



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO



TECNOLÓGICO NACIONAL DE MÉXICO

INSTITUTO TECNOLÓGICO DE LA PAZ
DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN
MAESTRIA EN SISTEMAS COMPUTACIONALES

Clasificador robusto de huellas digitales basado en redes neuronales artificiales

QUE PARA OBTENER EL GRADO DE
MAESTRO

PRESENTA:
Karla Gallardo Ruiz

DIRECTOR DE TESIS:
Dr. Saul Martínez Díaz

LA PAZ, BAJA CALIFORNIA SUR, MÉXICO, ABRIL 2025.

Agradecimientos

A lo largo de este recorrido académico, descubrí que una tesis no se construye en solitario, sino con el apoyo invaluable de personas que, de una u otra forma, apoyaron cada paso. Hoy, con profunda gratitud, dedico estas palabras:

A Dios, por darme vida, propósito y la oportunidad de alcanzar esta meta y por tejer los hilos invisibles que nos conectan con las personas y oportunidades adecuadas en el momento preciso.

A mi familia

A mis padres, por su amor incondicional, por creer en mí incluso cuando yo dudo. A mi compañero de vida Fernando Madrigal por su apoyo constante. Gracias por caminar a mi lado en esta etapa tan importante.

A mi director de tesis, **Dr. Saul Martínez Díaz** por sus clases de procesamiento de imágenes, reconocimientos de patrones y redes neuronales que germinaron en esta idea. Por su confianza en mi trabajo y guiarme con sabiduría, mucha paciencia y generosidad.

A mis amigos, personas extraordinarias que aparecen en el camino. A **Oscar Ruiz Bernal** que fue un gran apoyo e impulso cuando mi proyecto parecía estancarse. A **Cesar Higuera Verdugo**, por ser el empujón final que necesitaba. Tu ayuda concreta en la etapa de conclusión fue decisiva para que esta tesis no se quedara en el cajón, sino que llegara a su meta final. Gracias por animarme a cruzar la línea de meta y por apoyarme con tu experiencia.

A mis profesores y colegas, por compartir sus conocimientos y enriquecer mi formación. Y a todos aquellos que, de una manera u otra, contribuyeron a que este logro fuera posible.

Karla Gallardo Ruiz

Declaración de autenticidad

Por la presente declaro que, salvo cuando se haga referencia específica al trabajo de otras personas, el contenido de esta tesis es original y no se ha presentado total o parcialmente para su consideración para cualquier otro título o grado en esta o cualquier otra Universidad. Esta tesis es resultado de mi propio trabajo y no incluye nada que sea el resultado de algún trabajo realizado en colaboración, salvo que se indique específicamente en el texto.

Resumen

Esta tesis presenta el desarrollo de un clasificador robusto de huellas digitales basado en redes neuronales artificiales (RNA) y redes neuronales convolucionales (CNN), diseñado para mejorar la precisión en la identificación de huellas incluso en condiciones adversas, como imágenes con ruido o distorsiones. El sistema propuesto integra técnicas de procesamiento digital de imágenes para extraer características clave de las huellas, como minucias (bifurcaciones y terminaciones de crestas), y emplea modelos de aprendizaje automático para clasificarlas y compararlas con una base de datos previamente registrada.

Se exploraron dos enfoques principales:

- 1 **Redes Neuronales Artificiales (RNA)**: Se implementó un perceptrón multicapa (MLP) para clasificar huellas a partir de minucias extraídas, logrando una precisión del 88 % con configuraciones optimizadas.
- 2 **Redes Neuronales Convolucionales (CNN)**: Se ajustaron modelos preentrenados (AlexNet y GoogleNet) mediante transferencia de aprendizaje, demostrando un rendimiento superior, con GoogleNet alcanzando hasta un 94.96 % de precisión en validación.

El estudio destaca las ventajas de las CNN sobre las RNA tradicionales, especialmente en el manejo de variabilidad en las imágenes y la reducción de falsas aceptaciones. Los resultados experimentales confirman la hipótesis de que las redes neuronales mejoran significativamente la robustez en la clasificación, incluso con huellas de baja calidad.

Este trabajo contribuye al campo de la biometría al ofrecer un análisis comparativo de arquitecturas de redes neuronales, evaluando su eficiencia, tiempos de ejecución y aplicabilidad en escenarios reales. Como futuras líneas de investigación, se plantea ampliar el conjunto de datos y explorar modelos híbridos para incrementar aún más la precisión del sistema.

Palabras clave: clasificación de huellas digitales, redes neuronales artificiales, redes neuronales convolucionales, biometría, procesamiento de imágenes, aprendizaje automático.

Abstract

This thesis presents the development of a robust fingerprint classifier based on artificial neural networks (ANNs) and convolutional neural networks (CNNs), designed to improve the accuracy of fingerprint identification even under unfavorable image conditions, such as noise or distortions. The proposed system integrates digital image processing techniques to extract key fingerprint features, such as minutiae (bifurcations and ridge endings), and employs machine learning models to classify and match fingerprints against a pre-registered database.

Two main approaches were explored:

- 1 **Artificial Neural Networks (ANNs)**: A multilayer perceptron (MLP) was implemented to classify fingerprints based on extracted minutiae, achieving an accuracy of 88% with optimized configurations.
- 2 **Convolutional Neural Networks (CNNs)**: Pretrained models (AlexNet and GoogleNet) were fine-tuned using transfer learning, demonstrating superior performance, with GoogleNet reaching up to 94.96% accuracy in validation.

The study highlights the advantages of CNNs over traditional ANNs, particularly in handling image variability and reducing false acceptance rates. Experimental results confirm the hypothesis that neural networks significantly enhance classification robustness, especially in noisy or incomplete fingerprint images.

This work contributes to the field of biometrics by offering a comparative analysis of different neural network architectures, providing insights into their efficiency, execution times, and applicability in real-world scenarios. Future research directions include expanding the dataset and exploring hybrid models to further improve recognition rates.

Keywords: fingerprint classification, artificial neural networks, convolutional neural networks, biometrics, image processing, machine learning.

Índice general

1. Introducción	1
1.1. Introducción	1
1.2. Antecedentes	3
1.2.1. Estado de la investigación	4
1.2.2. Reconocimiento de patrones	4
1.2.3. Clasificación con redes neuronales artificiales	5
1.2.4. Clasificación de huellas digitales	7
1.2.5. Tendencias	7
1.3. Objetivo	8
1.4. Justificación	9
1.5. Alcance	10
1.6. Hipótesis	10
2. Marco teórico	11
2.1. Análisis y representación de la huella digital	11
2.1.1. Huella digital	12
2.1.2. Puntos característicos de las huellas digitales	12
2.1.3. Fases del reconocimiento mediante huellas dactilar	13
2.1.4. Técnicas de clasificación de huellas digitales	14
2.1.5. Procesamiento de la huella digital	16
2.1.6. Segmentación	16
2.1.7. Normalización	17

2.1.8.	Estimación de la orientación de las crestas en la imagen	18
2.1.9.	Estimación de la frecuencia de las crestas de la huella	20
2.1.10.	Filtrado	21
2.1.11.	Binarización	22
2.1.12.	Adelgazamiento	23
2.1.13.	Extracción de las características de la huella	24
2.1.13.1.	Identificación de minucias	24
2.1.13.2.	Análisis y obtención de minucias	25
2.1.13.3.	Medidas de rendimiento en reconocimiento de huellas digitales .	26
2.2.	Reconocimiento de patrones	27
2.2.1.	Clasificador	27
2.2.2.	Tipos de clasificadores	27
2.2.2.1.	Redes neuronales artificiales	27
2.2.2.2.	Deep Learning (Aprendizaje profundo)	27
2.2.2.3.	Redes neuronales convolucionales	28
2.2.3.	Algoritmos de clasificación de huella dactilar	28
2.2.3.1.	Técnicas de clasificación basadas en patrones de minucias . . .	29
2.3.	Redes neuronales artificiales	31
2.3.1.	Neurona biológica	31
2.3.2.	Neurona artificial	32
2.3.2.1.	Modelo general de la neurona artificial.	32
2.3.3.	Red neuronal artificial	34
2.3.4.	Ventajas de la red neuronal artificial	35
2.3.5.	Estructura de una red neuronal artificial	36
2.3.6.	Funcionamiento de una red neuronal artificial	37
2.3.7.	Tipos de redes neuronales artificiales	39
2.3.8.	Mecanismos de aprendizaje	40
2.3.8.1.	Aprendizaje supervisado.	41
2.3.8.2.	Aprendizaje no supervisado.	41
2.3.8.3.	Aprendizaje híbrido.	42

2.3.8.4.	Aprendizaje reforzado.	42
2.4.	Redes Neuronales Convolucionales (CNN)	43
2.4.1.	Modelos de aprendizaje profundo en las CNN.	45
2.4.2.	Transferencia de aprendizaje (transfer learning)	47
2.4.3.	Extracción de características	47
2.4.3.1.	CNN pre-entrenada Alexnet	49
2.4.3.2.	GoogLeNet (Inception V1).	52
3.	Desarrollo	57
3.1.	Procesamiento de la imagen.	57
3.1.1.	Segmentación	58
3.1.2.	Normalización	58
3.1.3.	Orientación	60
3.1.4.	Filtrado	60
3.1.5.	Binarización	60
3.1.6.	Adelgazamiento.	61
3.2.	Extracción de las características de la huella.	62
3.3.	Implementación de la red neuronal artificial.	63
3.3.1.	Descripción de la arquitectura de la red	63
3.3.2.	Configuración de la RNA	64
3.3.3.	Entrenamiento de la RNA	65
3.3.4.	Evaluación del modelo	66
3.3.5.	Herramientas y entorno de desarrollo	67
3.3.6.	Pruebas y resultados	68
3.3.7.	Modelo de clasificación multiclase.	69
3.4.	Implementación con redes neuronales convolucionales.	76
3.4.1.	Conjunto de datos utilizados.	76
3.4.2.	Implementación de clasificador con AlexNet.	76
3.4.2.1.	Preprocesamiento de los datos.	76
3.4.2.2.	División del conjunto de datos	77

3.4.2.3.	Modificación de la arquitectura de AlexNet	77
3.4.2.4.	Entrenamiento de la red	77
3.4.2.5.	Evaluación del modelo	79
3.4.2.6.	Prueba de clasificación	81
3.4.2.7.	Interpretación de resultados	82
3.4.2.8.	Ajustes para mejorar el rendimiento de la red Alexnet	82
3.4.2.9.	Evaluación del modelo ajustado	84
3.4.2.10.	Análisis visual	85
3.4.3.	Implementación de clasificador con GoogleNet	85
3.4.3.1.	Entrenamiento de la red en GoogleNet.	86
3.4.3.2.	Evaluación de la red con datos de validación.	88
4.	Resultados	94
4.1.	Resultados obtenidos	94
4.1.1.	Base de Datos	94
4.1.2.	Métricas de Evaluación	94
4.1.3.	Resultados Experimentales del Algoritmo Basado en Redes Neuronales Artificiales	95
4.1.3.1.	Tiempos de Ejecución	96
4.1.3.2.	Desempeño de la Red Neuronal	96
4.1.3.3.	Tasas de Reconocimiento y Errores de Clasificación	97
4.1.3.4.	Tasas Promedio de Reconocimiento	98
4.1.3.5.	Análisis de Resultados	98
4.1.4.	Resultados del algoritmo Basado en Redes Neuronales Convolucionales	99
4.1.4.1.	Resultados de GoogleNet	100
4.1.4.2.	Comparativa entre AlexNet y GoogleNet	100
4.1.4.3.	Conclusiones Parciales	101
4.1.5.	Conclusiones Generales y Futuras Líneas de Trabajo	101

Índice de figuras

2.1. Huella dactilar.	12
2.2. Ejemplo de los principales tipo de minucias.	13
2.3. Diferentes tipos característicos en una huella digital.	13
2.4. Proceso de reconocimiento biométrico.	14
2.5. Zonas de alto y bajo contraste.	17
2.6. Huella digitalizada y su imagen normalizada respectivamente.	18
2.7. Imagen después del proceso de orientación	20
2.8. Estimación de la frecuencia de la cresta.	21
2.9. Imagen Binarizada.	23
2.10. Image adelgazada.	24
2.11. Minucias en huella dactilar.	24
2.12. Detección en 3x3.	25
2.13. Detección de minucias.	26
2.14. Estructura de una Neurona Biológica.	32
2.15. Modelo de una Neurona Artificial.	33
2.16. Neurona Artificial.	33
2.17. Estructura jerárquica de un sistema basado en redes neuronales artificiales. . . .	34
2.18. Estructura de una Red Neuronal Artificial.	37
2.19. Arquitectura de una Red Neuronal convolucional.	43
2.20. Funcionamiento del operador de max_pooling en un mapa de características en 1D y 2D.	45
2.21. Ejemplo de red convolucional.	46

2.22. Proceso en la capa de convolución.	50
2.23. RELU (Rectified linear unit).	51
2.24. Operación Pooling.	52
2.25. Operación Softmax.	53
2.26. Arquitectura GoogleNet.	56
3.1. Segmentación.	59
3.2. Orientación de las crestas.	60
3.3. Imagen Filtrada.	61
3.4. Imagen Binarizada.	61
3.5. Esqueleto de imagen.	62
3.6. Minucias identificadas.	62
3.7. Arquitectura de la Red Neuronal.	64
3.8. Resumen del Entrenamiento de la Red Neuronal - Alcanzado Mínimo Gradiente.	66
3.9. Rendimiento del Modelo durante el Entrenamiento.	67
3.10. Matrices de confusión de la evaluación integral del modelo.	68
3.11. Huella identificada.	69
3.12. Gráfica del rendimiento del modelo en términos de precisión.	70
3.13. Regresión.	71
3.14. Esquema de la Red Neuronal Feed-Forward de clasificación multiclase.	72
3.15. Resumen entrenamiento modelo clasificación multiclase.	74
3.16. Matriz confusión modelo clasificación multiclase.	75
3.17. Progreso y resultados del entrenamiento del modelo	79
3.18. Resultados de Clasificación: Predicciones y Confianza en la Identificación de usuarios.	81
3.19. Progreso del Entrenamiento: Precisión del 100% en Validación y Detalles del Ciclo de Entrenamiento.	83
3.20. Resultados de clasificación: Predicción y niveles de confianza en la identificación de usuarios.	86
3.21. Progreso del Entrenamiento de Modelo GoogleNet.	88

3.22. Progreso del Entrenamiento del Modelo con Exactitud y Pérdida por Época e Iteración.	89
3.23. Ejemplos de clasificación del modelo GoogleNet	92

Índice de tablas

2.1. Ventana 3x3 análisis pixel P	25
2.2. Relación crossing number con tipo de minucia	25
2.3. Capas Red Alexnet	50
2.4. Un clasificador softmax con 1000 clases de salida similar al clasificador principal softmax.	55
3.1. Tiempo promedio	66
3.2. Porcentaje de aciertos y falsas aceptaciones en reconocimiento de huellas dactilares al variar el número de capas del modelo.(Muestra 1)	70
3.3. Porcentaje de aciertos y falsas aceptaciones en reconocimiento de huellas dactilares con diferentes configuraciones de capas.(Muestra 2)	70
3.4. Arquitectura de la red AlexNet modificada	78
4.1. Tiempos de ejecución	96
4.2. Relación entre profundidad del modelo y eficiencia computacional: Tiempos de entrenamiento vs. reconocimiento	97
4.3. Relación entre complejidad del modelo (Capas) y exactitud en identificación	97
4.4. Relación entre complejidad del modelo (Capas) y Exactitud en pruebas de Identificación de individuo P2	98
4.5. Porcentaje de éxito en reconocimiento según profundidad del modelo	98
4.6. Análisis comparativo entre AlexNet y GoogleNet: métricas de precisión y eficiencia	100

Capítulo 1

Introducción

1.1. Introducción

Las imágenes son ampliamente utilizadas como una de las fuentes más importantes de información, sobre todo en el contexto de las aplicaciones centradas en el ser humano, tales como: seguridad, desarrollo de sistemas biométricos, juegos multimedia de interacción hombre-máquina, robótica, y videoconferencias.

Debido al avance tecnológico en sensores que captan las imágenes digitales, se puede constatar el crecimiento de su utilización en diversos campos y aplicaciones. Uno de estos campos es el desarrollo de sistemas biométricos, el cual ha crecido en estos últimos años. La biometría es una tecnología de seguridad basada en el reconocimiento de una característica física e intransferible de las personas, tal como iris, retina, rostro, sistema vascular, palma de la mano, huella digital, y voz. Siendo la huella digital el interés de estudio en el presente trabajo.

Los sistemas biométricos se basan en el reconocimiento de patrones. Un patrón biométrico está compuesto por un conjunto de características fisiológicas que definen un vector asociado unívocamente a una persona. Utilizando esta información, es posible verificar e identificar un vector de características dentro de una base de datos con información de otras personas. Las aplicaciones de estos sistemas pueden ir desde intereses gubernamentales, militares, privados o comerciales, hasta las aplicaciones forenses o policiales [1].

En la actualidad se utilizan varios tipos de sistemas biométricos para la identificación en

tiempo real; dentro de los más populares se encuentran los de huella digital. Éstos básicamente son de dos tipos: Sistema Automático de Autenticación de Huellas Dactilares (AFAS) y Sistema de Identificación Automático de Huella Dactilar (AFIS). En un AFAS la entrada es la identidad de la persona y la imagen de la huella dactilar de esa persona; y la salida es una respuesta de SI o NO, indicando si la imagen de entrada pertenece a la persona cuya identidad es proporcionada. En un AFIS la entrada es solo la imagen de la huella dactilar y la salida es una lista de identidades de personas que pueden tener la huella dada, además de una puntuación de cada identidad indicando el grado de similitud entre ésta y la huella dada [2].

El rasgo biométrico más utilizado para la identificación de las personas es la huella digital; debido a que permite diferenciar perfectamente a los individuos y su grado de aceptabilidad es relativamente alto. Sin embargo se requiere de cierta cooperación por parte del individuo, para que la imagen adquirida de la huella tenga la suficiente calidad, así como para permitir el empleo de algoritmos de reconocimiento sencillos. En muchas aplicaciones, las condiciones de captación no son lo suficientemente buenas y, por tanto, la mala calidad de las huellas adquiridas obliga al empleo de algoritmos complejos, tanto en la etapa de extracción de características, como en la etapa de reconocimiento de patrones [3].

