna (gracias a *Flatten*) para que sea la entrada de la capa totalmente conectada:



Nótese que los nodos de la capa totalmente conectada se enlazan con los nodos de la capa de salida a través de las conexiones sinápticas w_{11} , w_{12} , w_{21} , w_{22} , w_{31} , w_{32} , w_{41} y w_{42} . Estos pesos se inicializan de la misma forma en que se inicializaron los pesos del filtro (con la inicialización He), entonces se procede a calcular el rango del cual se tomarán las muestras para los pesos:

$$limit_1 = \sqrt{\frac{6}{4}} = 1.2247$$

Los pesos se eligen de una distribución uniforme que está contenida en los rangos de -1.2247 a 1.2247; teniendo esto en cuenta, los pesos de la capa totalmente conectada se definen como:

- $w_{11} = 0.4$
- $w_{12} = 0.45$
- $w_{21} = 0.5$
- $w_{22} = 0.55$
- $w_{31} = 0.6$
- $w_{32} = 0.65$
- $w_{41} = 0.7$
- $w_{42} = 0.75$

Ahora que se tienen los pesos definidos se pueden calcular las entradas a nodos de salida O1 y O2 gracias a la ecuación 4.0:

- $netO_1 = OL_1w_{11} + OL_2w_{21} + OL_3w_{31} + OL_4w_{41} = (0.65)(0.4) + (0.5525)(0.5) + (0.8)(0.6) + (0.535)(0.7) = 1.39075$
- $netO_2 = OL_1w_{12} + OL_2w_{22} + OL_3w_{32} + OL_4w_{42} = (0.65)(0.45) + (0.5525)(0.55) + (0.8)(0.65) + (0.535)(0.75) = 1.51763$

Con los net-input obtenidos, se les aplica la función de activación Softmax para obtener así la predicción de la red neural:

- $O_1 = \frac{e^{netO_1}}{e^{netO_1} + e^{netO_2}} = \frac{e^{1.39075}}{e^{1.39075} + e^{1.51763}} = 0.46832$
- $O_2 = \frac{e^{netO_2}}{e^{netO_1} + e^{netO_2}} = \frac{e^{1.51763}}{e^{1.39075} + e^{1.51763}} = 0.53168$

Nótese que $O_1 + O_2 = 1$, con las predicciones de la red calculadas se procede a calcular el error de la red, para ello se aplica la ecuación 7.1 que corresponde al cálculo del error de Cross Entropy, supóngase un *target* de [0,1] entonces:

- $E = \sum_{i=1}^2 Y_{Ti} \ln(O_i)$
- $E = -0 \ln(O_1) - 1 \ln(O_2) = -\ln(0.53168) = 0.63171$

Ahora que se calculó el error de la red se procede a aplicar el *back-propagation* para actualizar los pesos de la red y así reducir la función de coste de manera que converja a cero, para ello primero se requiere calcular la gradiente del error con respecto a los pesos de la capa totalmente conectada (se emplea la ecuación 8.0 que corresponde a la regla de la cadena). Entonces se requiere saber las siguientes derivadas parciales:

- $\frac{\partial E}{\partial w_{11}} = \frac{\partial E}{\partial O_1} \frac{\partial O_1}{\partial \text{net}O_1} \frac{\partial \text{net}O_1}{\partial w_{11}} + \frac{\partial E}{\partial O_2} \frac{\partial O_2}{\partial \text{net}O_1} \frac{\partial \text{net}O_1}{\partial w_{11}}$
- $\frac{\partial E}{\partial w_{12}} = \frac{\partial E}{\partial O_1} \frac{\partial O_1}{\partial \text{net}O_2} \frac{\partial \text{net}O_2}{\partial w_{12}} + \frac{\partial E}{\partial O_2} \frac{\partial O_2}{\partial \text{net}O_2} \frac{\partial \text{net}O_2}{\partial w_{12}}$
- $\frac{\partial E}{\partial w_{21}} = \frac{\partial E}{\partial O_1} \frac{\partial O_1}{\partial \text{net}O_1} \frac{\partial \text{net}O_1}{\partial w_{21}} + \frac{\partial E}{\partial O_2} \frac{\partial O_2}{\partial \text{net}O_1} \frac{\partial \text{net}O_1}{\partial w_{21}}$
- $\frac{\partial E}{\partial w_{22}} = \frac{\partial E}{\partial O_1} \frac{\partial O_1}{\partial \text{net}O_2} \frac{\partial \text{net}O_2}{\partial w_{22}} + \frac{\partial E}{\partial O_2} \frac{\partial O_2}{\partial \text{net}O_2} \frac{\partial \text{net}O_2}{\partial w_{22}}$
- $\frac{\partial E}{\partial w_{31}} = \frac{\partial E}{\partial O_1} \frac{\partial O_1}{\partial \text{net}O_1} \frac{\partial \text{net}O_1}{\partial w_{31}} + \frac{\partial E}{\partial O_2} \frac{\partial O_2}{\partial \text{net}O_1} \frac{\partial \text{net}O_1}{\partial w_{31}}$
- $\frac{\partial E}{\partial w_{32}} = \frac{\partial E}{\partial O_1} \frac{\partial O_1}{\partial \text{net}O_2} \frac{\partial \text{net}O_2}{\partial w_{32}} + \frac{\partial E}{\partial O_2} \frac{\partial O_2}{\partial \text{net}O_2} \frac{\partial \text{net}O_2}{\partial w_{32}}$
- $\frac{\partial E}{\partial w_{41}} = \frac{\partial E}{\partial O_1} \frac{\partial O_1}{\partial \text{net}O_1} \frac{\partial \text{net}O_1}{\partial w_{41}} + \frac{\partial E}{\partial O_2} \frac{\partial O_2}{\partial \text{net}O_1} \frac{\partial \text{net}O_1}{\partial w_{41}}$
- $\frac{\partial E}{\partial w_{42}} = \frac{\partial E}{\partial O_1} \frac{\partial O_1}{\partial \text{net}O_2} \frac{\partial \text{net}O_2}{\partial w_{42}} + \frac{\partial E}{\partial O_2} \frac{\partial O_2}{\partial \text{net}O_2} \frac{\partial \text{net}O_2}{\partial w_{42}}$

Se comienza calculando $\frac{\partial E}{\partial O_1}$ y $\frac{\partial E}{\partial O_2}$, recordar que la función de error para este ejemplo está definida como:

- $E = -Y_{T1} \ln(O_1) - Y_{T2} \ln(O_2)$

Con lo que $\frac{\partial E}{\partial O_1}$ sería:

- $\frac{\partial E}{\partial O_1} = -Y_{T1} \left(\frac{1}{O_1} \right) = \frac{-Y_{t1}}{O_1}$

Pero recordar que el target es $[0,1]$, entonces $Y_{T1} = 0$ por lo que $\frac{\partial E}{\partial O_1} = 0$; ahora

se calcula $\frac{\partial E}{\partial O_2}$:

- $\frac{\partial E}{\partial O_2} = -Y_{T2} \left(\frac{1}{O_2} \right) = \frac{-Y_{t2}}{O_2} = \frac{-1}{0.53168} = -1.88083$

Se continúa calculando $\frac{\partial O_1}{\partial netO_1}$, $\frac{\partial O_2}{\partial netO_2}$, $\frac{\partial O_1}{\partial netO_2}$ y $\frac{\partial O_2}{\partial netO_1}$:

- $\frac{\partial O_1}{\partial netO_1} = \frac{\partial \left(\frac{e^{netO_1}}{e^{netO_1} + e^{netO_2}} \right)}{\partial netO_1} = \frac{(e^{netO_1} + e^{netO_2})e^{netO_1} - (e^{netO_1})^2}{(e^{netO_1} + e^{netO_2})^2} =$
 $\frac{e^{netO_1}e^{netO_2}}{(e^{netO_1} + e^{netO_2})^2} = \frac{(4.01786)(4.5614)}{8.57926^2} = \frac{18.32701}{68.48638} = 0.2676$

- $\frac{\partial O_2}{\partial netO_2} = \frac{\partial \left(\frac{e^{netO_2}}{e^{netO_1} + e^{netO_2}} \right)}{\partial netO_2} = \frac{(e^{netO_1} + e^{netO_2})e^{netO_2} - (e^{netO_2})^2}{(e^{netO_1} + e^{netO_2})^2} =$
 $\frac{e^{netO_1}e^{netO_2}}{(e^{netO_1} + e^{netO_2})^2} = \frac{(4.01786)(4.5614)}{8.57926^2} = \frac{18.32701}{68.48638} = 0.2676$

- $\frac{\partial O_1}{\partial netO_2} = \frac{\partial \left(\frac{e^{netO_1}}{e^{netO_1} + e^{netO_2}} \right)}{\partial netO_2} = e^{netO_1} \frac{\partial \left(\frac{1}{e^{netO_1} + e^{netO_2}} \right)}{\partial netO_2} = \frac{-e^{netO_1}e^{netO_2}}{(e^{netO_1} + e^{netO_2})^2} =$
 $\frac{-(4.01786)(4.5614)}{8.57926^2} = \frac{-18.32701}{68.48638} = -0.2676$

- $\frac{\partial O_2}{\partial netO_1} = \frac{\partial \left(\frac{e^{netO_2}}{e^{netO_1} + e^{netO_2}} \right)}{\partial netO_1} = e^{netO_2} \frac{\partial \left(\frac{1}{e^{netO_1} + e^{netO_2}} \right)}{\partial netO_1} = \frac{-e^{netO_1}e^{netO_2}}{(e^{netO_1} + e^{netO_2})^2} =$
 $\frac{-(4.01786)(4.5614)}{8.57926^2} = \frac{-18.32701}{68.48638} = -0.2676$

Ahora se calculan las derivadas parciales de las entradas a los nodos de la capa de salida con respecto a los pesos que inciden en ellos, es decir:

- $\frac{\partial netO_1}{\partial w_{11}} = \frac{\partial(OL_1w_{11}+OL_2w_{21}+OL_3w_{31}+OL_4w_{41})}{\partial w_{11}} = OL_1 = 0.65$
- $\frac{\partial netO_1}{\partial w_{12}} = \frac{\partial(OL_1w_{11}+OL_2w_{21}+OL_3w_{31}+OL_4w_{41})}{\partial w_{12}} = OL_2 = 0.5525$
- $\frac{\partial netO_1}{\partial w_{13}} = \frac{\partial(OL_1w_{11}+OL_2w_{21}+OL_3w_{31}+OL_4w_{41})}{\partial w_{13}} = OL_3 = 0.8$
- $\frac{\partial netO_1}{\partial w_{14}} = \frac{\partial(OL_1w_{11}+OL_2w_{21}+OL_3w_{31}+OL_4w_{41})}{\partial w_{14}} = OL_4 = 0.535$
- $\frac{\partial netO_2}{\partial w_{12}} = \frac{\partial(OL_1w_{11}+OL_2w_{21}+OL_3w_{31}+OL_4w_{41})}{\partial w_{12}} = OL_1 = 0.65$
- $\frac{\partial netO_2}{\partial w_{22}} = \frac{\partial(OL_1w_{11}+OL_2w_{21}+OL_3w_{31}+OL_4w_{41})}{\partial w_{22}} = OL_2 = 0.5525$
- $\frac{\partial netO_2}{\partial w_{32}} = \frac{\partial(OL_1w_{11}+OL_2w_{21}+OL_3w_{31}+OL_4w_{41})}{\partial w_{32}} = OL_3 = 0.8$
- $\frac{\partial netO_2}{\partial w_{42}} = \frac{\partial(OL_1w_{11}+OL_2w_{21}+OL_3w_{31}+OL_4w_{41})}{\partial w_{42}} = OL_4 = 0.535$

Obtenidos los valores de todas las derivadas parciales, se reemplazan dichos valores en las ecuaciones de la gradiente del error de la red con respecto a cada uno de los pesos de la capa totalmente conectada:

- $\frac{\partial E}{\partial w_{11}} = (0)(0.2676)(0.65) + (-1.88083)(-0.2676)(0.65) = 0.32715$
- $\frac{\partial E}{\partial w_{12}} = (0)(-0.2676)(0.65) + (-1.88083)(0.2676)(0.65) = -0.32715$
- $\frac{\partial E}{\partial w_{21}} = (0)(0.2676)(0.5525) + (-1.88083)(-0.2676)(0.5525) = 0.27808$
- $\frac{\partial E}{\partial w_{22}} = (0)(-0.2676)(0.5525) + (-1.88083)(0.2676)(0.5525) = -0.27808$

- $\frac{\partial E}{\partial w_{31}} = (0)(0.2676)(0.8) + (-1.88083)(-0.2676)(0.8) = 0.40265$
- $\frac{\partial E}{\partial w_{32}} = (0)(-0.2676)(0.8) + (-1.88083)(0.2676)(0.8) = -0.40265$
- $\frac{\partial E}{\partial w_{41}} = (0)(0.2676)(0.535) + (-1.88083)(-0.2676)(0.535) = 0.26927$
- $\frac{\partial E}{\partial w_{42}} = (0)(-0.2676)(0.535) + (-1.88083)(0.2676)(0.535) = -0.26927$

Una vez conocida la gradiente del error con respecto a cada uno de los pesos se puede actualizar dichos pesos con la optimización Adam (se emplean las ecuaciones 9.2, 9.3, 9.5, 9.6 y 9.7). Como el proceso se encuentra en la primera iteración los momentos m_0 y v_0 se inicializan en 0; con ello las ecuaciones 9.2, 9.3, 9.5 y 9.6 para el ejemplo de la figura 23 en la primera iteración ($t = 1$) se definen como:

- $m_1 = \beta_1 m_0 + (1 - \beta_1) \frac{\partial E}{\partial w_{ij}}$
- $v_1 = \beta_2 v_0 + (1 - \beta_2) \frac{\partial E^2}{\partial w_{ij}}$

Pero $m_0 = v_0 = 0$, entonces m_1 y v_1 se reescriben como:

- $m_1 = (1 - \beta_1) \frac{\partial E}{\partial w_{ij}}$
- $v_1 = (1 - \beta_2) \frac{\partial E^2}{\partial w_{ij}}$

Para $t = 1$ las respectivas correcciones de los momentos m_1 y v_1 son:

- $\widehat{m}_1 = \frac{m_1}{(1 - \beta_1^t)} = \frac{m_1}{(1 - \beta_1^1)} = \frac{(1 - \beta_1) \frac{\partial E}{\partial w_{ij}}}{(1 - \beta_1)} = \frac{\partial E}{\partial w_{ij}}$
- $\widehat{v}_1 = \frac{v_1}{(1 - \beta_2^t)} = \frac{v_1}{(1 - \beta_2^1)} = \frac{(1 - \beta_2) \frac{\partial E^2}{\partial w_{ij}}}{(1 - \beta_2^1)} = \frac{\partial E^2}{\partial w_{ij}}$

Se recuerda que la actualización de los pesos se lleva a cabo con las correcciones de los dos momentos de la siguiente manera para $t = 1$:

- $w_{ij}(new) = w_{ij}(old) - \eta \frac{\widehat{m}_1}{\sqrt{\widehat{v}_1 + \epsilon}}$

Aquí se considera $\eta = 0.001$ y $\epsilon = 0$ (por motivos de simplicidad de cálculos), entonces reemplazando estos valores en la ecuación anterior para $t = 1$ se tiene:

- $w_{ij}(new) = w_{ij}(old) - 0.001 \frac{\frac{\partial E}{\partial w_{ij}}}{\sqrt{\frac{\partial E}{\partial w_{ij}}^2}}$

Para w_{11} :

- $w_{11}(new) = w_{11}(old) - 0.001 \frac{\frac{\partial E}{\partial w_{11}}}{\sqrt{\frac{\partial E}{\partial w_{11}}^2}} = 0.4 - 0.001 \left(\frac{0.32715}{\sqrt{0.107027}} \right) = 0.4 - 0.001 \frac{0.32715}{0.32715} = 0.4 - 0.001 = 0.399$

Para w_{12} :

- $w_{12}(new) = w_{12}(old) - 0.001 \frac{\frac{\partial E}{\partial w_{12}}}{\sqrt{\frac{\partial E}{\partial w_{12}}^2}} = 0.45 - 0.001 \left(\frac{-0.32715}{\sqrt{0.107027}} \right) = 0.45 + 0.001 \frac{0.32715}{0.32715} = 0.45 + 0.001 = 0.451$

Para w_{21} :

- $$w_{21}(new) = w_{21}(old) - 0.001 \frac{\frac{\partial E}{\partial w_{21}}}{\sqrt{\frac{\partial E}{\partial w_{21}}^2}} = 0.5 - 0.001 \left(\frac{0.27808}{\sqrt{0.07733}} \right) = 0.5 -$$

$$0.001 \frac{0.27808}{0.27808} = 0.5 - 0.001 = 0.499$$

Para w_{22} :

- $$w_{22}(new) = w_{22}(old) - 0.001 \frac{\frac{\partial E}{\partial w_{22}}}{\sqrt{\frac{\partial E}{\partial w_{22}}^2}} = 0.55 - 0.001 \left(\frac{-0.27808}{\sqrt{0.07733}} \right) = 0.5 +$$

$$0.001 \frac{0.27808}{0.27808} = 0.55 + 0.001 = 0.551$$

Para w_{31} :

- $$w_{31}(new) = w_{31}(old) - 0.001 \frac{\frac{\partial E}{\partial w_{31}}}{\sqrt{\frac{\partial E}{\partial w_{31}}^2}} = 0.6 - 0.001 \left(\frac{0.40265}{\sqrt{0.16213}} \right) = 0.5 -$$

$$0.001 \frac{0.27808}{0.27808} = 0.6 - 0.001 = 0.599$$

Para w_{32} :

- $$w_{32}(new) = w_{32}(old) - 0.001 \frac{\frac{\partial E}{\partial w_{32}}}{\sqrt{\frac{\partial E}{\partial w_{32}}^2}} = 0.65 - 0.001 \left(\frac{-0.40265}{\sqrt{0.16213}} \right) = 0.5 +$$

$$0.001 \frac{0.27808}{0.27808} = 0.65 + 0.001 = 0.651$$

Para w_{41} :

- $$w_{41}(new) = w_{41}(old) - 0.001 \frac{\frac{\partial E}{\partial w_{41}}}{\sqrt{\frac{\partial E}{\partial w_{41}}^2}} = 0.7 - 0.001 \left(\frac{0.26927}{\sqrt{0.07251}} \right) = 0.7 -$$

$$0.001 \frac{0.26927}{0.26927} = 0.7 - 0.001 = 0.699$$

Para w_{42} :

- $$w_{42}(new) = w_{42}(old) - 0.001 \frac{\frac{\partial E}{\partial w_{42}}}{\sqrt{\frac{\partial E}{\partial w_{42}}^2}} = 0.7 - 0.001 \left(\frac{-0.26927}{\sqrt{0.07251}} \right) = 0.75 +$$

$$0.001 \frac{0.26927}{0.26927} = 0.75 + 0.001 = 0.751$$

Actualizados los pesos de la capa totalmente conectada, el *back-propagation* continúa hacia las capas de convolución y de reducción. Recordar que los pesos del filtro de la figura 23 afectan (por teoría) a todas las unidades del mapa de características (el bloque naranja de la figura 23) de manera que si no existiera una capa de reducción la gradiente del error de la red con respecto a F_{11} , F_{12} , F_{21} y F_{22} sería:

- $$\frac{\partial E}{\partial F_{11}} = \frac{\partial E}{\partial OC_{11}} \frac{\partial OC_{11}}{\partial netOC_{11}} \frac{\partial netOC_{11}}{\partial F_{11}} + \frac{\partial E}{\partial OC_{12}} \frac{\partial OC_{12}}{\partial netOC_{12}} \frac{\partial netOC_{12}}{\partial F_{11}} +$$

$$\frac{\partial E}{\partial OC_{13}} \frac{\partial OC_{13}}{\partial netOC_{13}} \frac{\partial netOC_{13}}{\partial F_{11}} + \frac{\partial E}{\partial OC_{14}} \frac{\partial OC_{14}}{\partial netOC_{14}} \frac{\partial netOC_{14}}{\partial F_{11}} + \frac{\partial E}{\partial OC_{21}} \frac{\partial OC_{21}}{\partial netOC_{21}} \frac{\partial netOC_{21}}{\partial F_{11}} +$$

$$\frac{\partial E}{\partial OC_{22}} \frac{\partial OC_{22}}{\partial netOC_{22}} \frac{\partial netOC_{22}}{\partial F_{11}} + \frac{\partial E}{\partial OC_{23}} \frac{\partial OC_{23}}{\partial netOC_{23}} \frac{\partial netOC_{23}}{\partial F_{11}} + \frac{\partial E}{\partial OC_{24}} \frac{\partial OC_{24}}{\partial netOC_{24}} \frac{\partial netOC_{24}}{\partial F_{11}} +$$

$$\frac{\partial E}{\partial OC_{31}} \frac{\partial OC_{31}}{\partial netOC_{31}} \frac{\partial netOC_{31}}{\partial F_{11}} + \frac{\partial E}{\partial OC_{32}} \frac{\partial OC_{32}}{\partial netOC_{32}} \frac{\partial netOC_{32}}{\partial F_{11}} + \frac{\partial E}{\partial OC_{33}} \frac{\partial OC_{33}}{\partial netOC_{33}} \frac{\partial netOC_{33}}{\partial F_{11}} +$$

$$\frac{\partial E}{\partial OC_{34}} \frac{\partial OC_{34}}{\partial netOC_{34}} \frac{\partial netOC_{34}}{\partial F_{11}} + \frac{\partial E}{\partial OC_{41}} \frac{\partial OC_{41}}{\partial netOC_{41}} \frac{\partial netOC_{41}}{\partial F_{11}} + \frac{\partial E}{\partial OC_{42}} \frac{\partial OC_{42}}{\partial netOC_{42}} \frac{\partial netOC_{42}}{\partial F_{11}} +$$

$$\frac{\partial E}{\partial OC_{43}} \frac{\partial OC_{43}}{\partial netOC_{43}} \frac{\partial netOC_{43}}{\partial F_{11}} + \frac{\partial E}{\partial OC_{44}} \frac{\partial OC_{44}}{\partial netOC_{44}} \frac{\partial netOC_{44}}{\partial F_{11}}$$
- $$\frac{\partial E}{\partial F_{12}} = \frac{\partial E}{\partial OC_{11}} \frac{\partial OC_{11}}{\partial netOC_{11}} \frac{\partial netOC_{11}}{\partial F_{12}} + \frac{\partial E}{\partial OC_{12}} \frac{\partial OC_{12}}{\partial netOC_{12}} \frac{\partial netOC_{12}}{\partial F_{12}} +$$

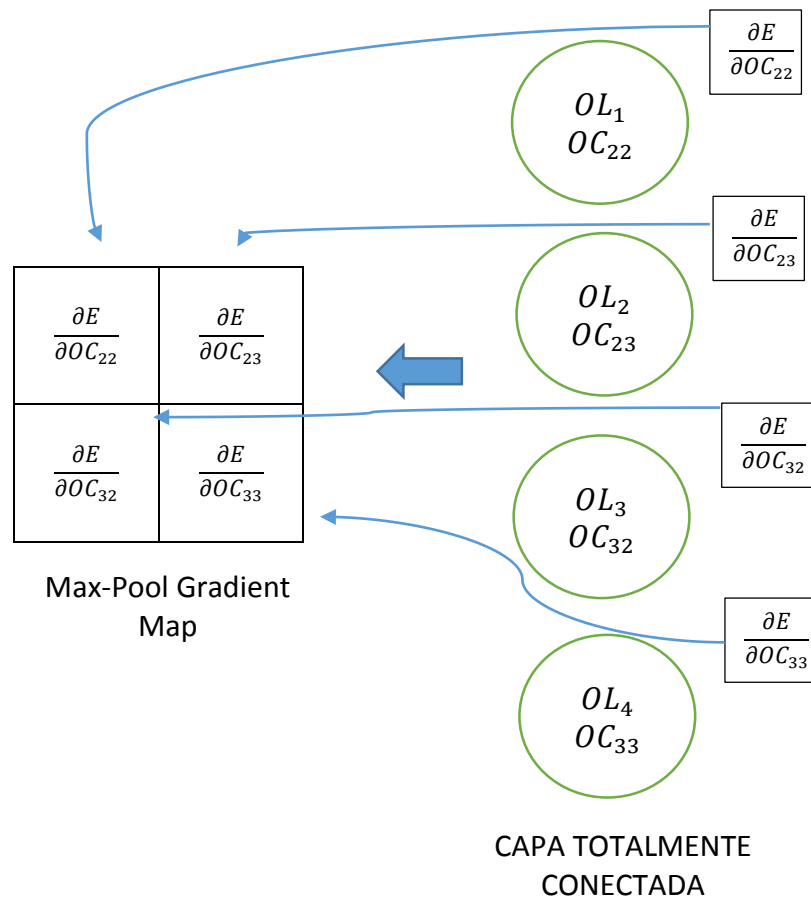
$$\frac{\partial E}{\partial OC_{13}} \frac{\partial OC_{13}}{\partial netOC_{13}} \frac{\partial netOC_{13}}{\partial F_{12}} + \frac{\partial E}{\partial OC_{14}} \frac{\partial OC_{14}}{\partial netOC_{14}} \frac{\partial netOC_{14}}{\partial F_{12}} + \frac{\partial E}{\partial OC_{21}} \frac{\partial OC_{21}}{\partial netOC_{21}} \frac{\partial netOC_{21}}{\partial F_{12}} +$$

$$\frac{\partial E}{\partial OC_{22}} \frac{\partial OC_{22}}{\partial netOC_{22}} \frac{\partial netOC_{22}}{\partial F_{12}} + \frac{\partial E}{\partial OC_{23}} \frac{\partial OC_{23}}{\partial netOC_{23}} \frac{\partial netOC_{23}}{\partial F_{12}} + \frac{\partial E}{\partial OC_{24}} \frac{\partial OC_{24}}{\partial netOC_{24}} \frac{\partial netOC_{24}}{\partial F_{12}} +$$

$$\frac{\partial E}{\partial OC_{31}} \frac{\partial OC_{31}}{\partial netOC_{31}} \frac{\partial netOC_{31}}{\partial F_{12}} + \frac{\partial E}{\partial OC_{32}} \frac{\partial OC_{32}}{\partial netOC_{32}} \frac{\partial netOC_{32}}{\partial F_{12}} + \frac{\partial E}{\partial OC_{33}} \frac{\partial OC_{33}}{\partial netOC_{33}} \frac{\partial netOC_{33}}{\partial F_{12}} +$$

$$\frac{\partial E}{\partial OC_{34}} \frac{\partial OC_{34}}{\partial netOC_{34}} \frac{\partial netOC_{34}}{\partial F_{12}} + \frac{\partial E}{\partial OC_{41}} \frac{\partial OC_{41}}{\partial netOC_{41}} \frac{\partial netOC_{41}}{\partial F_{12}} + \frac{\partial E}{\partial OC_{42}} \frac{\partial OC_{42}}{\partial netOC_{42}} \frac{\partial netOC_{42}}{\partial F_{12}} +$$

$$\frac{\partial E}{\partial OC_{43}} \frac{\partial OC_{43}}{\partial netOC_{43}} \frac{\partial netOC_{43}}{\partial F_{12}} + \frac{\partial E}{\partial OC_{44}} \frac{\partial OC_{44}}{\partial netOC_{44}} \frac{\partial netOC_{44}}{\partial F_{12}}$$



Nótese que, de acuerdo con el gráfico, el error se propaga solamente por las unidades que salieron ganadoras de Max-Pooling. Eso quiere decir que las gradientes de las demás unidades adquieren un valor nulo, ya que dichas unidades no influyeron en el error de la red, esto se aprecia de mejor manera en la siguiente matriz que representa el mapa de gradientes del bloque naranja de la figura 23:

0	0	0	0
0	$\frac{\partial E}{\partial OC_{22}}$	$\frac{\partial E}{\partial OC_{23}}$	0
0	$\frac{\partial E}{\partial OC_{32}}$	$\frac{\partial E}{\partial OC_{33}}$	0
0	0	0	0

$\frac{\partial E}{\partial OC_{22}}$	$\frac{\partial E}{\partial OC_{23}}$
$\frac{\partial E}{\partial OC_{32}}$	$\frac{\partial E}{\partial OC_{33}}$

Mapa de gradientes
Max-Pool
2x2.

Mapa de gradientes de
todas las unidades del
mapa de características.