Considerando lo antes mencionado, en este proyecto de investigación, se desarrolla un clasificador de huellas digitales que permite su correcta identificación aún con la presencia de condiciones desfavorables en la imagen captada. El sistema extrae los puntos característicos de la huella, realiza una comparación con una base de datos asociada con un conjunto de usuarios previamente almacenados.

Existen diversos métodos para el reconocimiento de patrones y de huellas, así como técnicas de clasificación, Esta investigación aborda el problema de clasificación de huellas digitales mediante el desarrollo de un clasificador robusto que combina técnicas de procesamiento digital de imágenes con dos enfoques principales: redes neuronales artificiales (RNA) y redes neuronales convolucionales (CNN). Las RNA son un paradigma de aprendizaje y procesamiento automático inspirado en la forma en que funciona el sistema nervioso de los animales. Se trata de un sistema de interconexión de neuronas en una red que colabora para producir un estímulo de salida. La importancia de las RNA es su útil comportamiento de aprender y reconocer, al aplicar relaciones entre objetos. En este sentido, se utilizan las RNA como una herramienta para resolver

problemas difíciles. Para la implementación de la RNA se ha definido el tipo, número de capas, la manera en que se conectan las neuronas, el tipo de entrenamiento o regla de aprendizaje.

Por otro lado, las CNN representan un avance significativo en el campo del aprendizaje profundo (deep learning), especialmente en el procesamiento de imágenes. A diferencia de las RNA tradicionales, las CNN están diseñadas para capturar patrones espaciales jerárquicos mediante operaciones de convolución, agrupamiento (pooling) y capas totalmente conectadas. Entre sus ventajas destacan la capacidad de extraer características automáticamente sin necesidad de un preprocesamiento manual exhaustivo, lo que las hace altamente eficientes para tareas de visión artificial. En este proyecto, se implementaron dos arquitecturas CNN preentrenadas, AlexNet y GoogleNet, adaptadas mediante transferencia de aprendizaje para clasificar huellas digitales. Estas redes demostraron un buen rendimiento al procesar imágenes con variaciones de escala, rotación o iluminación, superando en precisión a los métodos tradicionales basados en RNA.

La combinación de ambos enfoques permite un análisis comparativo integral, evaluando no solo la precisión en la clasificación, sino también el tiempo de ejecución y la robustez frente a imágenes de baja calidad. Los resultados obtenidos refuerzan la hipótesis de que el uso de redes neuronales, tanto artificiales como convolucionales, mejora significativamente la identificación de huellas digitales, incluso en presencia de ruido o distorsiones, ofreciendo así una solución viable para aplicaciones biométricas en entornos reales.

1.2. Antecedentes

En los últimos años, ha cobrado mayor importancia el uso de las técnicas de clasificación de imágenes o patrones, principalmente para aumentar la eficacia de los sistemas de identificación y verificación biométricos. Por esta razón actualmente se investiga en la obtención y desarrollo de algoritmos de clasificación de mayor eficacia y eficiencia. Numerosas investigaciones han abordado este problema a través de métodos de decisión estadística, de correlación y aprendizaje estadístico así como las redes neuronales artificiales e implementan la combinación de diferentes algoritmos.

Un área que se ha beneficiado con la aplicación de estos métodos es el reconocimiento automatizado de huellas digitales, por lo que una parte muy importante en los sistemas de reco-

nocimiento automatizado de huellas lo constituye el algoritmo de clasificación. A continuación, se da a conocer el estado del tema de investigación, y el estado del arte respecto al reconocimiento de patrones, clasificación de huellas digitales, uso de redes neuronales artificiales y redes neuronales convolucionales.

1.2.1. Estado de la investigación

Los sistemas para el reconocimiento de las huellas digitales cada vez son más prácticos y en la actualidad se han corregido muchos errores y se han logrado sistemas más confiables con la implementación de algoritmos de clasificación de patrones. El uso más común de la clasificación de huellas digitales es la validación, identificación y autenticación de usuarios en sistemas biométricos, como los que se mencionan a continuación.

- Sistema de identificación de visitantes en una prisión. Donde los visitantes e internos pasan por procedimientos de verificación para no poder intercambiar identidades durante la visita.
- Licencias de conducir.
- Sistemas de pago.
- Control de Frontera. Se usan terminales biométricas para comprobar identidad y de esta manera reducir el tiempo de inspección de inmigrantes para viajeros autorizados.

1.2.2. Reconocimiento de patrones

El Reconocimiento de patrones es un área de carácter multidisciplinario cuyo objetivo de estudio son los procesos de identificación, caracterización, clasificación y pronóstico sobre objetos, físicos o abstractos con el propósito de extraer información que permita establecer propiedades de o entre conjuntos de dichos objetos. Entre las áreas de aplicación del reconocimiento de patrones se encuentran actualmente:

- Biometría: Detección de voz, tratamiento de imágenes, clasificación de huellas digitales, procesamiento de lenguaje natural.

- Defensa: Reconocimiento automático de objetivos, guía y control de armamento.
- Medicina: Diagnósticos, análisis de imágenes y clasificación de enfermedades.
- Diseño de vehículos.
- Estudio de recursos naturales: agricultura, geología, recursos forestales.
- Industria: Diseño asistido por computadora, pruebas y control de calidad.
- Aplicaciones policiales.

1.2.3. Clasificación con redes neuronales artificiales

El uso de las redes neuronales artificiales en diferentes campos del conocimiento (control, procesado de la señal, sistemas expertos, predicción en series temporales, etc.) han sufrido un crecimiento exponencial en los últimos años [5], [6]. Este número tan elevado se debe a su aplicación en problemas de clasificación, modelización y predicción.

Las redes neuronales han emergido como una herramienta importante para clasificación. La reciente actividad investigadora en la clasificación neuronal ha establecido que las RNA son una alternativa prometedora en comparación a métodos de clasificación alternativos. La ventaja de las redes neuronales radica en los siguientes aspectos teóricos:

- Las redes neuronales son métodos adaptativos dirigidos por datos que pueden ajustarse a éstos sin ninguna especificación explícita de forma distribucional o funcional para el modelo subyacente.
- Las redes neuronales son aproximadores funcionales universales dado que puede ajustar cualquier función con una precisión arbitraria. Ya que cualquier procedimiento de clasificación busca una relación funcional entre el grupo al que pertenece y los atributos del objeto, una identificación precisa de esta función subyacente es de vital importancia.
- Las redes neuronales son modelos no lineales, lo que les hace flexibles en el modelado de las complejas relaciones del mundo real.

- Las redes neuronales son capaces de estimar las probabilidades a posteriori, lo que proporciona la base para establecer las reglas de clasificación y llevar a cabo un análisis estadístico.

La popularidad adquirida por las redes neuronales artificiales puede atribuirse a su capacidad para el reconocimiento de patrones, clasificación y aproximación de funciones; siendo las redes multicapa feedforward o perceptrones multicapa los clasificadores basados en redes neuronales más ampliamente estudiados y usados.

La computación neuronal provee un acercamiento mayor al reconocimiento y percepción humana que los métodos tradicionales de cálculo. Las redes neuronales artificiales presentan resultados razonables en aplicaciones donde las entradas presentan ruido o las entradas están incompletas [7]. Algunas de las áreas de aplicación de las RNA son las siguientes:

- Análisis y Procesado de señales
- Predicción
- Procesado natural del lenguaje
- Conversión de texto a voz
- Compresión de imágenes
- Reconocimiento de Imágenes
- Reconocimiento de caracteres
- Problemas de combinatoria
- Control de Procesos
- Filtrado de ruido
- Robótica
- Procesado del Lenguaje
- Diagnósticos médicos
- Modelados de sistemas

- Modelos económicos y financieros

Entre los avances o desarrollos tecnológicos que permiten la realización de la computación neuronal destacan el software de simulación, los aceleradores hardware, los chips de silicio y los procesadores ópticos.

1.2.4. Clasificación de huellas digitales

En los sistemas actuales de verificación de huella digital, la clasificación y comparación entre huellas es la fase más crítica; debido a que el proceso resulta muy complejo cuando las huellas sufren desplazamientos, rotaciones, distorsiones o su calidad es baja. Existen un gran número de algoritmos automáticos de clasificación los cuales presentan un buen rendimiento cuando las huellas digitales son de buena calidad, sin embargo comparar y clasificar huellas de baja calidad o incompletas sigue siendo un foco de interés científico. Los algoritmos de comparación desarrollados hasta la fecha pueden clasificarse en:

- Basados en correlación entre bloques de píxeles de la imagen
- Basados en minucias
- Basados en características del patrón de crestas o texturas
- Redes Neuronales Artificiales; un sistema de auto aprendizaje utiliza las direcciones de las crestas y otros aspectos de las imágenes de las huellas como entrada de entrenamiento para diferenciar las diferentes clases.

1.2.5. Tendencias

Los avances tecnológicos en el campo de las huellas digitales fueron dejados de lado durante algunos años ya que la tecnología del reconocimiento de individuos avanzó hacia otras fronteras, como por ejemplo el reconocimiento facial. Los investigadores han empezado a trabajar nuevamente con las huellas dactilares, algunos estudios van más allá del reconocimiento de individuos.

- Los científicos de la Universidad Sheffield Hallam han desarrollado un lector que a través del dedo, más específicamente de la huella dactilar puede indicar qué ha tocado el individuo e incluso qué ha ingerido últimamente. La nueva tecnología fue nombrada Mass Spectrometry Imaging, que ofrece una firma química de la huella dactilar [8].
- Sistema de predicción de la Huella Dactilar. Científicos de la Universidad de Göttingen, en Suecia, han desarrollado un nuevo procedimiento que, con la ayuda de la tecnología y los cálculos matemáticos, anticipa cómo será cualquier huella dactilar en el futuro.
- Tecnología MALDI-MSI («Espectrometría de masas por ionización/desorción por láser asistida por matriz»), que se utiliza normalmente para clasificar las moléculas de un tejido. Aplicada al desciframiento de huellas digitales, Investigadores de la Universidad de Sheffield Hallam, en el Reino Unido, han creado una novedosa técnica que permite establecer, a partir de una simple huella digital encontrada en la escena de un delito, información sobre la medicación, la dieta o el estilo de vida del sospechoso. Esta tecnología aporta mucha más información sobre los sospechosos que otros medios tradicionales.

1.3. Objetivo

Desarrollar un algoritmo robusto para identificar y clasificar huellas digitales, basado en técnicas de procesamiento digital de imágenes y redes neuronales artificiales, y así mostrar la utilidad de las RNA en el proceso de clasificación de patrones distorsionados y con ruido Los objetivos específicos planteados en este proyecto son:

- Investigar los avances en el tema de clasificación de huellas dactilares.
- Estudiar el Estado del arte de clasificadores de patrones con redes neuronales Artificiales.
- Implementar algoritmos basados en procesamiento digital de imágenes, los cuales permitan extraer características visibles de las huellas digitales.
- Diseñar e implementar el algoritmo clasificador de patrones con una red neuronal artificial
- Adquirir base de datos de huellas digitales.

- Implementar algoritmos de clasificación basados en Redes Convolucionales a través de transferencia de aprendizaje
- Hacer un estudio experimental comparativo entre el clasificador desarrollado basado en redes neuronales Artificiales y los basados en Redes convolucionales, en términos de tasas de error y tiempos de ejecución.

1.4. Justificación

La identificación dactilar debe ser muy precisa con un índice de error muy bajo, por lo que un sistema de identificación y verificación a través de huellas digitales debe tener la capacidad de clasificarlas con exactitud, aún con la presencia de variabilidad y ruido en las imágenes.

En una tecnología biométrica, la combinación de un algoritmo clasificador y los sensores de huellas dactilares es muy importante en relación a los índices de error. Los algoritmos de reconocimiento de huellas digitales utilizados actualmente suelen ser ineficientes en imágenes distorsionadas y con ruido.

Algunos problemas que presenta el reconocimiento de huellas digitales son:

- Identificación basada en fragmentos de huellas: En algunas ocasiones no se capta toda la huella digital.
- Variabilidad de las huellas digitales: esto se debe a diferencias de humedad, lubricación del dedo, presión aplicada al momento de tomar la huella.
- Variabilidad en el sensor: variabilidad en la escala, el pixelado y otras propiedades de la imagen.
- Ruido, variaciones de posición.

En el trabajo se ataca el problema de clasificar huellas digitales con la presencia de la variabilidad y ruido antes mencionado.

1.5. Alcance

Se centra en el desarrollo e implementación de un algoritmo clasificador orientado al proceso completo de la huella digital, las funciones que contiene el algoritmo permiten extraer los puntos característicos de la huella, para su comparación con una base de datos previamente almacenada. Así mismo se implementan las redes convolucionales Alexnet y Googlenet basados en Deep Learning para realizar un estudio comparativo

1.6. Hipótesis

El uso de redes neuronales artificiales en algoritmos de reconocimiento de huellas digitales permite mejorar significativamente los resultados de clasificación con presencia de ruido en la imagen captada.

Capítulo 2

Marco teórico

Este capítulo se divide en varias secciones que abordan los fundamentos teóricos necesarios para comprender el proceso de identificación y clasificación de huellas digitales, así como las técnicas y herramientas utilizadas en el desarrollo. En primer lugar, se introduce el análisis y representación de la huella digital, donde se describen los conceptos básicos relacionados, sus características principales y los puntos característicos que las hacen únicas. Se detallan las fases del reconocimiento, desde el procesamiento de la imagen de la huella digital hasta la extracción de características y la comparación con una base de datos. Además, se explican las técnicas de clasificación.

El capítulo concluye con una introducción a las redes neuronales convolucionales (CNN), un tipo de red neuronal profunda que ha demostrado ser especialmente efectiva en tareas de visión artificial y procesamiento de imágenes. Se describen los modelos de aprendizaje profundo en las CNN, el proceso de entrenamiento desde cero y la extracción de características. Además, se presentan dos modelos específicos de CNN preentrenados, AlexNet y GoogleNet, que se utilizan en este trabajo para la clasificación de huellas digitales.

2.1. Análisis y representación de la huella digital

Para poder profundizar en la identificación de los humanos a través de la huella digital y poder comprender en qué consiste dicha identificación, tenemos que conocer que es una huella dactilar, así como sus características respecto a la identificación y los procesos que pueden ser

utilizados para mejorar la calidad de las imágenes de dichas huellas.

2.1.1. Huella digital

Una huella digital es el patrón característico, único y diferente de un dedo; es la impresión visible o moldeada que produce el contacto de las crestas papilares de un dedo de la mano (generalmente se usan el dedo pulgar o el dedo índice) sobre una superficie [4]. (Figura 2.1a)

2.1.2. Puntos característicos de las huellas digitales

Una huella dactilar aparece como una serie de líneas oscuras en la cual la característica más evidente es un patrón de crestas (representan los relieves) y valles (espacios en blanco de bajo relieve) intercalados entre sí. (Figura 2.1b)



(a) Representación gráfica.



(b) Puntos característicos.

Figura 2.1: Huella dactilar.

De acuerdo a la investigación en el reconocimiento de la huella digital, éstas son distinguidas por las minucias denominados “puntos característicos”; que son algunos puntos anormales en las crestas. Entre la variedad de tipos de minucias mencionadas en las literaturas, son dos principalmente significativas y de amplio uso, uno se llama terminación, que es el fin inmediato de una cresta; el otro se llama bifurcación, que es el punto en la cresta desde la cual se derivan dos ramas. (Figura 2.2)

Sin embargo, en un proceso de identificación se pueden considerar los diversos tipos de minucias representadas en la figura 2.3.

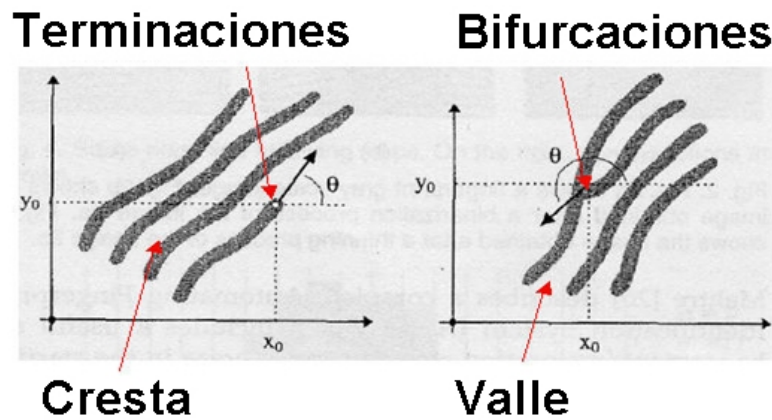


Figura 2.2: Ejemplo de los principales tipo de minucias.

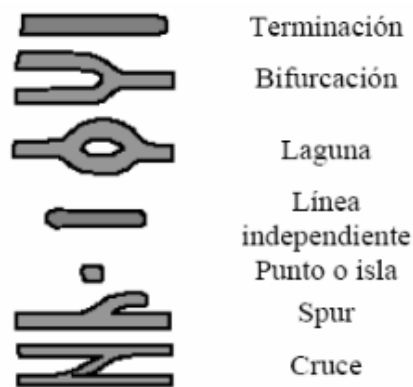


Figura 2.3: Diferentes tipos característicos en una huella digital.

2.1.3. Fases del reconocimiento mediante huellas dactilar

Las técnicas de identificación biométrica siguen el mismo esquema con alguna variante, en el caso de las huellas dactilares es muy semejante el esquema que sigue el proceso de identificación y el de verificación, dicho esquema se detalla a continuación. (figura 2.4)

Como muestra el diagrama de bloques después de adquirir la imagen de la huella dactilar de un usuario y extraer las características, se almacena el patrón en una base de datos para una posterior identificación o verificación. En cualquiera de los dos procesos, identificación o verificación, se genera un patrón siguiendo tres etapas: preprocesamiento de la imagen, que permitirá mejorar la calidad de la imagen y localizar la parte con información útil, la extracción de minucias en la cual se determinan los puntos característicos de la huella y el reconocimiento de patrones, donde se emplea una técnica o algoritmo de clasificación realizando una comparación con los datos almacenados en la base de datos.