Las ecuaciones de $\frac{\partial E}{\partial F_{11}}$, $\frac{\partial E}{\partial F_{12}}$, $\frac{\partial E}{\partial F_{21}}$ y $\frac{\partial E}{\partial F_{22}}$ quedan reescritas como:

- $$\frac{\partial E}{\partial F_{11}} = \frac{\partial E}{\partial OC_{22}} \frac{\partial OC_{22}}{\partial netOC_{22}} \frac{\partial netOC_{22}}{\partial F_{11}} + \frac{\partial E}{\partial OC_{23}} \frac{\partial OC_{23}}{\partial netOC_{23}} \frac{\partial netOC_{23}}{\partial F_{11}} + \frac{\partial E}{\partial OC_{32}} \frac{\partial OC_{32}}{\partial netOC_{32}} \frac{\partial netOC_{32}}{\partial F_{11}} + \frac{\partial E}{\partial OC_{33}} \frac{\partial OC_{33}}{\partial netOC_{33}} \frac{\partial netOC_{33}}{\partial F_{11}}$$
- $$\frac{\partial E}{\partial F_{12}} = \frac{\partial E}{\partial OC_{22}} \frac{\partial OC_{22}}{\partial netOC_{22}} \frac{\partial netOC_{22}}{\partial F_{12}} + \frac{\partial E}{\partial OC_{23}} \frac{\partial OC_{23}}{\partial netOC_{23}} \frac{\partial netOC_{23}}{\partial F_{12}} + \frac{\partial E}{\partial OC_{32}} \frac{\partial OC_{32}}{\partial netOC_{32}} \frac{\partial netOC_{32}}{\partial F_{12}} + \frac{\partial E}{\partial OC_{33}} \frac{\partial OC_{33}}{\partial netOC_{33}} \frac{\partial netOC_{33}}{\partial F_{12}}$$
- $$\frac{\partial E}{\partial F_{21}} = \frac{\partial E}{\partial OC_{22}} \frac{\partial OC_{22}}{\partial netOC_{22}} \frac{\partial netOC_{22}}{\partial F_{21}} + \frac{\partial E}{\partial OC_{23}} \frac{\partial OC_{23}}{\partial netOC_{23}} \frac{\partial netOC_{23}}{\partial F_{21}} + \frac{\partial E}{\partial OC_{32}} \frac{\partial OC_{32}}{\partial netOC_{32}} \frac{\partial netOC_{32}}{\partial F_{21}} + \frac{\partial E}{\partial OC_{33}} \frac{\partial OC_{33}}{\partial netOC_{33}} \frac{\partial netOC_{33}}{\partial F_{21}}$$

- $$\frac{\partial E}{\partial F_{22}} = \frac{\partial E}{\partial OC_{22}} \frac{\partial OC_{22}}{\partial netOC_{22}} \frac{\partial netOC_{22}}{\partial F_{22}} + \frac{\partial E}{\partial OC_{23}} \frac{\partial OC_{23}}{\partial netOC_{23}} \frac{\partial netOC_{23}}{\partial F_{22}} +$$

$$\frac{\partial E}{\partial OC_{32}} \frac{\partial OC_{32}}{\partial netOC_{32}} \frac{\partial netOC_{32}}{\partial F_{22}} + \frac{\partial E}{\partial OC_{33}} \frac{\partial OC_{33}}{\partial netOC_{33}} \frac{\partial netOC_{33}}{\partial F_{22}}$$

Se comienza calculando $\frac{\partial E}{\partial OC_{22}}$, $\frac{\partial E}{\partial OC_{23}}$, $\frac{\partial E}{\partial OC_{32}}$ y $\frac{\partial E}{\partial OC_{33}}$, para ello recordar que :

- $$netO_1 = OC_{22}w_{11} + OC_{23}w_{21} + OC_{32}w_{31} + OC_{33}w_{41}$$
- $$netO_2 = OC_{22}w_{12} + OC_{23}w_{22} + OC_{32}w_{32} + OC_{33}w_{42}$$

Nótese como las unidades ganadoras de Max-Pooling influyen en las net-inputs de los nodos de la capa de salida de manera que el error se propaga por dichas unidades, entonces para $\frac{\partial E}{\partial OC_{22}}$ se tiene lo siguiente:

- $$\frac{\partial E}{\partial OC_{22}} = \frac{\partial E}{\partial netO_1} \frac{\partial netO_1}{\partial OC_{22}} + \frac{\partial E}{\partial netO_2} \frac{\partial netO_2}{\partial OC_{22}}$$

Pero $\frac{\partial E}{\partial netO_1} = \frac{\partial E}{\partial O_1} \frac{\partial O_1}{\partial netO_1} + \frac{\partial E}{\partial O_2} \frac{\partial O_2}{\partial netO_1}$ y $\frac{\partial E}{\partial netO_2} = \frac{\partial E}{\partial O_1} \frac{\partial O_1}{\partial netO_2} + \frac{\partial E}{\partial O_2} \frac{\partial O_2}{\partial netO_2}$, de manera que $\frac{\partial E}{\partial OC_{22}}$ queda como:

- $$\frac{\partial E}{\partial OC_{22}} = \left(\frac{\partial E}{\partial O_1} \frac{\partial O_1}{\partial netO_1} + \frac{\partial E}{\partial O_2} \frac{\partial O_2}{\partial netO_1} \right) \frac{\partial netO_1}{\partial OC_{22}} + \left(\frac{\partial E}{\partial O_1} \frac{\partial O_1}{\partial netO_2} + \frac{\partial E}{\partial O_2} \frac{\partial O_2}{\partial netO_2} \right) \frac{\partial netO_2}{\partial OC_{22}} =$$

$$\frac{\partial E}{\partial O_1} \frac{\partial O_1}{\partial netO_1} \frac{\partial netO_1}{\partial OC_{22}} + \frac{\partial E}{\partial O_2} \frac{\partial O_2}{\partial netO_1} \frac{\partial netO_1}{\partial OC_{22}} + \frac{\partial E}{\partial O_1} \frac{\partial O_1}{\partial netO_2} \frac{\partial netO_2}{\partial OC_{22}} + \frac{\partial E}{\partial O_2} \frac{\partial O_2}{\partial netO_2} \frac{\partial netO_2}{\partial OC_{22}}$$

Sin embargo, hay que recordar que $\frac{\partial E}{\partial O_1} = 0$ de manera que la ecuación se simplifica de la siguiente forma:

- $$\frac{\partial E}{\partial OC_{22}} = \frac{\partial E}{\partial O_2} \frac{\partial O_2}{\partial netO_1} \frac{\partial netO_1}{\partial OC_{22}} + \frac{\partial E}{\partial O_2} \frac{\partial O_2}{\partial netO_2} \frac{\partial netO_2}{\partial OC_{22}}$$

En pasos previos se determinó que $\frac{\partial E}{\partial O_2} = -1.88083$,

$\frac{\partial O_2}{\partial netO_1} = -0.2676$ y $\frac{\partial O_2}{\partial netO_2} = 0.2676$, de manera que solo queda hallar $\frac{\partial netO_1}{\partial OC_{22}}$ y $\frac{\partial netO_2}{\partial OC_{22}}$, entonces:

- $\frac{\partial netO_1}{\partial OC_{22}} = \frac{\partial(OC_{22}w_{11}+OC_{23}w_{21}+OC_{32}w_{31}+OC_{33}w_{41})}{\partial OC_{22}} = w_{11} = 0.4$
- $\frac{\partial netO_2}{\partial OC_{22}} = \frac{\partial(OC_{22}w_{12}+OC_{23}w_{22}+OC_{32}w_{32}+OC_{33}w_{42})}{\partial OC_{22}} = w_{12} = 0.45$

Los valores de w_{11} y w_{12} corresponden a los valores con los que se inicializó la red (puesto que aún se está ejecutando la primera iteración), y no los valores actualizados por la optimización Adam (los valores actualizados se emplearán en la segunda iteración). Reemplazando los valores se tiene ahora:

- $\frac{\partial E}{\partial OC_{22}} = (-1.88083)(-0.2676)(w_{11}) + (-1.88083)(0.2676)(w_{12}) = (1.88083)(0.2676)(0.4 - 0.45) = -0.02517$

El procedimiento que se realizó para obtener $\frac{\partial E}{\partial OC_{22}}$ se puede generalizar para obtener $\frac{\partial E}{\partial OC_{23}}$, $\frac{\partial E}{\partial OC_{32}}$ y $\frac{\partial E}{\partial OC_{33}}$. Entonces dichas ecuaciones quedan como:

- $\frac{\partial E}{\partial OC_{23}} = (1.88083)(0.2676)(w_{21} - w_{22}) = (1.88083)(0.2676)(0.5 - 0.55) = (1.88083)(0.2676)(0.05) = -0.02517$
- $\frac{\partial E}{\partial OC_{32}} = (1.88083)(0.2676)(w_{31} - w_{32}) = (1.88083)(0.2676)(0.6 - 0.65) = (1.88083)(0.2676)(0.05) = -0.02517$
- $\frac{\partial E}{\partial OC_{33}} = (1.88083)(0.2676)(w_{41} - w_{42}) = (1.88083)(0.2676)(0.7 - 0.75) = (1.88083)(0.2676)(0.05) = -0.02517$

Obtenidos los valores de la gradiente del error con respecto a las unidades ganadoras de Max-Pool, se continúa calculando las derivadas parciales de dichas unidades vencedoras con respecto a sus *net-inputs*. Recordar que $netOC_{22}$, $netOC_{23}$, $netOC_{32}$ y $netOC_{33}$ son todos mayores a 0, por lo que se tiene:

- $\frac{\partial OC_{22}}{\partial netOC_{22}} = \frac{\partial(\max(0, netOC_{22}))}{\partial netOC_{22}} = \frac{\partial netOC_{22}}{\partial netOC_{22}} = 1$
- $\frac{\partial OC_{23}}{\partial netOC_{23}} = \frac{\partial(\max(0, netOC_{23}))}{\partial netOC_{23}} = \frac{\partial netOC_{23}}{\partial netOC_{23}} = 1$
- $\frac{\partial OC_{32}}{\partial netOC_{32}} = \frac{\partial(\max(0, netOC_{32}))}{\partial netOC_{32}} = \frac{\partial netOC_{32}}{\partial netOC_{32}} = 1$
- $\frac{\partial OC_{33}}{\partial netOC_{33}} = \frac{\partial(\max(0, netOC_{33}))}{\partial netOC_{33}} = \frac{\partial netOC_{33}}{\partial netOC_{33}} = 1$

Se prosigue calculando la derivada parcial de cada unidad ganadora con respecto a los pesos F_{11}, F_{12}, F_{21} y F_{22} :

- $\frac{\partial netOC_{22}}{\partial F_{11}} = \frac{\partial(x_{22}F_{11} + x_{23}F_{12} + x_{32}F_{21} + x_{33}F_{22})}{\partial F_{11}} = x_{22} = 0.5$
- $\frac{\partial netOC_{23}}{\partial F_{11}} = \frac{\partial(x_{23}F_{11} + x_{24}F_{12} + x_{33}F_{21} + x_{34}F_{22})}{\partial F_{11}} = x_{23} = 0.6$
- $\frac{\partial netOC_{32}}{\partial F_{11}} = \frac{\partial(x_{32}F_{11} + x_{33}F_{12} + x_{42}F_{21} + x_{43}F_{22})}{\partial F_{11}} = x_{32} = 0.8$
- $\frac{\partial netOC_{33}}{\partial F_{11}} = \frac{\partial(x_{33}F_{11} + x_{34}F_{12} + x_{43}F_{21} + x_{44}F_{22})}{\partial F_{11}} = x_{33} = 0.85$
- $\frac{\partial netOC_{22}}{\partial F_{12}} = \frac{\partial(x_{22}F_{11} + x_{23}F_{12} + x_{32}F_{21} + x_{33}F_{22})}{\partial F_{12}} = x_{23} = 0.6$
- $\frac{\partial netOC_{23}}{\partial F_{12}} = \frac{\partial(x_{23}F_{11} + x_{24}F_{12} + x_{33}F_{21} + x_{34}F_{22})}{\partial F_{12}} = x_{24} = 0.65$
- $\frac{\partial netOC_{32}}{\partial F_{12}} = \frac{\partial(x_{32}F_{11} + x_{33}F_{12} + x_{42}F_{21} + x_{43}F_{22})}{\partial F_{12}} = x_{33} = 0.85$

- $\frac{\partial netOC_{33}}{\partial F_{12}} = \frac{\partial(x_{33}F_{11}+x_{34}F_{12}+x_{43}F_{21}+x_{44}F_{22})}{\partial F_{12}} = x_{34} = 0.4$
- $\frac{\partial netOC_{22}}{\partial F_{21}} = \frac{\partial(x_{22}F_{11}+x_{23}F_{12}+x_{32}F_{21}+x_{33}F_{22})}{\partial F_{21}} = x_{32} = 0.8$
- $\frac{\partial netOC_{23}}{\partial F_{21}} = \frac{\partial(x_{23}F_{11}+x_{24}F_{12}+x_{33}F_{21}+x_{34}F_{22})}{\partial F_{21}} = x_{33} = 0.85$
- $\frac{\partial netOC_{32}}{\partial F_{21}} = \frac{\partial(x_{32}F_{11}+x_{33}F_{12}+x_{42}F_{21}+x_{43}F_{22})}{\partial F_{21}} = x_{42} = 0.9$
- $\frac{\partial netOC_{33}}{\partial F_{21}} = \frac{\partial(x_{33}F_{11}+x_{34}F_{12}+x_{43}F_{21}+x_{44}F_{22})}{\partial F_{21}} = x_{43} = 0.95$
- $\frac{\partial netOC_{22}}{\partial F_{22}} = \frac{\partial(x_{22}F_{11}+x_{23}F_{12}+x_{32}F_{21}+x_{33}F_{22})}{\partial F_{22}} = x_{33} = 0.85$
- $\frac{\partial netOC_{23}}{\partial F_{22}} = \frac{\partial(x_{23}F_{11}+x_{24}F_{12}+x_{33}F_{21}+x_{34}F_{22})}{\partial F_{22}} = x_{34} = 0.4$
- $\frac{\partial netOC_{32}}{\partial F_{22}} = \frac{\partial(x_{32}F_{11}+x_{33}F_{12}+x_{42}F_{21}+x_{43}F_{22})}{\partial F_{22}} = x_{43} = 0.95$
- $\frac{\partial netOC_{33}}{\partial F_{22}} = \frac{\partial(x_{33}F_{11}+x_{34}F_{12}+x_{43}F_{21}+x_{44}F_{22})}{\partial F_{22}} = x_{44} = 0.3$

Reemplazando los valores en las ecuaciones de $\frac{\partial E}{\partial F_{11}}$, $\frac{\partial E}{\partial F_{12}}$, $\frac{\partial E}{\partial F_{21}}$ y $\frac{\partial E}{\partial F_{22}}$ se tiene:

- $\frac{\partial E}{\partial F_{11}} = (-0.02517)(1)(0.5) - (0.02517)(1)(0.6) - (0.02517)(1)(0.8) - (0.02517)(1)(0.85) = (0.02517)(1)(0.6 + 0.5 + 0.8 + 0.85) = -0.06922$

Generalizando para $\frac{\partial E}{\partial F_{12}}$, $\frac{\partial E}{\partial F_{21}}$ y $\frac{\partial E}{\partial F_{22}}$:

- $\frac{\partial E}{\partial F_{12}} = (-0.02517)(1)(0.6 + 0.65 + 0.85 + 0.4) = -0.06293$
- $\frac{\partial E}{\partial F_{21}} = (-0.02517)(1)(0.8 + 0.85 + 0.9 + 0.95) = -0.08809$

- $\frac{\partial E}{\partial F_{22}} = (-0.02517)(1)(0.85 + 0.4 + 0.95 + 0.3) = -0.06293$

Finalmente, se procede a actualizar los pesos F_{11} , F_{12} , F_{21} y F_{22} con la optimización Adam. Recordar, que aún se está ejecutando la primera iteración, es decir $t = 1$, por lo que ambos momentos de la optimización se definen como:

- $m_1 = \beta_1 m_0 + (1 - \beta_1) \frac{\partial E}{\partial F_{ij}}$
- $v_1 = \beta_2 v_0 + (1 - \beta_2) \frac{\partial E^2}{\partial F_{ij}}$

Pero $m_0 = v_0 = 0$, entonces m_1 y v_1 se reescriben como:

- $m_1 = (1 - \beta_1) \frac{\partial E}{\partial F_{ij}}$
- $v_1 = (1 - \beta_2) \frac{\partial E^2}{\partial F_{ij}}$

Para $t = 1$ las respectivas correcciones de los momentos m_1 y v_1 son:

- $\widehat{m}_1 = \frac{m_1}{(1 - \beta_1^t)} = \frac{m_1}{(1 - \beta_1^1)} = \frac{(1 - \beta_1) \frac{\partial E}{\partial F_{ij}}}{(1 - \beta_1)} = \frac{\partial E}{\partial F_{ij}}$
- $\widehat{v}_1 = \frac{v_1}{(1 - \beta_2^t)} = \frac{v_1}{(1 - \beta_2^1)} = \frac{(1 - \beta_2) \frac{\partial E^2}{\partial F_{ij}}}{(1 - \beta_2^1)} = \frac{\partial E^2}{\partial F_{ij}}$

Los pesos del filtro actualizados en $t = 1$ tienen la forma:

- $F_{ij}(new) = F_{ij}(old) - \eta \frac{\widehat{m}_1}{\sqrt{\widehat{v}_1 + \epsilon}}$

Líneas arriba se explicó que $\eta = 0.001$ y $\epsilon = 0$ (para simplificar cálculos), de manera que se tiene lo siguiente para F_{11} :

- $F_{11}(new) = 0.15 - 0.001 \frac{-0.06922}{\sqrt{(-0.06922)^2}} = 0.15 + 0.001 = 0.151$

Para F_{12} :

- $F_{12}(new) = 0.2 - 0.001 \frac{-0.06293}{\sqrt{(-0.0629)^2}} = 0.2 + 0.001 = 0.201$

Para F_{21} :

- $F_{21}(new) = 0.25 - 0.001 \frac{-0.08809}{\sqrt{(-0.08809)^2}} = 0.25 + 0.001 = 0.251$

Para F_{22} :

- $F_{22}(new) = 0.3 - 0.001 \frac{-0.06293}{\sqrt{(-0.0629)^2}} = 0.3 + 0.001 = 0.301$

Una vez actualizados los pesos del filtro, se ejecuta nuevamente el proceso de *feed-forward* para calcular la nueva función de error con los valores actualizados en la primera iteración; dado que recién se ha completado 1 iteración para la arquitectura de la figura 23, es muy poco probable que el error haya sufrido un cambio significativo de manera que se debe volver a ejecutar el *back-propagation*. Este proceso de *feed-forward* seguido por *back-propagation* se repite tantas veces como sea necesario de manera que la función de error o coste converja a cero garantizando así un correcto aprendizaje de la red neuronal convolucional. Las tablas de los parámetros que se emplearon para los ejemplos de la etapa de preprocesamiento (sección 2.3.3.1) y la etapa de procesamiento (sección 2.3.4.4) se listan a continuación:

TABLA DE PARÁMETROS ETAPA DE PRE-PROCESAMIENTO				
Nombre del Parámetro	Símbolo	Dimensión	Rango o valor posible	Comentario
Intensidad del píxel en las coordenadas x_p, y_p	$V(x_p, y_p)$	adimensional	0 a 255	Se calcula por interpolación bilineal a partir de x_p, y_p y de los coeficientes bilineales.
Coordenadas del píxel a calcular por interpolación bilineal	x_p, y_p	adimensional	Entero positivo	Se utilizan para calcular la intensidad del píxel.
Coficiente bilineales	a_p, b_p, c_p, d_p	adimensional	-	Se calculan partir de las coordenadas y valor de intensidad de los 4 vecinos más cercanos a $V(x_p, y_p)$
Coordenadas de los 4 vecinos más cercanos a $V(x_p, y_p)$	(x_{p1}, y_{p1}) (x_{p2}, y_{p2}) (x_{p3}, y_{p3}) (x_{p4}, y_{p4})	adimensional	Entero positivo	Intervienen en el cálculo de los coeficientes bilineales.
Intensidad de los 4 vecinos más cercanos a $V(x_p, y_p)$	$f(Q_{11})$ $f(Q_{12})$ $f(Q_{21})$ $f(Q_{22})$	adimensional	0 a 255	Intervienen en el cálculo de los coeficientes bilineales.

Tabla (1). Tabla de parámetros empleados para el preprocesado

Elaboración: los autores

TABLA DE PARÁMETROS ETAPA DE PROCESAMIENTO				
Nombre del Parámetro	Símbolo	Dimensión	Rango o valor posible	Comentario
Unidad del mapa de entrada.	x	adimensional	<0,1>	Son los píxeles normalizados de la imagen de entrada
Entradas a las Unidades de las capas de convolución.	netOC	adimensional	-	Son el resultado de la convolucion de las unidades del mapa de entradas con el filtro
Salidas de las Unidades de las capas de convolución.	OC	adimensional	-	Son el resultado de haber aplicado la función de activación a las entradas a las unidades de las capas de convolución.
Unidades de las capas de reducción (pooling layer).	P	adimensional	-	Son el resultado de aplicar Max-Pooling a las unidades de salida de la capa convolutiva.
Entradas a las Unidades de las capas totalmente conectadas (fully connected layers)	netOL	adimensional	-	Son el resultado de haber aplicado Flatten al tensor de la capa de reducción.
Salidas de las Unidades de las capas totalmente conectadas (fully connected layers).	OL	adimensional	-	Sirven para calcular los nodos de la siguiente capa.
Entradas a las unidades de la capa de salida de la red.	netO	adimensional	-	Son afectadas por la función de activación Softmax para convertirlas en una distribución de probabilidad.
Salida o predicción de la red .	O	adimensional	<0,1>	Es la probabilidad que una imagen pertenezca acada una de las posibles clases
Valor o predicción esperada.	y_T	adimensional	0 ó 1	Es la probabilidad deseada de que una imagen corresponda a una de las clases.
Pesos de las fully conected layers.	w	adimensional	[-limit,limit]	Conectan a las neuronas de las capas totalmente conectadas.
Pesos de los filtros o kernels.	F	adimensional	[-limit,limit]	Convolucionan con el mapa de entrada
Error de la red por Cross Entropy	E	adimensional	-	Función de coste que se calcula al final del proceso de feed-forward.
Gradiente del Error con respecto a los pesos de la fully conected layer	$\frac{\partial E}{\partial w}$	adimensional	-	Empleado por la optimización Adam para calcular las estimaciones de primer y segundo momento
Gradiente del Error con respecto a los pesos de los filtros.	$\frac{\partial E}{\partial F}$	adimensional	-	Empleado por la optimización Adam para calcular las estimaciones de primer y segundo momento

Tabla (2). Tabla de parámetros empleados para el procesado

Elaboración: los autores

La continuación de la tabla 2 se muestra en la tabla 3:

Tasa de aprendizaje	η	adimensional	0.001 por recomendación de los creadores de Adam	Hiperparámetro
Estimación de primer momento	m	adimensional	-	Empleado por Adam para actualizar los pesos.
Tasa de decaimiento exponencial para la estimación del primer momento	β_1	adimensional	0.9 por recomendación de los creadores de Adam	Hiperparámetro
Estimación de segundo momento	v	adimensional	-	Empleado por Adam para actualizar los pesos.
Tasa de decaimiento exponencial para la estimación del segundo momento	β_2	adimensional	0.999 por recomendación de los creadores de Adam	Hiperparámetro
Corrección de la estimación del primer momento.	\hat{m}	adimensional	-	Empleado por Adam para actualizar los pesos.
Corrección de la estimación del segundo momento.	\hat{v}	adimensional	-	Empleado por Adam para actualizar los pesos.
Hiper-parámetro suavizante de la optimización Adam.	ϵ	adimensional	10^{-8} por recomendación de los creadores de Adam.	Hiperparámetro
Altura de un tensor.	H	adimensional	mayor a cero	constante
Ancho de un tensor.	W	adimensional	mayor a cero	constante
Profundidad de un tensor.	D	adimensional	mayor a cero	constante
Pasos del kernel o stride	s	adimensional	mayor a cero	Indica cada cuántos píxeles avanza un kernel o filtro en una imagen
Número de clases (o personas) a clasificar.	K	adimensional	mayor a cero	El número de clases es igual al número de nodos de salida en una red neuronal convolucional.
Rango mínimo y máximo de la distribución de probabilidad uniforme de la inicialización He	$[-limit, limit]$	adimensional	$limit = \sqrt{\frac{6}{N_{in}}}$	Distribución uniforme de donde se toman los pesos del filtro y de la capa totalmente conectada
Número de entradas al tensor de pesos.	N_{in}	adimensional	variable	Sirve para calcular los límites de la distribución uniforme de donde se toman las muestras

Tabla (3). Tabla de parámetros empleados para el procesamiento (continuación).

Elaboración: los autores

CAPÍTULO III METODOLOGÍA

3.1 Materiales

En este capítulo se detallan los materiales más importantes empleados para la elaboración del presente trabajo. Algunas condiciones de operación establecidas previo a la selección de los materiales fueron:

- El sistema debe identificar usuarios en tiempo real (para ello se requiere diseñar una estructura que albergue un sensor para la adquisición de imágenes).
- El sistema debe operar con un universo de alrededor de 80 individuos (para ello debe diseñarse una red neural adecuada).
- El sistema debe operar al aire libre (son necesarios LED's para dar uniformidad de iluminación en la estructura de adquisición de imágenes).
- El sistema debe ser instalado en una mesa de trabajo de dimensiones 50cm de largo, 60cm de ancho y 80cm de altura.
- El sistema debe reconocer inmediatamente al usuario cuando se adquiere una muestra palmar para su verificación.

3.1.1 Cámara Logitech C-922

Es el sensor elegido para ejecutar la adquisición de las imágenes de las palmas de las manos de los usuarios, este sensor se encontrará alojado en la base de una estructura (diseñada por los autores)

hecha de melamina la cual brindará soporte a las manos de los potenciales usuarios para facilitar y estandarizar la adquisición de las muestras palmares. Este dispositivo (regresar a la figura 9 para apreciar el sensor) es compatible con el sistema operativo de Windows (desde Windows 7 en adelante), integra una lente de cristal HD y cuenta con un sistema de enfoque automático. Otras especificaciones técnicas se muestran a continuación en la tabla 5:

Característica	Valor
Alimentación	5V DC
Máxima Resolución	1080p/30fps
Campo de visión	78 grados
Longitud de cable	1.5m
Peso (incluyendo clip y cable)	162gr
Altura (con clip)	14.22 cm
Ancho (con clip)	13.72cm
Profundidad (con clip)	8.13cm

Tabla (4). Especificaciones técnicas de la cámara digital Logitech C-922

Elaboración: los autores

Criterios de selección del sensor:

- **Disponibilidad:** el sensor está ampliamente disponible en el mercado peruano
- **Resolución:** el sensor es parecido al sensor de adquisición de imágenes (Logitech C-930) empleado por uno de los *papers* del estado del arte
- **Dimensiones:** las dimensiones reducidas del sensor los hacen ideal para ser alojado en un recinto de melamina
- **Precio:** el sensor costó S/271, lo cual es un monto que puede ser fácilmente asumido por los autores
- **Flexibilidad:** la estructura misma del sensor permite que el lente apunte verticalmente hacia el molde palmar del recinto

3.1.2 Tiras de LED de luz blanca

Las tiras de LED que emiten luz blanca acompañan a la cámara digital en la base del recinto de adquisición de muestras, se colocan alrededor de la cámara formando 4 aristas emitiendo su luz hacia la plataforma que contiene el molde donde se posiciona la palma de la persona (observar la figura 28), ello permite que haya uniformidad luminosa a la hora de tomar las muestras palmares independientemente de las condiciones luminosas en el exterior (recordar que el sistema debe ser capaz de operar al aire libre donde hay cambios constantes en la luminosidad. Sus especificaciones técnicas se describen en la tabla 6.

Característica	Valor
Alimentación	12 VDC
Alto	10mm
Ancho	15mm
Protección	IP65
Tipo de LED	5050
Angulo de luz	120 grados
Consumo por metro	14.4 W

Tabla (5). Especificaciones técnicas de los LED empleados en el presente trabajo

Elaboración: los autores

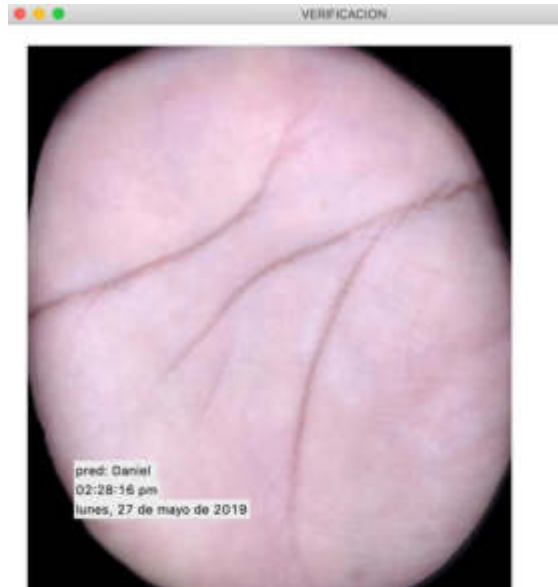


Figura (28). Región palmar iluminada por el arreglo de LED's posicionados alrededor de la cámara Logitech C-922

Elaboración: los autores

3.1.3 Tarjeta gráfica Nvidia GTX 1060 6GB

Una vez que se cuentan con las muestras palmares preprocesadas, se las debe ingresar a una arquitectura neuronal adecuada con el objetivo de poder clasificar a un número de individuos en función de los patrones palmares.

Para garantizar una clasificación adecuada, la red neuronal convolucional debe ser entrenada un determinado número de iteraciones (también llamadas épocas) de manera que la función de coste o error converja lo más posible a cero; tal y como se apreció en la sección 2.3.4.4 el cálculo requerido para una iteración de la arquitectura de una RNC muy básica es de hecho sustancial si es que es realizado por un humano; evidenciando así la inmensa cantidad de cálculos requeridos para arquitecturas mucho más complejas (como la que se presenta en el trabajo actual), dicha tarea resultaría imposible de realizar por parte de un ser humano por lo que se recurren a los CPU y GPU (y si es que se cuenta con los recursos económicos suficientes, una TPU) para ejecutar el entrenamiento de las redes neuronales.

Dado que no se cuentan con los recursos suficientes para adquirir una TPU (que es la mejor opción hoy en día para entrenar redes neuronales), se debió elegir entre una CPU y una GPU; las CPU se diseñaron para analizar e interpretar lógica compleja en un determinado código, razón por la cual poseen muy pocos núcleos y no las hacen opciones viables para entrenar redes neurales convolucionales (las cuales requieren de millones de cálculos). Las GPU, por otro lado, tienen muchos más núcleos que una CPU (aunque son de menor tamaño) y son capaces de procesar un conjunto de cálculos similares en forma paralela; de hecho Nvidia creó una arquitectura y plataforma de cálculo en paralelo para su GPU a la que llamó CUDA. Afortunadamente uno de los autores del presente trabajo ya contaba con una GPU de Nvidia correspondiente al modelo GTX 1060 6GB (observar la figura 29).



Figura (29). Tarjeta gráfica Nvidia modelo GTX 1060

Fuente: <https://www.nvidia.com/en-us/geforce/products/10series/geforce-gtx-1060/>

Esta GPU cuenta con 1280 de estos núcleos CUDA, lo que la convierte en un buen candidato para entrenar a la red neuronal propuesta. Otras especificaciones técnicas se muestran a continuación en la tabla 7:

Característica	Valor
Velocidad de Memoria	8 Gbps.
Núcleos CUDA	1280
Frame Buffer	6 GB GDDR5/X
Arquitectura de GPU	Pascal
Reloj (Base)	1506 Mhz
Altura	4.378 pulgadas
Ancho	9.823 pulgadas
Máxima temperatura de GPU	94 grados Celsius
Potencia de tarjeta gráfica	120 W

Tabla (6) Algunas especificaciones técnicas de la tarjeta gráfica Nvidia GTX 1060

Elaboración: los autores

Gracias al número de núcleos CUDA de la tarjeta gráfica Nvidia GTX 1060, el tiempo que le toma a la red neural completar 1 iteración es bastante reducido y se muestra en la figura 30:

```

WARNING:tensorflow:From C:\Users\John\Anaconda3\envs\tf-gpu\lib\site-packages\tensorflow\python\ops\math_ops.py:3066: to_int32
(from tensorflow.python.ops.math_ops) is deprecated and will be removed in a future version.
Instructions for updating:
Use tf.cast instead.
Train on 5760 samples, validate on 1440 samples
Epoch 1/10
5760/5760 [-----] - 37s 6ms/step - loss: 3.9621 - acc: 0.1457 - val_loss: 1.2786 - val_acc: 0.7590
Epoch 2/10
5760/5760 [-----] - 32s 6ms/step - loss: 0.7776 - acc: 0.7674 - val_loss: 0.8692 - val_acc: 0.9938
Epoch 3/10
5760/5760 [-----] - 32s 6ms/step - loss: 0.3183 - acc: 0.9066 - val_loss: 0.0245 - val_acc: 0.9986
Epoch 4/10

```

Figura (30). Tiempo (en el recuadro rojo) que le toma a la tarjeta gráfica completar una iteración de una RNC con 5760 muestras de entrenamiento.

Elaboración: los autores

3.2 Método

La primera parte del método consistió en diseñar la plataforma que contiene el molde de una mano adulta promedio, la plataforma está hecha con un material de melamina para darle durabilidad. Una vez obtenida la plataforma del molde palmar se diseña la estructura circundante (también hecha de melamina) en donde irá acoplada dicha plataforma y donde se

posicionarán las manos de los usuarios, para su posterior registro y verificación.

Una vez terminada la estructura de melamina, se procede a acoplar tanto la cámara Logitech C-922 en la base de dicha estructura, con el sensor fijado se procede a colocar las tiras de LED's alrededor del sensor (o cámara digital) de manera que se garantice una iluminación más uniforme en dirección a la plataforma palmar ubicada unos cuantos centímetros arriba de la cámara.

Con la estructura de melamina y sus componentes electrónicos ya acoplados se prosigue con la segunda parte del método la cual consiste en la obtención de las muestras palmares. En ese sentido los LED que emiten luz blanca deben ser encendidos antes que la cámara, ya que de lo contrario esta adquiere las imágenes con una tonalidad azul la cual sería detrimental para la correcta adquisición de las muestras palmares.

Una vez hecho esto, a cada usuario se le pidió que posicionara su mano derecha en el molde palmar estirando al máximo sus dedos de manera que se resalten los pliegues palmares; a continuación, se procede a realizar una filmación que dura poco más de 20 segundos (recordar que la Logitech C-922, graba a 30 tramas por segundo) durante los cuales al usuario se le solicitó que rotara ligeramente su palma derecha dentro de los límites del molde palmar. A este video de alrededor de 20 segundos se le desfragmenta en múltiples imágenes las cuales deberían totalizar alrededor de las 600 muestras.

Con estas 600 muestras por usuario, en el que cada segundo contenía 30 muestras, se seleccionó el 50%; quedando al final 300 muestras por cada sujeto de prueba y con ello se ingresa a la siguiente etapa que es la de preprocesamiento. Se debe mencionar que en paralelo a esta desfragmentación de vídeo se aplicó una máscara diseñada por los autores la cual tiene la forma del agujero palmar del molde para que de esta manera la región exterior a dicho agujero quede eliminada y solo se extraiga la región palmar de interés, la cual contiene los pliegues palmares discriminativos. A

esta nueva imagen se le aplica un algoritmo de recorte para que se delimite el área contenida en el agujero de la plantilla palmar. La etapa de preprocesamiento continúa con un reescalamiento por interpolación bilineal, de manera que las imágenes queden reducidas a un tamaño final de 88x80. Finalmente, las imágenes son normalizadas antes de ser ingresadas a la red neuronal convolucional.

Ahora se procede con la etapa de diseño de la arquitectura neural convolucional que permita la adecuada clasificación de los individuos en función de los patrones palmares mostrados en las imágenes preprocesadas de 88x80, para obtener la arquitectura adecuada. Los autores pensaron y probaron distintos modelos posibles, los cuales arrojaron distintas curvas de precisión y decaimiento de la función de pérdida para un determinado conjunto de condiciones iniciales.

La arquitectura neural cuyas curvas de pérdida de entrenamiento y validación muestren el mejor desempeño, es elegida para procesar las muestras de 88x80 que provienen de la etapa de preprocesamiento. Finalmente, la etapa de postprocesamiento incluye el diseño de una GUI con programación Python para que el usuario sea capaz de registrar su asistencia.

CAPÍTULO IV

DESARROLLO

4.1 Diagrama de bloques

El desarrollo del presente trabajo dividió el sistema propuesto en cuatro grandes bloques los cuales son:

- Bloque de Adquisición de imágenes: es la etapa donde el sensor elegido por los autores adquiere las muestras palmares de los usuarios.
- Bloque de Preprocesamiento: es la etapa donde se aplican técnicas de procesamiento digital de imágenes para preparar a las muestras palmares antes de entregarlas a la red neuronal convolucional
- Bloque de Procesamiento: es la etapa donde la RNC se encarga de extraer y clasificar los patrones principales de cada muestra palmar adquirida en las etapas anteriores
- Bloque de Postprocesamiento: es la etapa que se encarga brindarle una interfaz amigable con el usuario al sistema propuesto de manera que sea más fácil realizar los registros de asistencia

El lector puede seguir la secuencia del diagrama de bloques de la figura 31 para tener una mejor idea del desarrollo propuesto por los autores:

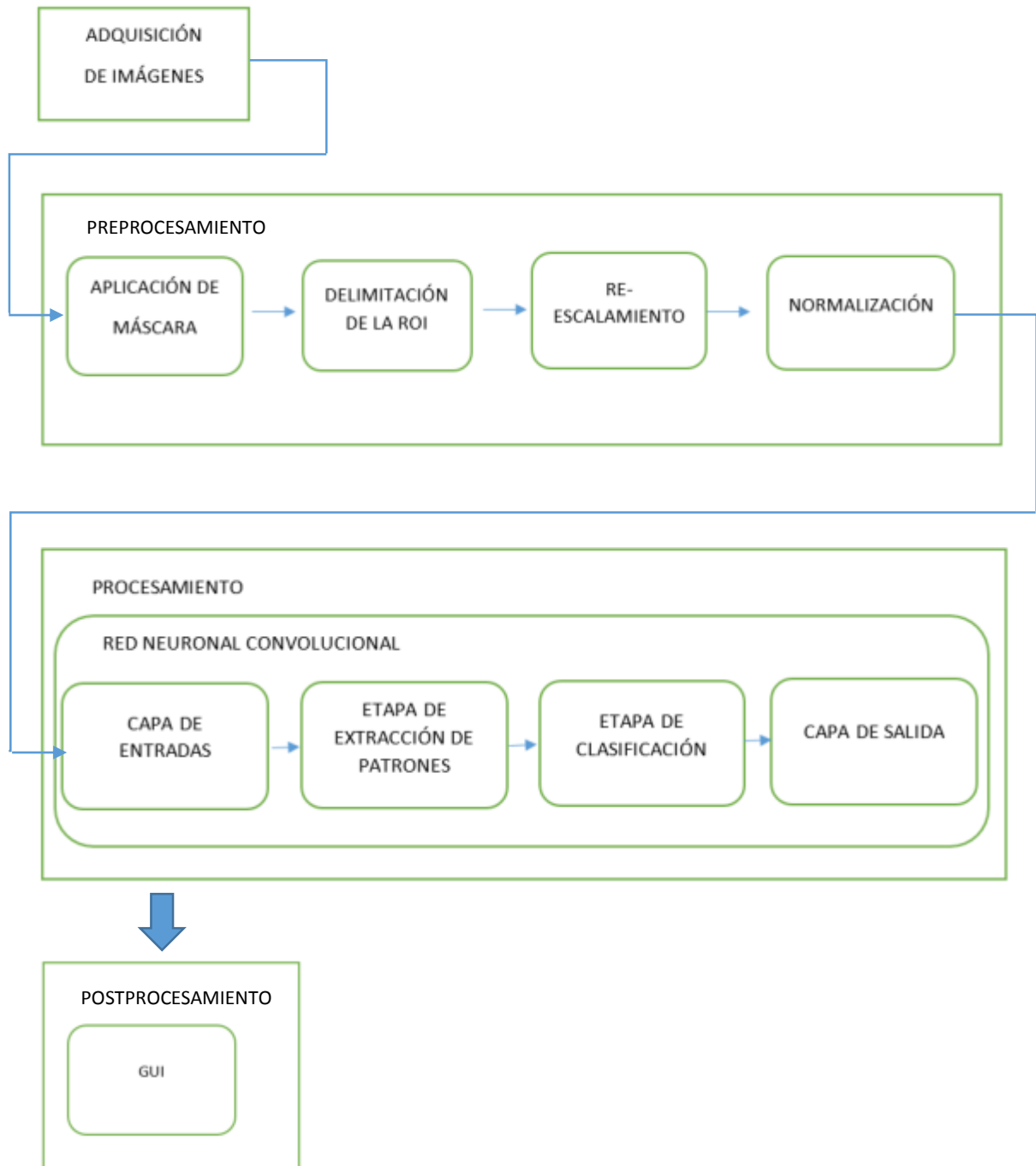


Figura (31). Diagrama de bloques del presente trabajo
Elaboración: los autores

4.2 Diseño del recinto de adquisición de imágenes

Para realizar la adquisición de imágenes no solo se requiere de un sensor (la cámara digital Logitech C-922) sino que también se requiere un

recinto que garantice la uniformidad de la adquisición de las muestras palmares.

4.2.1 Determinación de las dimensiones de la plantilla y la plataforma palmar

Con el fin de obtener muestras palmares que no presenten alteraciones de rotación se propone un molde palmar en donde se posicione la mano derecha de un usuario tal y como se aprecia en la figura 32:



Figura (32). Plantilla palmar donde se colocan las manos derechas de los usuarios

Elaboración: los autores

Para obtener el largo y ancho de una mano se siguen los siguientes criterios que van de acuerdo con lo especificado por Binivignat, Almagia, Lizana y Olave en su publicación titulada *Biometric Aspects of the Hand in Chilean Individuals* del año 2012:

- El largo de una mano se considerará como la distancia que separa el pliegue distal de la muñeca del extremo distal del dedo medio.
- El ancho de una mano representará a la distancia que separa el margen lateral de la palma (antes del pliegue entre el índice y el pulgar), del margen medial de la palma previo al pliegue digito palmar del dedo meñique.

Bingvinat et al. (2012) proporciona los promedios de las dimensiones de las manos (derecha e izquierda) de hombres (y también de mujeres) de distintas edades tal y como se evidencia en la tabla de la figura 33.

edades	Longitud		Ancho	
	derecha	mano izquierda	derecha	mano izquierda
18-19	182±10	183±10	95±19	94±19
20-24	186±9	188±11	92±19	91±19
25-29	188±10	188±10	92±21	90±21
30-34	183±12	186±12	89±10	89±9
35-39	184±8	183±9	97±13	96±14
40-44	184±15	184±15	89±10	88±8
45-49	183±12	184±6	91±8	89±6
50-54	186±13	186±13	90±8	89±9
55-59	187±16	187±20	92±13	91±13
60-64	186±12	187±12	87±4	87±4

Figura (33). Promedios de la longitud y ancho (en milímetros) de las manos de hombres chilenos cuyas edades oscilan entre los 18 y 64 años

Fuente: *Biometric Aspects of the Hand in Chilean Individuals*. Binignat, Almagia, Lizana y Olave. *International Journal of Morphology*.2012

Los autores del presente trabajo decidieron tomar como referencia estos valores para el diseño del molde palmar ya que no se encontraron publicaciones similares para el caso de Perú; sin embargo, se considera que al ser Perú y Chile países fronterizos latinoamericanos la diferencia antropomórfica de sus respectivas poblaciones no debe ser significativa.

Para determinar la longitud del molde palmar se considera lo siguiente en función a los valores de la mano derecha de un hombre de acuerdo con la tabla de Binignat et al. (2012):

$$L_m = \frac{182+186+188+183+184+184+183+186+187+186}{10} = 184.9 \text{ mm} = 18.49 \text{ cm} \approx 19 \text{ cm}$$

Para determinar el ancho se sigue un proceso similar:

$$A_m = \frac{95+92+92+89+97+89+91+90+92+87}{10} = 91.4mm = 9.14cm \approx 9cm$$

Teniendo en cuenta los valores previamente calculados, el molde palmar deberá tener las siguientes especificaciones de diseño siguientes:

- La plantilla palmar tendrá la forma de la mano derecha de una persona adulta
- El largo de la plantilla palmar deberá ser de 19 cm
- El ancho de la plantilla palmar deberá ser de 9 cm

Obtenidas las dimensiones de la plantilla palmar se pueden especificar las dimensiones de la plataforma, la que irá impresa (observar la figura 34):

- La plataforma tendrá forma rectangular
- La plataforma estará hecha con melamina
- El largo de la plataforma será de 21 cm
- El ancho de la plataforma será de 23 cm

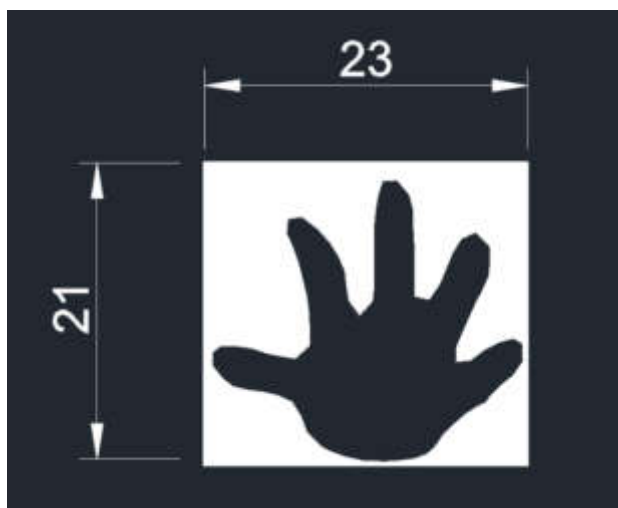


Figura (34). Vista superior de la plataforma palmar que contiene el molde o plantilla palmar (Diseño AutoCAD)

Elaboración: los autores

4.2.2 Determinación de las dimensiones del recinto de adquisición de imágenes

Una vez especificadas las dimensiones de la plataforma palmar, es posible especificar las dimensiones del recinto en donde va acoplada dicha plataforma hecha con melamina; en su base, el recinto alberga tanto a la cámara digital LogitechC-922 como a las tiras de LED's de luz blanca. Dicho recinto tiene las siguientes especificaciones de diseño:

- El recinto estará hecho de melamina
- El recinto tendrá una altura de 33cm
- La base del recinto tendrá 25 cm de ancho
- La base del recinto tendrá 25 cm de largo
- Una de las caras laterales del recinto se dejará abierta para poder acoplar y desacoplar la plataforma palmar cuando sea necesario.

El lector puede hacerse una mejor idea de las especificaciones antes mencionadas si recurre a la figura 35:

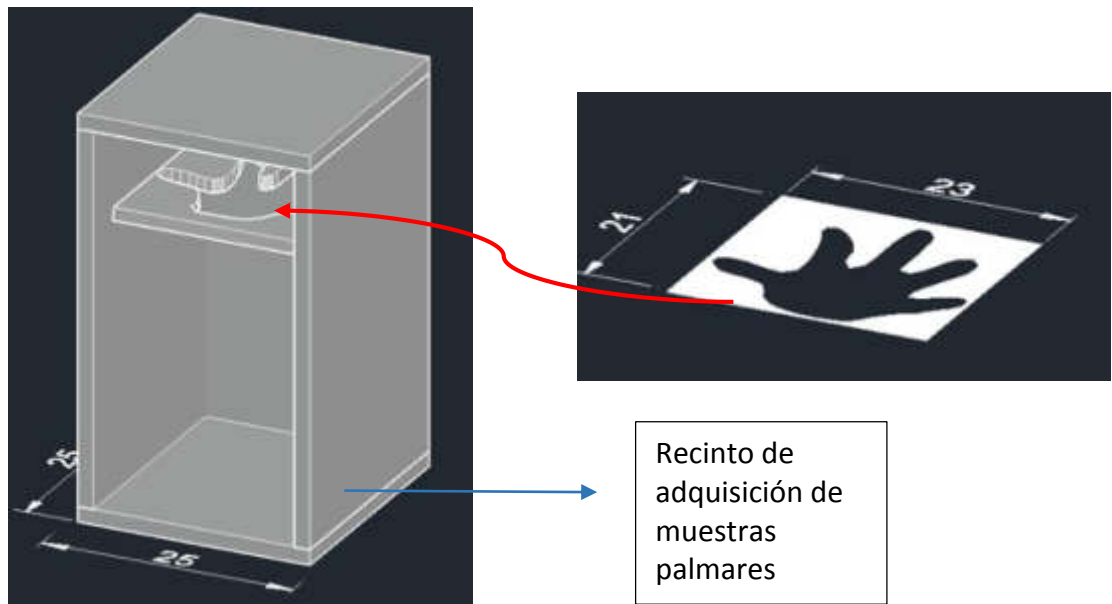


Figura (35) Vista tridimensional en AutoCAD (izquierda) del recinto de adquisición de muestras palmares (sin la puerta) con la plataforma palmar ya acoplada

Elaboración: los autores

4.2.3 Determinación de las dimensiones de la puerta que sella parcialmente una de las caras laterales del recinto

Para proteger a la cámara y los LED de luz blanca de factores externos que puedan dañarlos en cualquier capacidad, se debe diseñar una puerta que se acople al recinto de adquisición de imágenes de melamina; dicha puerta también evitará que se filtre la luz del exterior y permita así que exista uniformidad en la adquisición de las muestras palmares. Para ello se consideran las especificaciones de diseño siguientes:

- La puerta estará hecha de melamina al igual que el recinto de adquisición de imágenes y la plataforma palmar.
- La puerta deberá cubrir el área entre la plataforma palmar y la base del recinto de adquisición de muestras palmares.
- La puerta tendrá un ancho de 24 cm
- La puerta tendrá un largo de 32 cm

Todo lo expuesto anteriormente se puede apreciar mejor con ayuda de la figura 36:

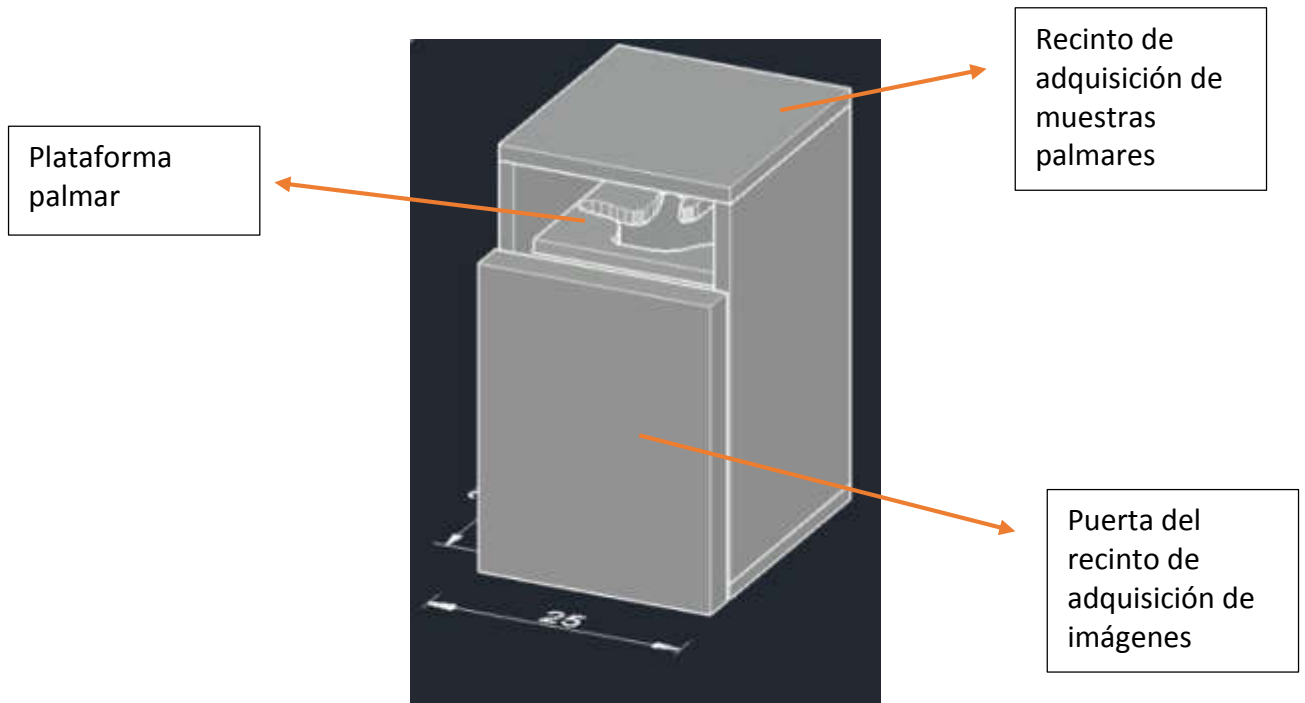


Figura (36) Vista tridimensional en AutoCAD del modelo del recinto de adquisición de imágenes con la puerta y la plataforma palmar ya acopladas

Elaboración: los autores

Limitaciones:

- La plataforma acoplada al recinto solo acepta las muestras palmares provenientes de las manos derechas de los usuarios.
- La plataforma acoplada al recinto solo acepta manos de personas adultas.

4.3 Desarrollo de la Etapa de Preprocesamiento

A continuación, se detallan los pasos seguidos en el preprocesamiento, tras haber adquirido el video de 20 segundos de la mano derecha del usuario posicionada en el molde o plantilla palmar.

4.3.1 Diseño de la máscara para extraer la región palmar de interés

Una vez que se desfragmenta el video que contiene las distintas posiciones de la mano del usuario dentro de la plantilla palmar; se tomó una de las tantas imágenes desfragmentadas (de tamaño 960x720) y, a continuación, fue convertida de RGB a HSV para que con la ayuda de un *Track Bar* se seleccionen los valores HSV con los cuales la región de interés queda anulada temporalmente, tal y como se muestra en la figura 37.

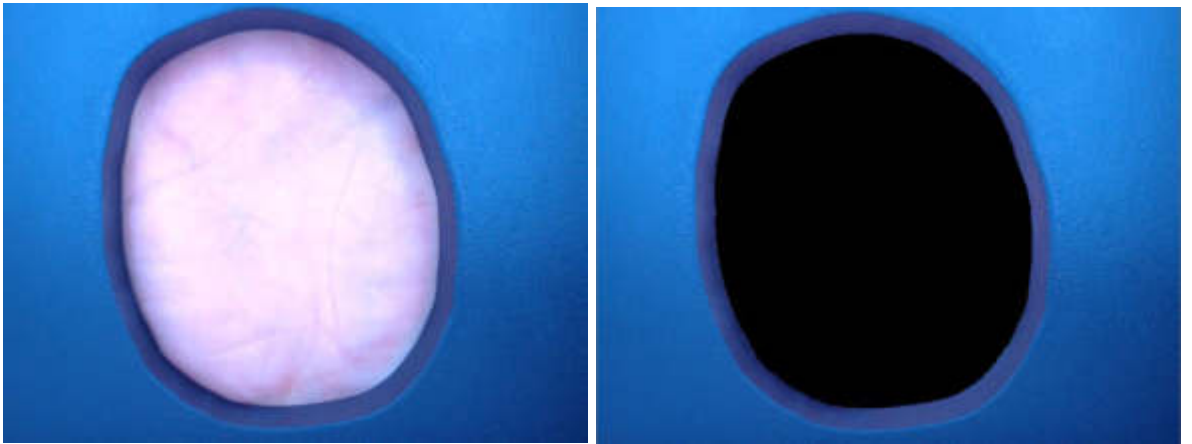


Figura (37). Imagen HSV de una muestra palmar con su región de interés intacta (izquierda) y la versión HSV de la misma imagen con su región de interés anulada (derecha)

Elaboración: los autores

Para obtener la imagen de la derecha de la figura, se debieron fijar los siguientes valores HSV con la ayuda del *Track Bar*:

- H *Low* deberá ser 87
- H *High* deberá ser 120
- S *Low* deberá ser 115
- S *High* deberá ser 255
- V *Low* deberá ser 0
- V *High* deberá ser 255

Todo esto se puede apreciar aún mejor si se recurre al diagrama de flujo correspondiente a la figura 38:

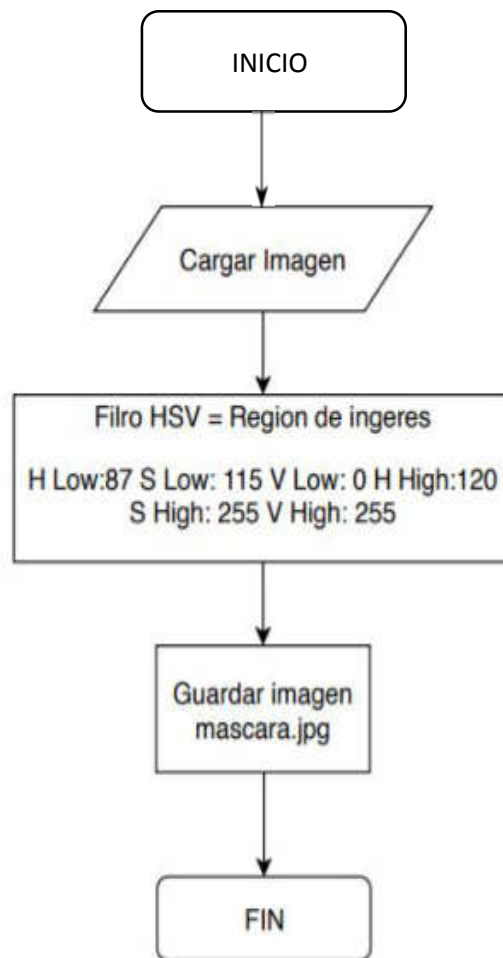


Figura (38). Diagrama de flujo que sigue el proceso de para anular la región central de la palma de una mano

Elaboración: los autores

Una vez obtenida la imagen en donde los píxeles de la región palmar de interés están seteados en cero, se le aplica un algoritmo de cambio de píxeles que analiza el valor de cada píxel de la imagen y consiste en cambiar los píxeles iniciales en 0, por píxeles en 1, y aquellos píxeles diferentes de cero serán cambiados a píxeles en 0, tal y como se muestra en el diagrama de flujo de la figura 39:

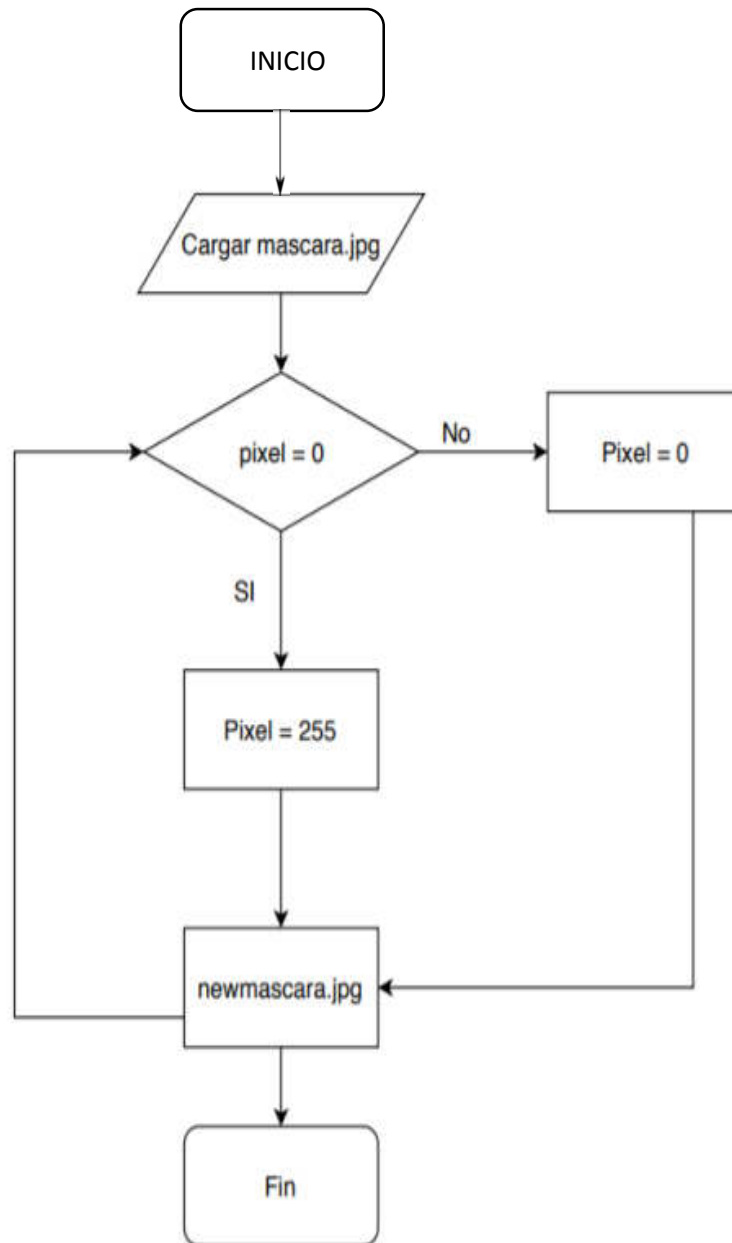


Figura (39). Diagrama de flujo para el cambio del valor de los píxeles de mascara.jpg

Elaboración: los autores

Los resultados de binarizar los píxeles en mascara.jpg se pueden observar mucho mejor en la figura 40:

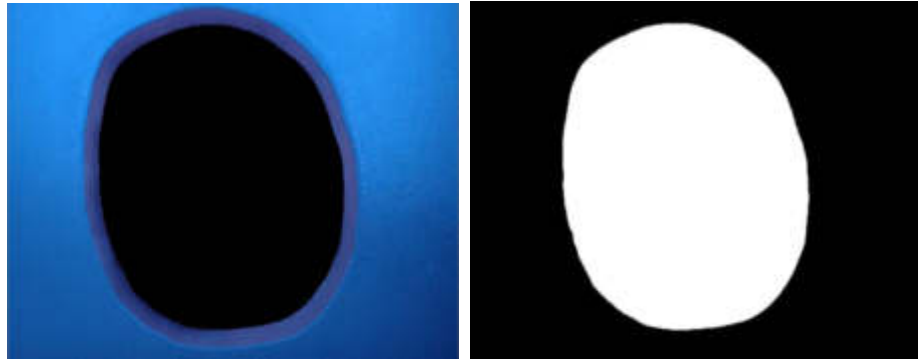


Figura (40). Comparación entre mascara.jpg (izquierda) y la versión binarizada de la misma imagen (derecha) tras aplicarse el algoritmo de cambio de píxeles (Nota: Ambas imágenes tienen las mismas dimensiones)

Elaboración: los autores

La imagen de la parte izquierda de la figura se aplica entonces a cada una de las muestras palmares en forma de una multiplicación punto a punto (ya que tanto la máscara como la imagen palmar adquirida con la cámara digital tienen las mismas dimensiones de 960x720); esto origina que la región central de la máscara (con píxeles de valor 255) extraiga la región de interés palmar, mientras que los píxeles fuera de la región central (con valor 0) anularán cualquier información irrelevante para el proceso de discriminación tal y como se aprecia en la figura 41.

Esta máscara es aplicada a cada una de las 24000 muestras palmares de la data set, obteniéndose nuevas muestras palmares con menos información innecesaria y manteniendo el tamaño de adquisición que es de 960x720.

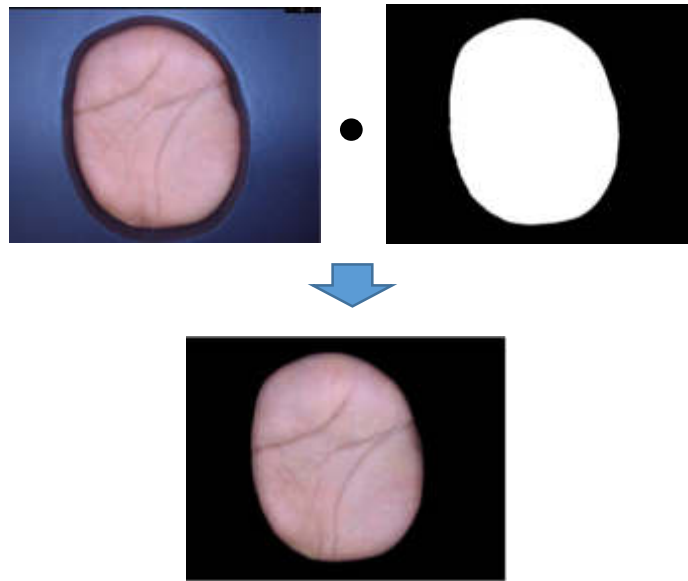


Figura (41). Multiplicación entre la imagen de la izquierda y la máscara (derecha) para obtener la nueva muestra palmar solo con la región de interés

Elaboración: los autores

Limitaciones:

- La máscara tiene la forma del orificio palmar del molde
- Para obtener la máscara se debe emplear un *Track Bar* HSV para poder anular la región central de interés ya que no es posible realizar dicho proceso en imágenes con el formato RGB

4.3.2 Recorte de imagen y delimitación de la ROI (Región de interés)

Ahora que se obtuvieron muestras de 960x720 que solo incluyen la región central palmar, se procede a emplear un algoritmo de recorte a la totalidad de dichas imágenes para delimitar aún más la ROI debido a que se desea eliminar el exceso de información innecesaria que quedó del paso anterior (ver figura 42).

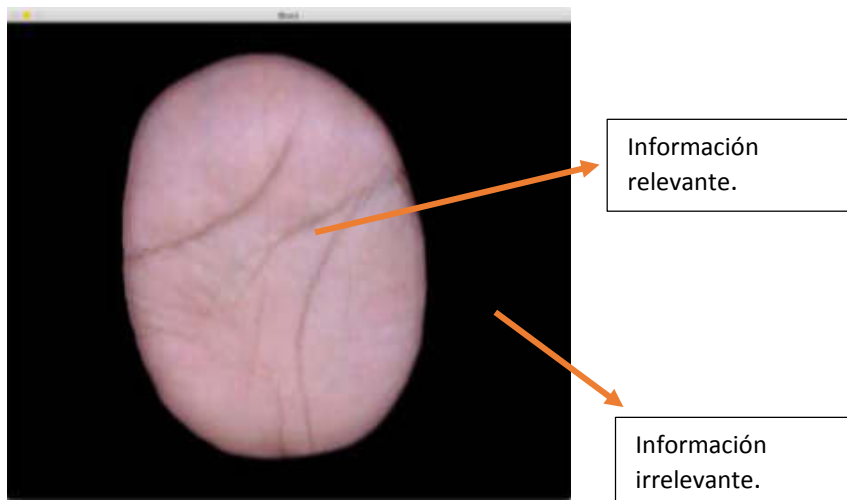


Figura (42). Imagen de una muestra palmar de dimensiones 960x720 (a la que se le aplicó la máscara), en donde se distingue la información relevante de la irrelevante.