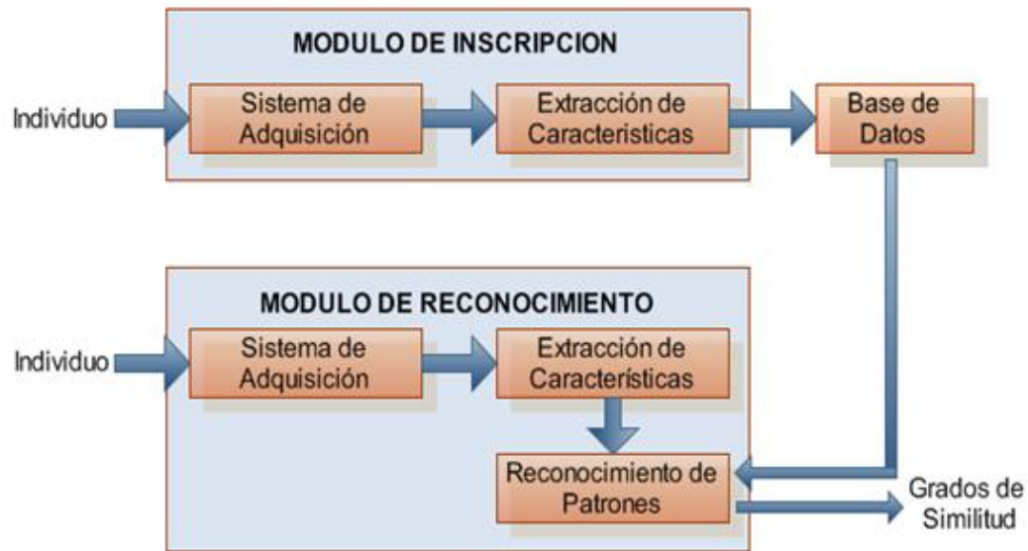


Figura 2.4: Proceso de reconocimiento biométrico.

Las comparaciones entre patrones se realizan por semejanza, siendo un valor de umbral especificado, el que determina el grado de similitud que deben tener las imágenes comparadas para que la identificación se dé por válida o no.

2.1.4. Técnicas de clasificación de huellas digitales

La clasificación de huellas se realiza normalmente en sistemas de identificación que manejan grandes bases de datos, con el fin de limitar en lo posible el campo de búsqueda de las huellas que se desean conocer. Dicha clasificación se basa en la particular distribución de las crestas y valles de la huella en la zona central de la imagen. Las características extraídas de las huellas, que han sido empleadas por las diferentes técnicas de clasificación son [6].

- Flujo de líneas de cresta. Son líneas que fluyen paralelas a las líneas de las crestas las cuales mantienen la orientación local de la estructura de las crestas.
- Campo de orientación. Es una de las características de la imagen utilizada por la mayoría de las técnicas, ya que contiene toda la información necesaria para poder efectuar la clasificación y puede calcularse con suficiente precisión.

- Puntos singulares. Se determinan analizando el campo de orientación. existen varias aproximaciones para determinar los puntos con exactitud.
- Respuesta a filtros de Gabor. El filtrado de la estructura de crestas con filtros localmente orientados de Gabor, pueden proporcionar información de textura útil para la clasificación.

Las técnicas de clasificación de huellas digitales pueden dividirse en:

- Técnicas basadas en reglas. Establecen la clasificación de las huellas a partir del número de puntos característicos que presentan y su distribución espacial. Esta forma automática de clasificación responde al esquema empleado tradicionalmente mediante inspección visual. La mayoría de estas técnicas emplean el índice de Poincaré para determinar los puntos característicos, a partir del campo de orientación [6], [7]. Esta técnica es apropiada en el caso de huellas tintadas ya que se adquiere una huella completa a diferencia de huellas adquiridas on line donde estas técnicas son poco eficientes dado que gran parte de la estructura de crestas que contiene las minucias no es adquirida.
- Técnicas estructurales. estos esquemas se basan en la representación de las características de bajo nivel mediante estructuras de alto nivel. Dicha representación utiliza estructuras de datos simbólicas, como los grafos y los árboles, ya que permiten organizar jerárquicamente la información.
- Técnicas estadísticas. Los vectores de características, de longitud fija, extraídos de cada huella son clasificados estadísticamente. El clasificador más utilizado es el k- Vecinos más cercanos.
- Técnicas basadas en Redes Neuronales Artificiales. La mayoría de estos métodos se basan en la utilización de perceptores multicapa y vectores de características derivados del campo de orientación. Algunas técnicas también incluyen los puntos característicos.
- Técnicas basadas en Redes Neuronales Convolucionales (CNN). Las CNN han revolucionado la clasificación de huellas al automatizar la extracción de características mediante operaciones de convolución, agrupamiento (pooling) y capas totalmente conectadas. A diferencia de las RNA tradicionales, las CNN procesan la imagen directamente, aprendiendo

jerárquicamente desde bordes y texturas hasta patrones globales de crestas. Arquitecturas como AlexNet, GoogleNet o ResNet —adaptadas mediante transfer learning— han demostrado superioridad en precisión y robustez, especialmente con huellas parciales o distorsionadas. Además, técnicas como data augmentation y dropout mejoran su generalización.

- Técnicas basadas en múltiples clasificadores. Representan la tendencia actual de los sistemas de clasificación de huellas digitales. Estas técnicas hacen uso de los esquemas de combinación de clasificadores, ya que la elección adecuada de las diferentes estrategias de clasificación individuales puede proporcionar información complementaria para mejorar el funcionamiento de un clasificador global.

2.1.5. Procesamiento de la huella digital

En el procesamiento de la huella es de mucha importancia mejorar la calidad de la imagen, es decir eliminar las señales indeseadas que aparecen por las condiciones con las que fue captada dicha imagen, este proceso recibe el nombre de filtrado de imágenes, en él se pueden aplicar una o varias técnicas como: segmentar, normalizar la imagen, estimar la orientación de las crestas, filtrar, binarizar y obtener el área de interés, fases que se detallan a continuación.

2.1.6. Segmentación

Permite separar la región del frente de la imagen de la región del fondo. La región del frente corresponde al área donde se imprimió la huella que contiene las crestas y valles, la cual es el área de interés. El fondo corresponde a la región que se encuentra por afuera de los bordes del área de la huella, la cual no contiene ninguna información válida de la huella. Cuando el algoritmo de extracción de características se aplica al fondo de la imagen, da como resultado la extracción de falsas características.

Esta segmentación es empleada para descartar la región del fondo, con lo cual se facilita la confiabilidad de la extracción de características de la huella.

En una imagen de huella, la región del fondo generalmente exhibe un muy bajo nivel de variación en el valor en la escala de grises, mientras que la región del frente tiene una alta variación, según muestra la Figura 2.5. Por lo tanto, un método basado en un umbral de varianza es utilizado para realizar la segmentación.



Figura 2.5: Zonas de alto y bajo contraste.

Primeramente, la imagen es dividida en bloques y se calcula la varianza de la escala de grises para cada bloque de la imagen. Si la varianza es menos que el umbral global, entonces el bloque es asignado a la región del fondo, de otra manera es asignada a la parte del frente.

La varianza para un bloque de tamaño $W \times W$ es definida por:

$$V(k) = \frac{1}{W^2} \sum_{i=0}^{W-1} \sum_{j=0}^{W-1} (I(i, j) - M(k))^2 \quad (2.1)$$

Donde $V(k)$ es la varianza del bloque k . $I(i, j)$ es el valor en la escala de grises del pixel de la posición (i, j) y $M(k)$ es la media de la escala de grises del bloque k .

2.1.7. Normalización

La normalización es usada para estandarizar la intensidad de la imagen, ajustando el rango de la escala de grises de forma que quede dentro del rango de valores deseados. Donde $I(i, j)$ representa el valor en la escala de grises del pixel (i, j) y $N(i, j)$ representa el valor normalizado en la escala de grises del pixel (i, j) . La normalización de la imagen está definida por.

$$N(i, j) = \begin{cases} M_o + \sqrt{\frac{V_o(I(i, j) - M)^2}{V}} & \text{si } I(i, j) > M \\ M_o - \sqrt{\frac{V_o(I(i, j) - M)^2}{V}} & \text{de lo contrario} \end{cases} \quad (2.2)$$

Donde M y V son la media y varianza estimada para $I(i, j)$ respectivamente, M_o y V_o son la media y la varianza deseadas respectivamente. La normalización no cambia la estructura de las crestas en la imagen de una huella, es realizado para estandarizar el nivel de la variación en la escala de grises, lo cual facilita los subsecuentes procesos para el mejoramiento de la imagen [8].

La Figura 2.6 muestran un ejemplo de cómo el proceso de normalización modifica la intensidad de los píxeles.



Figura 2.6: Huella digitalizada y su imagen normalizada respectivamente.

2.1.8. Estimación de la orientación de las crestas en la imagen

La estimación de la orientación de las crestas es fundamental para los procesos subsecuentes, como el filtrado de la imagen; donde se requiere la orientación local de las crestas para realizar la mejora de la imagen efectivamente. Los ángulos del campo de orientación representan la dirección en que fluyen las crestas de la huella. Esto nos permite revelar características intrínsecas de la topología de las huellas [8].

Los pasos para calcular la orientación del píxel (i, j) son los siguientes:

- a. En un bloque de tamaño $W \times W$ es centrado en el píxel (i, j) de la imagen normalizada.

- b. Para cada pixel en el bloque se calcula el gradiente ; es la magnitud del gradientes en la dirección x y y respectivamente. Aplicando las máscaras G_x y G_y se realiza una convolución de cada pixel con una máscara de 3 x 3 conocida como el operador Sobel, el cual está definido como.

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} * \mathbf{A} \quad \text{and} \quad G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * \mathbf{A} \quad (2.3)$$

Valores de la máscara G_y y valores de la máscara G_x respectivamente. Para cada punto de la imagen un estimado de la magnitud del gradiente resultante está dado por.

$$G = \sqrt{G_x^2 + G_y^2} \quad (2.4)$$

- c. La orientación local para el pixel (i, j) es estimada utilizando la siguiente ecuación.

$$V_x(i, j) = \sum_{u=i-\frac{W}{2}}^{i+\frac{W}{2}} \sum_{v=j-\frac{W}{2}}^{j+\frac{W}{2}} 2\partial x(u, v)\partial y(u, v) \quad (2.5)$$

$$V_y(i, j) = \sum_{u=i-\frac{W}{2}}^{i+\frac{W}{2}} \sum_{v=j-\frac{W}{2}}^{j+\frac{W}{2}} \partial x(u, v)2\partial y(u, v) \quad (2.6)$$

$$\theta(i, j) = \frac{1}{2} \tan^{-1} \frac{V_y(i, j)}{V_x(i, j)} \quad (2.7)$$

donde $\theta(i, j)$ es la estimación de los mínimos cuadrados de la orientación local para el pixel (i, j) .

- d. Se alisa el campo de vectores de orientación local utilizando un filtro de Gauss. La orientación de la imagen es primeramente en un campo de vectores continuos, lo que se define como.

$$\phi x(i, j) = \cos(2\theta)(i, j) \quad (2.8)$$

$$\phi y(i, j) = \text{sen}(2\theta)(i, j) \quad (2.9)$$

donde ϕ_x y ϕ_y son los componentes x y y del campo de vectores, respectivamente. Después de que el campo de vectores es calculado, el filtro de Gauss es entonces ejecutado de la siguiente manera.

$$\phi'_x(i, j) = \sum_{w=-\frac{W_\phi}{2}}^{\frac{W_\phi}{2}} \sum_{v=-\frac{W_\phi}{2}}^{\frac{W_\phi}{2}} G(u, v) \phi_x(i - vw, j - vw) \quad (2.10)$$

$$\phi'_y(i, j) = \sum_{w=-\frac{W_\phi}{2}}^{\frac{W_\phi}{2}} \sum_{v=-\frac{W_\phi}{2}}^{\frac{W_\phi}{2}} G(u, v) \phi_y(i - vw, j - vw) \quad (2.11)$$

donde G es el filtro de Gauss de tamaño $W_\phi \times W_\phi$

e. Finalmente el campo de orientación alisado O de un pixel (i, j) está definido por:

$$O(i, j) = \frac{1}{2} \tan^{-1} \frac{\phi'_y(i, j)}{\phi'_x(i, j)} \quad (2.12)$$

En la Figura 2.7 puede apreciarse un ejemplo de la etapa de orientación, mostrando la dirección principal de cada bloque.



Figura 2.7: Imagen después del proceso de orientación

2.1.9. Estimación de la frecuencia de las crestas de la huella

En adición a la orientación local de la imagen de la huella, otro de los parámetros importantes para la construcción del filtro de Gabor es la frecuencia local de las crestas; es el número de

veces que aparecen las crestas y los valles donde se forman minucias. El primer paso para la estimación de la frecuencia es dividir la imagen en bloques de tamaño $W \times W$. Lo siguiente es proyectar el valor en la escala de grises de cada pixel en el bloque en dirección ortogonal de la dirección del vector de orientación local de las crestas. Esta proyección forma una onda casi sinusoidal que corresponde con las crestas locales del bloque, Figura 2.8.

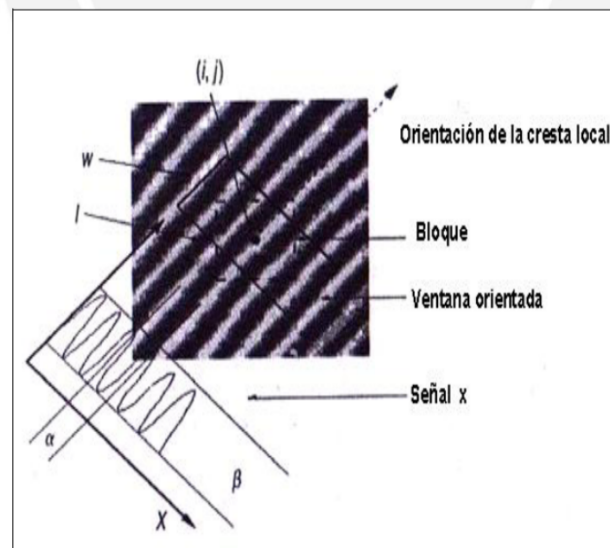


Figura 2.8: Estimación de la frecuencia de la cresta.

2.1.10. Filtrado

Una herramienta muy utilizada en el procesado de imágenes es el filtro de Gabor. Se caracteriza por tener propiedades de frecuencia selectiva y orientación selectiva y tienen óptimas resoluciones tanto en el dominio de la frecuencia como en el dominio del espacio [4].

Una vez que el vector de orientación y la información de la frecuencia de las crestas han sido determinadas, estos parámetros son utilizados para construir el filtro de Gabor. Un filtro de Gabor de dos dimensiones consiste en un plano de ondas sinusoides de particular orientación y frecuencia, moduladas sobre una envoltura Gaussiana.

El filtro de Gabor es utilizado debido a que tiene una selección de propiedades de frecuencias y orientaciones. Estas propiedades permiten que el filtro obtenga la máxima respuesta de las crestas en base a una orientación y frecuencia específica en la imagen de la huella. Este filtro

permite preservar la estructura de las crestas y reduciendo el ruido.

La parte real de una función de un filtro de Gabor, está dada por la siguiente expresión:

$$\begin{aligned} G(x, y, \theta, f) &= \exp \\ x_\theta &= x \cos \theta + y \sin \theta \\ y_\theta &= x \sin \theta + y \cos \theta \end{aligned} \quad (2.13)$$

θ es la orientación del filtro de Gabor, f es la frecuencia de la onda del coseno, σ_x y σ_y son la desviación estándar de la envoltura Gausseana a través de los ejes x , y , respectivamente, x_θ , y_θ definen el eje x , y de las coordenadas del marco del filtro respectivamente [8].

El filtro de Gabor es aplicado a la imagen de la huella por una convolución espacial de la imagen con el filtro. La convolución del pixel (i, j) en la imagen requiere de la orientación $O(i, j)$ y la frecuencia $F(i, j)$ correspondientes al pixel. Lo cual se aplica en la siguiente expresión.

$$E(i, j) = \sum_{w=-\frac{W_x}{2}}^{\frac{W_x}{2}} \sum_{v=-\frac{W_y}{2}}^{\frac{W_y}{2}} G(u, v; \phi(i, j), F(i, j))N(i - u, j - v) \quad (2.14)$$

Donde O es el campo vectorial de orientación de la imagen, F es la frecuencia de las crestas, N es la imagen normalizada, W_x y W_y son el ancho y alto de la máscara del filtro de Gabor.

2.1.11. Binarización

La imagen mejorada, se encuentra en escala de grises, es decir una imagen que contiene 8 bits por cada píxel, y con 256 posibilidades diferentes de tonos de gris.

En este paso se transforma la imagen a un formato binario, unos o ceros, lo que permite diferenciar claramente y procesar las minucias en la imagen.

La mayor parte de los algoritmos de extracción de características de las huellas digitales, trabaja con imágenes binarias, donde hay solo dos niveles que interesan: los pixeles negros que representan las crestas y los pixeles blancos que representan los valles. Este proceso incrementa el contraste entre las crestas y los valles en la imagen de la huella y consecuentemente facilita la extracción de las características.

Una propiedad del filtro de Gabor es que la imagen resultante tiene un valor de la media

igual a cero, por lo que la binarización de la imagen puede realizarse con un umbral o límite de cero. El proceso de binarización requiere examinar el nivel en la escala de grises de cada pixel y si el valor es mayor que el umbral global, entonces el valor del pixel es asignado a uno de lo contrario es asignado a cero. La salida es una imagen binaria que contiene dos niveles de información las crestas con uno y los valles con cero.

El resultado final del proceso de Binarización se puede apreciar en la Figura 2.9



Figura 2.9: Imagen Binarizada.

2.1.12. Adelgazamiento

Proceso también denominado esqueletización, por el cual, las crestas en la imagen, que se representan mediante unos binarios, son reducidas en espesor, para de esta manera obtener una imagen en la que todas las crestas tengan solamente 1 píxel de ancho.

Es el paso final del proceso de mejora de las imágenes que es realizado antes de la extracción de características. El adelgazamiento es una operación que sucesivamente erosiona los píxeles de las crestas hasta que tengan un ancho de un píxel [9].

La aplicación del algoritmo de adelgazamiento sobre la imagen de huella digital preserva la conectividad de las crestas como formando una versión del esqueleto de la imagen binaria.

En la Figura 2.10 se puede apreciar un ejemplo de cómo el grosor de las líneas se reduce después de aplicar un algoritmo de adelgazamiento.

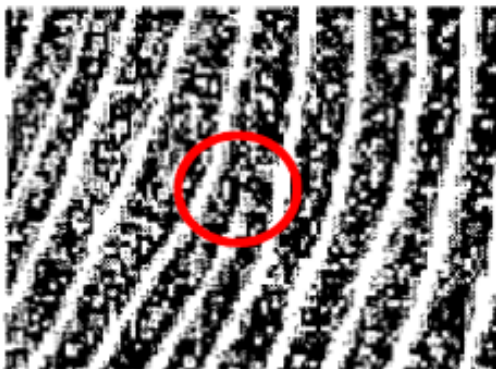


Figura 2.10: Image adelgazada.

2.1.13. Extracción de las características de la huella

2.1.13.1. Identificación de minucias

Una vez obtenida la imagen “esqueleto” de la huella digital, todas las crestas tienen un espesor de un píxel, lo que permite realizar una búsqueda de las minucias, es decir las bifurcaciones (Figura 2.11a) y terminaciones (Figura 2.11b).



(a) Terminación.



(b) Bifurcación.

Figura 2.11: Minucias en huella dactilar.

Para la identificación de minucias, se toma ventanas de la imagen de 3 por 3 píxeles (Figura 2), y sobre estas aplicar la siguiente fórmula de Crossing Number:

P_4	P_3	P_2
P_5	P	P_1
P_6	P_7	P_8

Tabla 2.1: Ventana 3x3 análisis pixel P

Valor de CN	Minucia
0	Punto aislado
1	Terminación
2	Cresta continua
3	Bifurcación

Tabla 2.2: Relación crossing number con tipo de minucia

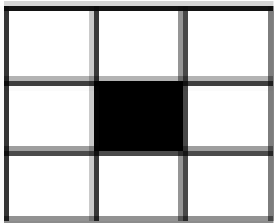
$$CN(p) = \frac{1}{2} \sum_{i=1}^{i=8} |p_{i \bmod 8} - p_{i+1}| \quad (2.15)$$

donde p_i es el valor de cada píxel adyacente al píxel central de la ventana. Dependiendo del resultado de esta ecuación, el píxel central de la ventana puede ser identificado como terminación, bifurcación o cresta continua.

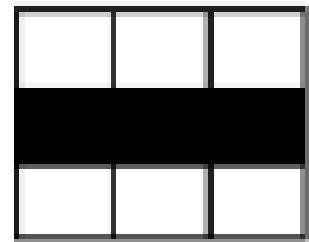
El resultado de aplicar Crossing Number determina el tipo de minucia que se aprecia en la siguiente tabla:

2.1.13.2. Análisis y obtención de minucias

Se puede observar las minucias en una ventana de 3 x 3 en las siguientes figuras

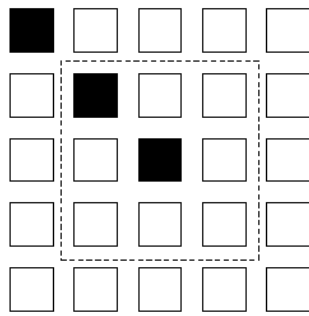


(a) Punto aislado.

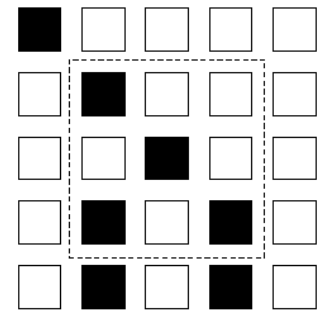


(b) Cresta continua.

Figura 2.12: Detección en 3x3.

(a) $CN = 1$

(a) Terminación.



(b) Bifurcación.

Figura 2.13: Detección de minucias.

2.1.13.3. Medidas de rendimiento en reconocimiento de huellas digitales

El comportamiento de un sistema biométrico basado en huellas digitales con umbral de decisión específico puede ser mostrado a través de las siguientes tasas de medida básicas:

- FAR. “False Accept Rate” o tasa de falsa aceptación, una medida de probabilidad que la huella de un impostor sea aceptada por el sistema. Representa el porcentaje de usuarios no autorizados que son incorrectamente identificados como usuarios válidos y aceptados por un sistema.
- FRR. “False Reject Rate” o tasa de falso rechazo, es una medida de probabilidad que la huella de un individuo sea rechazada. Representa el porcentaje de usuarios autorizados que son incorrectamente rechazados por el sistema.

Los dos parámetros anteriores son funciones de grado de seguridad deseado. Las medidas de rendimiento para los sistemas de clasificación consisten en principalmente de dos indicadores:

- Matriz de confusión. Describe los errores cometidos por cada clase, es una matriz comparativa entre la clase verdadera (filas) versus las clases asignadas (columnas) por el clasificador. Una matriz de confusión de un clasificador ideal contendrá valores numéricos altos solamente en la diagonal principal, siendo los valores restantes cero.
- Precisión. Es un indicador global que nos muestra el porcentaje de la base de datos que ha sido correctamente clasificada, este valor depende de si se ha considerado algún criterio de rechazo a las imágenes de huellas que contiene información muy pobre para ser clasificada.

2.2. Reconocimiento de patrones

Dentro de los trabajos de reconocimiento y clasificación de patrones se han desarrollado varias metodologías y algoritmos para clasificar, algunos desde la década de los 50's, y aún son tema de estudio. La tarea de clasificación consistió en hacer particiones del espacio de características para formar regiones, donde a cada región se le asignará una categoría o clase.

2.2.1. Clasificador

Es un elemento que proporciona una clase etiquetada como salida a partir de un conjunto de características tomadas como entradas. Una manera de construir un clasificador es tomar un conjunto de ejemplos etiquetados y tratar de definir una regla que pueda asignar una etiqueta a cualquier otro dato de entrada.

2.2.2. Tipos de clasificadores

2.2.2.1. Redes neuronales artificiales

Las redes neuronales son una técnica de aproximación paramétrica útil para construir modelos de densidad. El modelo de red neuronal habitual que utiliza este algoritmo consiste en una red con una capa de entrada con tantos nodos como entradas tengan, una capa oculta con un número de nodos variable que dependerá de las características del problema, y una capa de salida con tantos nodos como posibles salidas tenga.

2.2.2.2. Deep Learning (Aprendizaje profundo)

El aprendizaje profundo es una rama del aprendizaje automático (machine learning) que utiliza varias capas de procesamiento no lineal para aprender representaciones útiles de características directamente a partir de los datos. Los modelos de aprendizaje automático pueden lograr una precisión extrema en la clasificación de objetos, que en ocasiones supera el nivel del rendimiento humano. Los modelos se entrenan mediante un amplio conjunto de datos etiquetados y arquitecturas de redes neuronales que contienen muchas capas [10]. La presión de un modelo de aprendizaje profundo depende en gran medida de la cantidad de datos utilizados

para su entrenamiento. Los modelos más precisos pueden requerir miles o millones de muestras, y el entrenamiento puede llevar mucho tiempo [10].

El aprendizaje profundo se puede aplicar a diferentes problemas, como:

- Clasificación de imágenes.
- Reconocimiento de Voz.
- Procesamiento de Lenguaje natural.

2.2.2.3. Redes neuronales convolucionales

Uno de los tipos más populares de redes neuronales profundas son las conocidas como redes neuronales convolucionales (CNN). Una CNN convoluciona las características aprendidas con los datos de entrada y emplea capas convolucionales 2D, lo cual hace que esta arquitectura resulte adecuada para procesar datos 2D, tales como imágenes.

Las CNN eliminan la necesidad de una extracción de características manual, por lo que no es necesario identificar las características utilizadas para clasificar las imágenes. La CNN funciona mediante la extracción de características directamente de las imágenes. Las características relevantes no se entrenan previamente; se aprenden mientras la red se entrena con una colección de imágenes. Esta extracción de características automatizada hace que los modelos de Deep Learning sean muy precisos para tareas de visión artificial, tales como la clasificación de objetos.

2.2.3. Algoritmos de clasificación de huella dactilar

La clasificación de huellas dactilares es un procedimiento en el cual las huellas son agrupadas en una consistente y fiable forma, tal que diferentes impresiones de una misma huella son seleccionadas en un mismo grupo [11].

Existen diversos métodos para el reconocimiento automático de huella dactilar. Las técnicas empleadas dependen de los tipos de patrones comparados. Estas son:

- Técnicas basadas en la comparación de patrones de puntos, en los casos en los que se trabaja con las minucias extraídas de la estructura de las crestas.
- Técnicas basadas en las características de la estructura de crestas y valles.

- Técnicas basadas en la textura de la imagen.

Para el desarrollo de este trabajo se empleó la técnica de basada en patrones de minucias o puntos característicos aplicando Redes Neuronales Artificiales y Redes Neuronales convolucionales.

2.2.3.1. Técnicas de clasificación basadas en patrones de minucias

En los casos en los que los patrones biométricos están definidos por minucias, los métodos de reconocimiento se basan generalmente en técnicas de comparación euclídea. Estos comparadores realizan transformaciones matemáticas entre los patrones de puntos para poder estimar el grado de similitud entre ellos. La mayoría de ellos, mediante procesos iterativos, pueden comparar los patrones con suficiente exactitud, a pesar de que en el proceso de extracción de características se hayan podido perder minucias verdaderas o se hayan podido introducir minucias espurias. Algunos ejemplos importantes de estas técnicas son:

- Técnica de relajación. Es un método de comparación de patrones de puntos mediante el procedimiento matemático de “relajación”. Se trata de un método iterativo en el que, mediante aproximaciones sucesivas, se va desplazando un patrón de puntos sobre el otro, al tiempo que se van ajustando las distancias entre las parejas de minucias más o menos coincidentes. Presentan el inconveniente de que no funcionan bien cuando los patrones a comparar proceden de imágenes muy distorsionadas; situación frecuente en el caso de las huellas dactilares [9].
- Técnicas basadas en el alineamiento de patrones. Según el método anterior la comparación entre dos huellas dactilares suponen la comparación de todas las posibles combinaciones que pueden hacerse con las agrupaciones de minucias de dos huellas. Por este motivo, el tiempo de respuesta del sistema puede hacerse muy grande. Como solución a este problema se han propuesto métodos de reconocimiento que incluyen el alineamiento de las huellas antes de efectuar la comparación. Gracias al proceso de alineamiento se consigue reducir el número de comparaciones necesarias para establecer el grado de similitud entre las huellas, reduciéndose significativamente el tiempo de respuesta. Un procedimiento muy frecuente es alinear las agrupaciones de las huellas con respecto a sus puntos singulares.

La determinación de los puntos singulares conlleva siempre la estimación previa del campo de orientación local de la estructura de las crestas [12].

- Técnicas basadas en la transformada de Hough. Se utiliza esta transformación para el caso de la comparación de patrones de minucias. La transformada de Hough generalizada transforma el problema de la comparación de patrones de puntos en un problema de detección de máximos en el espacio de Hough de los parámetros de transformación. El método busca la mejor transformación posible para que el conjunto de minucias de la huella de entrada se solape con el conjunto de minucias de la huella de la base de datos. Para que dos puntos sean considerados coincidentes deben tener las mismas coordenadas espaciales dentro de unos márgenes de tolerancia, y el mismo ángulo de orientación. El procedimiento permite la coexistencia de puntos no coincidentes. Se hace discreto el conjunto de todas las posibles transformaciones, y para cada transformación se obtiene una puntuación. El espacio de transformaciones está formado por cuádruplas de parámetros del tipo $(s, \theta, \Delta x, \Delta y)$, donde cada parámetro se hace discreto según un conjunto de valores; donde: s , es un factor de escala, Δx y Δy , son las traslaciones de los ejes x y y , respectivamente; y θ , es el ángulo de rotación. El algoritmo se desarrolla en dos partes: en la primera, para cada par de minucias, pertenecientes a los dos patrones que se comparan, se calculan todas las posibles transformaciones que las hacen coincidentes. En la segunda fase, se alinean los dos conjuntos de minucias con los parámetros estimados, y se hace el recuento de todas las parejas de minucias coincidentes dentro de una caja de tolerancia. El valor de la puntuación se escala entre 0 y 99. La transformación cuya puntuación es mayor se toma como la óptima, deduciéndose de ella, el valor de similitud entre los dos patrones comparados. El método proporciona un valor de similitud no muy fiable cuando las huellas contienen un número reducido de minucias, ya que, en este caso, resulta difícil acumular suficiente evidencia en el espacio de la transformada de Hough [12].
- Técnicas basadas en grafos. Existen también varios comparadores topológicos basados en grafos. Estos comparadores toleran bien las transformaciones originadas entre los patrones y los errores de posición de las minucias, las minucias espurias y las minucias perdidas. Las características de las huellas, tales como la posición de los núcleos, la frecuencia de crestas,

el número de crestas entre minucias y toda información de alineamiento son utilizadas para efectuar las comparaciones [12].

- Otras técnicas. La técnica de comparación de patrones puede realizarse también mediante la aplicación de algoritmos genéticos, redes neuronales, simulated annealing, etc. Estos métodos establecen correspondencias iniciales entre las minucias de los patrones que se van a comparar que, durante el proceso de comparación, van modificándose con el fin de minimizar, en cada fase del proceso, una función de ajuste, previamente definida.

2.3. Redes neuronales artificiales

2.3.1. Neurona biológica

Las estructuras de los sistemas biológicos pueden proporcionar bases para el diseño de modelos de sistemas inteligentes. Estos procesan información de una forma no convencional, no requieren modelos de referencia, y se desempeñan de manera exitosa en presencia de incertidumbre, aprenden a realizar nuevas tareas.

A finales del siglo XIX el científico español Santiago Ramón y Cajal demostró que la estructura del sistema nervioso está conformada por una red de células individuales interconectadas entre sí, las neuronas. Además de establecer que dentro de la neurona la información fluye desde las dendritas, pasa por el soma que es el cuerpo de la célula y llega hasta el axón para conectarse a otra neurona, es decir que el flujo de la información es siempre de un mismo sentido. Este descubrimiento fue fundamental para el desarrollo de la neurociencia [13].

Desde un punto de vista funcional cada neurona actúa como un procesador con pequeño poder de procesamiento, donde las entradas al sistema son las dendritas y el axón las salidas. A diferencia de un procesador normal, las neuronas establecen relaciones con las neuronas contiguas traspasando información y trabajando en paralelo 2.14.

La unión o comunicación entre dos neuronas se llama sinapsis. Este es un proceso químico o eléctrico que va en un solo sentido. Esta comunicación o falta de comunicación entre las neuronas contiguas es la que forma la estructura de red neuronal.

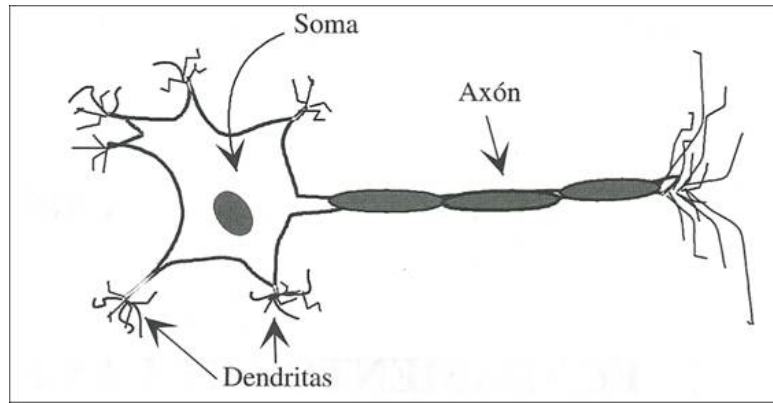


Figura 2.14: Estructura de una Neurona Biológica.

El aprendizaje de una red neuronal se debe a la variación que sufre la intensidad y duración de la sinapsis, esto es, durante la vida de la neurona ésta recibe unos estímulos sinápticos y genera otros. Las neuronas moderan su sinapsis en intensidad y duración para que ésta sea la que presente mejores resultados al sistema en general, de acuerdo a lo aprendido.

2.3.2. Neurona artificial

Una neurona es un procesador elemental tal que a partir de un vector de entrada procedente del exterior o de otras neuronas, proporciona una única respuesta o salida.

2.3.2.1. Modelo general de la neurona artificial.

Los elementos de una neurona artificial se pueden ver en la Figura 2.15, estos son:

- Conjunto de entradas: Representa todos los valores que recibe la neurona. De acuerdo al problema que se esté representando con el modelo neuronal puede ser un sólo valor o varios, de la misma forma puede recibir datos binarios o reales.
- Los pesos sinápticos W_{ij} : Indican la intensidad o fuerza de interacción entre las neuronas presinápticas “ i ” y postsinápticas “ j ”, se representa como un factor que multiplica (para incrementar o disminuir la intensidad), a cada valor del conjunto de entrada cuando este viene de otra neurona.

- La regla de propagación. Proporciona el potencial postsináptico de la neurona, lo más común es utilizar la suma ponderada del producto escalar del conjunto de entradas y los pesos sinápticos.
- La función de activación: Genera el estado actual de la activación de la neurona en función de su estado anterior y su potencial sináptico actual.
- La función de salida: Proporciona la salida de la neurona, normalmente es una función de identidad o se utiliza una función umbral para activar el valor de salida.

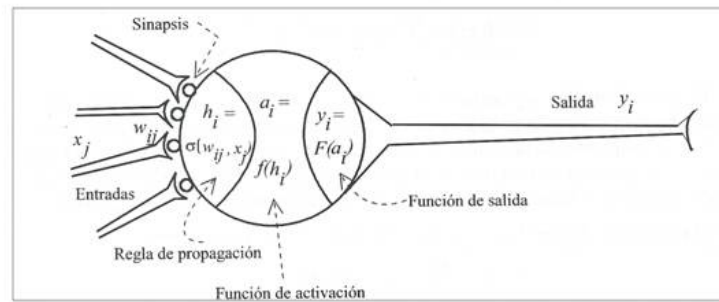
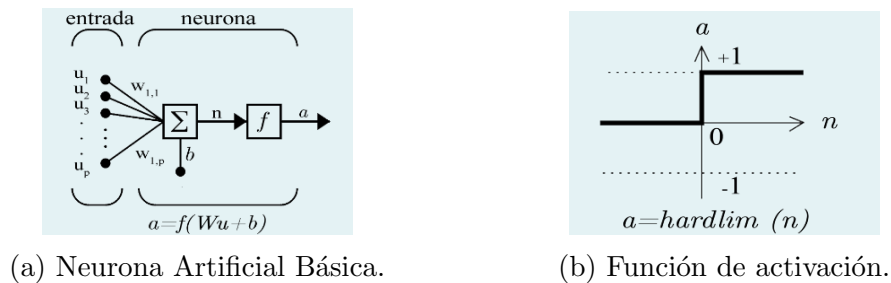


Figura 2.15: Modelo de una Neurona Artificial.