Elaboración: los autores

Para delimitar aún más la imagen de la figura, se trazaron líneas verticales y horizontales imaginarias que coincidan con el agujero de la plantilla palmar, dado que dicho agujero se encuentra en la misma posición para cualquiera de las 24000 muestras. Por prueba y error se encontró que las imágenes se pueden recortar adecuadamente (y sin riesgo perder información relevante) en las coordenadas de la figura 43.

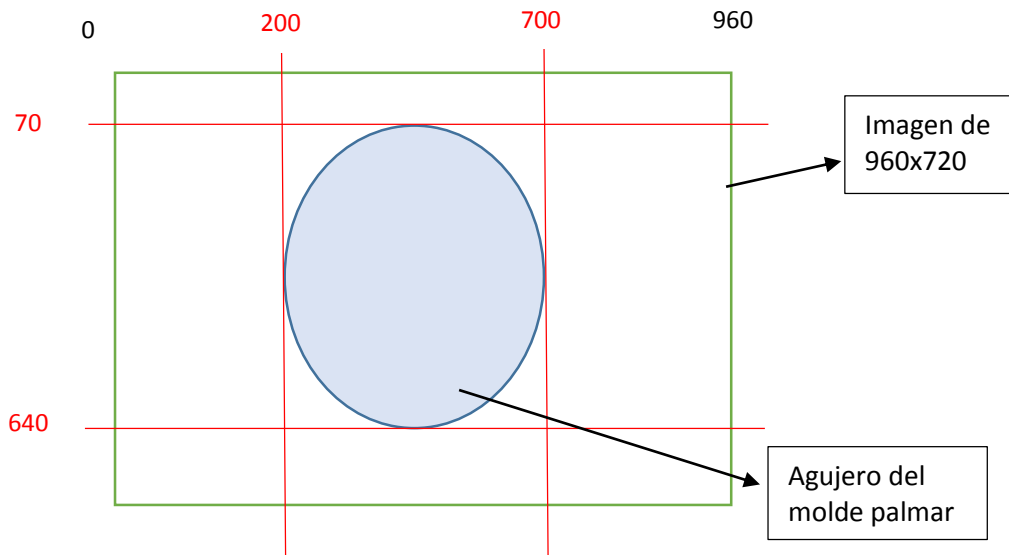


Figura (43). Representación del recorte de una muestra palmar trazando líneas verticales y horizontales en las coordenadas de color rojo

Elaboración: los autores

Para realizar esos cortes, se realizó el procedimiento descrito por el diagrama de flujo de la figura 44.

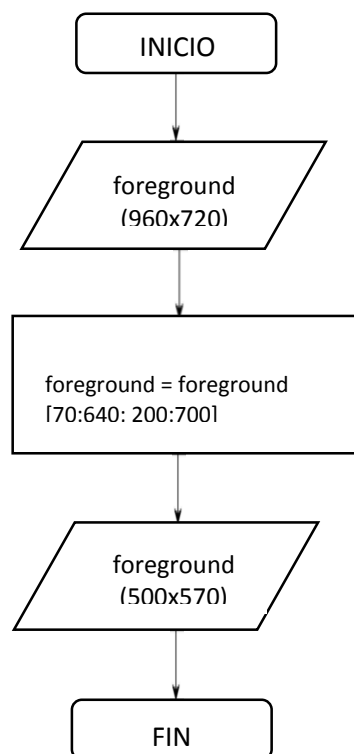


Figura (44). Diagrama de flujo para el recorte de las muestras palmares de 960x720

Elaboración: los autores

Los resultados se aprecian en la figura 45:

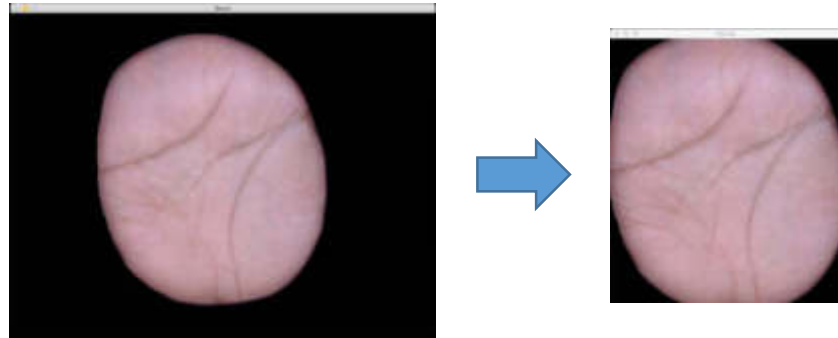


Figura (45). Resultado del recorte de la muestra de 960x720

Elaboración: los autores

4.3.3 Reescalamiento de muestras y normalización

La etapa de preprocesamiento culmina tomando los criterios que se mencionan a continuación:

- Las muestras de la etapa anterior (con tamaño 500x570) deberán ser re-escaladas a un tamaño de 88x80 utilizando interpolación bilineal (se aplica la ecuación 3.0 con el comando *resize*) debido a que tiene menor carga procesal que la interpolación bicúbica.
- Las imágenes de 88x80 serán normalizadas de manera que sus píxeles obtengan valores entre 0 y 1, lo cual también reduce carga procesal.

Estos criterios son entonces aplicados en el procedimiento expuesto en el diagrama de flujo de la figura 46:

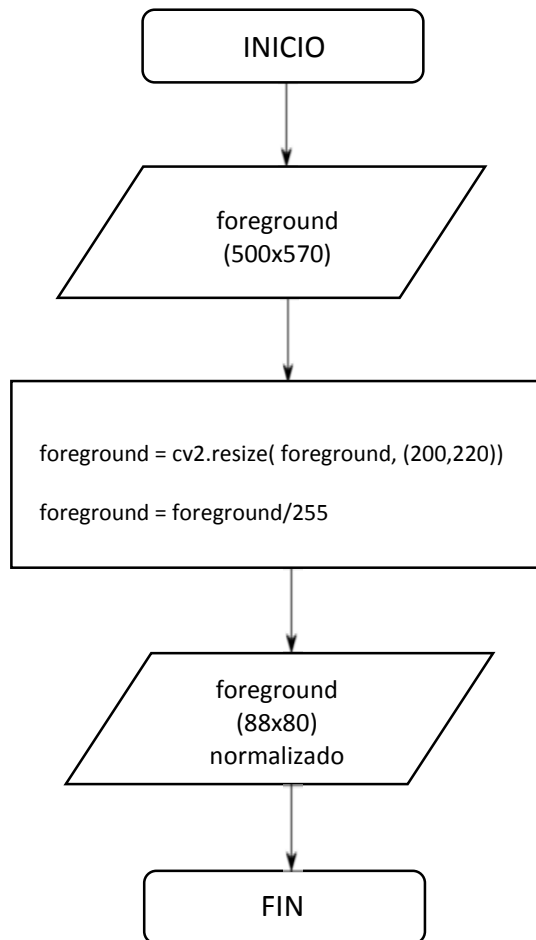


Figura (46). Secuencia para reescalar y normalizar las muestras de 500x570
Elaboración: los autores

4.4 Diseño de la arquitectura neuronal (Etapa de Procesamiento)

Se busca encontrar la arquitectura neural adecuada para la clasificación de 80 individuos, para ello se comienza diseñando arquitecturas neuronales básicas para evaluar su desempeño y a partir de ello decidir qué tanto incrementar la complejidad de dicho modelo o arquitectura.

Para obtener el número de capas adecuado para la tarea de clasificación se consideran las siguientes condiciones iniciales como especificaciones de diseño:

- El tamaño de los filtros de convolución deberá ser simétrico y ha de ser representado por un número impar (por convención)
- El número de pasos o *stride* de los filtros a la hora de recorrer la imagen ha de ser el mínimo posible (*stride*=1) de manera que se garantice que los filtros capten las líneas más sutiles de la región palmar del individuo
- Se utilizarán filtros de tamaño 2x2 para efectuar el proceso de reducción y así obtener los mapas de la capa de reducción. La función para aplicar será la de Max-Pooling debido a que es la más empleada en el estado del arte
- El número de filtros será incrementará a medida que se avance hacia las capas totalmente conectadas y a medida que la imagen vaya reduciéndose en tamaño. Además, el número de filtros serán potencias de 2 (por convención)
- La función de activación que se ha de emplear es la función ReLU por ser computacionalmente más eficiente que la función tangente hiperbólica o la función sigmoide. La capa de salida, sin embargo, empleará la función de activación Softmax debido a que esta función proporciona una distribución de probabilidad a la salida de la capa que está siendo afectada por dicha función
- El número de neuronas de las capas totalmente conectadas serán potencias de dos (por convención). Para la capa de salida el número de unidades debe coincidir con el número de clases o sujetos a clasificar

- La red será entrenada utilizando la optimización Adam por tratarse de una optimización perteneciente al estado del arte
- La función de pérdida será la de entropía cruzada, ya que es capaz de cuantificar la diferencia entre dos distribuciones de probabilidad
- Los hiper parámetros β_1, β_2 y ϵ serán fijados en 0.9, 0.99 y 10^{-8} respectivamente (para la optimización Adam), tal y como se recomienda por Kingma et al. (2015)
- El factor de aprendizaje será fijado (inicialmente) en un valor lo suficientemente bajo de manera que garantice que la función de pérdida converja a su mínimo local y no lo saltee como se aprecia en la figura 48, por ello se considera $\eta = 0.001$ como punto de partida

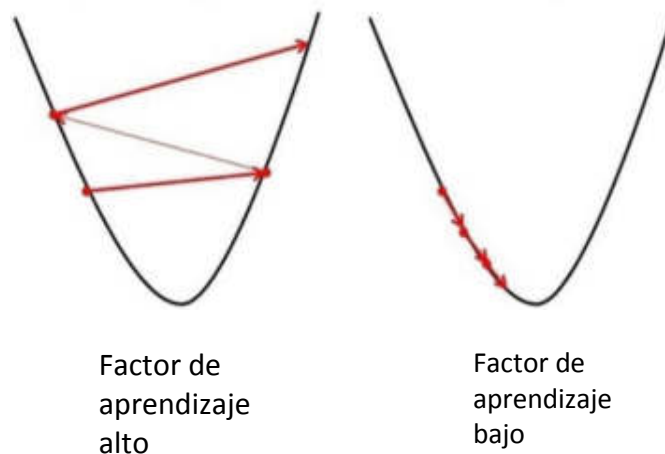


Figura (47). Factor de aprendizaje alto (izquierda) genera divergencia, mientras que un factor de aprendizaje bajo (derecha) permite la convergencia de la función de coste hacia el punto mínimo global

Fuente: <https://towardsdatascience.com/gradient-descent-in-a-nutshell-eaf8c18212f0>

Teniendo en cuenta todo lo anterior, se probaron las siguientes arquitecturas de red que fueron implementadas en *Python*:

➤ Secuencia de Modelo 1:

- 1 capa de convolución (32 filtros de convolución de tamaño 3x3)
- 1 capa de reducción (*Max-Pooling* de tamaño 2x2)
- 1 capa totalmente conectada con 128 neuronas
- 80 unidades en la capa de salida
- Número de épocas = 20

El modelo 1, tras 20 épocas de entrenamiento produce las curvas de la figura 48:

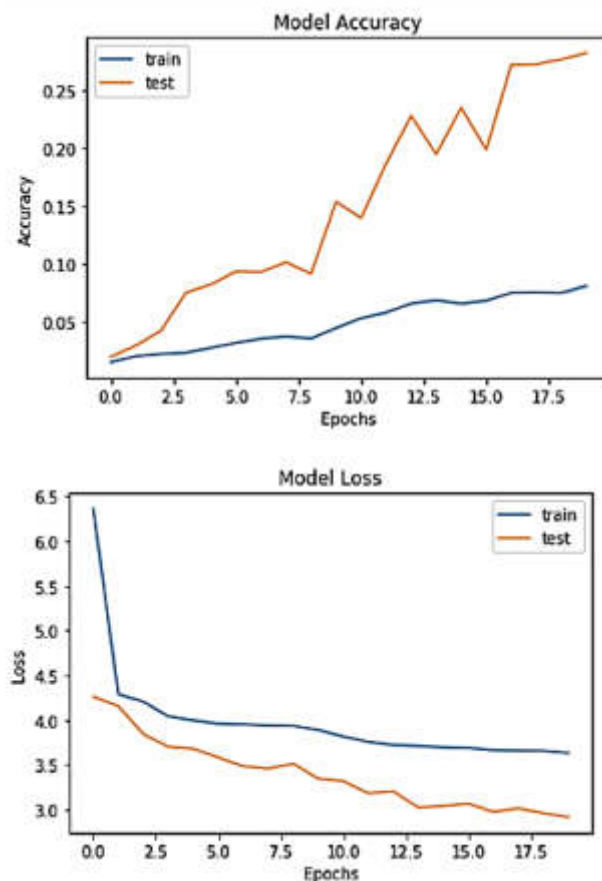


Figura (48). Curvas de precisión (parte superior) y curvas de la función de pérdida (parte inferior) correspondientes al Modelo 1

Elaboración: los autores

Obsérvese en la figura 49 que el modelo contiene más de 7 millones de parámetros y que no converge en ninguna de sus curvas, es decir no es capaz de clasificar adecuadamente a 80 sujetos bajo esa arquitectura y condiciones iniciales por lo que queda descartado.

Layer (type)	Output Shape	Param #
conv2d_26 (Conv2D)	(None, 88, 88, 32)	896
max_pooling2d_22 (MaxPooling)	(None, 44, 44, 32)	0
flatten_9 (Flatten)	(None, 56320)	0
dense_17 (Dense)	(None, 128)	7209088
dropout_9 (Dropout)	(None, 128)	0
dense_18 (Dense)	(None, 83)	10707
Total params: 7,220,691		
Trainable params: 7,220,691		
Non-trainable params: 0		

Figura (49). Resumen de la arquitectura del modelo neural 1

Elaboración: los autores

➤ Secuencia de Modelo 2:

- 1 capa de convolución (32 filtros de convolución de tamaño 3x3)
- 1 capa de reducción (*Max-Pooling* de tamaño 2x2)
- 1 capa de convolución (64 filtros de convolución de tamaño 3x3)
- 1 capa de reducción (*Max-Pooling* de tamaño 2x2)
- 1 capa totalmente conectada con 128 neuronas
- Número de épocas = 20

El modelo 2, tras 20 épocas de entrenamiento produce las curvas de la figura 50:

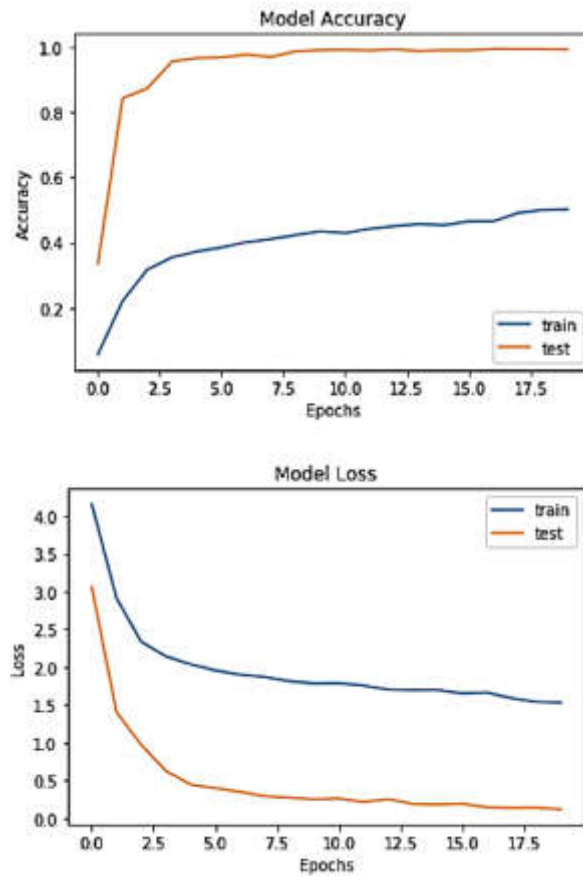


Figura (50). Curvas de precisión (parte superior) y curvas de la función de pérdida (parte inferior) correspondientes al Modelo 2

Elaboración: los autores

Obsérvese que las curvas de entrenamiento y de validación tanto para la pérdida como para la precisión presentan grandes diferencias entre sí por lo que puede ser una señal de *overfitting*, con lo que dicha arquitectura queda descartada a pesar de tener menos parámetros que el modelo 1 (más de 3 millones como se aprecia en la figura 51).

Layer (type)	Output Shape	Param #
conv2d_27 (Conv2D)	(None, 88, 88, 32)	896
max_pooling2d_23 (MaxPooling)	(None, 44, 44, 32)	0
conv2d_28 (Conv2D)	(None, 42, 38, 64)	18496
max_pooling2d_24 (MaxPooling)	(None, 21, 19, 64)	0
flatten_10 (Flatten)	(None, 25536)	0
dense_19 (Dense)	(None, 128)	3268736
dropout_10 (Dropout)	(None, 128)	0
dense_20 (Dense)	(None, 83)	10707

Total params: 3,298,835
 Trainable params: 3,298,835
 Non-trainable params: 0

Figura (51). Resumen de la arquitectura del modelo neural 2

Elaboración: los autores

➤ Secuencia de Modelo 3:

- 1 capa de convolución (32 filtros de convolución de tamaño 3x3)
- 1 capa de convolución (64 filtros de convolución de tamaño 3x3)
- 1 capa de reducción (*Max-Pooling* de tamaño 2x2)
- 1 capa totalmente conectada con 128 neuronas
- Número de épocas = 20

El modelo 3, tras 20 épocas de entrenamiento produce las curvas de la figura 52:

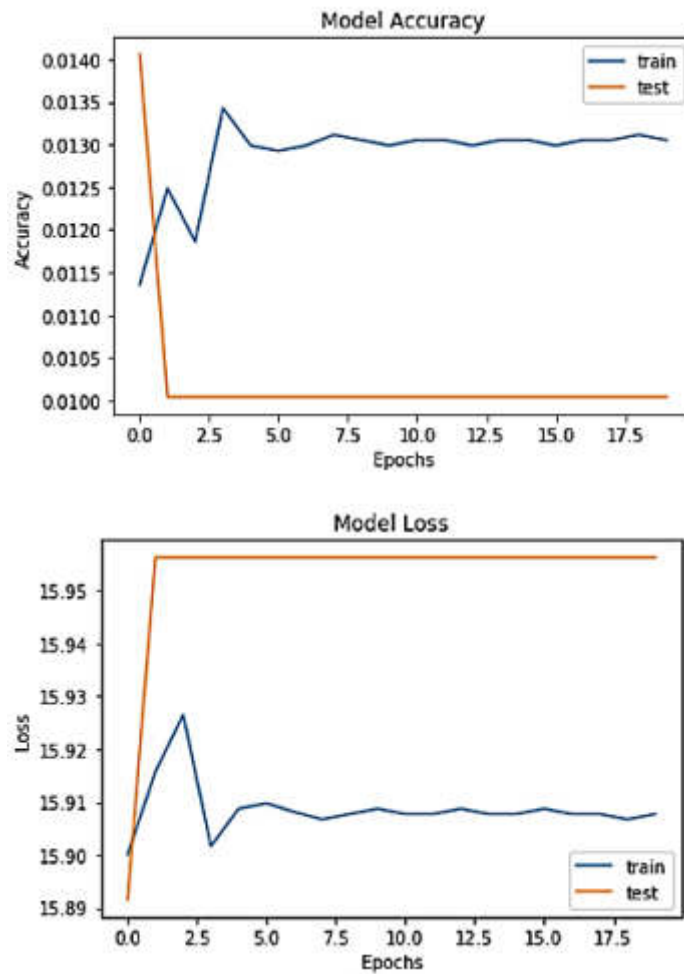


Figura (52). Curvas de precisión (parte superior) y curvas de la función de pérdida (parte inferior) correspondientes al Modelo 3

Elaboración: los autores

Tal y como se aprecia, las curvas correspondientes a este modelo no son capaces de converger, lo que quiere decir que dicha arquitectura no es apta para clasificar a 80 individuos. Este modelo, además, tiene más de 13 millones de parámetros tal y como se evidencia en la figura 53:

Layer (type)	Output Shape	Param #
conv2d_29 (Conv2D)	(None, 88, 88, 32)	896
conv2d_30 (Conv2D)	(None, 86, 78, 64)	18496
max_pooling2d_25 (MaxPooling)	(None, 43, 39, 64)	0
flatten_11 (Flatten)	(None, 107328)	0
dense_21 (Dense)	(None, 128)	13738112
dropout_11 (Dropout)	(None, 128)	0
dense_22 (Dense)	(None, 83)	10707
Total params: 13,768,211		
Trainable params: 13,768,211		
Non-trainable params: 0		

Figura (53). Resumen de la arquitectura del modelo neural 3

Elaboración: los autores

➤ Secuencia de Modelo 4:

- 1 capa de convolución (32 filtros de convolución de tamaño 3x3)
- 1 capa de reducción (*Max-Pooling* de tamaño 2x2)
- 1 capa de convolución (64 filtros de convolución de tamaño 3x3)
- 1 capa de reducción (*Max-Pooling* de tamaño 2x2)
- 1 capa de convolución (128 filtros de convolución de tamaño 3x3)
- 1 capa de reducción (*Max-Pooling* de tamaño 2x2)
- 1 capa totalmente conectada con 128 neuronas
- Número de épocas = 20

El modelo 4, tras 20 épocas de entrenamiento produce las curvas de la figura 54:

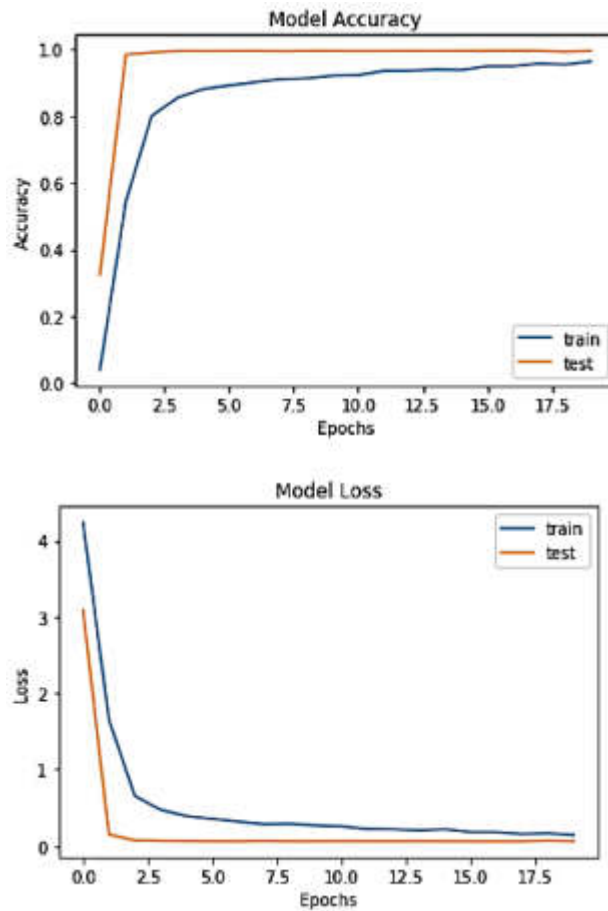


Figura (54). Curvas de precisión (parte superior) y curvas de la función de pérdida convergiendo a cero (parte inferior) correspondientes al Modelo 4

Elaboración: los autores

Se puede apreciar en la figura 54 que las curvas tienen tendencia a converger, por lo que la hace una candidata a ser elegida. Este modelo cuenta con más de 1 millón de parámetros tal y como se observa en la figura 55:

Layer (type)	Output Shape	Param #
conv2d_31 (Conv2D)	(None, 88, 80, 32)	896
max_pooling2d_26 (MaxPooling)	(None, 44, 40, 32)	0
conv2d_32 (Conv2D)	(None, 42, 38, 64)	18496
max_pooling2d_27 (MaxPooling)	(None, 21, 19, 64)	0
conv2d_33 (Conv2D)	(None, 19, 17, 128)	73856
max_pooling2d_28 (MaxPooling)	(None, 9, 8, 128)	0
flatten_12 (Flatten)	(None, 9216)	0
dense_23 (Dense)	(None, 128)	1179776
dropout_12 (Dropout)	(None, 128)	0
dense_24 (Dense)	(None, 83)	10707
Total params: 1,283,731		
Trainable params: 1,283,731		
Non-trainable params: 0		

Figura (55). Resumen de la arquitectura del modelo neural 4

Elaboración: los autores

➤ Secuencia de Modelo 5:

- 1 capa de convolución (32 filtros de convolución de tamaño 5x5)
- 1 capa de convolución (64 filtros de convolución de tamaño 3x3)
- 1 capa de convolución (128 filtros de convolución de tamaño 3x3)
- 1 capa de reducción (*Max-Pooling* de tamaño 2x2)
- 1 capa totalmente conectada con 256 neuronas
- Número de épocas = 20

El modelo 5, tras 20 épocas de entrenamiento produce las curvas de la figura 56:

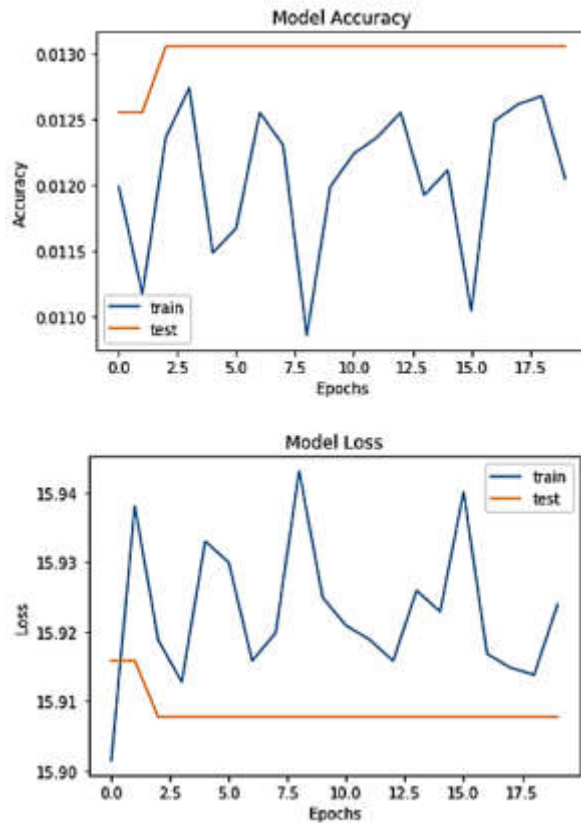


Figura (56). Curvas de precisión (parte superior) y curvas de la función de pérdida (parte inferior) correspondientes al Modelo 5

Elaboración: los autores

Nótese que para el modelo 5 ninguna de sus curvas converge ni tiene tendencia a converger, por lo que queda descartada.

➤ Secuencia de Modelo 6:

- 1 capa de convolución (32 filtros de convolución de tamaño 5x5)
- 1 capa de convolución (64 filtros de convolución de tamaño 3x3)
- 1 capa de reducción (*Max-Pooling* de tamaño 2x2)
- 1 capa de convolución (64 filtros de convolución de tamaño 3x3)
- 1 capa de convolución (128 filtros de convolución de tamaño 3x3)
- 1 capa de reducción (*Max-Pooling* de tamaño 2x2)

- 1 capa totalmente conectada con 256 neuronas
- Número de épocas = 20

El modelo 6, tras 20 épocas de entrenamiento produce las curvas de la figura 57:

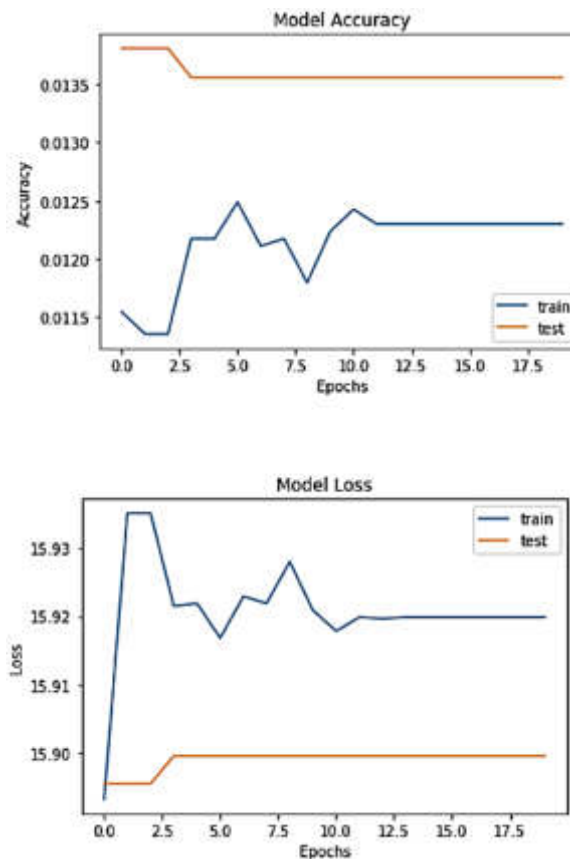


Figura (57). Curvas de precisión (parte superior) y curvas de la función de pérdida (parte inferior) correspondientes al Modelo 6

Elaboración: los autores

Nótese que para el modelo 6 ninguna de sus curvas converge ni tiene tendencia a converger, por lo que queda descartada.

➤ Secuencia de Modelo 7:

- 1 capa de convolución (32 filtros de convolución de tamaño 3x3)
- 1 capa de reducción (*Max-Pooling* de tamaño 2x2)

- 1 capa de convolución (64 filtros de convolución de tamaño 3x3)
- 1 capa de reducción (*Max-Pooling* de tamaño 2x2)
- 1 capa de convolución (128 filtros de convolución de tamaño 3x3)
- 1 capa de reducción (*Max-Pooling* de tamaño 2x2)
- 1 capa totalmente conectada con 64 neuronas
- Número de épocas = 20

El modelo 7, tras 20 épocas de entrenamiento produce las curvas de la figura 58:

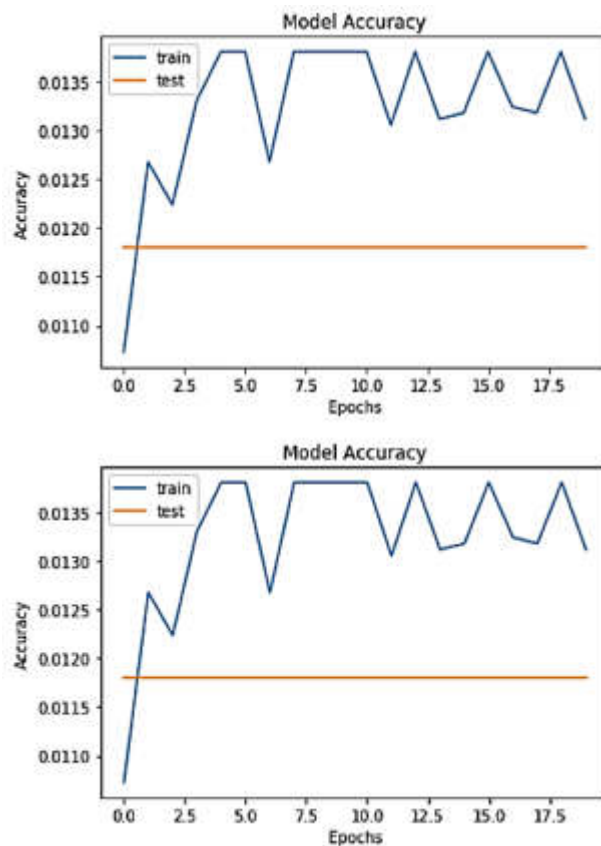


Figura (58). Curvas de precisión (parte superior) y curvas de la función de pérdida (parte inferior) correspondientes al Modelo 7

Elaboración: los autores

Nótese que para el modelo 7 ninguna de sus curvas converge ni tiene tendencia a converger, por lo que queda descartada.