Una neurona básica se muestra en la Figura 5.2. Su comportamiento se puede resumir en que la suma de sus entradas ponderadas activa su salida cuando se supera un umbral fijado por una determinada función de activación 2.16b.



(a) Neurona Artificial Básica.

(b) Función de activación.

Figura 2.16: Neurona Artificial.

La verdadera potencia de las neuronas artificiales radica en la combinación de cientos o miles de ellas, organizadas en capas, varias capas constituyen una red neuronal y una red neuronal o conjunto de ellas pueden constituir un sistema global de proceso, emulando paralelismo de cálculo, memoria distribuida, y adaptabilidad al entorno. (Figura 2.17).

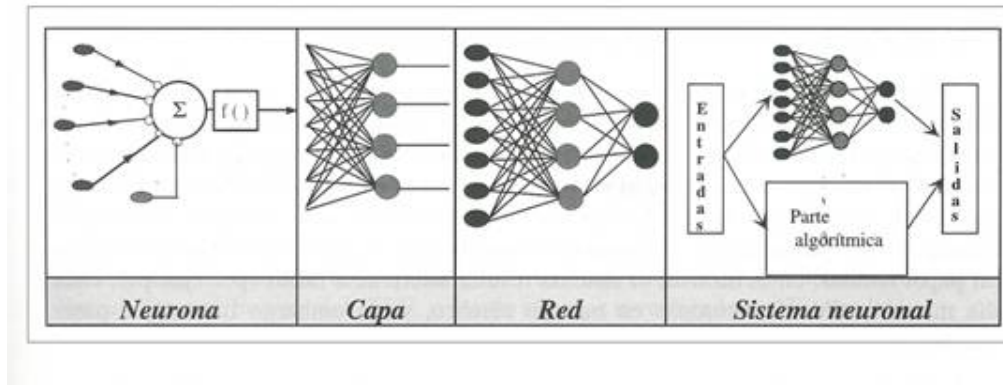


Figura 2.17: Estructura jerárquica de un sistema basado en redes neuronales artificiales.

2.3.3. Red neuronal artificial

Existen numerosas formas de definir a las redes neuronales; desde las definiciones cortas y genéricas hasta las que intentan explicar más detalladamente qué son las redes neuronales. Por ejemplo:

1. Una forma de computación, inspirada en modelos biológicos.
2. Un modelo matemático compuesto por un gran número de elementos procesales organizados en niveles.
3. Un sistema de computación compuesto por un gran número de elementos simples, elementos de procesos muy interconectados, los cuales procesan información por medio de su estado dinámico como respuesta a entradas externas.
4. Redes neuronales artificiales son redes de elementos simples (usualmente adaptativos) interconectados masivamente en paralelo y con organización jerárquica, los cuales intentan interactuar con los objetos del mundo real del mismo modo que lo hace el sistema nervioso biológico [15].

Para una definición formal de red neuronal se tiene que hacer uso del concepto matemático de grafo. a través de ese término se puede definir una red neuronal de la siguiente forma:

Una red neuronal es un grafo dirigido con las siguientes propiedades [16]:

1. A cada nodo i se le asocia un variable de estado X_i .

2. A cada conexión (i, j) de los nodos i y j se le asocia un peso $W_{ij} \in \mathbb{R}$.
3. A cada nodo i se le asocia un umbral θ_i .
4. Para cada nodo i se define una función $f(X_i, W_{ij}, \theta_i)$, que depende de los pesos de sus conexiones, del umbral y de los estados de los nodos j a él conectados. Esta función proporciona el nuevo estado del nodo.

2.3.4. Ventajas de la red neuronal artificial

Debido a su constitución y a sus fundamentos, las redes neuronales artificiales presentan un gran número de características semejantes a las del cerebro. Por ejemplo, son capaces de aprender de la experiencia, de generalizar de casos anteriores a nuevos casos, de abstraer características esenciales a partir de entradas que representan información irrelevante, etc. Esto hace que ofrezcan numerosas ventajas y que éste tipo de tecnología se esté aplicando en múltiples áreas.

Entre las ventajas se incluyen:

- *Aprendizaje Adaptativo.* Capacidad de aprender a realizar tareas basadas en un entrenamiento. Son adaptables debido a la capacidad de autoajuste de los elementos procesales (neuronas) que componen el sistema para obtener resultados específicos.
- *Auto-organización.* Una red neuronal puede crear su propia organización o representación de la información que recibe mediante una etapa de aprendizaje. Cuando las redes neuronales se usan para reconocer ciertas clases de patrones, ellas autoorganizan la información usada. Por ejemplo, la red llamada backpropagation, creará su propia representación característica, mediante la cual puede reconocer ciertos patrones.
- *Tolerancia a fallos.* La destrucción parcial de una red conduce a una degradación de su estructura; sin embargo, algunas capacidades de la red se pueden retener y puede seguir realizando su función. Las redes pueden aprender a reconocer patrones con ruido, distorsionados o incompletos. Esta es una tolerancia a fallos respecto a los datos.
- *Operación en tiempo real.* Una de las mayores prioridades, casi en la totalidad de las áreas de aplicación, es la necesidad de realizar procesos con datos de forma muy rápida. Las

redes neuronales se adaptan bien a esto debido a su implementación paralela. Para que la mayoría de las redes puedan operar en un entorno de tiempo real, la necesidad de cambio en los pesos de las conexiones o entrenamiento es mínimo.

- *Fácil inserción dentro de la tecnología existente.* Con las herramientas computacionales existentes (no del tipo PC), una red puede ser rápidamente entrenada, comprobada, verificada y trasladada a una implementación hardware de bajo coste. Por lo tanto, no se presentan dificultades para la inserción de redes neuronales en aplicaciones específicas, por ejemplo, de control, dentro de los sistemas existentes. De esta manera, las redes neuronales se pueden utilizar para mejorar sistemas en forma incremental y cada paso puede ser evaluado antes de acometer un desarrollo más amplio [15].

2.3.5. Estructura de una red neuronal artificial

Una RNA se compone de un conjunto masivamente paralelo de unidades de proceso muy simples y es en las conexiones entre estas unidades donde reside la inteligencia de la red.

Tienen un gran número de procesadores virtuales interconectados que de forma simplificada simulan la funcionalidad de las neuronas biológicas. En esta simulación, la reorganización de las conexiones sinápticas biológicas se modela mediante un mecanismo de pesos, que son ajustados durante la fase de aprendizaje. En una RNA entrenada, el conjunto de los pesos determina el conocimiento de esa RNA y tiene la propiedad de resolver el problema para el que la RNA ha sido entrenada. Además de los pesos y las conexiones, cada neurona tiene asociada una función matemática denominada función de transferencia. Dicha función genera la señal de salida de la neurona a partir de las señales de entrada. La entrada de la función es la suma de todas las señales de entrada por el peso asociado a la conexión de entrada de la señal.

A continuación se puede ver en la Figura 2.18, la estructura de una red neuronal.

Está constituida por neuronas interconectadas y distribuidas en tres capas (esto último puede variar); el conjunto de una o más capas constituyen una red neuronal. Los datos ingresan por medio de la “capa de entrada”, pasan a través de la “capa oculta” y salen por la “capa de salida”. Cabe mencionar que la capa oculta puede estar constituida por varias capas.

Capa de entrada: es la capa que recibe directamente la información proveniente de las fuentes

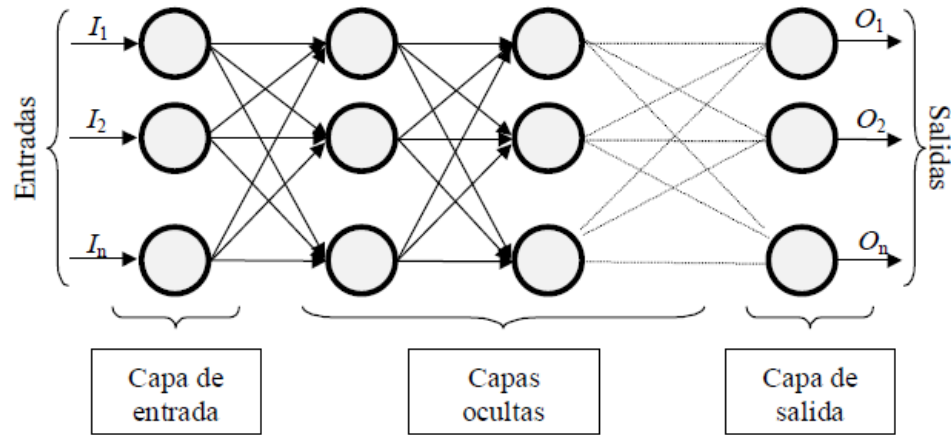


Figura 2.18: Estructura de una Red Neuronal Artificial.

externas de la red.

Capas Ocultas: son internas a la red y no tienen contacto directo con el entorno exterior. El número de niveles ocultos puede estar entre cero y un número elevado. Las neuronas de las capas ocultas pueden estar interconectadas de distintas maneras, lo que determina, junto con su número, las distintas topologías de redes neuronales.

Capa de salidas: transfieren información de la red hacia el exterior; proporcionando la respuesta.

2.3.6. Funcionamiento de una red neuronal artificial

En un RNA, los nodos se conectan por medio de sinapsis; esta estructura de conexiones sinápticas determina el comportamiento de la red. Las conexiones sinápticas son direccionales, es decir, la información sólo puede fluir en un sentido (desde la neurona presináptica a la neurona postsináptica).

Las conexiones entre las neuronas pueden ser excitatorias o inhibitoras, según el signo del peso sináptico asociado a la conexión.

Las conexiones pueden clasificarse también en conexiones intracapa y conexiones intercapa. Las primeras se corresponden con las conexiones entre las neuronas de una misma capa y, la segunda se corresponde a las conexiones entre neuronas de distintas capas [16].

La respuesta o salida de la RNA está dada por tres funciones:

1. Una función de propagación (también conocida como función de excitación), que por lo general consiste en el sumatorio de cada entrada multiplicada por el peso de su interconexión (valor neto). Si el peso es positivo, la conexión se denomina excitatoria; si es negativo, se denomina inhibitoria.
2. Una función de activación, que modifica a la anterior. Puede no existir, siendo en este caso la salida leconocimiento de patronesa misma función de propagación.
3. Una función de transferencia, que se aplica al valor devuelto por la función de activación. Se utiliza para acotar la salida de la neurona y generalmente viene dada por la interpretación que queramos darle a dichas salidas. Algunas de las más utilizadas son la función sigmoidea (para obtener valores en el intervalo $[0,1]$) y la tangente hiperbólica (para obtener valores en el intervalo $[-1,1]$).

Una neurona artificial tiene dos modos de funcionamiento:

1. Las neuronas deben ser previamente entrenadas para reconocer patrones. Esto se hace de la siguiente manera: Se coloca como valores de entrada de la neurona el patrón a aprender (p.ej. usando la neurona del ejemplo, el patrón 1 1 0) y después se incrementan los pesos de las entradas “utilizadas” (en este caso la 1 y la 2, que son las que están a 1) y se decrementan los de las no utilizadas. Los valores de los pesos oscilan generalmente entre -1 y 1, aunque estos valores pueden ser otros.

La velocidad a la que se ajustan esos pesos varía según el número de ejemplos con los que se vaya a entrenar a la red. Cuantos más ejemplos se le vayan a poner a la red, con mayor lentitud debe variar la tasa de aprendizaje. Puede parecer que cuanto más rápido se ajusten los pesos de las neuronas más rápido va a aprender a reconocer un determinado patrón, pero esto, que puede ser cierto a veces para una neurona, generalmente no lo es para una red neuronal con muchas neuronas. Además, incluso en el caso de una neurona sola, siempre es mejor poner muchos ejemplos con una tasa de aprendizaje lenta que poner unos pocos con una tasa de aprendizaje rápida.

2. Una vez entrenada la neurona, se utiliza para reconocer ese patrón entre otros. Ahora basta con presentar un patrón a la neurona, que devolverá un valor dependiendo del patrón. Entonces, basta con interpretar el resultado; la neurona habrá reconocido el patrón cuando produzca una salida mayor que una determinada cantidad, y no lo habrá reconocido cuando la salida sea menor que esa cantidad.

2.3.7. Tipos de redes neuronales artificiales

Un modelo neuronal se caracteriza principalmente por el aprendizaje, la arquitectura de Red y según el tipo de información que procesan.

En función del patrón de conexiones entre neuronas que presenta se pueden clasificar en:

Perceptrón monocapa. Red neuronal más sencilla ya que se tiene una capa de neuronas que proyectan las entradas a una capa de neuronas de salida donde se realizan diferentes cálculos. Se utilizan generalmente en tareas relacionadas con lo que se conoce como autoasociación (regenerar información de entrada que se presenta a la red de forma incompleta o distorsionada).

Perceptrón multicapa. Es un tipo de RNA no realimentada que consta de una capa de entrada, una de salida y por lo menos una capa oculta. Esta arquitectura se entrena de manera supervisada mediante el algoritmo de retropropagación (backpropagation).

En el algoritmo backpropagation la actualización de los valores de los pesos sinápticos de la capa de salida se hace tomando en cuenta el error de la salida actual de la red (señal de error), después, se propagan hacia atrás los errores a través de las sinapsis, proporcionando así señales de error a las neuronas de la capa oculta, que con esta información actualizan sus pesos y propagan nuevamente la señal de error hasta llegar a la capa de entrada.

Según el tipo de aprendizaje se pueden clasificar en:

Redes supervisadas. Se presenta a la red un conjunto de patrones junto con las salidas esperadas para cada uno de ellos y, a través de esta información la red va ajustando los pesos sinápticos con una regla de aprendizaje que debe converger. Este tipo de aprendizaje se utiliza en procesos en los que se busca que la red sea un estimador universal de funciones, es decir que la red se aproxime a una función conocida o desconocida.

Redes no supervisadas. En este tipo de aprendizaje se presenta a la red multitud de patrones

sin adjuntar la respuesta deseada. Se busca que la red reconozca regularidades en el conjunto de datos de entrada.

Redes híbridas. Son un enfoque mixto en el que se utiliza una función de mejora para facilitar la convergencia.

Aprendizaje reforzado: se sitúa a medio camino entre el supervisado y el no supervisado.

Finalmente, también se pueden clasificar las RNA's según sean capaces de procesar información de distinto tipo en:

Redes analógicas: procesan datos de entrada con valores continuos y, habitualmente, acotados.

Ejemplos de este tipo de redes son: Hopfield, Kohonen y las redes de aprendizaje competitivo.

Redes discretas: procesan datos de entrada de naturaleza discreta; habitualmente valores lógicos booleanos. Ejemplos de este segundo tipo de redes son: las máquinas de Boltzmann y Cauchy, y la red discreta de Hopfield [15].

2.3.8. Mecanismos de aprendizaje

Se ha visto que los datos de entrada se procesan a través de la red neuronal con el propósito de lograr una salida, por lo que una red neuronal debe aprender a calcular la salida correcta para cada vector de entrada en el conjunto de ejemplos. Este proceso de aprendizaje se denomina: proceso de entrenamiento. El conjunto de datos sobre el cual este proceso se basa es llamado: conjunto de datos de entrenamiento.

En otras palabras, el aprendizaje es el proceso por el cual una red neuronal modifica sus pesos en respuesta a una información de entrada. Los cambios que se producen durante el mismo se reducen a la destrucción, modificación y creación de conexiones entre las neuronas. En los sistemas biológicos existe una continua destrucción y creación de conexiones entre las neuronas. En los modelos de redes neuronales artificiales, la creación de una nueva conexión implica que el peso de la misma pasa a tener un valor distinto de cero. De la misma manera, una conexión se destruye cuando su peso pasa a ser cero.

Durante el proceso de aprendizaje, los pesos de las conexiones de la red sufren modificaciones, por lo tanto, se puede afirmar que este proceso ha terminado (la red ha aprendido) cuando los valores de los pesos permanecen estables ($dw_{ij}/dt = 0$). Un aspecto importante respecto al

aprendizaje de las redes neuronales es el conocer cómo se modifican los valores de los pesos, es decir, cuáles son los criterios que se siguen para cambiar el valor asignado a las conexiones cuando se pretende que la red aprenda una nueva información.

Hay dos métodos de aprendizaje importantes que pueden distinguirse:

- a. Aprendizaje supervisado.
- b. Aprendizaje no supervisado.
- c. Híbrido.
- d. Reforzado.

2.3.8.1. Aprendizaje supervisado.

El aprendizaje supervisado se caracteriza porque el proceso se realiza mediante un entrenamiento controlado por un agente externo (supervisor, maestro) que determina la respuesta que debería generar la red a partir de una entrada determinada. El supervisor controla la salida de la red y en caso de que ésta no coincida con la deseada, se procederá a modificar los pesos de las conexiones, con el fin de conseguir que la salida obtenida se aproxime a la deseada.

En este tipo de aprendizaje se suelen considerar, a su vez, tres formas de llevarlo a cabo, que dan lugar a los siguientes aprendizajes supervisados [15]:

- a. Aprendizaje por corrección de error.
- b. Aprendizaje por refuerzo.
- c. Aprendizaje estocástico.

2.3.8.2. Aprendizaje no supervisado.

Las redes con aprendizaje no supervisado no requieren influencia externa para ajustar los pesos de las conexiones entre sus neuronas. La red no recibe ninguna información por parte del

entorno que le indique si la salida generada en respuesta a una determinada entrada es o no correcta. Estas redes deben encontrar las características, regularidades, correlaciones o categorías que se puedan establecer entre los datos que se presenten en su entrada.

Existen varias posibilidades en cuanto a la interpretación de la salida de estas redes, que dependen de su estructura y del algoritmo de aprendizaje empleado. En algunos casos, la salida representa el grado de familiaridad o similitud entre la información que se le está presentando en la entrada y las informaciones que se le han mostrado hasta entonces (en el pasado). En otro caso, podría realizar un agrupamiento o establecimiento de categorías, indicando la red a la salida a qué categoría pertenece la información presentada a la entrada, siendo la propia red quien debe encontrar las categorías apropiadas a partir de las correlaciones entre las informaciones presentadas [15].