➤ Secuencia de Modelo 8:

- 1 capa de convolución (32 filtros de convolución de tamaño 7x7)
- 1 capa de reducción (*Max-Pooling* de tamaño 2x2)
- 1 capa de convolución (64 filtros de convolución de tamaño 5x5)
- 1 capa de reducción (*Max-Pooling* de tamaño 2x2)
- 1 capa de convolución (128 filtros de convolución de tamaño 3x3)
- 1 capa de reducción (*Max-Pooling* de tamaño 2x2)
- 1 capa totalmente conectada con 64 neuronas
- Número de épocas = 20

El modelo 8, tras 20 épocas de entrenamiento produce las curvas de la figura 59:

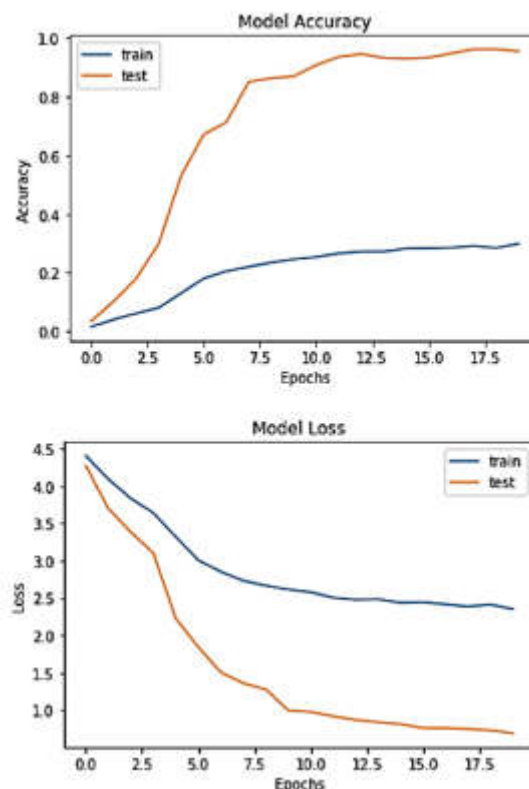


Figura (59). Curvas de precisión (parte superior) y curvas de la función de pérdida (parte inferior) correspondientes al Modelo 8

Elaboración: los autores

Las curvas de la figura 59 tienen tendencia baja a converger, aunque dicha tendencia no es superior a la del modelo 4 por lo que también queda descartada.

➤ Secuencia de Modelo 9:

- 1 capa de convolución (64 filtros de convolución de tamaño 7x7)
- 1 capa de reducción (*Max-Pooling* de tamaño 2x2)
- 1 capa de convolución (128 filtros de convolución de tamaño 5x5)
- 1 capa de reducción (*Max-Pooling* de tamaño 2x2)
- 1 capa de convolución (256 filtros de convolución de tamaño 3x3)
- 1 capa de reducción (*Max-Pooling* de tamaño 2x2)
- 1 capa totalmente conectada con 64 neuronas
- Número de épocas = 20

El modelo 9, tras 20 épocas de entrenamiento produce las curvas de la figura 60:

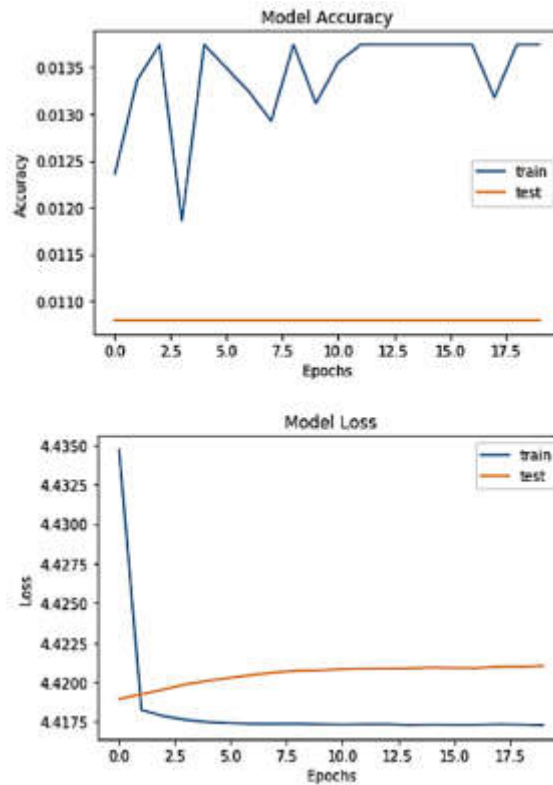


Figura (60). Curvas de precisión (parte superior) y curvas de la función de pérdida (parte inferior) correspondientes al Modelo 9

Elaboración: los autores

Nótese que para el modelo 9 ninguna de sus curvas converge ni tiene tendencia a converger, por lo que queda descartada.

➤ Secuencia de Modelo 10:

- 1 capa de convolución (32 filtros de convolución de tamaño 11x11)
- 1 capa de reducción (*Max-Pooling* de tamaño 2x2)
- 1 capa de convolución (64 filtros de convolución de tamaño 3x3)
- 1 capa de reducción (*Max-Pooling* de tamaño 2x2)
- 1 capa de convolución (128 filtros de convolución de tamaño 3x3)
- 1 capa de reducción (*Max-Pooling* de tamaño 2x2)
- 1 capa totalmente conectada con 128 neuronas

- Número de épocas = 20

El modelo 10, tras 20 épocas de entrenamiento produce las curvas de la figura 61

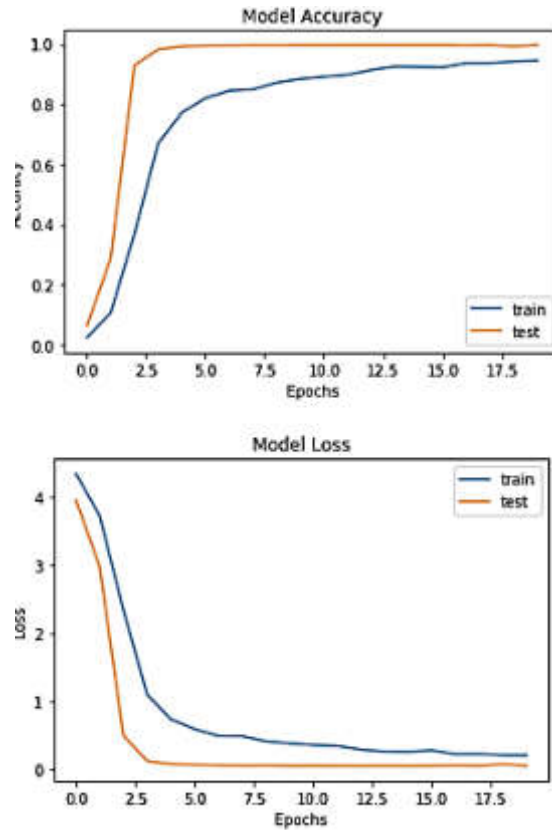


Figura (61). Curvas de precisión (parte superior) y curvas de la función de pérdida convergiendo a cero (parte inferior) correspondientes al Modelo 10

Elaboración: los autores

Se puede apreciar que las curvas del modelo 10 tienen tendencia a la convergencia y muestran un comportamiento similar a las curvas del modelo 4, aunque no logran superarlo (las curvas del modelo 4 convergen un poco antes) por lo que esta red también queda descartada.

- Secuencia de Modelo 11:
 - 1 capa de convolución (32 filtros de convolución de tamaño 11x11)
 - 1 capa de reducción (*Max-Pooling* de tamaño 2x2)

- 1 capa de convolución (64 filtros de convolución de tamaño 9x9).
- 1 capa de reducción (*Max-Pooling* de tamaño 2x2)
- 1 capa de convolución (128 filtros de convolución de tamaño 3x3)
- 1 capa de reducción (*Max-Pooling* de tamaño 2x2)
- 1 capa totalmente conectada con 128 neuronas
- Número de épocas = 20

El modelo 11, tras 20 épocas de entrenamiento produce las curvas de la figura 62:

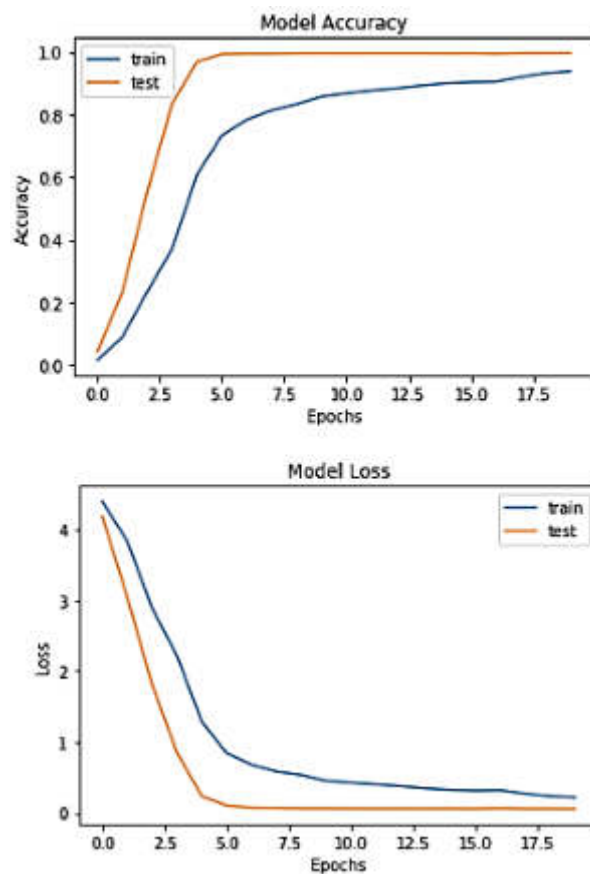


Figura (62). Curvas de precisión (parte superior) y curvas de la función de pérdida convergiendo a cero (parte inferior) correspondientes al Modelo 11

Elaboración: los autores

Se puede apreciar que las curvas del modelo 11 tienen tendencia a converger, pero no logran superar el desempeño de las curvas de los modelos 4 y 10, por lo que también queda descartada.

➤ Secuencia de Modelo 12:

- 1 capa de convolución (32 filtros de convolución de tamaño 13x13)
- 1 capa de reducción (*Max-Pooling* de tamaño 2x2)
- 1 capa de convolución (64 filtros de convolución de tamaño 7x7)
- 1 capa de reducción (*Max-Pooling* de tamaño 2x2)
- 1 capa totalmente conectada con 128 neuronas
- Número de épocas = 20

El modelo 12, tras 20 épocas de entrenamiento produce las curvas de la figura 63:

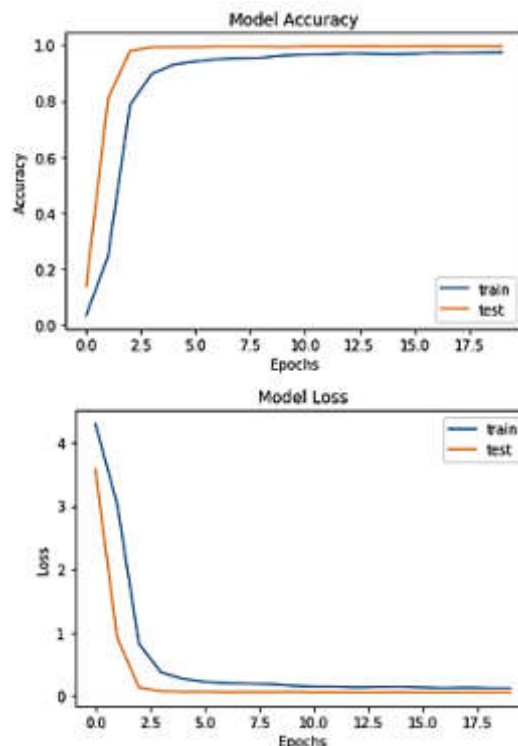


Figura (63). Curvas de precisión (parte superior) y curvas de la función de pérdida convergiendo a cero (parte inferior) correspondientes al Modelo 12

Elaboración: los autores

Las curvas de la figura 63 correspondientes al modelo 12 tienen mejor capacidad de converger que cualquiera de las curvas anteriores. Este modelo consta de más de 2 millones de parámetros como se observa en la figura 64:

Layer (type)	Output Shape	Param #
conv2d_7 (Conv2D)	(None, 88, 80, 32)	16256
max_pooling2d_7 (MaxPooling2)	(None, 44, 40, 32)	0
conv2d_8 (Conv2D)	(None, 38, 34, 64)	100416
max_pooling2d_8 (MaxPooling2)	(None, 19, 17, 64)	0
flatten_3 (Flatten)	(None, 20672)	0
dense_5 (Dense)	(None, 128)	2646144
dropout_3 (Dropout)	(None, 128)	0
dense_6 (Dense)	(None, 83)	10707
Total params: 2,773,523		
Trainable params: 2,773,523		
Non-trainable params: 0		

Figura (64). Resumen de la arquitectura del modelo neural 12

Elaboración: los autores

Dicho modelo es, además, capaz de completar una iteración entre 16 y 17 segundos aproximadamente tal y como se puede apreciar en la figura

```

epoch 18/20
15936/15936 [-----] - 15s 943us/step - loss: 0.1338 - acc: 0.9713 - val_loss: 0.0555 - val_acc: 0.9957
Epoch 19/20
15936/15936 [-----] - 16s 980us/step - loss: 0.1267 - acc: 0.9730 - val_loss: 0.0573 - val_acc: 0.9960
Epoch 20/20
15936/15936 [-----] - 15s 934us/step - loss: 0.1243 - acc: 0.9741 - val_loss: 0.0558 - val_acc: 0.9957

```

65:

Figura (65). Tiempo que le toma a la arquitectura 12 para completar una iteración

Elaboración: los autores

- Secuencia de Modelo 13:
 - 1 capa de convolución (32 filtros de convolución de tamaño 13x13)

- 1 capa de convolución (64 filtros de convolución de tamaño 7x7)
- 1 capa de reducción (*Max-Pooling* de tamaño 2x2)
- 1 capa de convolución (128 filtros de convolución de tamaño 3x3)
- 1 capa de reducción (*Max-Pooling* de tamaño 2x2)
- 1 capa totalmente conectada con 128 neuronas
- Número de épocas = 20

El modelo 13, tras 20 épocas de entrenamiento produce las curvas de la figura 66:

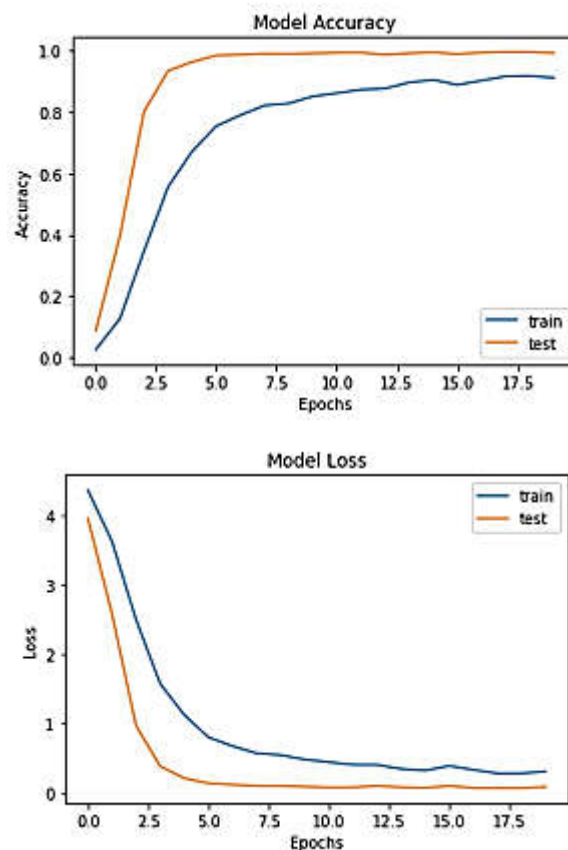


Figura (66). Curvas de precisión (parte superior) y curvas de la función de pérdida convergiendo a cero (parte inferior) correspondientes al Modelo 13

Elaboración: los autores

Las curvas del modelo 13 tienen tendencia a converger, pero su desempeño es inferior a las curvas del modelo 12 por lo que la arquitectura queda descartada.

➤ Secuencia de Modelo 14:

- 1 capa de convolución (32 filtros de convolución de tamaño 15x15)
- 1 capa de reducción (*Max-Pooling* de tamaño 2x2)
- 1 capa de convolución (64 filtros de convolución de tamaño 5x5)
- 1 capa de reducción (*Max-Pooling* de tamaño 2x2)
- 1 capa de convolución (128 filtros de convolución de tamaño 3x3)
- 1 capa de reducción (*Max-Pooling* de tamaño 2x2)
- 1 capa totalmente conectada con 128 neuronas
- Número de épocas = 20

El modelo 14, tras 20 épocas de entrenamiento produce las curvas de la figura 67:

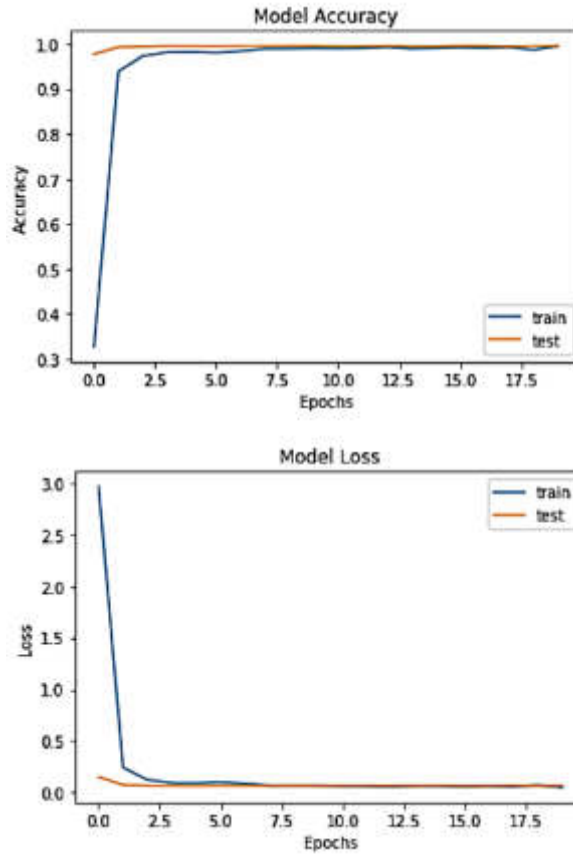


Figura (67). Curvas de precisión (parte superior) y curvas de la función de pérdida convergiendo a cero (parte inferior) correspondientes al Modelo 14
Elaboración: los autores

Las curvas del modelo 14 son capaces de converger en 20 iteraciones, mostrando un desempeño superior a cualquiera de las curvas de los otros modelos, por lo que la red es elegible. Este modelo tiene casi 5 millones de parámetros y logra completar una iteración en poco más de 1 segundos tal y como se aprecia en las figuras 68 y 69:

```
15936/15936 [=====] - 17s 1ms/step - loss: 0.0589 - acc: 0.9938 - val_loss: 0.0686 - val_acc: 0.9945
Epoch 19/20
15936/15936 [=====] - 17s 1ms/step - loss: 0.0748 - acc: 0.9873 - val_loss: 0.0651 - val_acc: 0.9957
Epoch 20/20
15936/15936 [=====] - 17s 1ms/step - loss: 0.0497 - acc: 0.9952 - val_loss: 0.0689 - val_acc: 0.9957
```

Figura (68). Tiempo que le toma a la arquitectura 14 para completar una iteración

Elaboración: los autores

Layer (type)	Output Shape	Param #
conv2d_23 (Conv2D)	(None, 88, 80, 32)	21632
max_pooling2d_19 (MaxPooling)	(None, 44, 40, 32)	0
conv2d_24 (Conv2D)	(None, 40, 36, 64)	51264
max_pooling2d_20 (MaxPooling)	(None, 20, 18, 64)	0
conv2d_25 (Conv2D)	(None, 18, 16, 128)	73856
max_pooling2d_21 (MaxPooling)	(None, 9, 8, 128)	0
flatten_8 (Flatten)	(None, 9216)	0
dense_15 (Dense)	(None, 512)	4719104
dropout_8 (Dropout)	(None, 512)	0
dense_16 (Dense)	(None, 83)	42579

Total params: 4,908,435		
Trainable params: 4,908,435		
Non-trainable params: 0		

Figura (69). Número de parámetros para la arquitectura del modelo 14

Elaboración: los autores

4.5 Diseño de la interfaz gráfica (GUI)

Una vez completadas las partes de preprocesamiento (donde se aplicó PDI) y la etapa de procesamiento (donde se aplicó el concepto de redes neurales convolucionales) e procese a implementar la etapa de post-procesamiento. Esta etapa consiste en diseñar una interfaz gráfica que le permita al usuario manipular el sistema de manera que pueda registrar su asistencia.

Para desarrollar la interfaz gráfica se siguieron las especificaciones de diseño siguientes:

- La interfaz será diseñada utilizando código de programación *Python*.
- La GUI deberá mostrar un botón de captura de imagen en una ventana principal donde también se pueda apreciar la fecha y hora transcurriendo en tiempo real.

- La GUI deberá ser capaz de tomar la muestra palmar recién adquirida e ingresarla a la red neuronal convolucional para su clasificación.
- La GUI deberá mostrar la muestra palmar de interés del usuario junto con la predicción que haga la red neuronal convolucional en una ventana secundaria.
- La GUI deberá mostrar la fecha y día en que se realizó la predicción en pantalla.
- Una vez hecha la predicción por parte de la red neuronal, la ventana se deberá cerrar tras unos 5 segundos.
- La interfaz deberá contar con una capa de seguridad de manera que personas ajenas a la base de datos manipulen el sistema y se eviten así falsos positivos.

Observar el diagrama de flujo de la figura 70 para tener una idea del procedimiento descrito líneas arriba:

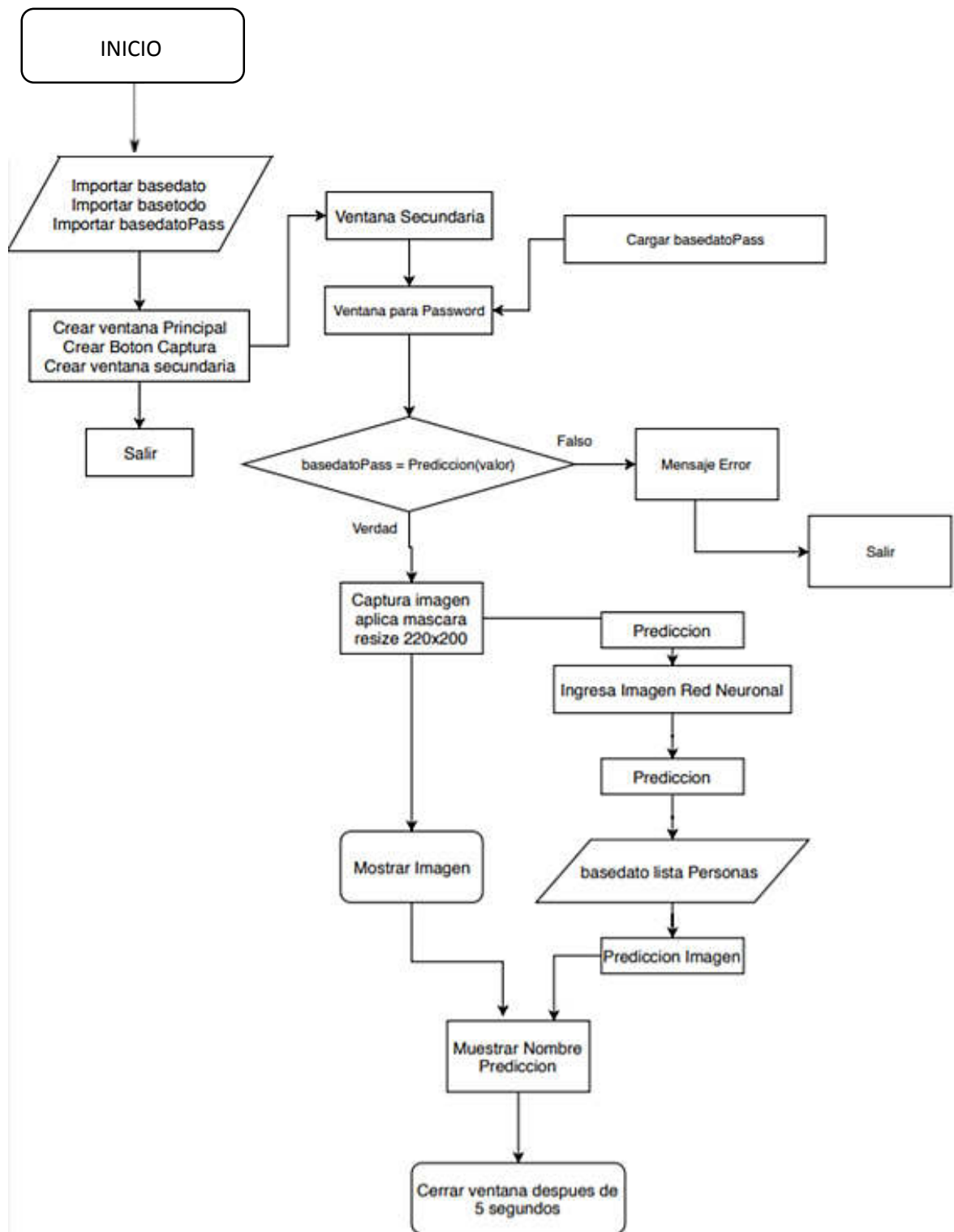


Figura (70). Diagrama de flujo de la GUI del presente trabajo.

Fuente: Propia.

Elaboración: los autores

La capa de seguridad consiste en que cada usuario de la base de datos tenga una contraseña que le permita validar la predicción hecha por la red una vez que dicho usuario haya pasado su mano por la plataforma palmar, esto se puede apreciar en la figura 71:

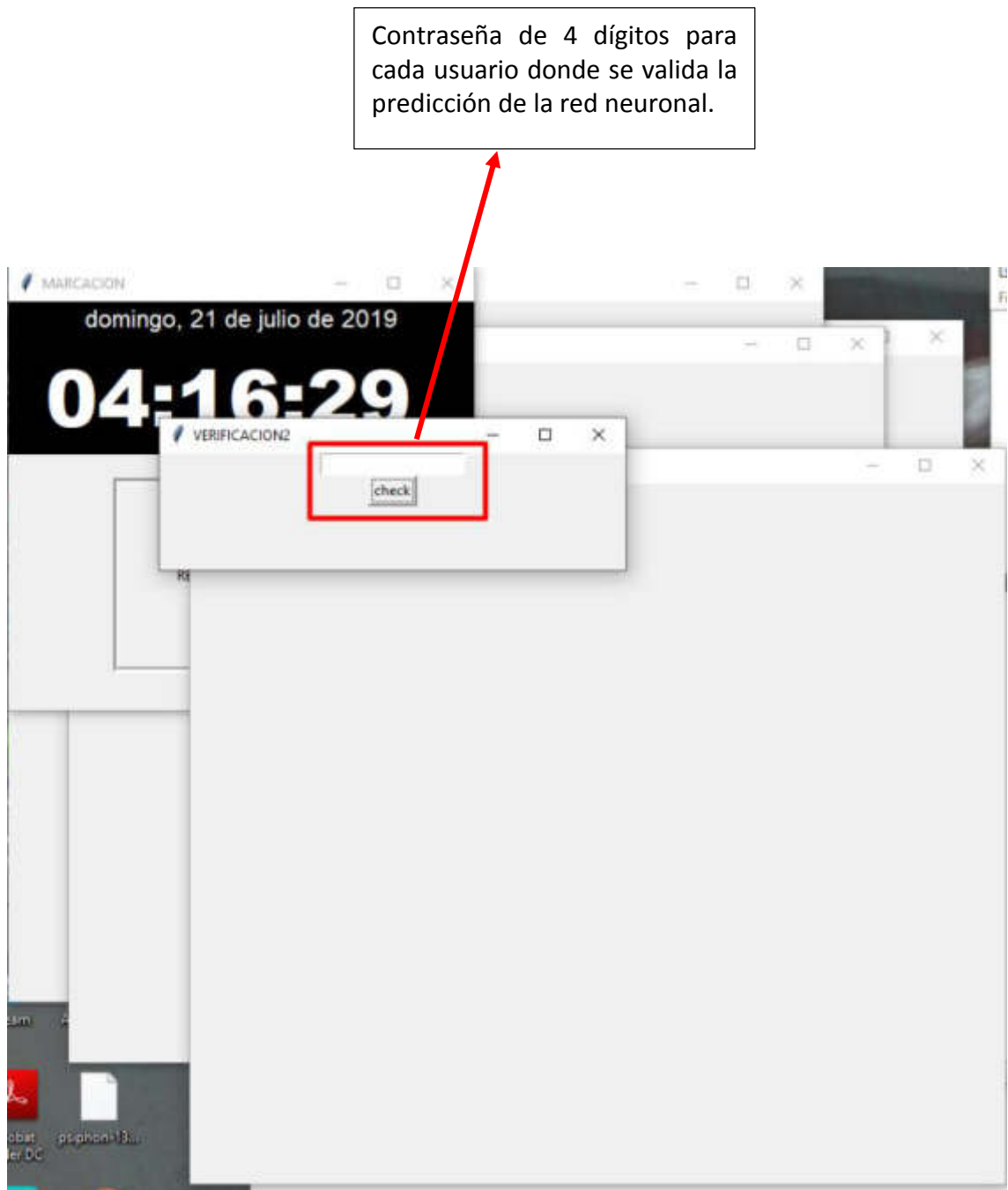


Figura (71). Representación gráfica de la capa de seguridad implementada en la GUI para validar el registro de asistencia de un usuario

Elaboración: los autores

La predicción de la red se puede apreciar en la figura 72:

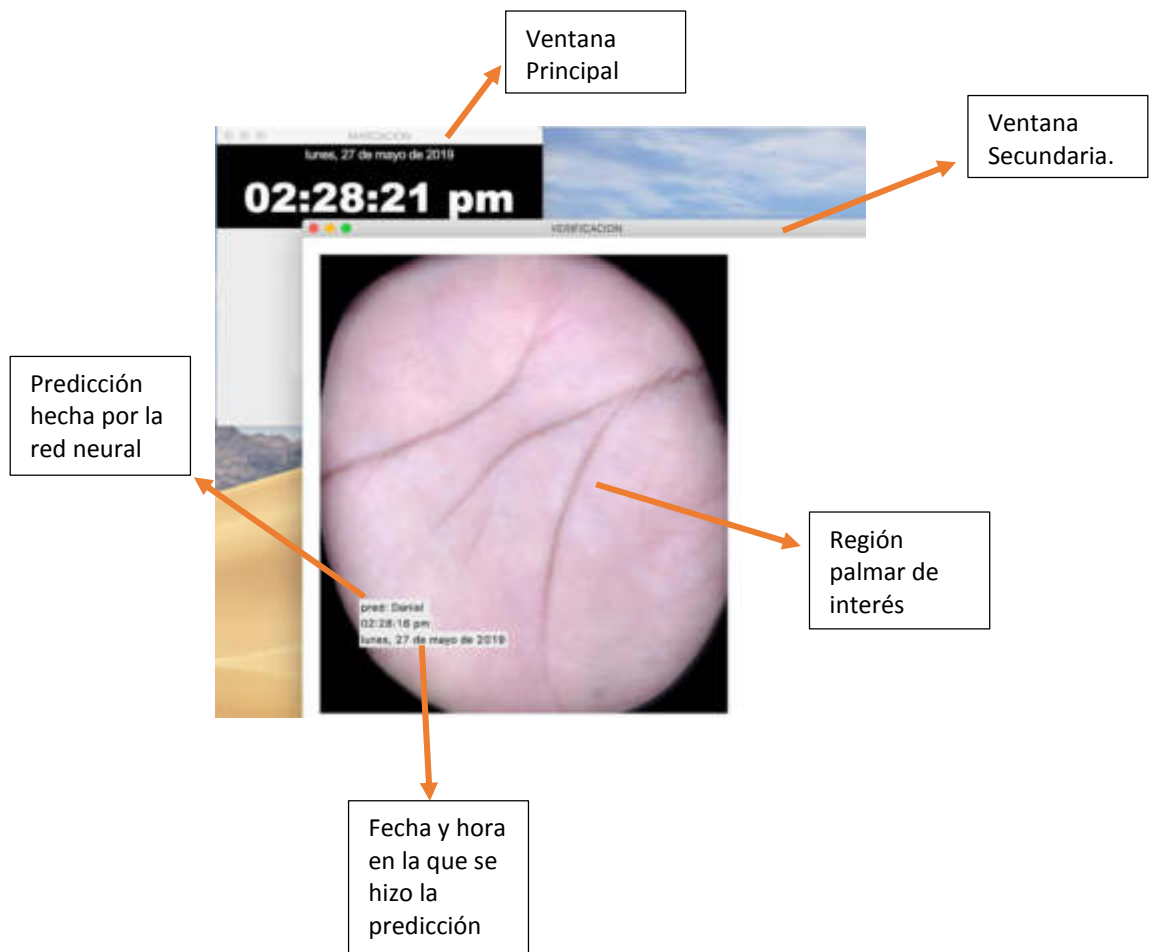


Figura (72). Representación gráfica de la GUI y su predicción, siendo utilizada por un usuario
Elaboración: los autores