En cuanto a los algoritmos de aprendizaje no supervisado, en general se suelen considerar dos tipos, que dan lugar a los siguientes aprendizajes:

- a. Aprendizaje hebbiano.
- b. Aprendizaje competitivo y comparativo.

2.3.8.3. Aprendizaje híbrido.

En este caso existen en la red los dos tipos de aprendizaje básicos, supervisado y autoorganizado, normalmente en distintas capas de neuronas [16].

2.3.8.4. Aprendizaje reforzado.

Se sitúa a medio camino entre al aprendizaje supervisado y el autoorganizado. Como en el primero, se emplea información sobre el error cometido, pero en este caso existe una única señal de error, que representa un índice global del rendimiento de la red (solamente le indicamos lo bien o lo mal que está actuando). Como en el caso del no supervisado, no se suministra explícitamente la salida deseada [16].

2.4. Redes Neuronales Convolucionales (CNN)

Las redes neuronales convolucionales o CNNs de su nombre en inglés, son la parte de la inteligencia artificial que más se ha visto inspirada por la biología, en concreto por la Neurociencia, ya que funciona como lo haría el córtex visual primario.

Las CNNs representan un modelo profundo muy exitoso, y hoy en día con utilizadas en gran cantidad de aplicaciones, obteniendo por lo general muy buenos resultados. En particular son muy usadas en tareas de visión artificial y procesamiento de imágenes, por lo que son de gran interés para este trabajo.

Se puede definir como un tipo de red neuronal multicapa jerárquica similares a las redes neuronales simples, descritas en el punto anterior. En cuanto a su estructura las CNNs están formadas por tres tipos de capas, la capa de convolución, la de agrupamiento (reducción o pooling), con la cual se reducirá la cantidad de parámetros quedándose con las características más comunes y la de clasificación totalmente conectadas, con la que obtendrá el resultado final de la red [17], [18].

A continuación, se muestra en la Figura 2.19 una arquitectura típica de la red neuronal convolucional.

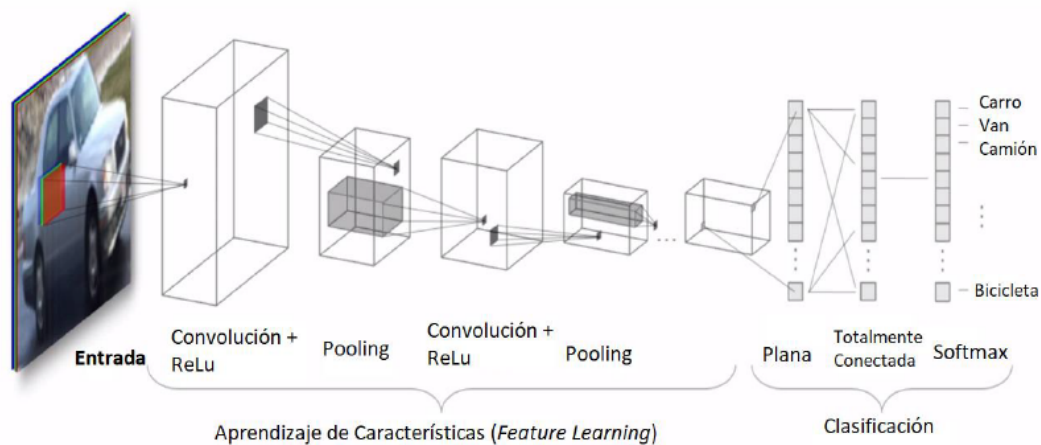


Figura 2.19: Arquitectura de una Red Neuronal convolucional.

El primer conjunto de capas de aprendizaje de características, está detectando características, mientras que las capas finales realizarán la clasificación de las imágenes. Como se observa

en la Figura 2.19 en la entrada de la red se tiene el primer conjunto de capas cuyos parámetros serán determinados por un conjunto de imágenes con un determinado ancho y alto y tres canales: rojo, verde y azul (RGB). En seguida se tienen varias capas de tipo convolución, capa de Unidad Lineal Rectificada (ReLU) que actúa como una función de activación y capa Pooling que reduce el tamaño de las capas. Tienen la funcionalidad del aprendizaje de características de imágenes. Cuando se llega al último volumen de capas se la aplanan en un vector, para de esta manera trabajar con redes totalmente conectadas y finalmente la capa Softmax que permite determinar a qué etiqueta corresponde la imagen de entrada [20].

A continuación, se explica con detalle cada una de las capas:

Capas de convolución. Es la que le da el nombre a la arquitectura y se trata de una ventana deslizante que se emplea en al menos una de sus capas, tanto a la imagen completa como a los mapas de características intermedias.

Capas de agrupamiento o Pooling. Se sitúan después de las capas de convolución con el objetivo de reducir progresivamente las dimensiones de los mapas de características. La función más utilizada en estas capas es la de max_pooling, que reduce la dimensionalidad de la entrada tomando para cada conjunto rectangular de tamaño fijo y seleccionado, el valor máximo de los píxeles de esa región y haciendo de ese valor el nuevo valor del píxel de salida [17].

Estas capas reciben un volumen de entrada de $W_1 \times H_1 \times D_1$, que corresponden al ancho, altura y profundidad de los datos de entrada. Producen un volumen de salida $W_2 \times H_2 \times D_2$, que corresponden al ancho, altura y profundidad del volumen de salida, estos quedan determinados por las ecuaciones 2.4 a 2.4, respectivamente.

$$W_2 = \frac{(W_1 - F)}{S} + 1 \quad (2.16)$$

$$H_2 = \frac{(H_1 - F)}{S} + 1 \quad (2.17)$$

$$D_2 = D_1 \quad (2.18)$$

En la Figura 2.20 se muestra el caso donde un mapa de características de 4 x 4 queda reducido a 2 x 2 al aplicar un max_pooling de 2 x 2. Se puede ver que el número de capas permanece intacto (64).

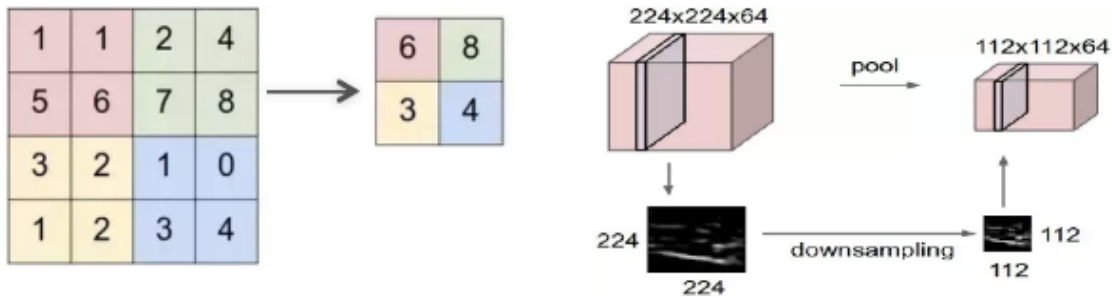


Figura 2.20: Funcionamiento del operador de max_pooling en un mapa de características en 1D y 2D.

Capas totalmente conectadas (FC). Este tipo de capas se sitúan tras la última de las capas de agrupación para convertir los mapas de características en un vector. Las capas FC trabajan como una red neuronal tradicional y contienen aproximadamente el 90% de los parámetros de la red [17], [18].

El vector de salida de la red suele ser de longitud predefinida, por ejemplo, para una tarea de clasificación de imágenes será de longitud igual al número de categorías que te tengan. En cada nodo se hace una suma ponderada de sus parámetros por el valor de entrada, el que tenga mayor puntuación, será el que tenga mayor probabilidad.

En la figura se muestra un ejemplo de funcionamiento de una red con varias capas convolucionales. Se aplican filtros a cada imagen de entrenamiento con distintas resoluciones, y la salida de cada imagen convolucionada se emplea como entrada para la siguiente capa.

2.4.1. Modelos de aprendizaje profundo en las CNN.

Las tres formas más habituales de utilizar el aprendizaje profundo para llevar a cabo la clasificación de objetos son las siguientes [10].

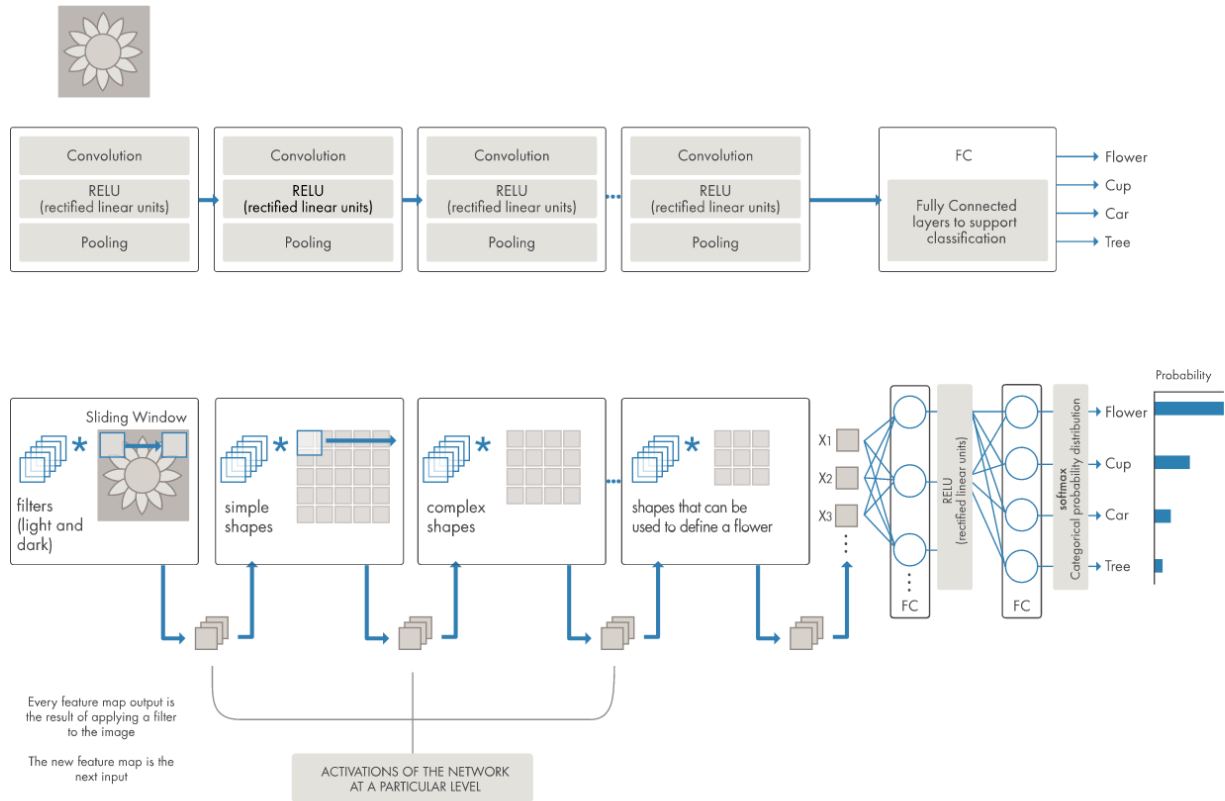


Figura 2.21: Ejemplo de red convolucional.

Para entrenar una red profunda desde cero, se recopila un conjunto de datos etiquetados muy amplio y se diseña una arquitectura de red que aprenda las características y el modelo. Esto resulta útil para las aplicaciones nuevas o las aplicaciones que tendrán un número muy elevado de categorías de salida. Este enfoque es menos común porque, debido a la gran cantidad de datos y a la velocidad de aprendizaje, se suele tardar días o semanas en entrenar estas redes.

2.4.2. Transferencia de aprendizaje (transfer learning)

Un proceso que implica el ajuste detallado de un modelo previamente entrenado. Se empieza con la red existente, como por ejemplo AlexNet o GoogLeNet, y se le proporcionan datos nuevos que contienen clases previamente desconocidas. Tras realizar algunos ajustes en la red, es posible realizar una tarea nueva, por ejemplo, categorizar solo los perros o los gatos en lugar de 1000 objetos distintos. Esto también tiene la ventaja de necesitar muchos menos datos (se procesan miles de imágenes en lugar de millones), de forma que el tiempo de cálculo se reduce a horas o minutos.

La transferencia del aprendizaje requiere una interfaz con los elementos internos de la red previamente existente, de forma que se pueda modificar y mejorar de forma muy detallada de acuerdo con la nueva tarea. MATLAB[®], que es el software de desarrollo utilizado en este trabajo, dispone de herramientas y funciones diseñadas para ayudarle con la transferencia del aprendizaje.

2.4.3. Extracción de características

Un enfoque ligeramente menos habitual y más especializado en relación con el aprendizaje profundo es el uso de un extractor de características. Como todas las capas tienen tareas asignadas con ciertas características de aprendizaje de las imágenes, estas características se pueden extraer de la red en cualquier momento durante el proceso de entrenamiento. A continuación, se pueden usar estas características como entrada para un modelo de aprendizaje automático, por ejemplo, las máquinas de vectores de soporte (SVM) [10].

Existen distintos modelos de CNNs con transferencia de aprendizaje (previamente entrenadas), de los que sobresalen los siguientes:

Alexnet. Es una de las arquitecturas CNN más conocidas, que sigue la estructura típica de 5 capas convolucionales y 3 FC, hoy en día es empleado para la tarea de clasificación de imágenes. Debido a su simplicidad y eficiencia es uno de los modelos utilizados en este trabajo por lo que se describirá más adelante.

VGG. Es una red convolucional que emplea entre 13 y 15 capas de convolución, de ahí su profundidad. VGG obtiene muy buenos resultados a la hora de la clasificación de imágenes demostrando que al aumentar la profundidad de la red puede aumentar la precisión.

GoogleNet. fue propuesta por una investigación en Google (con la colaboración de varias universidades) en 2014 en el trabajo de investigación titulado “Going Deeper with Convolutions”

Es una red todavía más profunda formada por 22 capas de convolución y por una capa totalmente conectada (FC). Al emplear esta red en trabajos realizados de clasificación de imágenes también se ha observado un aumento en la precisión de los resultados.

ResidualNetworks(ResNet). Los modelos de ResNet fueron desarrollados por He et al. que surgieron como una familia de arquitecturas profundas que mostraron buenos comportamientos de convergencia y una precisión bastante convincente. Ganaron el primer lugar en el desafío de clasificación de ILSVRC y Common Objects in Context (COCO) en 2015. ResNet se construyó con varias unidades residuales aplicadas y se desarrolló con diferentes números de capas: 18,34,50,101, 152 y 1202. Sin embargo, el número de operaciones puede variar dependiendo de las diferentes arquitecturas. ResNet es similar a VGG net, pero ResNet es aproximadamente ocho veces más profunda. La arquitectura de Res Net representa una buena compensación entre la profundidad y rendimiento, esta red está compuesta por cinco capas convolucionales, una de agrupación promedio y una capa totalmente conectada con un softmax. ResNet 50 contiene 49 capas convolucionales y una capa completamente conectada al final de la red [21], [22].

2.4.3.1. CNN pre-entrenada Alexnet

Es uno de los modelos implementados en este trabajo. Es una red convolucional pre-entrenada que consta de 25 capas, de las cuales 8 se destinan a la parte de aprendizaje (5 capas convolucionales y 3 totalmente conectadas). Las capas restantes son de ajuste de las imágenes y extracción de características. La primera capa define las dimensiones de la imagen de entrada a un tamaño de 227 x 227. La capa final es la capa de clasificación y sus propiedades dependen de la tarea de clasificación.

Alexnet cuenta con una estructura de datos llamada imdb. La estructura contiene los datos de entrenamiento y validación, la media de los datos, un conjunto de datos que define las imágenes para validar e incluye las clases y sus nombres [21].

Como se observa en la Figura 2.3, Alexnet está constituida por una serie de capas en las que sobresalen: capas de convolución, RELU, pooling y una capa clasificadora totalmente conectada, la cual define el resultado final de la red.

A continuación, se describen algunas de las capas que conforman la Red Alexnet:

Capa de entrada (Input layer). En primer lugar, se localiza la capa de entrada, que va a definir el tamaño de las imágenes que se deseen utilizar en la red. Dicho tamaño se corresponderá con la altura, la anchura y la cantidad de canales de color de que disponga la imagen. En este caso concreto, se ha diseñado para imágenes de 227x227 en la escala de color RGB.

Capa convolucional (Convolution layer). Esta capa recibe como entrada la imagen y luego aplica sobre ella un filtro o kernel que devuelve un mapa de características de la imagen original, de esta forma se logra reducir el tamaño de los parámetros que se deben aprender en la siguientes capas [23].

Este tipo de capas constan de neuronas que se conectan a pequeñas regiones o filtros de la entrada, tal y como muestra la Figura 2.22 Dicho filtro se mueve a lo largo de toda la imagen, y, en cada caso, se calcula el producto de los pesos por las entradas al que posteriormente se le añade un offset o sesgo.