Limitaciones:

- La aplicación solo puede lidiar con una mano a la vez
- La contraseña de cada usuario para la capa de seguridad solo cuenta con 4 dígitos
- Antes de volver a tomar un nuevo registro se debe esperar a que la ventana secundaria se cierre (tiempo de cierre es 5 segundos)

CAPÍTULO V. PRUEBAS Y RESULTADOS

5.1 Pruebas

Se emplearon 30 muestras palmares de cada usuario que no fueron entrenadas por la red para realizar las pruebas y validaciones, dichas muestras fueron ingresadas a la red elegida (entrenada con 20 épocas). Las pruebas se muestran en las tablas 7 a 89, **todas las tablas fueron elaboradas por los autores.**

Nombre del Individuo	José Dueñas			
	N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
	1	Acierto	16	Acierto
	2	Acierto	17	Acierto
	3	Acierto	18	Fallo
	4	Acierto	19	Acierto
	5	Acierto	20	Acierto
	6	Acierto	21	Acierto
	7	Acierto	22	Acierto
	8	Acierto	23	Acierto
	9	Acierto	24	Acierto
	10	Acierto	25	Acierto
	11	Acierto	26	Acierto
	12	Acierto	27	Acierto
	13	Acierto	28	Acierto
	14	Acierto	29	Acierto
	15	Fallo	30	Acierto

Tabla (7) Resultados de las pruebas para el usuario José Dueñas

Nombre del Individuo	John Lopez		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Acierto
2	Acierto	17	Acierto
3	Acierto	18	Acierto
4	Acierto	19	Acierto
5	Acierto	20	Acierto
6	Fallo	21	Acierto
7	Acierto	22	Acierto
8	Acierto	23	Acierto
9	Acierto	24	Acierto
10	Acierto	25	Acierto
11	Acierto	26	Acierto
12	Acierto	27	Acierto
13	Acierto	28	Acierto
14	Acierto	29	Acierto
15	Acierto	30	Acierto

Tabla (8) Resultados de las pruebas para el usuario John López

Nombre del Individuo	Abigail Alvarez		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Acierto
2	Acierto	17	Acierto
3	Acierto	18	Acierto
4	Acierto	19	Acierto
5	Acierto	20	Acierto
6	Acierto	21	Acierto
7	Acierto	22	Acierto
8	Fallo	23	Acierto
9	Acierto	24	Acierto
10	Acierto	25	Acierto
11	Acierto	26	Fallo
12	Acierto	27	Acierto
13	Acierto	28	Acierto
14	Acierto	29	Acierto
15	Acierto	30	Acierto

Tabla (9) Resultados de las pruebas para el usuario Abigail Álvarez

Nombre del Individuo	Alejandro Rios		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Acierto
2	Acierto	17	Acierto
3	Acierto	18	Acierto
4	Acierto	19	Acierto
5	Acierto	20	Acierto
6	Acierto	21	Acierto
7	Acierto	22	Acierto
8	Acierto	23	Acierto
9	Acierto	24	Acierto
10	Acierto	25	Acierto
11	Acierto	26	Acierto
12	Acierto	27	Acierto
13	Acierto	28	Acierto
14	Acierto	29	Acierto
15	Acierto	30	Acierto

Tabla (10) Resultados de las pruebas para el usuario Alejandro Ríos

Nombre del Individuo	Alex Moreno		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Fallo
2	Acierto	17	Acierto
3	Acierto	18	Acierto
4	Acierto	19	Acierto
5	Acierto	20	Fallo
6	Acierto	21	Acierto
7	Acierto	22	Acierto
8	Acierto	23	Acierto
9	Acierto	24	Acierto
10	Acierto	25	Acierto
11	Acierto	26	Acierto
12	Acierto	27	Acierto
13	Acierto	28	Acierto
14	Acierto	29	Acierto
15	Acierto	30	Acierto

Tabla (11) Resultados de las pruebas para el usuario Alex Moreno

Nombre del Individuo	Angie Villalba		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Acierto
2	Acierto	17	Acierto
3	Acierto	18	Acierto
4	Fallo	19	Acierto
5	Acierto	20	Acierto
6	Acierto	21	Acierto
7	Acierto	22	Acierto
8	Acierto	23	Acierto
9	Acierto	24	Acierto
10	Acierto	25	Acierto
11	Acierto	26	Acierto
12	Acierto	27	Acierto
13	Fallo	28	Acierto
14	Acierto	29	Acierto
15	Acierto	30	Acierto

Tabla (12) Resultados de las pruebas para el usuario Angie Villalba

Nombre del Individuo	Antony Salas		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Acierto
2	Acierto	17	Acierto
3	Acierto	18	Acierto
4	Acierto	19	Acierto
5	Acierto	20	Acierto
6	Acierto	21	Acierto
7	Acierto	22	Acierto
8	Acierto	23	Acierto
9	Acierto	24	Acierto
10	Acierto	25	Acierto
11	Fallo	26	Acierto
12	Acierto	27	Acierto
13	Acierto	28	Acierto
14	Acierto	29	Acierto
15	Acierto	30	Acierto

Tabla (13) Resultados de las pruebas para el usuario Anthony Salas

Nombre del Individuo	Christian Mendoza		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Acierto
2	Fallo	17	Acierto
3	Acierto	18	Acierto
4	Acierto	19	Acierto
5	Acierto	20	Acierto
6	Acierto	21	Acierto
7	Acierto	22	Acierto
8	Acierto	23	Acierto
9	Fallo	24	Acierto
10	Acierto	25	Fallo
11	Acierto	26	Acierto
12	Acierto	27	Acierto
13	Acierto	28	Acierto
14	Acierto	29	Acierto
15	Acierto	30	Acierto

Tabla (14) Resultados de las pruebas para el usuario Christian Mendoza

Nombre del Individuo	Clarisa Penedo		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Acierto
2	Acierto	17	Acierto
3	Acierto	18	Acierto
4	Acierto	19	Acierto
5	Acierto	20	Acierto
6	Acierto	21	Acierto
7	Acierto	22	Fallo
8	Acierto	23	Acierto
9	Acierto	24	Acierto
10	Acierto	25	Acierto
11	Acierto	26	Acierto
12	Acierto	27	Acierto
13	Acierto	28	Acierto
14	Acierto	29	Acierto
15	Acierto	30	Acierto

Tabla (15) Resultados de las pruebas para el usuario Clarisa Penedo

Nombre del Individuo	Danitza Alcantara		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Acierto
2	Acierto	17	Fallo
3	Acierto	18	Acierto
4	Acierto	19	Acierto
5	Fallo	20	Acierto
6	Acierto	21	Acierto
7	Acierto	22	Acierto
8	Acierto	23	Acierto
9	Acierto	24	Fallo
10	Acierto	25	Acierto
11	Acierto	26	Acierto
12	Acierto	27	Acierto
13	Acierto	28	Acierto
14	Acierto	29	Acierto
15	Acierto	30	Acierto

Tabla (16) Resultados de las pruebas para el usuario Danitza Alcántara

Nombre del Individuo	Carmen Sanchez		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Acierto
2	Acierto	17	Acierto
3	Acierto	18	Acierto
4	Acierto	19	Acierto
5	Acierto	20	Acierto
6	Acierto	21	Acierto
7	Acierto	22	Acierto
8	Acierto	23	Acierto
9	Acierto	24	Acierto
10	Acierto	25	Acierto
11	Acierto	26	Acierto
12	Acierto	27	Acierto
13	Acierto	28	Acierto
14	Acierto	29	Acierto
15	Acierto	30	Acierto

Tabla (17) Resultados de las pruebas para el usuario Carmen Sánchez

Nombre del Individuo	Emanuel Salas		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Acierto
2	Acierto	17	Acierto
3	Acierto	18	Acierto
4	Acierto	19	Acierto
5	Acierto	20	Acierto
6	Acierto	21	Acierto
7	Acierto	22	Acierto
8	Acierto	23	Acierto
9	Acierto	24	Acierto
10	Acierto	25	Acierto
11	Acierto	26	Fallo
12	Acierto	27	Acierto
13	Acierto	28	Acierto
14	Acierto	29	Acierto
15	Acierto	30	Fallo

Tabla (18) Resultados de las pruebas para el usuario Emanuel Salas

Nombre del Individuo	Gaston Navas		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Acierto
2	Acierto	17	Fallo
3	Acierto	18	Acierto
4	Acierto	19	Acierto
5	Acierto	20	Acierto
6	Acierto	21	Acierto
7	Acierto	22	Acierto
8	Fallo	23	Acierto
9	Acierto	24	Acierto
10	Acierto	25	Acierto
11	Acierto	26	Acierto
12	Acierto	27	Acierto
13	Acierto	28	Acierto
14	Acierto	29	Acierto
15	Acierto	30	Acierto

Tabla (19) Resultados de las pruebas para el usuario Gastón Navas

Nombre del Individuo	Ines Mori		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Acierto
2	Acierto	17	Acierto
3	Acierto	18	Acierto
4	Acierto	19	Acierto
5	Acierto	20	Acierto
6	Acierto	21	Acierto
7	Acierto	22	Acierto
8	Acierto	23	Acierto
9	Acierto	24	Acierto
10	Acierto	25	Acierto
11	Acierto	26	Acierto
12	Acierto	27	Acierto
13	Acierto	28	Acierto
14	Acierto	29	Fallo
15	Acierto	30	Acierto

Tabla (20) Resultados de las pruebas para el usuario Inés Mori

Nombre del Individuo	Jose Carlos Roman		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Acierto
2	Acierto	17	Fallo
3	Acierto	18	Acierto
4	Acierto	19	Acierto
5	Fallo	20	Acierto
6	Fallo	21	Acierto
7	Acierto	22	Acierto
8	Acierto	23	Acierto
9	Acierto	24	Acierto
10	Acierto	25	Acierto
11	Acierto	26	Acierto
12	Acierto	27	Acierto
13	Acierto	28	Acierto
14	Acierto	29	Acierto
15	Acierto	30	Acierto

Tabla (21) Resultados de las pruebas para el usuario José Carlos Román

Nombre del Individuo	Juan Zapata		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Fallo	16	Acierto
2	Acierto	17	Acierto
3	Acierto	18	Acierto
4	Acierto	19	Acierto
5	Acierto	20	Acierto
6	Acierto	21	Acierto
7	Acierto	22	Acierto
8	Acierto	23	Acierto
9	Acierto	24	Acierto
10	Acierto	25	Acierto
11	Acierto	26	Acierto
12	Acierto	27	Acierto
13	Acierto	28	Fallo
14	Fallo	29	Acierto
15	Acierto	30	Acierto

Tabla (22) Resultados de las pruebas para el usuario Juan Zapata

Nombre del Individuo	Luis Cornejo		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Acierto
2	Acierto	17	Acierto
3	Acierto	18	Acierto
4	Acierto	19	Acierto
5	Acierto	20	Acierto
6	Acierto	21	Acierto
7	Acierto	22	Acierto
8	Acierto	23	Fallo
9	Fallo	24	Acierto
10	Acierto	25	Acierto
11	Acierto	26	Fallo
12	Acierto	27	Acierto
13	Acierto	28	Acierto
14	Acierto	29	Acierto
15	Acierto	30	Acierto

Tabla (23) Resultados de las pruebas para el usuario Luis Cornejo

Nombre del Individuo	Maria Bandera		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Acierto
2	Acierto	17	Acierto
3	Acierto	18	Acierto
4	Fallo	19	Acierto
5	Acierto	20	Acierto
6	Acierto	21	Acierto
7	Acierto	22	Acierto
8	Acierto	23	Acierto
9	Acierto	24	Acierto
10	Acierto	25	Acierto
11	Acierto	26	Acierto
12	Acierto	27	Acierto
13	Acierto	28	Acierto
14	Acierto	29	Acierto
15	Acierto	30	Acierto

Tabla (24) Resultados de las pruebas para el usuario María Bandera

Nombre del Individuo	Milagrito Cabezas		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Acierto
2	Fallo	17	Acierto
3	Acierto	18	Acierto
4	Acierto	19	Acierto
5	Acierto	20	Acierto
6	Acierto	21	Acierto
7	Acierto	22	Acierto
8	Fallo	23	Acierto
9	Acierto	24	Acierto
10	Acierto	25	Acierto
11	Acierto	26	Acierto
12	Acierto	27	Acierto
13	Acierto	28	Fallo
14	Acierto	29	Acierto
15	Acierto	30	Acierto

Tabla (25) Resultados de las pruebas para el usuario Milagrito Cabezas

Nombre del Individuo	Patricia Villar		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Acierto
2	Acierto	17	Acierto
3	Acierto	18	Acierto
4	Acierto	19	Acierto
5	Fallo	20	Acierto
6	Acierto	21	Acierto
7	Acierto	22	Acierto
8	Acierto	23	Acierto
9	Acierto	24	Acierto
10	Acierto	25	Acierto
11	Acierto	26	Acierto
12	Acierto	27	Acierto
13	Acierto	28	Acierto
14	Acierto	29	Acierto
15	Acierto	30	Acierto

Tabla (26) Resultados de las pruebas para el usuario Patricia Villar

Nombre del Individuo	Pierinna Sanchez		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Acierto
2	Acierto	17	Acierto
3	Acierto	18	Acierto
4	Acierto	19	Fallo
5	Acierto	20	Acierto
6	Acierto	21	Acierto
7	Acierto	22	Acierto
8	Acierto	23	Acierto
9	Acierto	24	Acierto
10	Acierto	25	Acierto
11	Acierto	26	Acierto
12	Acierto	27	Acierto
13	Fallo	28	Acierto
14	Acierto	29	Acierto
15	Acierto	30	Acierto

Tabla (27) Resultados de las pruebas para el usuario Pierinna Sánchez

Nombre del Individuo	Ricardo Tapullima		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Acierto
2	Acierto	17	Acierto
3	Acierto	18	Acierto
4	Acierto	19	Acierto
5	Acierto	20	Acierto
6	Acierto	21	Acierto
7	Fallo	22	Acierto
8	Acierto	23	Acierto
9	Acierto	24	Acierto
10	Acierto	25	Acierto
11	Acierto	26	Acierto
12	Acierto	27	Acierto
13	Acierto	28	Acierto
14	Acierto	29	Acierto
15	Acierto	30	Acierto

Tabla (28) Resultados de las pruebas para el usuario Ricardo Tapullima

Nombre del Individuo	Rosario Quinonez		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Fallo
2	Acierto	17	Acierto
3	Fallo	18	Acierto
4	Acierto	19	Acierto
5	Acierto	20	Acierto
6	Acierto	21	Acierto
7	Acierto	22	Acierto
8	Acierto	23	Fallo
9	Acierto	24	Acierto
10	Acierto	25	Acierto
11	Acierto	26	Acierto
12	Acierto	27	Acierto
13	Acierto	28	Acierto
14	Acierto	29	Acierto
15	Acierto	30	Acierto

Tabla (29) Resultados de las pruebas para el usuario Rosario Quiñonez

Nombre del Individuo	Santiago Quiñonez		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Acierto
2	Acierto	17	Acierto
3	Acierto	18	Acierto
4	Acierto	19	Acierto
5	Acierto	20	Acierto
6	Acierto	21	Fallo
7	Acierto	22	Acierto
8	Acierto	23	Acierto
9	Acierto	24	Acierto
10	Acierto	25	Acierto
11	Acierto	26	Acierto
12	Acierto	27	Acierto
13	Acierto	28	Acierto
14	Acierto	29	Acierto
15	Acierto	30	Acierto

Tabla (30) Resultados de las pruebas para el usuario Santiago Quiñonez

Nombre del Individuo	Luis Laurente		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Acierto
2	Acierto	17	Acierto
3	Acierto	18	Acierto
4	Acierto	19	Acierto
5	Acierto	20	Acierto
6	Acierto	21	Acierto
7	Acierto	22	Acierto
8	Acierto	23	Acierto
9	Acierto	24	Acierto
10	Acierto	25	Acierto
11	Acierto	26	Acierto
12	Acierto	27	Acierto
13	Acierto	28	Acierto
14	Acierto	29	Acierto
15	Fallo	30	Acierto

Tabla (31) Resultados de las pruebas para el usuario Luis Laurente

Nombre del Individuo	Wilman Fernandez		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Acierto
2	Acierto	17	Acierto
3	Acierto	18	Acierto
4	Acierto	19	Acierto
5	Acierto	20	Acierto
6	Acierto	21	Acierto
7	Acierto	22	Acierto
8	Acierto	23	Acierto
9	Acierto	24	Acierto
10	Acierto	25	Acierto
11	Fallo	26	Acierto
12	Acierto	27	Acierto
13	Acierto	28	Acierto
14	Acierto	29	Acierto
15	Acierto	30	Acierto

Tabla (32) Resultados de las pruebas para el usuario Wilman Fernandez

Nombre del Individuo	Yeng Inga Bereche		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Acierto
2	Acierto	17	Acierto
3	Acierto	18	Acierto
4	Acierto	19	Acierto
5	Acierto	20	Acierto
6	Acierto	21	Acierto
7	Acierto	22	Acierto
8	Acierto	23	Acierto
9	Acierto	24	Acierto
10	Acierto	25	Acierto
11	Acierto	26	Acierto
12	Acierto	27	Acierto
13	Acierto	28	Acierto
14	Acierto	29	Acierto
15	Acierto	30	Acierto

Tabla (33) Resultados de las pruebas para el usuario Yeng Inga

Nombre del Individuo	Yocce Jupa		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Fallo
2	Acierto	17	Acierto
3	Acierto	18	Acierto
4	Acierto	19	Acierto
5	Acierto	20	Acierto
6	Acierto	21	Acierto
7	Acierto	22	Acierto
8	Acierto	23	Acierto
9	Acierto	24	Acierto
10	Acierto	25	Acierto
11	Acierto	26	Acierto
12	Acierto	27	Acierto
13	Acierto	28	Acierto
14	Acierto	29	Acierto
15	Acierto	30	Acierto

Tabla (34) Resultados de las pruebas para el usuario Yocce Jupa

Nombre del Individuo	Aracelli Pariona		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Acierto
2	Acierto	17	Acierto
3	Acierto	18	Acierto
4	Acierto	19	Acierto
5	Acierto	20	Acierto
6	Acierto	21	Acierto
7	Acierto	22	Acierto
8	Acierto	23	Acierto
9	Acierto	24	Acierto
10	Acierto	25	Acierto
11	Acierto	26	Acierto
12	Acierto	27	Acierto
13	Acierto	28	Acierto
14	Acierto	29	Acierto
15	Acierto	30	Acierto

Tabla (35) Resultados de las pruebas para el usuario Aracelli Pariona

Nombre del Individuo	Aron Bustamante		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Acierto
2	Acierto	17	Acierto
3	Acierto	18	Acierto
4	Acierto	19	Acierto
5	Fallo	20	Acierto
6	Acierto	21	Acierto
7	Acierto	22	Acierto
8	Acierto	23	Acierto
9	Acierto	24	Acierto
10	Acierto	25	Acierto
11	Fallo	26	Acierto
12	Acierto	27	Acierto
13	Acierto	28	Acierto
14	Acierto	29	Acierto
15	Acierto	30	Acierto

Tabla (36) Resultados de las pruebas para el usuario Aron Bustamante

Nombre del Individuo	Benjamina Ruiz		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Acierto
2	Acierto	17	Acierto
3	Acierto	18	Acierto
4	Acierto	19	Acierto
5	Acierto	20	Acierto
6	Acierto	21	Acierto
7	Acierto	22	Fallo
8	Acierto	23	Acierto
9	Fallo	24	Acierto
10	Acierto	25	Acierto
11	Acierto	26	Acierto
12	Acierto	27	Acierto
13	Acierto	28	Acierto
14	Acierto	29	Acierto
15	Acierto	30	Acierto

Tabla (37) Resultados de las pruebas para el usuario Benjamina Ruiz

Nombre del Individuo	Brandon Norabuena		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Acierto
2	Acierto	17	Acierto
3	Acierto	18	Fallo
4	Acierto	19	Acierto
5	Acierto	20	Fallo
6	Acierto	21	Acierto
7	Acierto	22	Acierto
8	Acierto	23	Acierto
9	Acierto	24	Acierto
10	Acierto	25	Acierto
11	Acierto	26	Acierto
12	Acierto	27	Acierto
13	Acierto	28	Acierto
14	Acierto	29	Acierto
15	Acierto	30	Acierto

Tabla (38) Resultados de las pruebas para el usuario Brandon Norabuena

Nombre del Individuo	Celia Curi		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Acierto
2	Acierto	17	Acierto
3	Acierto	18	Acierto
4	Acierto	19	Acierto
5	Acierto	20	Acierto
6	Acierto	21	Acierto
7	Acierto	22	Acierto
8	Acierto	23	Acierto
9	Acierto	24	Acierto
10	Acierto	25	Acierto
11	Acierto	26	Acierto
12	Acierto	27	Acierto
13	Acierto	28	Acierto
14	Acierto	29	Acierto
15	Acierto	30	Acierto

Tabla (39) Resultados de las pruebas para el usuario Celia Curi

Nombre del Individuo	Charley Lovaton		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Acierto
2	Acierto	17	Acierto
3	Acierto	18	Acierto
4	Acierto	19	Acierto
5	Acierto	20	Acierto
6	Acierto	21	Acierto
7	Acierto	22	Acierto
8	Acierto	23	Acierto
9	Acierto	24	Acierto
10	Acierto	25	Acierto
11	Acierto	26	Acierto
12	Acierto	27	Acierto
13	Acierto	28	Acierto
14	Fallo	29	Acierto
15	Acierto	30	Acierto

Tabla (40) Resultados de las pruebas para el usuario Charley Lovatón

Nombre del Individuo	Claudia Carbajal		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Acierto
2	Acierto	17	Acierto
3	Acierto	18	Acierto
4	Acierto	19	Acierto
5	Acierto	20	Acierto
6	Acierto	21	Acierto
7	Acierto	22	Acierto
8	Fallo	23	Acierto
9	Fallo	24	Fallo
10	Acierto	25	Acierto
11	Acierto	26	Acierto
12	Acierto	27	Acierto
13	Acierto	28	Acierto
14	Acierto	29	Acierto
15	Acierto	30	Acierto

Tabla (41) Resultados de las pruebas para el usuario Claudia Carbajal

Nombre del Individuo	Daniel Colquichahua		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Acierto
2	Acierto	17	Acierto
3	Acierto	18	Acierto
4	Acierto	19	Fallo
5	Acierto	20	Acierto
6	Acierto	21	Acierto
7	Acierto	22	Acierto
8	Acierto	23	Acierto
9	Acierto	24	Acierto
10	Acierto	25	Acierto
11	Acierto	26	Acierto
12	Acierto	27	Acierto
13	Acierto	28	Acierto
14	Acierto	29	Acierto
15	Acierto	30	Acierto

Tabla (42) Resultados de las pruebas para el usuario Daniel Colquichahua

Nombre del Individuo	Deimer Ortiz		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Acierto
2	Acierto	17	Acierto
3	Acierto	18	Acierto
4	Acierto	19	Acierto
5	Fallo	20	Acierto
6	Acierto	21	Acierto
7	Acierto	22	Acierto
8	Acierto	23	Acierto
9	Acierto	24	Acierto
10	Acierto	25	Acierto
11	Acierto	26	Acierto
12	Acierto	27	Fallo
13	Acierto	28	Acierto
14	Acierto	29	Acierto
15	Acierto	30	Acierto

Tabla (43) Resultados de las pruebas para el usuario Deimer Ortiz

Nombre del Individuo	Deysi Mozombite		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Acierto
2	Acierto	17	Fallo
3	Acierto	18	Fallo
4	Acierto	19	Acierto
5	Acierto	20	Acierto
6	Acierto	21	Acierto
7	Acierto	22	Acierto
8	Acierto	23	Acierto
9	Acierto	24	Acierto
10	Acierto	25	Acierto
11	Acierto	26	Acierto
12	Acierto	27	Acierto
13	Acierto	28	Acierto
14	Acierto	29	Acierto
15	Acierto	30	Acierto

Tabla (44) Resultados de las pruebas para el usuario Deysi Mozombite

Nombre del Individuo	Dr Ballona		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Acierto
2	Acierto	17	Acierto
3	Acierto	18	Acierto
4	Acierto	19	Acierto
5	Acierto	20	Acierto
6	Acierto	21	Acierto
7	Fallo	22	Acierto
8	Acierto	23	Acierto
9	Acierto	24	Acierto
10	Acierto	25	Acierto
11	Acierto	26	Acierto
12	Fallo	27	Acierto
13	Acierto	28	Acierto
14	Acierto	29	Acierto
15	Acierto	30	Acierto

Tabla (45) Resultados de las pruebas para el usuario Doctor Ballona

Nombre del Individuo	Dra Rocia Gonzales		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Acierto
2	Acierto	17	Acierto
3	Acierto	18	Acierto
4	Acierto	19	Acierto
5	Acierto	20	Acierto
6	Acierto	21	Acierto
7	Acierto	22	Acierto
8	Acierto	23	Acierto
9	Acierto	24	Acierto
10	Acierto	25	Acierto
11	Acierto	26	Acierto
12	Acierto	27	Acierto
13	Acierto	28	Acierto
14	Acierto	29	Acierto
15	Acierto	30	Acierto

Tabla (46) Resultados de las pruebas para el usuario Rocía Gonzales

Nombre del Individuo	Edson Avellaneda		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Acierto
2	Acierto	17	Acierto
3	Fallo	18	Acierto
4	Acierto	19	Acierto
5	Acierto	20	Acierto
6	Acierto	21	Acierto
7	Acierto	22	Acierto
8	Acierto	23	Acierto
9	Acierto	24	Acierto
10	Acierto	25	Acierto
11	Acierto	26	Acierto
12	Acierto	27	Acierto
13	Acierto	28	Acierto
14	Acierto	29	Acierto
15	Acierto	30	Acierto

Tabla (47) Resultados de las pruebas para el usuario Edson Avellaneda

Nombre del Individuo	Edwin Ramirez		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Acierto
2	Acierto	17	Acierto
3	Acierto	18	Acierto
4	Acierto	19	Acierto
5	Acierto	20	Acierto
6	Acierto	21	Fallo
7	Acierto	22	Acierto
8	Acierto	23	Acierto
9	Acierto	24	Acierto
10	Acierto	25	Acierto
11	Acierto	26	Acierto
12	Acierto	27	Acierto
13	Acierto	28	Acierto
14	Acierto	29	Acierto
15	Acierto	30	Acierto

Tabla (48) Resultados de las pruebas para el usuario Edson Avellaneda

Nombre del Individuo	Enrique Alva		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Acierto
2	Acierto	17	Acierto
3	Acierto	18	Acierto
4	Acierto	19	Acierto
5	Acierto	20	Acierto
6	Acierto	21	Acierto
7	Acierto	22	Acierto
8	Acierto	23	Acierto
9	Fallo	24	Acierto
10	Acierto	25	Acierto
11	Acierto	26	Fallo
12	Acierto	27	Acierto
13	Acierto	28	Acierto
14	Acierto	29	Acierto
15	Acierto	30	Acierto

Tabla (49) Resultados de las pruebas para el usuario Enrique Alva

Nombre del Individuo	Enrique Belleza		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Acierto
2	Acierto	17	Acierto
3	Acierto	18	Acierto
4	Acierto	19	Acierto
5	Acierto	20	Acierto
6	Acierto	21	Acierto
7	Acierto	22	Acierto
8	Acierto	23	Fallo
9	Acierto	24	Acierto
10	Acierto	25	Acierto
11	Acierto	26	Acierto
12	Fallo	27	Acierto
13	Acierto	28	Acierto
14	Fallo	29	Acierto
15	Acierto	30	Acierto

Tabla (50) Resultados de las pruebas para el usuario Enrique Belleza

Nombre del Individuo	Frank Salas		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Acierto
2	Acierto	17	Acierto
3	Acierto	18	Acierto
4	Acierto	19	Acierto
5	Acierto	20	Acierto
6	Acierto	21	Acierto
7	Acierto	22	Acierto
8	Acierto	23	Acierto
9	Acierto	24	Acierto
10	Acierto	25	Acierto
11	Acierto	26	Acierto
12	Acierto	27	Acierto
13	Acierto	28	Acierto
14	Acierto	29	Acierto
15	Acierto	30	Acierto

Tabla (51) Resultados de las pruebas para el usuario Frank Salas

Nombre del Individuo	Gian Carlos Lupo		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Acierto
2	Acierto	17	Acierto
3	Acierto	18	Acierto
4	Acierto	19	Acierto
5	Acierto	20	Fallo
6	Acierto	21	Acierto
7	Acierto	22	Acierto
8	Acierto	23	Acierto
9	Acierto	24	Acierto
10	Acierto	25	Acierto
11	Acierto	26	Acierto
12	Acierto	27	Acierto
13	Acierto	28	Acierto
14	Acierto	29	Fallo
15	Acierto	30	Acierto

Tabla (52) Resultados de las pruebas para el usuario Gian Carlos Lupo

Nombre del Individuo	Gian Carlos Moya		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Acierto
2	Acierto	17	Fallo
3	Acierto	18	Acierto
4	Acierto	19	Acierto
5	Acierto	20	Acierto
6	Acierto	21	Acierto
7	Acierto	22	Acierto
8	Acierto	23	Acierto
9	Acierto	24	Acierto
10	Acierto	25	Acierto
11	Acierto	26	Acierto
12	Acierto	27	Acierto
13	Acierto	28	Acierto
14	Acierto	29	Acierto
15	Acierto	30	Acierto

Tabla (53) Resultados de las pruebas para el usuario Gian Carlos Moya

Nombre del Individuo	Gian Franco Mori		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Acierto
2	Acierto	17	Fallo
3	Acierto	18	Acierto
4	Acierto	19	Acierto
5	Fallo	20	Acierto
6	Acierto	21	Acierto
7	Acierto	22	Acierto
8	Acierto	23	Acierto
9	Acierto	24	Acierto
10	Acierto	25	Acierto
11	Acierto	26	Fallo
12	Acierto	27	Fallo
13	Acierto	28	Acierto
14	Acierto	29	Acierto
15	Acierto	30	Acierto

Tabla (54) Resultados de las pruebas para el usuario Gian Franco Mori

Nombre del Individuo	Gerson Calle		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Acierto
2	Acierto	17	Acierto
3	Acierto	18	Acierto
4	Acierto	19	Acierto
5	Acierto	20	Acierto
6	Acierto	21	Acierto
7	Acierto	22	Acierto
8	Acierto	23	Fallo
9	Fallo	24	Acierto
10	Acierto	25	Acierto
11	Acierto	26	Acierto
12	Acierto	27	Acierto
13	Acierto	28	Acierto
14	Acierto	29	Acierto
15	Acierto	30	Fallo

Tabla (55) Resultados de las pruebas para el usuario Gerson Calle

Nombre del Individuo	Gustavo Heredia		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Acierto
2	Acierto	17	Acierto
3	Acierto	18	Acierto
4	Acierto	19	Acierto
5	Acierto	20	Acierto
6	Acierto	21	Acierto
7	Acierto	22	Acierto
8	Acierto	23	Acierto
9	Acierto	24	Acierto
10	Fallo	25	Acierto
11	Acierto	26	Acierto
12	Acierto	27	Acierto
13	Acierto	28	Acierto
14	Acierto	29	Acierto
15	Acierto	30	Acierto