RELU (Rectified linear unit). La función RELU es la función de activación habitual en las capas ocultas de una red neuronal profunda, la cual consiste en devolver el valor max (0,

1	'data'	Image Input
2	'conv1'	Convolution
3	'relu1'	ReLU
4	'norm1'	Cross Channel Normalization
5	'pool1'	Max pooling
6	'conv2'	Convolution
7	'relu2'	ReLU
8	'norm2'	Cross Channel Normalization
9	'pool2'	Max Pooling
10	'conv3'	Convolution
11	'relu3'	ReLU
12	'conv4'	Convolution
13	'relu4'	ReLU
14	'conv5'	Convolution
15	'relu5'	ReLU
16	'pool5'	Max Pooling
17	'fc6'	Fully Connected
18	'relu6'	ReLU
19	'drop6'	Dropout
20	'fc7'	Fully Connected
21	'relu7'	ReLU
22	'drop7'	Dropout
23	'fc8'	Fully Connected
24	'prob'	Softmax
25	'output'	Classification

Tabla 2.3: Capas Red Alexnet

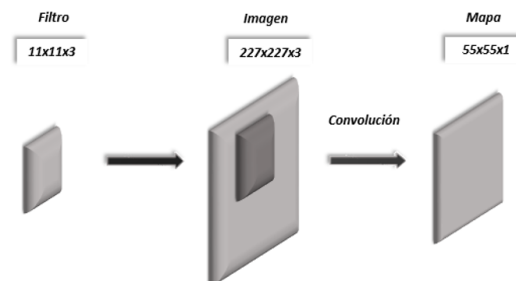


Figura 2.22: Proceso en la capa de convolución.

x). Se realiza una operación de umbral para cada elemento, dónde cualquier entrada de valor menor que cero se establece en cero.

$$f(x) = \begin{cases} X, & x \geq 0 \\ 0, & x < 0 \end{cases} \quad (2.19)$$

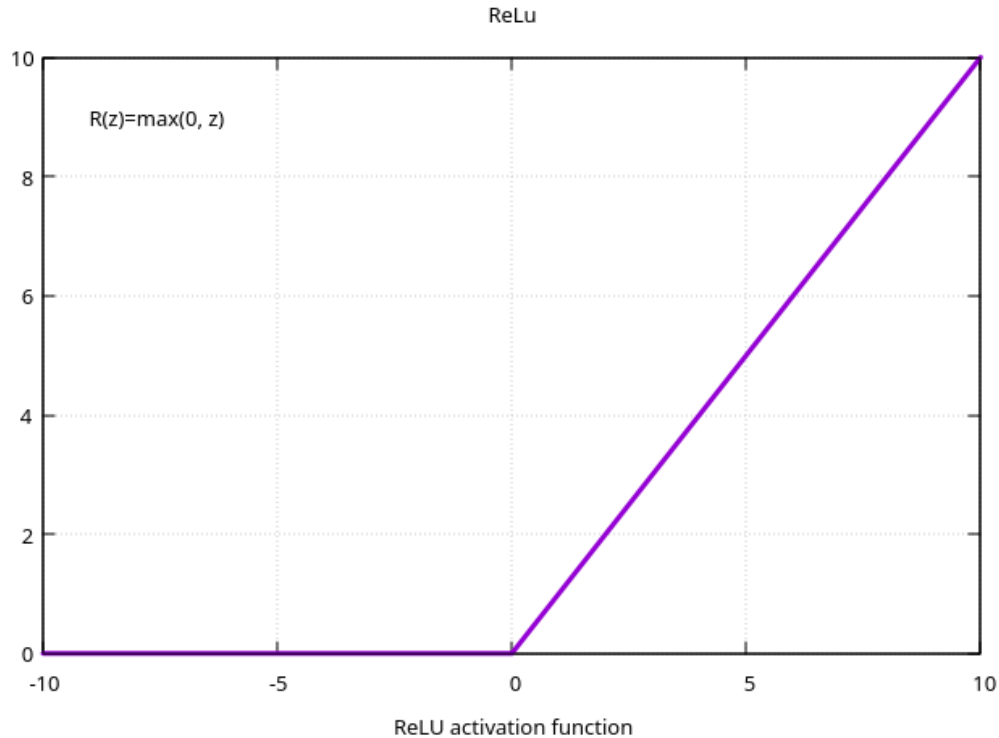


Figura 2.23: RELU (Rectified linear unit).

Capa Pooling. La capa de reducción o pooling se emplea generalmente después de la capa convolucional. Su utilidad principal radica en la reducción de las dimensiones espaciales del volumen de entrada para la siguiente capa convolucional. La operación realizada por esta capa también se llama reducción de muestreo, ya que la reducción de tamaño conduce también a la pérdida de información. Adicional a esto, también trabaja para reducir el sobreajuste. La operación que se suele utilizar es max-pooling, que divide a la imagen de entrada en un conjunto de rectángulos y, respecto de cada subregión, se va quedando con el máximo valor [23].

El tamaño de dichas regiones podrá ser establecido, de forma que, si a este se le da un valor de 3x3, la capa devolverá el valor máximo de cada región de la imagen de entrada tal y como

se muestra en la Figura 2.24.

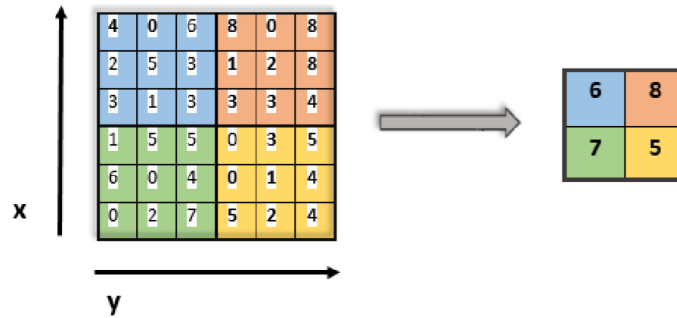


Figura 2.24: Operación Pooling.

Capa Clasificadora. Al final de las capas convolucionales, RELU y de pooling, las redes utilizan generalmente capas completamente conectadas en las que cada pixel se considera como una neurona separada al igual que en una red neuronal regular. Esta última capa clasificadora tendrá tantas neuronas como el número de clases que se debe predecir [23].

Capa Dropout. Crea una capa de deserción, “dropoutLayer” devuelve una capa de abandono que establece al azar elementos de entrada a cero con una probabilidad de 0.5. La capa de salida solo funciona en el tiempo de entrenamiento [10].

Capa Softmax. Este tipo de capa trunca las salidas de cada elemento de la matriz formada en la última capa totalmente conectada, de forma que, tal y como se muestra en la Figura 7, estas se encuentren contenidas entre 0 y 1 y que, además, la suma de todas ellas sea la unidad. Para ello, aplica la función sigmoide o función logística tal que:

$$\delta(x)_j = \frac{e^{x_j}}{\sum_{k=1}^k e^{x_k}} \quad (2.20)$$

Donde x es un vector compuesto por las entradas a la capa, y j indexa cada elemento de salida, $j = 1, 2, \dots, k$.

2.4.3.2. GoogLeNet (Inception V1).

GoogLeNet es una red neuronal convolucional que tiene 22 capas de profundidad. Puede utilizarse una versión pre entrenada de la red con los conjuntos de datos ImageNet. La red ca-

$$\begin{bmatrix} 4 \\ 2.1 \\ 0.5 \end{bmatrix} \rightarrow \text{SoftMax} \rightarrow \begin{bmatrix} 0.85 \\ 0.13 \\ 0.02 \end{bmatrix}$$

Figura 2.25: Operación Softmax.

pacitada en ImageNet clasifica las imágenes en 1000 categorías de objetos, como teclado, mouse, lápiz y muchos animales. Estas redes han aprendido diferentes representaciones de características para una amplia gama de imágenes y puede ser entrenada para realizar una nueva tarea utilizando el aprendizaje de transferencia.

GoogleNet posee siete millones de parámetros y contiene nueve módulos iniciales, cuatro capas convolucionales, cuatro capas de agrupación máxima, tres capas de agrupamiento promedio, cinco capas completamente conectadas y tres capas de softmax para los principales clasificadores auxiliares de la red. Además, utiliza la regularización de dropout en la capa totalmente conectada y aplica la activación ReLU en todas las capas convolucionales [22].

La arquitectura GoogLeNet es muy diferente de las arquitecturas de vanguardia anteriores, como AlexNet. Utiliza diferentes tipos de métodos, como la convolución 1×1 y la agrupación promedio global que le permite crear una arquitectura más profunda.

En la arquitectura, se observan los siguientes métodos:

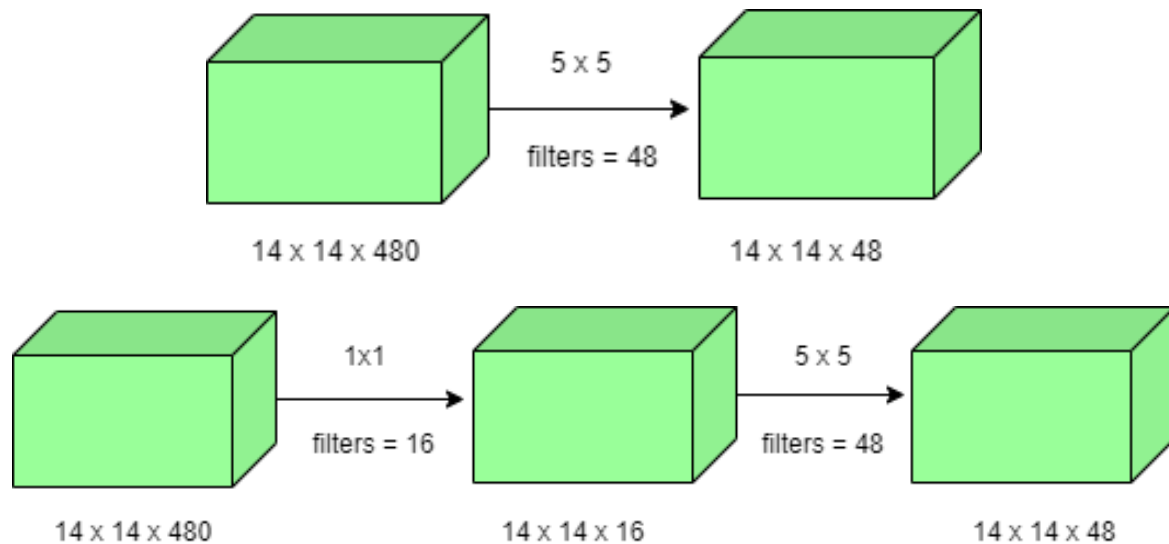
Convolución 1×1 . De inicio utiliza convolución 1×1 en su arquitectura. Estas convoluciones se utilizan para disminuir el número de parámetros (pesos y sesgos) de la arquitectura. Al reducir los parámetros también aumentamos la profundidad de la arquitectura.

Ejemplo:

Al realizar una convolución de 5×5 con 48 filtros sin usar la convolución de 1×1 , el número de operaciones es: $(14 \times 14 \times 48) \times (5 \times 5 \times 480) = 112,9 \text{ M}$ $(14 \times 14 \times 16) \times (1 \times 1 \times 480) + (14 \times 14 \times 48) \times (5 \times 5 \times 16) = 1.5\text{M} + 3.8\text{M} = 5.3\text{M}$, que es mucho más pequeño que 112.9M.

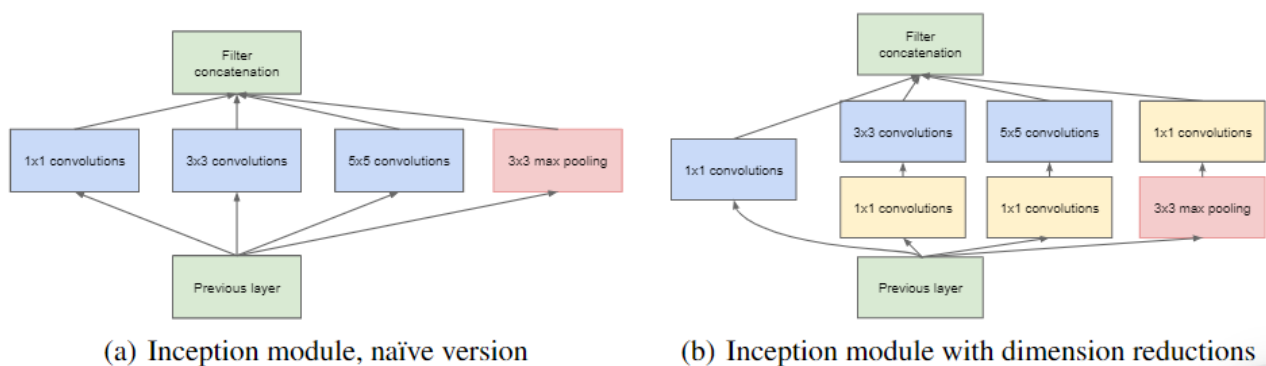
Agrupación promedio global. En la arquitectura anterior, como AlexNet, las capas completamente conectadas se utilizan al final de la red. Estas capas completamente conectadas contienen la mayoría de los parámetros de muchas arquitecturas que causan un aumento en el costo de cálculo.

En la arquitectura GoogLeNet, hay un método llamado agrupación promedio global que se



utiliza al final de la red. Esta capa toma un mapa de características de 7×7 y lo promedia a 1×1 . Esto también disminuye el número de parámetros entrenables a 0 y mejora la precisión del top 1 en un 0.6%.

Módulo de inicio(Inception): El módulo de inicio es diferente de las arquitecturas anteriores como AlexNet. En esta arquitectura, hay un tamaño de convolución fijo para cada capa. En el módulo Inception 1×1 , 3×3 , en convolución 5×5 y en agrupación máxima (max pooling) 3×3 , realizadas de forma paralela en la entrada y la salida de estas que se apilan juntas para generar la salida final.



Clasificador auxiliar para entrenamiento: La arquitectura inception usa algunas ramas clasificadoras intermedias en el medio de la arquitectura, estas ramas se usan solo durante el entrenamiento. Estas consisten en una capa de agrupación promedio de 5×5 con una zancada

type	patch size/ stride	output size	depth	# 1x1	# 3x3 reduce	# 3x3	# 5x5 reduce	# 5x5	pool proj	params	ops
convolution	7x7/2	112x112x64	1							2.7K	34M
max pool	3x3/2	56x56x64	0								
convolution	3x3/1	56x56x192	2		64	192				112K	360M
max pool	3x3/2	28x28x192	0								
inception (3a)		28x28x256	2	64	96	128	16	32	32	159K	128M
inception (3b)		28x28x480	2	128	128	192	32	96	64	380K	304M
max pool	3x3/2	14x14x480	0								
inception (4a)		14x14x512	2	192	96	208	16	48	64	364K	73M
inception (4b)		14x14x512	2	160	112	224	24	64	64	437K	88M
inception (4c)		14x14x512	2	128	128	256	24	64	64	463K	100M
inception (4d)		14x14x528	2	112	144	288	32	64	64	580K	119M
inception (4e)		14x14x832	2	256	160	320	32	128	128	840K	170M
max pool	3x3/2	7x7x832	0								
inception (5a)		7x7x832	2	256	160	320	32	128	128	1072K	54M
inception (5b)		7x7x1024	2	384	192	384	48	128	128	1388K	71M
avg pool	7x7/1	1x1x1024	0								
dropout (40%)		1x1x1024	0								
linear		1x1x1000	1							1000K	1M
softmax		1x1x1000	0								

Tabla 2.4: Un clasificador softmax con 1000 clases de salida similar al clasificador principal softmax.

de 3, una circunvolución de 1×1 con 128 filtros, dos capas completamente conectadas de 1024 salidas y 1000 salidas y una capa de clasificación softmax.

GoogLeNet tiene dos capas de clasificador auxiliar conectadas a la salida de las capas Inception (4a) e Inception (4d) con las siguientes características:

- Una capa de agrupación promedio de tamaño de filtro 5×5 y zancada 3.
- Una convolución 1×1 con 128 filtros para reducción de dimensiones y activación ReLU
- Una capa totalmente conectada con 1025 salidas y activación ReLU
- Regularización de dropout con un índice de 0.7

Esta arquitectura Figura 2.26 toma imágenes de tamaño 224×224 con canales de color RGB. Todas las circunvoluciones dentro de esta arquitectura utilizan unidades lineales rectificadas (ReLU) como sus funciones de activación.

Este ejemplo muestra cómo clasificar una imagen usando la red pre entrenada GoogLeNet.

Este trabajo se presenta con el objetivo de desarrollar un algoritmo que sea capaz de identificar huellas digitales, utilizando técnicas de procesamiento de imágenes e implementando Redes

Capítulo 3

Desarrollo

El desarrollo de este trabajo se centra en el procesamiento y clasificación de huellas dactilares, donde cada etapa cumple un rol fundamental en la precisión final del algoritmo. Entre estas etapas, el procesamiento de imágenes destaca como uno de los puntos claves del desarrollo, ya que la selección y adecuación de las técnicas aplicadas determinan directamente la eficacia en la extracción de características y, en el rendimiento del clasificador.

En este trabajo, la etapa de mejoramiento de la imagen se implementa mediante seis algoritmos ejecutados secuencialmente en MATLAB, siguiendo un orden predefinido para optimizar los resultados.

La entrada al sistema clasificador es la imagen de una huella dactilar proveniente de una base de datos. Es procesada para adquirir sus características generando un vector con los detalles esenciales (bifurcaciones y terminaciones) el cual es utilizado como entrada a la red neuronal artificial para su verificación e identificación dentro de la base de datos de huellas dactilares previamente almacenadas.

Este proceso se ha dividido en las siguientes etapas:

3.1. Procesamiento de la imagen.

a. Mejoramiento de la huella Digital.

a) Segmentación.

- b) Normalización.
 - c) Orientación.
 - d) Filtrado.
 - e) Binarización.
 - f) Adelgazamiento.
- b. Extracción de las Características de la Huella.
- a) Identificación de minucias.
 - b) Eliminación de falsas minucias.

3.1.1. Segmentación

El algoritmo de segmentación se basa en la varianza local asociada a un pixel. Generalmente la zona que contiene la huella dactilar, tiene una varianza elevada (contraste elevado), dada la diferencia entre la intensidad de los píxeles relacionados con los valles y las crestas.

Primero la imagen es dividida en bloques de tamaño $h \times a$ (16 x 16) y se calcula la varianza (desviación estándar) de la escala de grises para cada bloque de la imagen. Si la varianza es menor que el umbral global 0.1 (valor tomado de forma empírica), entonces el bloque es asignado a la región del fondo, si es igual mayor que el umbral es asignado a la parte del frente y se le considera parte de la huella.

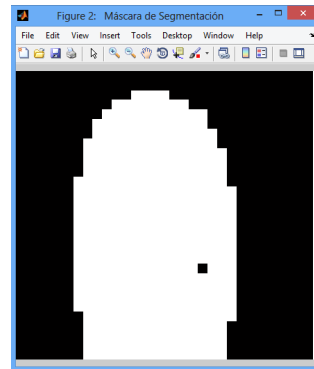
3.1.2. Normalización

Se estandariza la intensidad de la imagen, ajustando el rango de la escala de grises de forma que quede dentro del rango de valores deseados. Donde se calcula la media y la varianza a la máscara de la imagen obtenida en la segmentación, Figura 3.1b.

La normalización no cambia la estructura de las crestas en la imagen de una huella, es realizada para estandarizar el nivel de la variación en la escala de grises. Lo cual facilita los subsecuentes procesos para el mejoramiento de la imagen.