Tabla (56) Resultados de las pruebas para el usuario Gustavo Heredia

Nombre del Individuo	Heather Garcia		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Acierto
2	Acierto	17	Acierto
3	Acierto	18	Acierto
4	Acierto	19	Acierto
5	Acierto	20	Acierto
6	Acierto	21	Acierto
7	Acierto	22	Acierto
8	Acierto	23	Acierto
9	Acierto	24	Acierto
10	Acierto	25	Acierto
11	Acierto	26	Acierto
12	Acierto	27	Fallo
13	Acierto	28	Acierto
14	Acierto	29	Acierto
15	Acierto	30	Acierto

Tabla (57) Resultados de las pruebas para el usuario Heather García

Nombre del Individuo	Hilda Caceres		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Acierto
2	Acierto	17	Acierto
3	Acierto	18	Acierto
4	Acierto	19	Fallo
5	Acierto	20	Acierto
6	Acierto	21	Acierto
7	Fallo	22	Acierto
8	Acierto	23	Acierto
9	Acierto	24	Acierto
10	Acierto	25	Acierto
11	Fallo	26	Acierto
12	Acierto	27	Acierto
13	Acierto	28	Acierto
14	Acierto	29	Acierto
15	Acierto	30	Acierto

Tabla (58) Resultados de las pruebas para el usuario Hilda Cáceres

Nombre del Individuo	Jacky Poma		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Acierto
2	Acierto	17	Acierto
3	Acierto	18	Acierto
4	Acierto	19	Acierto
5	Acierto	20	Acierto
6	Acierto	21	Acierto
7	Acierto	22	Acierto
8	Acierto	23	Acierto
9	Acierto	24	Acierto
10	Acierto	25	Acierto
11	Acierto	26	Acierto
12	Acierto	27	Acierto
13	Acierto	28	Acierto
14	Acierto	29	Acierto
15	Acierto	30	Acierto

Tabla (59) Resultados de las pruebas para el usuario Jacky Poma

Nombre del Individuo	Jaime Lozada		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Acierto
2	Acierto	17	Acierto
3	Acierto	18	Fallo
4	Acierto	19	Acierto
5	Acierto	20	Acierto
6	Acierto	21	Acierto
7	Fallo	22	Acierto
8	Acierto	23	Acierto
9	Acierto	24	Acierto
10	Acierto	25	Acierto
11	Acierto	26	Acierto
12	Acierto	27	Acierto
13	Acierto	28	Acierto
14	Acierto	29	Acierto
15	Acierto	30	Acierto

Tabla (60) Resultados de las pruebas para el usuario Jaime Lozada

Nombre del Individuo	Jessica Girano		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Acierto
2	Acierto	17	Acierto
3	Acierto	18	Acierto
4	Acierto	19	Acierto
5	Acierto	20	Acierto
6	Acierto	21	Acierto
7	Acierto	22	Acierto
8	Acierto	23	Acierto
9	Acierto	24	Fallo
10	Acierto	25	Acierto
11	Acierto	26	Acierto
12	Acierto	27	Acierto
13	Acierto	28	Acierto
14	Acierto	29	Acierto
15	Acierto	30	Acierto

Tabla (61) Resultados de las pruebas para el usuario Jessica Girano

Nombre del Individuo	Jorge Garcia		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Acierto
2	Acierto	17	Acierto
3	Acierto	18	Acierto
4	Acierto	19	Acierto
5	Acierto	20	Acierto
6	Acierto	21	Acierto
7	Acierto	22	Acierto
8	Acierto	23	Acierto
9	Acierto	24	Acierto
10	Acierto	25	Acierto
11	Acierto	26	Acierto
12	Acierto	27	Acierto
13	Acierto	28	Acierto
14	Acierto	29	Acierto
15	Acierto	30	Acierto

Tabla (62) Resultados de las pruebas para el usuario Jorge García

Nombre del Individuo	Juan Diego Miljanovich		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Fallo
2	Acierto	17	Acierto
3	Acierto	18	Acierto
4	Acierto	19	Acierto
5	Acierto	20	Fallo
6	Acierto	21	Acierto
7	Acierto	22	Acierto
8	Acierto	23	Acierto
9	Acierto	24	Acierto
10	Acierto	25	Acierto
11	Acierto	26	Acierto
12	Acierto	27	Acierto
13	Acierto	28	Acierto
14	Acierto	29	Acierto
15	Fallo	30	Acierto

Tabla (63) Resultados de las pruebas para el usuario Juan Diego Miljanovich

Nombre del Individuo	Luis Bautista		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Acierto
2	Acierto	17	Acierto
3	Acierto	18	Acierto
4	Acierto	19	Acierto
5	Acierto	20	Acierto
6	Acierto	21	Acierto
7	Acierto	22	Acierto
8	Acierto	23	Acierto
9	Acierto	24	Acierto
10	Acierto	25	Acierto
11	Acierto	26	Acierto
12	Acierto	27	Acierto
13	Acierto	28	Acierto
14	Acierto	29	Acierto
15	Acierto	30	Acierto

Tabla (64) Resultados de las pruebas para el usuario Luis Bautista

Nombre del Individuo	Luis Fernado Coral		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Acierto
2	Acierto	17	Acierto
3	Acierto	18	Acierto
4	Fallo	19	Acierto
5	Acierto	20	Acierto
6	Acierto	21	Acierto
7	Acierto	22	Acierto
8	Acierto	23	Acierto
9	Acierto	24	Acierto
10	Acierto	25	Acierto
11	Acierto	26	Acierto
12	Acierto	27	Acierto
13	Acierto	28	Acierto
14	Acierto	29	Acierto
15	Acierto	30	Acierto

Tabla (65) Resultados de las pruebas para el usuario Luis Coral

Nombre del Individuo	Luis Flores		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Acierto
2	Acierto	17	Fallo
3	Acierto	18	Acierto
4	Acierto	19	Acierto
5	Acierto	20	Acierto
6	Fallo	21	Acierto
7	Acierto	22	Acierto
8	Acierto	23	Acierto
9	Acierto	24	Acierto
10	Acierto	25	Acierto
11	Fallo	26	Acierto
12	Acierto	27	Acierto
13	Acierto	28	Acierto
14	Acierto	29	Acierto
15	Acierto	30	Acierto

Tabla (66) Resultados de las pruebas para el usuario Luis Flores

Nombre del Individuo	Manuel Ortiz		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Acierto
2	Acierto	17	Acierto
3	Acierto	18	Acierto
4	Acierto	19	Acierto
5	Acierto	20	Acierto
6	Acierto	21	Acierto
7	Acierto	22	Acierto
8	Acierto	23	Acierto
9	Acierto	24	Acierto
10	Acierto	25	Acierto
11	Acierto	26	Acierto
12	Acierto	27	Acierto
13	Acierto	28	Acierto
14	Acierto	29	Acierto
15	Acierto	30	Acierto

Tabla (67) Resultados de las pruebas para el usuario Manuel Ortiz

Nombre del Individuo	Marco Julcapoma		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Acierto
2	Acierto	17	Acierto
3	Acierto	18	Fallo
4	Acierto	19	Acierto
5	Acierto	20	Acierto
6	Acierto	21	Acierto
7	Acierto	22	Acierto
8	Acierto	23	Acierto
9	Acierto	24	Acierto
10	Acierto	25	Acierto
11	Acierto	26	Acierto
12	Acierto	27	Acierto
13	Acierto	28	Acierto
14	Acierto	29	Acierto
15	Acierto	30	Acierto

Tabla (68) Resultados de las pruebas para el usuario Marco Julcapoma

Nombre del Individuo	Marco Mendoza		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Acierto
2	Acierto	17	Acierto
3	Acierto	18	Acierto
4	Acierto	19	Acierto
5	Acierto	20	Acierto
6	Acierto	21	Acierto
7	Acierto	22	Acierto
8	Acierto	23	Acierto
9	Acierto	24	Acierto
10	Acierto	25	Acierto
11	Acierto	26	Acierto
12	Acierto	27	Acierto
13	Acierto	28	Acierto
14	Acierto	29	Acierto
15	Acierto	30	Acierto

Tabla (69) Resultados de las pruebas para el usuario Marco Mendoza

Nombre del Individuo	Miguel Carhuapuma		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Acierto
2	Acierto	17	Acierto
3	Acierto	18	Acierto
4	Acierto	19	Acierto
5	Acierto	20	Acierto
6	Acierto	21	Acierto
7	Acierto	22	Acierto
8	Acierto	23	Acierto
9	Acierto	24	Acierto
10	Acierto	25	Acierto
11	Acierto	26	Acierto
12	Acierto	27	Acierto
13	Acierto	28	Acierto
14	Acierto	29	Acierto
15	Acierto	30	Acierto

Tabla (70) Resultados de las pruebas para el usuario Miguel Carhuapuma

Nombre del Individuo	Nat Palace		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Acierto
2	Acierto	17	Acierto
3	Acierto	18	Acierto
4	Acierto	19	Acierto
5	Acierto	20	Acierto
6	Fallo	21	Acierto
7	Fallo	22	Acierto
8	Acierto	23	Acierto
9	Acierto	24	Acierto
10	Acierto	25	Fallo
11	Acierto	26	Acierto
12	Acierto	27	Acierto
13	Acierto	28	Acierto
14	Acierto	29	Acierto
15	Acierto	30	Acierto

Tabla (71) Resultados de las pruebas para el usuario Nat Palace

Nombre del Individuo	Natividad Aguirre		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Acierto
2	Acierto	17	Acierto
3	Acierto	18	Acierto
4	Fallo	19	Acierto
5	Acierto	20	Acierto
6	Acierto	21	Acierto
7	Acierto	22	Fallo
8	Acierto	23	Acierto
9	Fallo	24	Acierto
10	Acierto	25	Acierto
11	Acierto	26	Acierto
12	Acierto	27	Acierto
13	Acierto	28	Acierto
14	Acierto	29	Acierto
15	Acierto	30	Acierto

Tabla (72) Resultados de las pruebas para el usuario Natividad Aguirre

Nombre del Individuo	Nelly Teran		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Acierto
2	Acierto	17	Acierto
3	Acierto	18	Acierto
4	Acierto	19	Fallo
5	Acierto	20	Acierto
6	Acierto	21	Acierto
7	Acierto	22	Acierto
8	Acierto	23	Acierto
9	Acierto	24	Acierto
10	Acierto	25	Acierto
11	Acierto	26	Acierto
12	Acierto	27	Acierto
13	Acierto	28	Acierto
14	Acierto	29	Acierto
15	Fallo	30	Acierto

Tabla (73) Resultados de las pruebas para el usuario Nelly Terán

Nombre del Individuo	Nicolas Vega		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Acierto
2	Acierto	17	Acierto
3	Acierto	18	Acierto
4	Acierto	19	Acierto
5	Acierto	20	Acierto
6	Acierto	21	Acierto
7	Acierto	22	Acierto
8	Acierto	23	Acierto
9	Acierto	24	Acierto
10	Acierto	25	Acierto
11	Acierto	26	Fallo
12	Acierto	27	Acierto
13	Acierto	28	Acierto
14	Acierto	29	Acierto
15	Acierto	30	Acierto

Tabla (74) Resultados de las pruebas para el usuario Nicolás Vega

Nombre del Individuo	Perci Tineo		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Acierto
2	Acierto	17	Acierto
3	Acierto	18	Acierto
4	Acierto	19	Acierto
5	Acierto	20	Acierto
6	Acierto	21	Acierto
7	Acierto	22	Acierto
8	Acierto	23	Acierto
9	Acierto	24	Acierto
10	Acierto	25	Acierto
11	Acierto	26	Acierto
12	Acierto	27	Acierto
13	Acierto	28	Acierto
14	Acierto	29	Acierto
15	Acierto	30	Acierto

Tabla (75) Resultados de las pruebas para el usuario Perci Tineo

Nombre del Individuo	Javier Perleche		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Acierto
2	Acierto	17	Acierto
3	Acierto	18	Acierto
4	Acierto	19	Acierto
5	Acierto	20	Acierto
6	Acierto	21	Acierto
7	Acierto	22	Acierto
8	Acierto	23	Acierto
9	Acierto	24	Acierto
10	Acierto	25	Acierto
11	Acierto	26	Acierto
12	Acierto	27	Acierto
13	Acierto	28	Acierto
14	Acierto	29	Acierto
15	Acierto	30	Acierto

Tabla (76) Resultados de las pruebas para el usuario Javier Perleche

Nombre del Individuo	Polo Paredes		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Acierto
2	Acierto	17	Acierto
3	Acierto	18	Acierto
4	Acierto	19	Acierto
5	Acierto	20	Acierto
6	Acierto	21	Acierto
7	Acierto	22	Acierto
8	Acierto	23	Acierto
9	Acierto	24	Acierto
10	Acierto	25	Acierto
11	Acierto	26	Fallo
12	Acierto	27	Acierto
13	Acierto	28	Acierto
14	Acierto	29	Acierto
15	Acierto	30	Acierto

Tabla (77) Resultados de las pruebas para el usuario Polo Paredes

Nombre del Individuo	Profesor Guillermo Kemper		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Acierto
2	Acierto	17	Acierto
3	Acierto	18	Fallo
4	Acierto	19	Acierto
5	Acierto	20	Acierto
6	Acierto	21	Acierto
7	Fallo	22	Acierto
8	Acierto	23	Acierto
9	Acierto	24	Acierto
10	Acierto	25	Acierto
11	Acierto	26	Acierto
12	Acierto	27	Acierto
13	Acierto	28	Acierto
14	Acierto	29	Acierto
15	Acierto	30	Acierto

Tabla (78) Resultados de las pruebas para el usuario Guillermo Kemper

Nombre del Individuo	Renato Li Toso		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Acierto
2	Acierto	17	Acierto
3	Acierto	18	Acierto
4	Acierto	19	Acierto
5	Fallo	20	Acierto
6	Fallo	21	Acierto
7	Acierto	22	Acierto
8	Acierto	23	Fallo
9	Acierto	24	Acierto
10	Acierto	25	Acierto
11	Acierto	26	Acierto
12	Acierto	27	Acierto
13	Acierto	28	Acierto
14	Acierto	29	Acierto
15	Acierto	30	Acierto

Tabla (79) Resultados de las pruebas para el usuario Renato Li Toso

Nombre del Individuo	Roger Cuzcano		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Acierto
2	Acierto	17	Acierto
3	Acierto	18	Acierto
4	Acierto	19	Acierto
5	Acierto	20	Acierto
6	Acierto	21	Acierto
7	Acierto	22	Acierto
8	Acierto	23	Acierto
9	Acierto	24	Acierto
10	Acierto	25	Acierto
11	Acierto	26	Acierto
12	Fallo	27	Acierto
13	Acierto	28	Acierto
14	Acierto	29	Acierto
15	Acierto	30	Acierto

Tabla (80) Resultados de las pruebas para el usuario Roger Cuzcano

Nombre del Individuo	Rosario Sialer		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Acierto
2	Fallo	17	Acierto
3	Acierto	18	Acierto
4	Acierto	19	Acierto
5	Acierto	20	Acierto
6	Acierto	21	Acierto
7	Acierto	22	Acierto
8	Acierto	23	Acierto
9	Acierto	24	Acierto
10	Acierto	25	Acierto
11	Acierto	26	Acierto
12	Acierto	27	Fallo
13	Acierto	28	Acierto
14	Acierto	29	Acierto
15	Acierto	30	Acierto

Tabla (81) Resultados de las pruebas para el usuario Rosario Sialer

Nombre del Individuo	Rosita Martínez		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Acierto
2	Acierto	17	Acierto
3	Acierto	18	Acierto
4	Acierto	19	Acierto
5	Acierto	20	Fallo
6	Acierto	21	Fallo
7	Acierto	22	Fallo
8	Acierto	23	Acierto
9	Acierto	24	Acierto
10	Fallo	25	Acierto
11	Acierto	26	Acierto
12	Acierto	27	Acierto
13	Acierto	28	Fallo
14	Acierto	29	Acierto
15	Acierto	30	Acierto

Tabla (82) Resultados de las pruebas para el usuario Rosita Martínez

Nombre del Individuo	Sr Loiza		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Acierto
2	Acierto	17	Acierto
3	Acierto	18	Acierto
4	Acierto	19	Acierto
5	Acierto	20	Acierto
6	Acierto	21	Acierto
7	Acierto	22	Acierto
8	Acierto	23	Acierto
9	Acierto	24	Acierto
10	Acierto	25	Acierto
11	Acierto	26	Acierto
12	Acierto	27	Acierto
13	Acierto	28	Acierto
14	Acierto	29	Acierto
15	Acierto	30	Acierto

Tabla (83) Resultados de las pruebas para el usuario Sr. Loiza

Nombre del Individuo	Juan Lorgio		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Acierto
2	Acierto	17	Acierto
3	Acierto	18	Acierto
4	Acierto	19	Acierto
5	Acierto	20	Fallo
6	Acierto	21	Acierto
7	Acierto	22	Acierto
8	Acierto	23	Acierto
9	Acierto	24	Acierto
10	Acierto	25	Acierto
11	Acierto	26	Acierto
12	Acierto	27	Acierto
13	Acierto	28	Acierto
14	Acierto	29	Acierto
15	Acierto	30	Acierto

Tabla (84) Resultados de las pruebas para el usuario Juan Lorgio

Nombre del Individuo	Alejandro Reaño		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Acierto
2	Acierto	17	Acierto
3	Acierto	18	Acierto
4	Fallo	19	Acierto
5	Acierto	20	Acierto
6	Acierto	21	Acierto
7	Acierto	22	Acierto
8	Acierto	23	Acierto
9	Acierto	24	Acierto
10	Acierto	25	Acierto
11	Acierto	26	Acierto
12	Acierto	27	Acierto
13	Acierto	28	Acierto
14	Acierto	29	Acierto
15	Acierto	30	Acierto

Tabla (85) Resultados de las pruebas para el usuario Alejandro Reaño

Nombre del Individuo	Renzo Tello		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Acierto
2	Acierto	17	Acierto
3	Acierto	18	Acierto
4	Acierto	19	Acierto
5	Acierto	20	Acierto
6	Acierto	21	Acierto
7	Acierto	22	Acierto
8	Acierto	23	Acierto
9	Acierto	24	Acierto
10	Acierto	25	Acierto
11	Acierto	26	Acierto
12	Acierto	27	Acierto
13	Acierto	28	Acierto
14	Acierto	29	Acierto
15	Acierto	30	Acierto

Tabla (86) Resultados de las pruebas para el usuario Renzo Tello

Nombre del Individuo	Viky Quispe		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Fallo
2	Acierto	17	Acierto
3	Acierto	18	Acierto
4	Acierto	19	Acierto
5	Acierto	20	Fallo
6	Acierto	21	Acierto
7	Acierto	22	Acierto
8	Acierto	23	Acierto
9	Acierto	24	Acierto
10	Acierto	25	Acierto
11	Acierto	26	Acierto
12	Acierto	27	Acierto
13	Acierto	28	Acierto
14	Acierto	29	Acierto
15	Fallo	30	Acierto

Tabla (87) Resultados de las pruebas para el usuario Vicky Quispe

Nombre del Individuo	Yolanda Sanchez		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Acierto
2	Acierto	17	Acierto
3	Acierto	18	Fallo
4	Acierto	19	Acierto
5	Acierto	20	Acierto
6	Fallo	21	Acierto
7	Acierto	22	Acierto
8	Acierto	23	Acierto
9	Acierto	24	Acierto
10	Acierto	25	Acierto
11	Acierto	26	Acierto
12	Acierto	27	Acierto
13	Acierto	28	Acierto
14	Acierto	29	Acierto
15	Acierto	30	Acierto

Tabla (88) Resultados de las pruebas para el usuario Yolanda Sánchez

Nombre del Individuo	Stefany Onofre		
N° de prueba.	Resultado de la predicción hecha por la red	N° de prueba.	Resultado de la predicción hecha por la red
1	Acierto	16	Acierto
2	Acierto	17	Acierto
3	Acierto	18	Acierto
4	Acierto	19	Acierto
5	Acierto	20	Acierto
6	Acierto	21	Acierto
7	Acierto	22	Acierto
8	Acierto	23	Acierto
9	Fallo	24	Acierto
10	Acierto	25	Fallo
11	Acierto	26	Acierto
12	Fallo	27	Acierto
13	Fallo	28	Acierto
14	Acierto	29	Acierto
15	Acierto	30	Acierto

Tabla (89) Resultados de las pruebas para el usuario Stefany Onofre

5.2 Cálculo de las tasas de verdaderos reconocimientos y falsos rechazos para cada usuario

Se calcula la métrica de los verdaderos reconocimientos (TAR) y falsos rechazos (FRR) en función del número de pruebas realizadas que equivale a la suma de treinta.

- Para José Dueñas:
 - Número de verdaderos reconocimientos: 28
 - Número de falsos rechazos: 2
 - $TAR_1 = \frac{28(100)}{30} = 93.33\%$
 - $FRR_1 = \frac{2}{30} = 0.067$

- Para John López:
 - Número de verdaderos reconocimientos: 29
 - Número de falsos rechazos: 1
 - $TAR_2 = \frac{29(100)}{30} = 96.67\%$
 - $FRR_2 = \frac{1}{30} = 0.033$

- Para Abigail Álvarez:
 - Número de verdaderos reconocimientos: 28
 - Número de falsos rechazos: 2
 - $TAR_3 = \frac{28(100)}{30} = 93.33\%$
 - $FRR_3 = \frac{2}{30} = 0.067$

- Para Alejandro Ríos:
 - Número de verdaderos reconocimientos: 30
 - Número de falsos rechazos: 0
 - $TAR_4 = \frac{28(100)}{30} = 100\%$
 - $FRR_4 = \frac{0}{30} = 0$

- Para Alex Moreno:
 - Número de verdaderos reconocimientos: 28
 - Número de falsos rechazos: 2
 - $TAR_5 = \frac{28(100)}{30} = 93.33\%$
 - $FRR_5 = \frac{2}{30} = 0.067$

- Para Angie Villalba:
 - Número de verdaderos reconocimientos: 28
 - Número de falsos rechazos: 2
 - $TAR_6 = \frac{28(100)}{30} = 93.33\%$
 - $FRR_6 = \frac{2}{30} = 0.067$

- Para Anthony Salas:
 - Número de verdaderos reconocimientos: 29
 - Número de falsos rechazos: 1
 - $TAR_7 = \frac{29(100)}{30} = 96.67\%$
 - $FRR_7 = \frac{1}{30} = 0.033$

- Para Christian Mendoza:
 - Número de verdaderos reconocimientos: 27
 - Número de falsos rechazos: 3
 - $TAR_8 = \frac{27(100)}{30} = 90\%$
 - $FRR_8 = \frac{3}{30} = 0.1$

- Para Clarisa Penedo:
 - Número de verdaderos reconocimientos: 29
 - Número de falsos rechazos: 1
 - $TAR_9 = \frac{29(100)}{30} = 96.67\%$
 - $FRR_9 = \frac{1}{30} = 0.033$

- Para Danitza Alcántara:
 - Número de verdaderos reconocimientos: 27
 - Número de falsos rechazos: 3
 - $TAR_{10} = \frac{27(100)}{30} = 90\%$
 - $FRR_{10} = \frac{3}{30} = 0.1$

- Carmen Sánchez:
 - Número de verdaderos reconocimientos: 30
 - Número de falsos rechazos: 0
 - $TAR_{11} = \frac{28(100)}{30} = 100\%$
 - $FRR_{11} = \frac{0}{30} = 0$

- Para Emanuel Salas:
 - Número de verdaderos reconocimientos: 28
 - Número de falsos rechazos: 2
 - $TAR_{12} = \frac{28(100)}{30} = 93.33\%$
 - $FRR_{12} = \frac{2}{30} = 0.067$

- Para Gastón Navas:
 - Número de verdaderos reconocimientos: 28
 - Número de falsos rechazos: 2
 - $TAR_{13} = \frac{28(100)}{30} = 93.33\%$
 - $FRR_{13} = \frac{2}{30} = 0.067$

- Para Inés Mori:
 - Número de verdaderos reconocimientos: 29
 - Número de falsos rechazos: 1
 - $TAR_{14} = \frac{29(100)}{30} = 96.67\%$
 - $FRR_{14} = \frac{1}{30} = 0.033$

- Para José Carlos Román:
 - Número de verdaderos reconocimientos: 27
 - Número de falsos rechazos: 3
 - $TAR_{15} = \frac{27(100)}{30} = 90\%$
 - $FRR_{15} = \frac{3}{30} = 0.1$

- Para José Juan Zapata:

- Número de verdaderos reconocimientos: 27
- Número de falsos rechazos: 3
- $TAR_{16} = \frac{27(100)}{30} = 90\%$
- $FRR_{16} = \frac{3}{30} = 0.1$

- Para Luis Cornejo:
 - Número de verdaderos reconocimientos: 27
 - Número de falsos rechazos: 3
 - $TAR_{17} = \frac{27(100)}{30} = 90\%$
 - $FRR_{17} = \frac{3}{30} = 0.1$

- Para Inés María Bandera:
 - Número de verdaderos reconocimientos: 29
 - Número de falsos rechazos: 1
 - $TAR_{18} = \frac{29(100)}{30} = 96.67\%$
 - $FRR_{18} = \frac{1}{30} = 0.033$

- Para Milagrito Cabezas:
 - Número de verdaderos reconocimientos: 27
 - Número de falsos rechazos: 3
 - $TAR_{19} = \frac{27(100)}{30} = 90\%$
 - $FRR_{19} = \frac{3}{30} = 0.1$

- Patricia Villar:
 - Número de verdaderos reconocimientos: 29
 - Número de falsos rechazos: 1
 - $TAR_{20} = \frac{29(100)}{30} = 96.67\%$
 - $FRR_{20} = \frac{1}{30} = 0.033$

- Para Pierina Sánchez:

- Número de verdaderos reconocimientos: 28
- Número de falsos rechazos: 2
- $TAR_{21} = \frac{28(100)}{30} = 93.33\%$
- $FRR_{21} = \frac{2}{30} = 0.067$

- Ricardo Tapullima:
 - Número de verdaderos reconocimientos: 29
 - Número de falsos rechazos: 1
 - $TAR_{22} = \frac{29(100)}{30} = 96.67\%$
 - $FRR_{22} = \frac{1}{30} = 0.033$

- Para Rosario Quiñonez:
 - Número de verdaderos reconocimientos: 27
 - Número de falsos rechazos: 3
 - $TAR_{23} = \frac{27(100)}{30} = 90\%$
 - $FRR_{23} = \frac{3}{30} = 0.1$

- Para Santiago Quiñonez:
 - Número de verdaderos reconocimientos: 29
 - Número de falsos rechazos: 1
 - $TAR_{24} = \frac{29(100)}{30} = 96.67\%$
 - $FRR_{24} = \frac{1}{30} = 0.033$

- Para Luis Laurente:
 - Número de verdaderos reconocimientos: 29
 - Número de falsos rechazos: 1
 - $TAR_{25} = \frac{29(100)}{30} = 96.67\%$
 - $FRR_{25} = \frac{1}{30} = 0.033$

- Para Wilman Fernandez:
 - Número de verdaderos reconocimientos: 29
 - Número de falsos rechazos: 1
 - $TAR_{26} = \frac{29(100)}{30} = 96.67\%$
 - $FRR_{26} = \frac{1}{30} = 0.033$

- Para Yeng Inga Bereche:
 - Número de verdaderos reconocimientos: 30
 - Número de falsos rechazos: 0
 - $TAR_{27} = \frac{28(100)}{30} = 100\%$
 - $FRR_{27} = \frac{0}{30} = 0$

- Para Yocce Jupa:
 - Número de verdaderos reconocimientos: 29
 - Número de falsos rechazos: 1
 - $TAR_{28} = \frac{29(100)}{30} = 96.67\%$
 - $FRR_{28} = \frac{1}{30} = 0.033$

- Para Aracelli Pariona:
 - Número de verdaderos reconocimientos: 30
 - Número de falsos rechazos: 0
 - $TAR_{29} = \frac{28(100)}{30} = 100\%$
 - $FRR_{29} = \frac{0}{30} = 0$

- Para Aron Bustamante:
 - Número de verdaderos reconocimientos: 28
 - Número de falsos rechazos: 2
 - $TAR_{30} = \frac{28(100)}{30} = 93.33\%$
 - $FRR_{30} = \frac{2}{30} = 0.067$

- Para Benjamina Ruiz:
 - Número de verdaderos reconocimientos: 28
 - Número de falsos rechazos: 2
 - $TAR_{31} = \frac{28(100)}{30} = 93.33\%$
 - $FRR_{31} = \frac{2}{30} = 0.067$

- Para Brandon Norabuena:
 - Número de verdaderos reconocimientos: 28
 - Número de falsos rechazos: 2
 - $TAR_{32} = \frac{28(100)}{30} = 93.33\%$
 - $FRR_{32} = \frac{2}{30} = 0.067$

- Para Celia Curi:
 - Número de verdaderos reconocimientos: 30
 - Número de falsos rechazos: 0
 - $TAR_{33} = \frac{28(100)}{30} = 100\%$
 - $FRR_{33} = \frac{0}{30} = 0$

- Para Charley Lovatón:
 - Número de verdaderos reconocimientos: 29
 - Número de falsos rechazos: 1
 - $TAR_{34} = \frac{29(100)}{30} = 96.67\%$
 - $FRR_{34} = \frac{1}{30} = 0.033$

- Para Claudia Carbajal:
 - Número de verdaderos reconocimientos: 27
 - Número de falsos rechazos: 3
 - $TAR_{35} = \frac{27(100)}{30} = 90\%$
 - $FRR_{35} = \frac{3}{30} = 0.1$

- Para Daniel Colquichahua:
 - Número de verdaderos reconocimientos: 29
 - Número de falsos rechazos: 1
 - $TAR_{36} = \frac{29(100)}{30} = 96.67\%$
 - $FRR_{36} = \frac{1}{30} = 0.033$

- Para Deimer Ortiz:
 - Número de verdaderos reconocimientos: 28
 - Número de falsos rechazos: 2
 - $TAR_{37} = \frac{28(100)}{30} = 93.33\%$
 - $FRR_{37} = \frac{2}{30} = 0.067$

- Para Deisy Mozombite:
 - Número de verdaderos reconocimientos: 28
 - Número de falsos rechazos: 2
 - $TAR_{38} = \frac{28(100)}{30} = 93.33\%$
 - $FRR_{38} = \frac{2}{30} = 0.067$

- Para Dr. Ballona:
 - Número de verdaderos reconocimientos: 28
 - Número de falsos rechazos: 2
 - $TAR_{39} = \frac{28(100)}{30} = 93.33\%$
 - $FRR_{39} = \frac{2}{30} = 0.067$

- Para Dra. Rocía Gonzales:
 - Número de verdaderos reconocimientos: 30
 - Número de falsos rechazos: 0
 - $TAR_{40} = \frac{28(100)}{30} = 100\%$
 - $FRR_{40} = \frac{0}{30} = 0$

- Para Edson Avellaneda:
 - Número de verdaderos reconocimientos: 29
 - Número de falsos rechazos: 1
 - $TAR_{41} = \frac{29(100)}{30} = 96.67\%$
 - $FRR_{41} = \frac{1}{30} = 0.033$

- Para Edwin Ramírez:
 - Número de verdaderos reconocimientos: 29
 - Número de falsos rechazos: 1
 - $TAR_{42} = \frac{29(100)}{30} = 96.67\%$
 - $FRR_{42} = \frac{1}{30} = 0.033$

- Para Enrique Alva:
 - Número de verdaderos reconocimientos: 28
 - Número de falsos rechazos: 2
 - $TAR_{43} = \frac{28(100)}{30} = 93.33\%$
 - $FRR_{43} = \frac{2}{30} = 0.067$

- Para Enrique Belleza:
 - Número de verdaderos reconocimientos: 27
 - Número de falsos rechazos: 3
 - $TAR_{44} = \frac{27(100)}{30} = 90\%$
 - $FRR_{44} = \frac{3}{30} = 0.1$

- Para Frank Salas:
 - Número de verdaderos reconocimientos: 30
 - Número de falsos rechazos: 0
 - $TAR_{45} = \frac{30(100)}{30} = 100\%$
 - $FRR_{45} = \frac{0}{30} = 0$

- Para Gian Carlos Lupo:
 - Número de verdaderos reconocimientos: 28
 - Número de falsos rechazos: 2
 - $TAR_{46} = \frac{28(100)}{30} = 93.33\%$
 - $FRR_{46} = \frac{2}{30} = 0.067$

- Para Gian Carlos Moya:
 - Número de verdaderos reconocimientos: 29
 - Número de falsos rechazos: 1
 - $TAR_{47} = \frac{29(100)}{30} = 96.67\%$
 - $FRR_{47} = \frac{1}{30} = 0.033$

- Para Gian Franco Mori:
 - Número de verdaderos reconocimientos: 26
 - Número de falsos rechazos: 4
 - $TAR_{48} = \frac{26(100)}{30} = 86.67\%$
 - $FRR_{48} = \frac{4}{30} = 0.133$

- Para Gerson Calle:
 - Número de verdaderos reconocimientos: 27
 - Número de falsos rechazos: 3
 - $TAR_{49} = \frac{27(100)}{30} = 90\%$
 - $FRR_{49} = \frac{3}{30} = 0.1$

- Para Gustavo Heredia:
 - Número de verdaderos reconocimientos: 29
 - Número de falsos rechazos: 1
 - $TAR_{50} = \frac{29(100)}{30} = 96.67\%$
 - $FRR_{50} = \frac{1}{30} = 0.033$

- Para Heather García:
 - Número de verdaderos reconocimientos: 29
 - Número de falsos rechazos: 1
 - $TAR_{51} = \frac{29(100)}{30} = 96.67\%$
 - $FRR_{51} = \frac{1}{30} = 0.033$

- Para Hilda Cáceres:
 - Número de verdaderos reconocimientos: 27
 - Número de falsos rechazos: 3
 - $TAR_{52} = \frac{27(100)}{30} = 90\%$
 - $FRR_{52} = \frac{3}{30} = 0.1$

- Para Jacky Poma:
 - Número de verdaderos reconocimientos: 29
 - Número de falsos rechazos: 1
 - $TAR_{53} = \frac{29(100)}{30} = 96.67\%$
 - $FRR_{53} = \frac{1}{30} = 0.033$

- Para Jaime Lozada:
 - Número de verdaderos reconocimientos: 28
 - Número de falsos rechazos: 2
 - $TAR_{54} = \frac{28(100)}{30} = 93.33\%$
 - $FRR_{54} = \frac{2}{30} = 0.067$

- Para Jessica Girano:
 - Número de verdaderos reconocimientos: 29
 - Número de falsos rechazos: 1
 - $TAR_{55} = \frac{29(100)}{30} = 96.67\%$
 - $FRR_{55} = \frac{1}{30} = 0.033$

- Para Jorge García:
 - Número de verdaderos reconocimientos: 30
 - Número de falsos rechazos: 0
 - $TAR_{56} = \frac{28(100)}{30} = 100\%$
 - $FRR_{56} = \frac{0}{30} = 0$

- Para Juan Diego Miljanovich:
 - Número de verdaderos reconocimientos: 27
 - Número de falsos rechazos: 3
 - $TAR_{57} = \frac{27(100)}{30} = 90\%$
 - $FRR_{57} = \frac{3}{30} = 0.1$

- Para Luis Bautista:
 - Número de verdaderos reconocimientos: 30
 - Número de falsos rechazos: 0
 - $TAR_{58} = \frac{28(100)}{30} = 100\%$
 - $FRR_{58} = \frac{0}{30} = 0$

- Para Luis Fernando Coral:
 - Número de verdaderos reconocimientos: 29
 - Número de falsos rechazos: 1
 - $TAR_{59} = \frac{29(100)}{30} = 96.67\%$
 - $FRR_{59} = \frac{1}{30} = 0.033$

- Para Luis Flores:
 - Número de verdaderos reconocimientos: 27
 - Número de falsos rechazos: 3
 - $TAR_{60} = \frac{27(100)}{30} = 90\%$
 - $FRR_{60} = \frac{3}{30} = 0.1$

- Para Manuel Ortiz:
 - Número de verdaderos reconocimientos: 30
 - Número de falsos rechazos: 0
 - $TAR_{61} = \frac{28(100)}{30} = 100\%$
 - $FRR_{61} = \frac{0}{30} = 0$

- Para Marco Julcapoma:
 - Número de verdaderos reconocimientos: 29
 - Número de falsos rechazos: 1
 - $TAR_{62} = \frac{29(100)}{30} = 96.67\%$
 - $FRR_{62} = \frac{1}{30} = 0.033$

- Para Marco Mendoza:
 - Número de verdaderos reconocimientos: 30
 - Número de falsos rechazos: 0
 - $TAR_{63} = \frac{28(100)}{30} = 100\%$
 - $FRR_{63} = \frac{0}{30} = 0$

- Para Miguel Carhuapuma:
 - Número de verdaderos reconocimientos: 30
 - Número de falsos rechazos: 0
 - $TAR_{64} = \frac{28(100)}{30} = 100\%$
 - $FRR_{64} = \frac{0}{30} = 0$

- Para Juan Nat Palace:
 - Número de verdaderos reconocimientos: 27
 - Número de falsos rechazos: 3
 - $TAR_{65} = \frac{27(100)}{30} = 90\%$
 - $FRR_{65} = \frac{3}{30} = 0.1$

- Para Natividad Aguirre:
 - Número de verdaderos reconocimientos: 27
 - Número de falsos rechazos: 3
 - $TAR_{66} = \frac{27(100)}{30} = 90\%$
 - $FRR_{66} = \frac{3}{30} = 0.1$

- Para Nelly Terán:
 - Número de verdaderos reconocimientos: 28
 - Número de falsos rechazos: 2
 - $TAR_{67} = \frac{28(100)}{30} = 93.33\%$
 - $FRR_{67} = \frac{2}{30} = 0.067$

- Para Nicolás Vega:
 - Número de verdaderos reconocimientos: 29
 - Número de falsos rechazos: 1
 - $TAR_{68} = \frac{29(100)}{30} = 96.67\%$
 - $FRR_{68} = \frac{1}{30} = 0.033$

- Para Perci Tineo:
 - Número de verdaderos reconocimientos: 30
 - Número de falsos rechazos: 0
 - $TAR_{69} = \frac{28(100)}{30} = 100\%$
 - $FRR_{69} = \frac{0}{30} = 0$

- Para Javier Perleche:
 - Número de verdaderos reconocimientos: 30
 - Número de falsos rechazos: 0
 - $TAR_{70} = \frac{28(100)}{30} = 100\%$
 - $FRR_{70} = \frac{0}{30} = 0$

- Para Polo Paredes:
 - Número de verdaderos reconocimientos: 29
 - Número de falsos rechazos: 1
 - $TAR_{71} = \frac{29(100)}{30} = 96.67\%$
 - $FRR_{71} = \frac{1}{30} = 0.033$

- Para Profesor Guillermo Kemper:
 - Número de verdaderos reconocimientos: 28
 - Número de falsos rechazos: 2
 - $TAR_{72} = \frac{28(100)}{30} = 93.33\%$
 - $FRR_{72} = \frac{2}{30} = 0.067$

- Para Renato Li Toso
 - Número de verdaderos reconocimientos: 27
 - Número de falsos rechazos: 3
 - $TAR_{73} = \frac{27(100)}{30} = 90\%$
 - $FRR_{73} = \frac{3}{30} = 0.1$

- Para Roger Cuzcano:
 - Número de verdaderos reconocimientos: 29
 - Número de falsos rechazos: 1
 - $TAR_{74} = \frac{29(100)}{30} = 96.67\%$
 - $FRR_{74} = \frac{1}{30} = 0.033$

- Para Rosario Sialer:
 - Número de verdaderos reconocimientos: 28
 - Número de falsos rechazos: 2
 - $TAR_{75} = \frac{28(100)}{30} = 93.33\%$
 - $FRR_{75} = \frac{2}{30} = 0.067$

- Para Rosita Martínez:
 - Número de verdaderos reconocimientos: 25
 - Número de falsos rechazos: 5
 - $TAR_{76} = \frac{25(100)}{30} = 83.33\%$
 - $FRR_{76} = \frac{5}{30} = 0.167$

- Para Sr. Loaiza:
 - Número de verdaderos reconocimientos: 30
 - Número de falsos rechazos: 0
 - $TAR_{77} = \frac{28(100)}{30} = 100\%$
 - $FRR_{77} = \frac{0}{30} = 0$

- Para Juan Lorgio:
 - Número de verdaderos reconocimientos: 29
 - Número de falsos rechazos: 1
 - $TAR_{78} = \frac{29(100)}{30} = 96.67\%$
 - $FRR_{78} = \frac{1}{30} = 0.033$

- Para Alejandro Reaño:
 - Número de verdaderos reconocimientos: 29
 - Número de falsos rechazos: 1
 - $TAR_{79} = \frac{29(100)}{30} = 96.67\%$
 - $FRR_{79} = \frac{1}{30} = 0.033$

- Para Renzo Tello:
 - Número de verdaderos reconocimientos: 30
 - Número de falsos rechazos: 0
 - $TAR_{80} = \frac{28(100)}{30} = 100\%$
 - $FRR_{80} = \frac{0}{30} = 0$

- Para Vicky Quispe:
 - Número de verdaderos reconocimientos: 27
 - Número de falsos rechazos: 3
 - $TAR_{81} = \frac{27(100)}{30} = 90\%$
 - $FRR_{81} = \frac{3}{30} = 0.1$

- Para Yolanda Sánchez:
 - Número de verdaderos reconocimientos: 28
 - Número de falsos rechazos: 2
 - $TAR_{82} = \frac{28(100)}{30} = 93.33\%$
 - $FRR_{82} = \frac{2}{30} = 0.067$

- Para Stefany Onofre:
 - Número de verdaderos reconocimientos: 26
 - Número de falsos rechazos: 4
 - $TAR_{83} = \frac{26(100)}{30} = 86.67\%$
 - $FRR_{83} = \frac{4}{30} = 0.133$

5.3 Cálculo de la tasa de verdaderos reconocimientos y falsos rechazos globales para la red neural

Lo calculado anteriormente se puede apreciar de forma más práctica en los gráficos de las figuras 73, 74, 75 y 76:

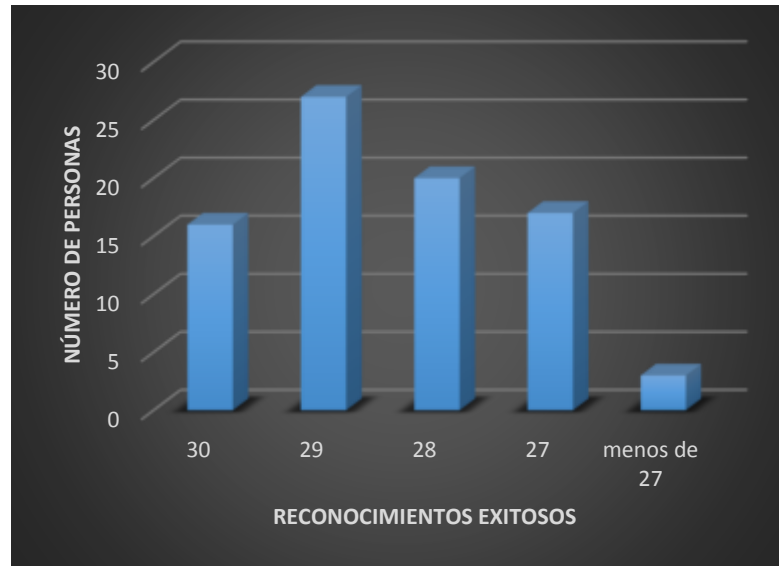


Figura (73) Gráfico de barras para los verdaderos reconocimientos en 83 usuarios

Elaboración: los autores

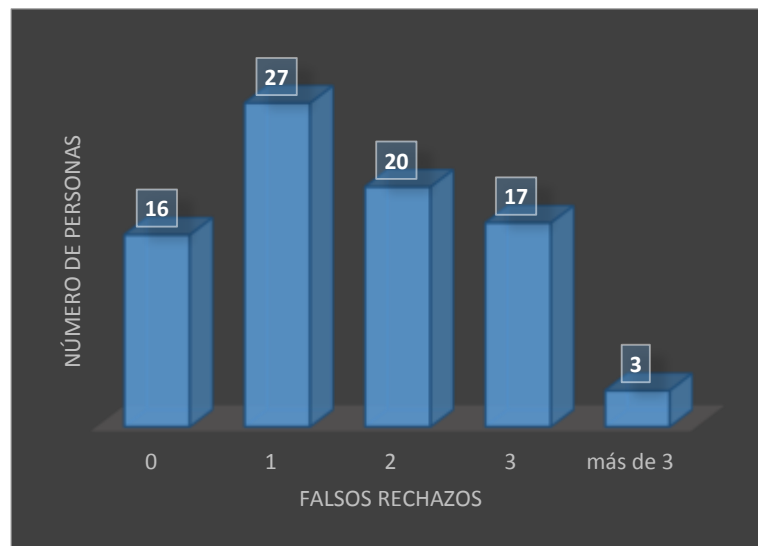


Figura (74) Gráfico de barras para los falsos rechazos en 83 usuarios

Elaboración: los autores

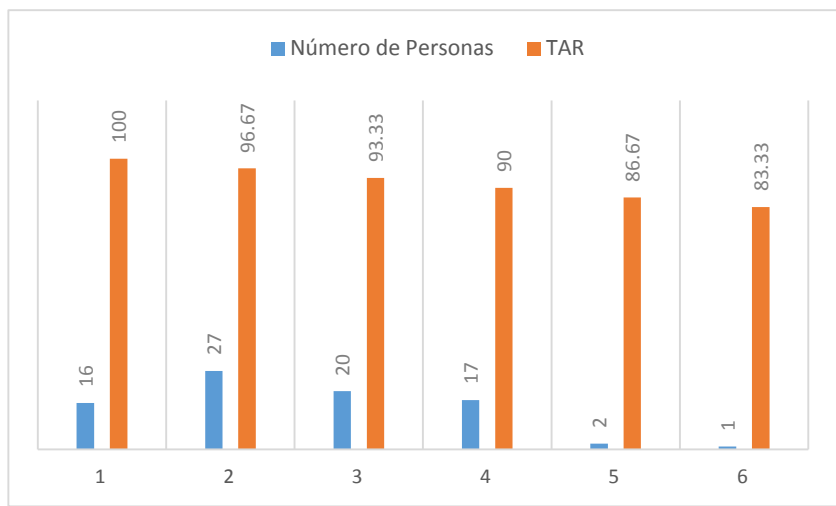


Figura (75) Gráfico de barras para el TAR de cada usuario

Elaboración: los autores

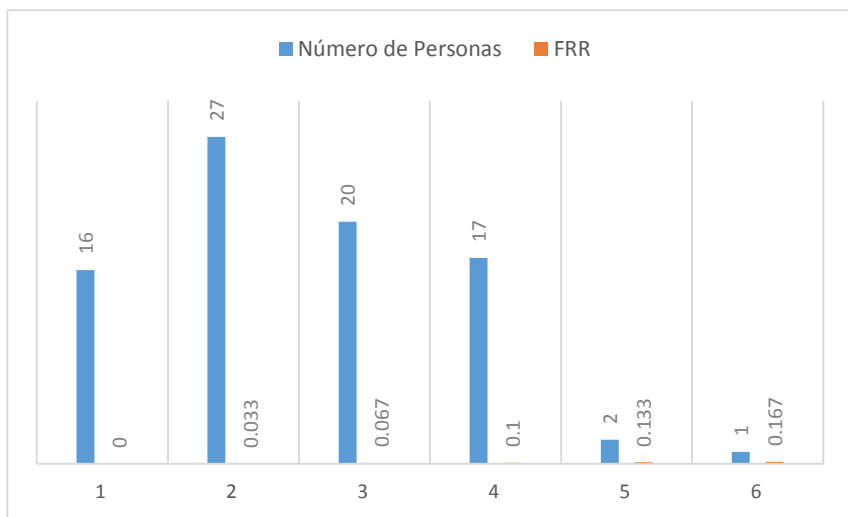


Figura (76) Gráfico de barras para el FRR de cada usuario

Elaboración: los autores

$$\begin{aligned} \text{➤ } TAR_{global} &= \frac{TAR_1 + TAR_2 + TAR_3 \dots + TAR_{83}}{83} = \\ &= \frac{16(100) + 27(96.67) + 20(93.33) + 17(90) + 2(86.67) + 83.33}{83} = 94.74\% \end{aligned}$$

$$\begin{aligned} \text{➤ } FRR_{global} &= \frac{FRR_1 + FRR_2 + FRR_3 + \dots + FRR_{83}}{83} = \\ &= \frac{16(0) + 27(0.033) + 20(0.067) + 17(0.1) + 2(0.133) + 0.167}{83} = 0.0526 \end{aligned}$$

5.4 Costos en inversión del producto final

En la tabla 90 se aprecia el costo e inversión total del trabajo, en el cual se declara los componentes que se emplearon para la realización del sistema. Aquellos ítems resaltados con color amarillo ya se encontraban en posesión de los autores.

Nombre	Función	Costo unitario (S/.)	Cantidad	Precio (S/.)
Cámara digital	Adquirir muestras palmares.	271	1	271
Tira de LED's	Brindar iluminación uniforme.	5 soles el metro	1 metro	5
Placas de melamina	Para elaborar el recinto de adquisición de imágenes	30	4	120
GPU	Para entrenar la red.	1100	1	1100
CPU	Para programar en Python	600	1	600
Disco Sólido	Para programar en Python	150	1	150
RAM	Para programar en Python	225	2	450
Mainboard	Para programar en Python	350	1	350
Sistema Operativo	Para programar en Python	69	1	69
Mouse y Teclado	Para programar en Python	220	1	220
Fuente de alimentación 12 V	Para alimentar a los LED's	30	1	30
TOTAL				3365
Total Invertido por los autores				426

Tabla (90) Tabla del costo final del trabajo

Elaboración: los autores

CONCLUSIONES

Conclusiones sobre el sensor de adquisición de imágenes:

1. Una buena capacidad de clasificación de imágenes depende de la calidad de las imágenes adquiridas por el sensor, en ese sentido se eligió la cámara Logitech C-922 que es muy parecida a la cámara Logitech C-930 la cual fue empleada por Jalali et. al (2015).

Conclusiones sobre la etapa de preprocesamiento:

2. Se utilizó interpolación bilineal para reducir las dimensiones de las imágenes adquiridas porque genera menos carga computacional ya que ingresar las imágenes en 960x680 a la capa de entradas de la red neuronal convolucional aumentaría de forma dramática el número de parámetros.
3. La aplicación de la máscara para extraer la región de interés permitió también reducir la carga computacional además de facilitarle la tarea de clasificación a la red neural.

Conclusiones sobre la etapa de procesamiento:

4. La red neuronal convolucional elegida tiene casi 5 millones de parámetros, si se considera que las redes neuronales convolucionales tienen la ventaja de poseer parámetros compartidos se concluye que si si hubiese empleado una red neuronal común (que no tiene parámetros compartidos) hubiese requerido muchos más parámetros lo que se hubiese traducido en una mayor carga computacional.
5. A pesar de utilizar la optimización Adam, se muestra que un cambio brusco del factor de aprendizaje provoca que la red ya no converja (ejemplo: pasar el factor de aprendizaje de 0.001 a 0.1).

6. El comportamiento de las curvas (su capacidad de converger) le da una idea al diseñador de red sobre qué tan capaz es dicho modelo de red neuronal para realizar la tarea de clasificar a 80 individuos.
7. Una vez entrenada la red neural, se genera un archivo de aproximadamente 5 MB, el cual puede ser importado y ejecutado por cualquier ordenador lo que le da flexibilidad en su aplicación y distribución.

Conclusiones sobre el recinto de adquisición de imágenes:

8. El recinto tiene dimensiones y un peso (5 KG aproximadamente) que le dan la capacidad de ser transportado a cualquier ambiente.
9. Debido a su material de melamina, el recinto tiene resistencia frente a golpes y accidentes similares, lo que le da durabilidad en el tiempo.

Conclusiones sobre los resultados:

10. Con un TAR de más de 94% y un FRR de 0.053 se concluye que el sistema cumple con el objetivo principal.

RECOMENDACIONES

1. Encender primero los LED de luz blanca antes que la cámara digital, de lo contrario la cámara comenzará a adquirir las muestras con una tonalidad azulada lo cual no es deseable.
2. Posicionar adecuadamente sus manos en el molde palmar, de lo contrario no se garantiza una correcta adquisición de muestras o una correcta predicción por parte de la red.
3. Tener una correcta higiene por parte de los usuarios en sus manos, ya que partículas de tierra o manchas impregnadas en estas, pueden inducir a que la red emita falsos rechazos.
4. No manipular la cámara en ninguna circunstancia, ya que podría desalinearse con el agujero palmar del molde, donde se posiciona la mano del usuario.
5. No manipular la plataforma palmar porque podría desalinearse con respecto a la cámara alojada en la base del recinto.

GLOSARIO

Backpropagation: algoritmo que permite actualizar los pesos o sinapsis de la red de manera que la función de error decrezca.

Biometría: es el reconocimiento automático de individuos en función de sus características físicas y/o de comportamiento.

Capa de Convolución: es la capa que se obtiene como resultado una matriz de entrada con un filtro convolucional.

Capa de entradas: capa que contiene a las unidades de entrada de una red neural.

Capas escondidas: capas que se encuentran entre la capa de entrada y la capa de salida, y son las encargadas de llevar la carga procesal de la red.

Capa de reducción: son capas que simplifican la información contenida en un mapa de características, la hacen a través de una función llamada Max-Pool. Normalmente esta capa se utiliza después de una capa de convolución.

Capa de salidas: capa que contiene los nodos de salida de una red neural.

Convolución Discreta 2D: proceso con el cual los filtros convolucionales interactúan con el mapa de características, aquí el píxel de salida se calcula mediante la suma ponderada de píxeles contenidos en un foco receptivo local sobre el cual el filtro convolucional se superpone.

Cuantificación: es aquel proceso por el cual los valores de un píxel son forzados a un conjunto determinado.

Dermatoglifos: pequeñas impresiones presentes en las palmas de manos y pies.

Entropía Cruzada: función de error que cuantifica la diferencia entre dos distribuciones de probabilidad.

Espectro Electromagnético: distribución energética del conjunto de ondas electromagnéticas.

Factor de aprendizaje: hiperparámetro de una red neuronal que determina que tan rápido puede converger la función de coste a cero.

Filtro Convolutivo: matriz que convoluciona y recorre a la matriz de entrada por medio de pasos o strides.

FRR: *false rejection rate*, se refiere a la métrica de los falsos rechazos.

Función de activación: función aplicada a una entrada lineal para obtener una salida no lineal en redes neuronales.

Función de error o coste: función que calcula el error producido por una red en cada una de sus iteraciones.

GPU: unidad de procesamiento gráfico o tarjeta gráfica que es empleada en el entrenamiento de la red por su capacidad de realizar cálculos similares en paralelo.

Gradiente del error: es el cambio del error con respecto a los pesos de la red neural.

Imagen digital: resultado de los procesos de muestreo y cuantificación, se representa como una matriz bidimensional.

Interpolación bilineal: es un método de la variación de la resolución espacial, es usado en el trabajo para re-escalar las imágenes adquiridas con la cámara digital (sensor).

Líneas o pliegues palmares: líneas robustas ubicadas en la superficie (epidermis) palma de la mano.

Luz: parte de la radiación espectro electromagnético que puede ser percibida por el ojo humano.

Max-Pool: función con la cual se obtiene una capa de reducción, toman la unidad o activación más fuerte de un área de la matriz que se quiere reducir.

Muestreo: proceso de tomar muestras discretas de aquellos valores lumínicos correspondientes a una imagen.

Neurona artificial: unidad de procesamiento básica de una red neuronal contenida en cualquiera de las capas de una red.

Normalización: proceso por el cual los píxeles de una imagen son forzados a tener valores entre 0 y 1.

Optimización: es el algoritmo por el cual los pesos de la red se actualizan en función de parámetros como la gradiente del error, el factor de aprendizaje, el peso anterior, etc.

Optimización Adam: es la optimización utilizada en el presente trabajo para actualizar los pesos de la red, utiliza estimaciones de primer y segundo momento.

Pesos: representan a las sinapsis o conexiones que unen a las neuronas artificiales de una capa con las demás neuronas de otras capas en una red neural.

Rectified Linear Unir (ReLU): es una función de activación aplicada a los nodos de las capas escondidas y usada comúnmente en redes neurales convolucionales.

Red neuronal: es un procesador masivamente distribuido en paralelo conformado por unidades de procesamiento simples, que tiene una propensión natural a almacenar conocimiento experimental y hacerlo disponible para su uso.

Redes neuronales convolucionales: variantes de las redes neuronales normales, presentan conexiones parciales y utilizan convoluciones para obtener las unidades de las capas de convolución. Son utilizadas para resolver problemas del campo de la visión.

Regla de la cadena: es la fórmula para la derivada de dos funciones y que es empleada en el cálculo de la gradiente del error con respecto a los pesos de la red neuronal.

Sensor: en el trabajo se refiere al elemento que interactúa con la luz para adquirir imágenes.

Softmax: función de activación aplicada a los nodos de la capa de salida de una red neuronal, tiene el objetivo de transformar sus salidas en una distribución de probabilidad, es usada para tareas de clasificación.

Stride: hiperparámetro que le permite al filtro convolucional avanzar por todo el mapa de entradas o mapa de características.

TAR: *true acceptance rate*, se refiere a la métrica de los verdaderos reconocimientos.

Tensor: matriz tridimensional que representa a múltiples mapas de características y reducción apilados unos sobre otros en una determinada capa.

FUENTES DE INFORMACIÓN

FUENTE BIBLIOGRÁFICAS

Gonzalez, R & Woods, R. (2007). *Digital Image Processing*. 3era edición.
Prentice Hall

Goodfellow, I., Bengio, Y. & Courville, A. (2017). *Deep Learning*. The MIT
Press.

Haykin, S (2008). *Neural Networks and Learning Machines*. 3era edición.
Prentice Hall.

Nielsen, M. (2015). *Neural Networks and Deep Learning*. Determination Press.

Viñuela, P. & Galván, I. (2004). *Redes Neuronales Artificiales: Un Enfoque
Práctico*. Prentice Hall.

FUENTE ELECTRÓNICAS

Albawi, S., Mohammed, T. & Al-Zawi, Z. (2017). *Understanding of a
Convolutional Neural Network*. 2017 International Conference on
Engineering and Technology. Recuperado de
<https://ieeexplore.ieee.org/document/8308186>

- Ambert, K. (8 de Julio de 2017). *Convolutional Neural Networks, Part 1: Historical Significance*. Intel AI. Recuperado de <https://www.intel.ai/convolutional-neural-networks-historical-significance/#gs.44c47g>
- Binvignat, Almagia, Lizana, Olave. (2012) *Biometric Aspects of the Hand in Chilean Individuals*. International Journal of Morphology. Recuperado de https://www.researchgate.net/publication/289040290_Biometric_aspects_of_the_hand_in_Chilean_individuals
- Cios, K. (2017). *Deep Neural Networks – A Brief History*. Virginia Commonwealth University and IITiS Polish Academy of Sciences. Recuperado de <https://arxiv.org/ftp/arxiv/papers/1701/1701.05549.pdf>
- Chatfield, K., Simonyan, K., Vedaldi, A. & Zisserman, A. (2014) *Return of the Devil in the Details: Delving Deep into Convolutional Nets*. Cornell University. Recuperado de <https://arxiv.org/abs/1405.3531>
- Cheng, J., Sun, Q., Zhang, J. & Zhang, Q. (2017). *Supervised Hashing with Deep Convolutional Features for Palmprint Recognition*. CCBR. Lecture Notes in Computer Science, vol 10568. Recuperado de https://link.springer.com/chapter/10.1007/978-3-319-69923-3_28
- Cortez, P. M. (2017). *Espectroscopia de Infrarrojo para Todos... y 51 espectros de alimentos consumidos en México*. Recuperado de https://ciatej.mx/files/divulgacion/divulgacion_5a43b7c09fdc1.pdf
- Dian, Dongmei, Liu & Sun. (2016). *Contactless Palmprint Recognition Based On Convolutional Neural Network*. 2016 IEEE 13th International Conference on Signal Processing (ICSP). Recuperado de <https://ieeexplore.ieee.org/document/7878049>

Dong, X., Mei, L. & Zhang, J. (2018). *Palmpoint Recognition Based on Deep Convolutional Neural Networks*. 2nd International Conference on Computer Science and Intelligent Communication. Recuperado de <https://www.clausiuspress.com/conferences/ACSS/CSIC%202018/G4149.pdf>

Fukushima, K. (1980). *Neocognitron: A Self-organizing Neural Network Model*. Biological Cybernetics. Recuperado de <https://www.rctn.org/bruno/public/papers/Fukushima1980.pdf>

Genovese, A., Piuri, V., Plataniotis, K. N. & Scotti, F. (2019). *PalmNet: Gabor-PCA Convolutional Networks for Touchless Palmpoint Recognition*. IEEE Transactions on Information Forensics and Security (Volume: 14, Issue: 12 , Dec. 2019). Recuperado de <https://ieeexplore.ieee.org/document/8691498>

He, K., Ren, S., Sun, J. & Zhang, X. (2015). *Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification*. Microsoft Research. Recuperado de https://www.cv-foundation.org/openaccess/content_iccv_2015/papers/He_Delving_Deep_into_ICCV_2015_paper.pdf

Jalali, A., Mallipeddi, H. & Lee, M. (2015) *Deformation Invariant and Contactless Palmpoint Recognition Using Convolutional Neural Network*. Proceedings of the 3rd International Conference on Human-Agent Interaction. Recuperado de <https://dl.acm.org/citation.cfm?id=2814977&dl=ACM&coll=DL>

- Kingma, D. P. & Lei Ba, J. (2015). *Adam: A Method for Stochastic Optimization*. 3rd International Conference for Learning Representations. Recuperado de <https://arxiv.org/abs/1412.6980>
- Krizhevsky, A., Sutskever, I. & Hinton, G. E. (2012) *ImageNet Classification with Deep Convolutional Neural Networks*. Universidad de Toronto. Recuperado de <https://www.cs.toronto.edu/~fritz/absps/imagenet.pdf>
- LeCun, Y., Haffner, P., Bottou, L. & Bengio, Y. (1998). *Object Recognition with Gradient-Based Learning*. AT&T Shanon Lab. Recuperado de <http://yann.lecun.com/exdb/publis/pdf/lecun-99.pdf>
- LeCun, Y. & Ranzato, M. A. (2013). *Deep Learning Tutorial*. ICML Atlanta. Recuperado de <https://speakerdeck.com/baojie/deep-learning-tutorial-at-icml-2013-by-yann-lecun-and-marcaurelio-ranzato>
- Martínez, L., Portilla, A., Rojas, S. & Oviedo, K. (2016). *Relación entre la Clasificación de los Dermatoglifos y la presencia de Labio y Paladar Hendido no Sindrómico en pacientes de la Clínica de Ortodoncia de la USTA* (tesis de grado). Recuperado de <https://repository.usta.edu.co/handle/11634/9671>
- Sun, Q., Zhang, J., Yang, A. & Zhang, Q. (2017). *Palmpoint Recognition with Deep Convolutional Features*. *Advances in Image and Graphics Technologies*. IGTA 2017. Communications in Computer and Information Science, vol 757. Recuperado de https://link.springer.com/chapter/10.1007/978-981-10-7389-2_2
- Svoboda, J., Masci, J. & Bronstein, M. M. (2016). *Palmpoint Recognition Via Discriminative Index Learning*. 23rd International Conference on Pattern Recognition. Recuperado de <https://ieeexplore.ieee.org/document/7900298>