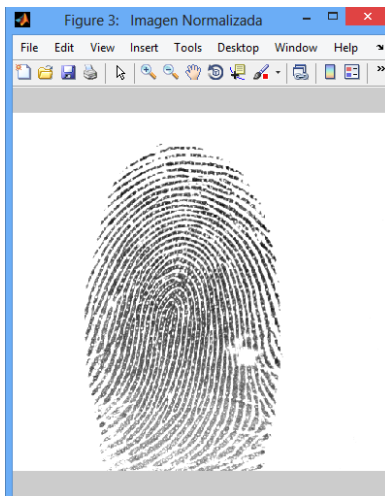


(a) Imagen original.



(b) Imagen segmentada.

Figura 3.1: Segmentación.



3.1.3. Orientación

Se determina la orientación de las crestas de la huella, calculando la orientación del pixel (i , j):

1. Se divide la imagen en bloques $w \times w$, calculando la orientación al pixel central.
2. Se calcula el gradiente para cada pixel en el bloque, Se realiza la convolución de cada pixel con una máscara de 3×3 .

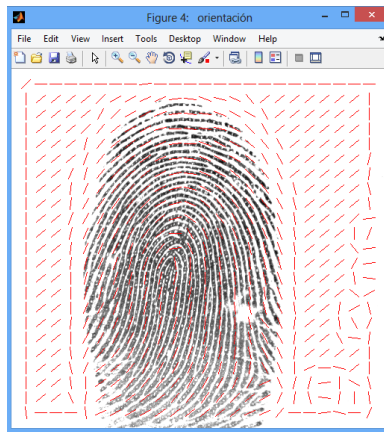


Figura 3.2: Orientación de las crestas.

3.1.4. Filtrado

Se realiza la estimación de la frecuencia dividiendo la imagen en bloques de tamaño $W \times W$. Lo siguiente es proyectar el valor en la escala de grises de cada pixel en el bloque en dirección ortogonal de la dirección del vector de orientación local de las crestas.

Una vez que el vector de orientación y la información de la frecuencia de las crestas han sido determinada, se utilizan estos parámetros para construir el filtro de Gabor

3.1.5. Binarización

Se convierte la imagen de escala de grises a un formato binario de unos y ceros donde se puede diferenciar los valles y las crestas. En este proceso se examina el nivel en la escala de grises de cada pixel y si el valor es mayor que el umbral global, entonces el valor del pixel es

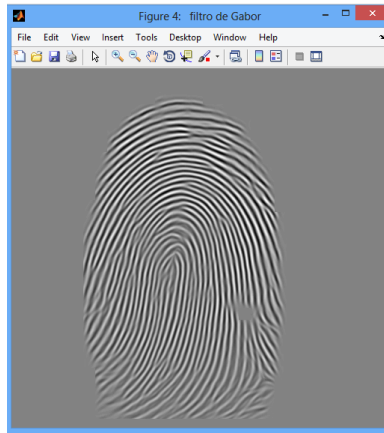


Figura 3.3: Imagen Filtrada.

asignado a uno de lo contrario es asignado a cero. La salida es una imagen binaria que contiene dos niveles de información las crestas con uno y los valles con cero.

El resultado final del proceso de Binarización se puede apreciar en la Figura 3.4.

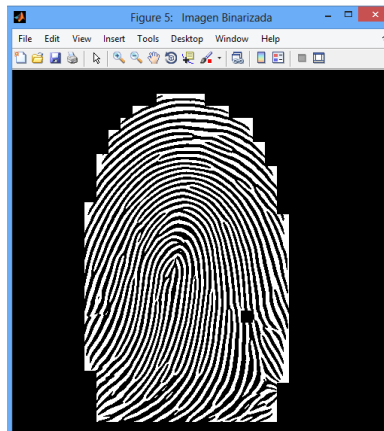


Figura 3.4: Imagen Binarizada.

3.1.6. Adelgazamiento.

Se reducen en espesor las crestas en la imagen, que se representan mediante unos binarios, obteniendo una imagen en la que todas las crestas tengan un píxel de ancho, como se muestra en la Figura 3.5.

La aplicación del algoritmo de adelgazamiento sobre la imagen de huella digital preserva la conectividad de las crestas formando un esqueleto de la imagen binaria. Éste esqueleto es

utilizado para la subsecuente extracción de características.

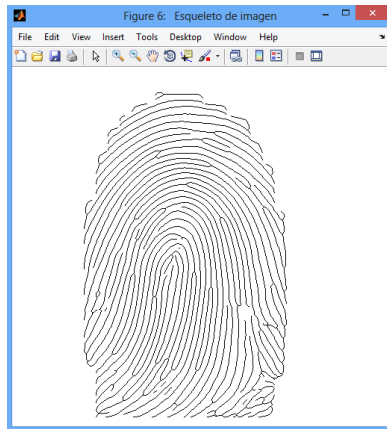


Figura 3.5: Esqueleto de imagen.

3.2. Extracción de las características de la huella.

Identificación de minucias. Después de ejecutar el proceso de adelgazamiento se obtiene una imagen binaria de un tamaño de 256 x 256. Se toman ventanas de la imagen de 3 por 3 píxeles y sobre ésta se aplica la fórmula de Crossing Number para determinar el tipo de minucias (bifurcación o final) que contiene la imagen. El resultado es un vector con las minucias encontradas, donde cada elemento tiene cuatro características (x,y,O,t) , coordenadas x,y , en el eje horizontal y eje vertical, la orientación del píxel (O) y el tipo de minucia (t).

En la Figura 3.6 se puede observar cómo el algoritmo detecta y marca las minucias.

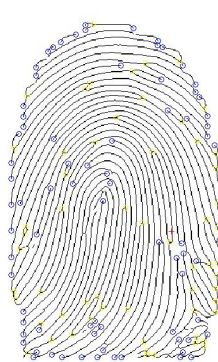


Figura 3.6: Minucias identificadas.

Eliminar falsas minucias. Una vez identificadas y extraídas las minucias se analizan y se determinan las posibles minucias erróneas resultando un vector solamente con las minucias que cumplen los criterios y características establecidas en los procesos.

3.3. Implementación de la red neuronal artificial.

En esta sección se describe el proceso de implementación de una red neuronal artificial (RNA) diseñada para la clasificación de huellas dactilares. La RNA fue desarrollada con el objetivo de extraer características relevantes de las imágenes de huellas (como bifurcaciones y terminaciones) y realizar la clasificación basándose en los patrones identificados. A continuación, se detallan los aspectos clave de la implementación, incluyendo la arquitectura de la red, su configuración, el proceso de entrenamiento y los resultados obtenidos.

3.3.1. Descripción de la arquitectura de la red

La red neuronal implementada es un perceptrón multicapa (MLP) compuesto por tres capas principales: una capa de entrada, una capa oculta (ajustadas durante la experimentación) y una capa de salida. La arquitectura de la red se describe a continuación:

- **Capa de entrada:** Consta de 30 neuronas, que corresponden a las características extraídas de las huellas dactilares (coordenadas x, y y ángulo de orientación de las minucias como bifurcaciones y terminaciones).
- **Capa Oculta:** Se definió el número de capas y neuronas de forma experimental. Iniciando con 1 capa incrementando a configuraciones de 10, 15, 20, 30 y 40 capas ocultas y con diferentes números de neuronas (30, 60 y 120) para analizar su impacto en el rendimiento del modelo. Se utilizó la función de activación **ReLU (Rectified Linear Unit)** en las capas ocultas debido a su eficiencia y capacidad para evitar el problema de **descenso del gradiente**.
- **Capa de Salida:** Consta de una sola neurona que proporciona una salida binaria. Se utilizó la función **tangente hiperbólica (tansig)** en la capa de salida, la cual produce

valores en el rango $[-1, 1]$. Esto permite interpretar la salida de la red de la siguiente manera:

- o 1: Huella positiva (perteneciente al individuo identificado).
- o -1: Huella negativa (no perteneciente al individuo identificado)

En la Figura 3.7 se muestra el esquema de la arquitectura de la red neuronal utilizada, con una configuración de 30 neuronas en la capa de entrada, 60 en la capa oculta y 1 en la capa de salida.

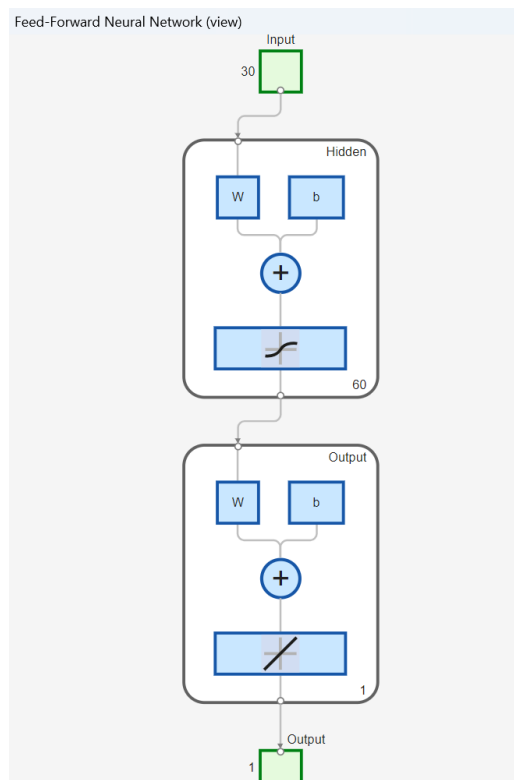


Figura 3.7: Arquitectura de la Red Neuronal.

3.3.2. Configuración de la RNA

La red fue configurada con los siguientes parámetros:

- **Función de pérdida:** Se utilizó la **función de pérdida de entropía cruzada (cross-entropy)** para medir el error durante el entrenamiento. Esta función es adecuada para problemas de clasificación binaria.

- **Optimizador:** Se empleó el algoritmo **descenso de gradiente estocástico (SGD)** con una tasa de aprendizaje de 0.01 para ajustar los pesos de la red durante el entrenamiento.
- **Regularización:** Se aplicó **dropout** con una tasa del 50% en las capas ocultas para evitar el sobreajuste.

3.3.3. Entrenamiento de la RNA

El entrenamiento de la red se llevó a cabo utilizando la función `train` de MATLAB, con un conjunto de 432 imágenes de huellas dactilares, divididas en un 70% para entrenamiento y un 30% para validación. El proceso de entrenamiento incluyó los siguientes pasos:

- **Entrado de datos:**
 - o Los datos de entrenamiento (`Min_train`) y las etiquetas correspondientes (`pers_train`) se introdujeron en la red neuronal.
 - o La red ajustó sus pesos iterativamente para minimizar la función de pérdida (entropía cruzada).
- **Iteraciones y épocas:**
 - o Se evaluaron (30 pruebas) diferentes números de épocas: 10, 20, 30 y 40 con las diferentes configuraciones de capas ocultas (1, 10, 15, 20, 30, 40).
 - o Durante cada época, la red procesó todas las muestras de entrenamiento y ajustó los pesos utilizando el algoritmo de descenso de gradiente.
 - o El entrenamiento se detuvo después de completar las épocas especificadas.
- **Métricas de evaluación:** Se midieron el tiempo de entrenamiento, la precisión en el conjunto de entrenamiento y la precisión en el conjunto de validación.

El tiempo total de entrenamiento, se muestra en la Tabla [3.1](#)

En la Figura [3.8](#) se presenta un resumen del entrenamiento de la red neuronal, donde se observa el comportamiento del gradiente durante las iteraciones. Además, en la Figura [3.9](#) se

Capas Ocultas	Tiempo promedio de entrenamiento (min)	Tiempo promedio de reconocimiento (s)
1	15.6789(s)	0.0123
10	10.1378665	0.02093713
15	23.3145971	0.01559347
20	44.4804757	0.01476033
30	108.211823	0.01443103
40	228.534584	0.014403

Tabla 3.1: Tiempo promedio

muestra el rendimiento del modelo durante el entrenamiento, incluyendo la precisión en los conjuntos de entrenamiento y validación.

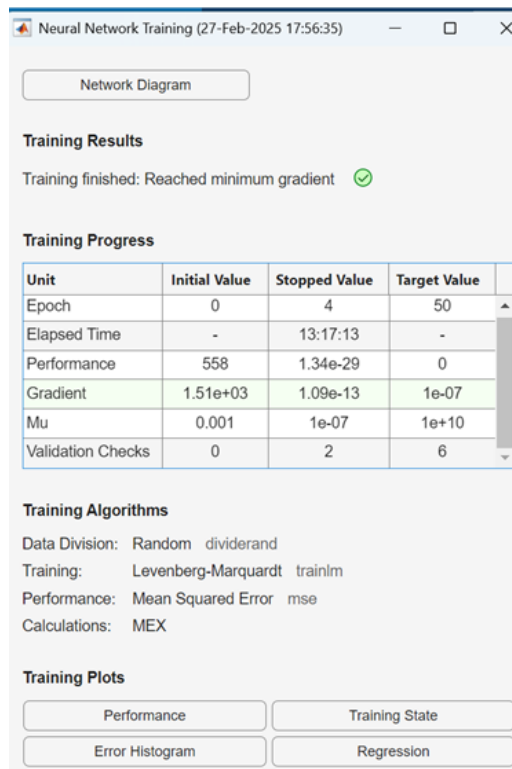


Figura 3.8: Resumen del Entrenamiento de la Red Neuronal - Alcanzado Mínimo Gradiente.

3.3.4. Evaluación del modelo

El rendimiento de la red se evaluó utilizando las siguientes métricas:

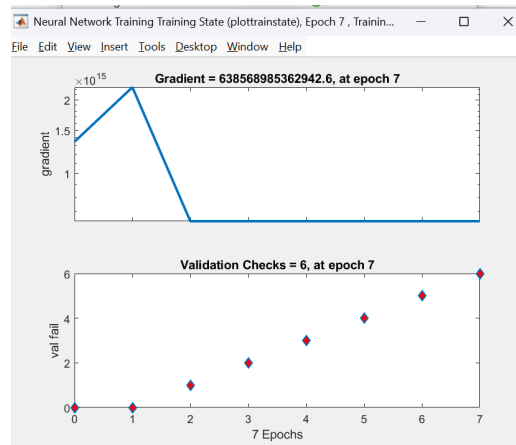


Figura 3.9: Rendimiento del Modelo durante el Entrenamiento.

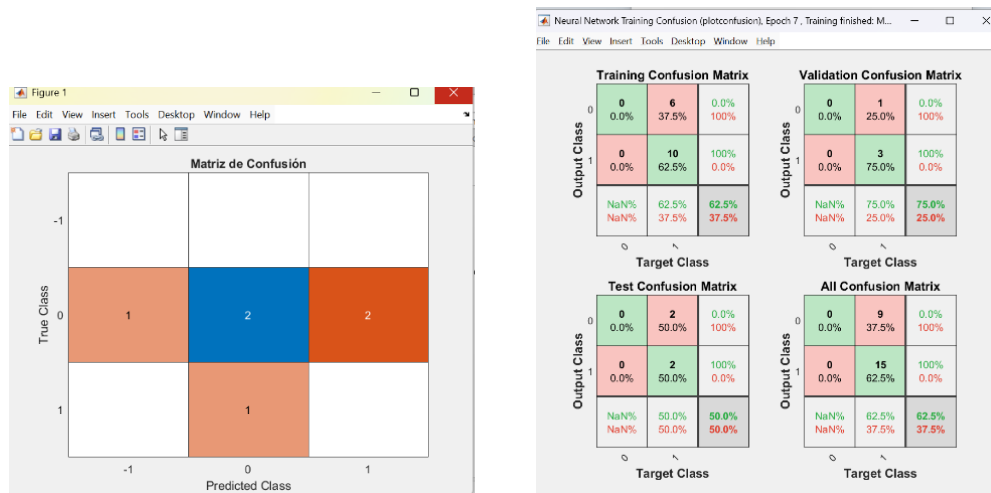
- **Precisión:** Se calculó la precisión en el conjunto de prueba, obteniendo un valor del 88 % con 60 neuronas en la capa oculta.
- **Matriz de Confusión:** Se generó una matriz de confusión para visualizar los aciertos y errores en la clasificación. La matriz permitió evaluar la precisión del modelo en términos de:
 - o **Verdaderos Positivos (TP):** Huellas correctamente identificadas como positivas.
 - o **Falsos Positivos (FP):** Huellas incorrectamente identificadas como positivas.
 - o **Verdaderos Negativos (TN):** Huellas correctamente identificadas como negativas.
 - o **Falsos Negativos (FN):** Huellas incorrectamente identificadas como negativas.

En la Figura 3.10 se muestra un ejemplo de la matriz de confusión generada para la evaluación del modelo, donde se pueden observar los aciertos y errores en la clasificación de las huellas dactilares.

3.3.5. Herramientas y entorno de desarrollo

La implementación de la red neuronal se realizó utilizando las siguientes herramientas:

- **Lenguaje de Programación:** MATLAB.
- **Bibliotecas:** Neural Network Toolbox de MATLAB.



(a) Comparación entre clases reales y pre- (b) Comparación de rendimiento en en-
dichas. trenamiento, validación y prueba.

Figura 3.10: Matrices de confusión de la evaluación integral del modelo.

- **Hardware:** Equipo con procesador Intel i5 1.60GHz 1.80 GHz

3.3.6. Pruebas y resultados

Para la evaluación de resultados se utilizó un conjunto de prueba que no se utilizaron durante el entrenamiento y validación y se realizaron 30 pruebas. Los resultados obtenidos durante la implementación y evaluación del modelo se ejemplifican continuación:

- Tiempo de extracción de minucias: 0.3456 segundos.
- Tiempo de procesamiento de minucias: 0.2789 segundos.
- Tiempo de preparación de datos: 0.0321 segundos.
- Dimensiones de Min_train: 19x24.
- Dimensiones de pers_train: 1x24.
- Tiempo de entrenamiento: 15.6789 segundos.
- La persona identificada es: 57
- Tiempo de identificación: 0.0123 segundos.

En la Figura 3.11 se muestra un ejemplo de una huella dactilar identificada por el sistema. La imagen presenta la huella testada junto con su identificador único (ID: 57), lo que demuestra

la capacidad del modelo para reconocer y clasificar huellas dactilares de manera efectiva. Este resultado visual refuerza la precisión del sistema en la identificación de individuos.

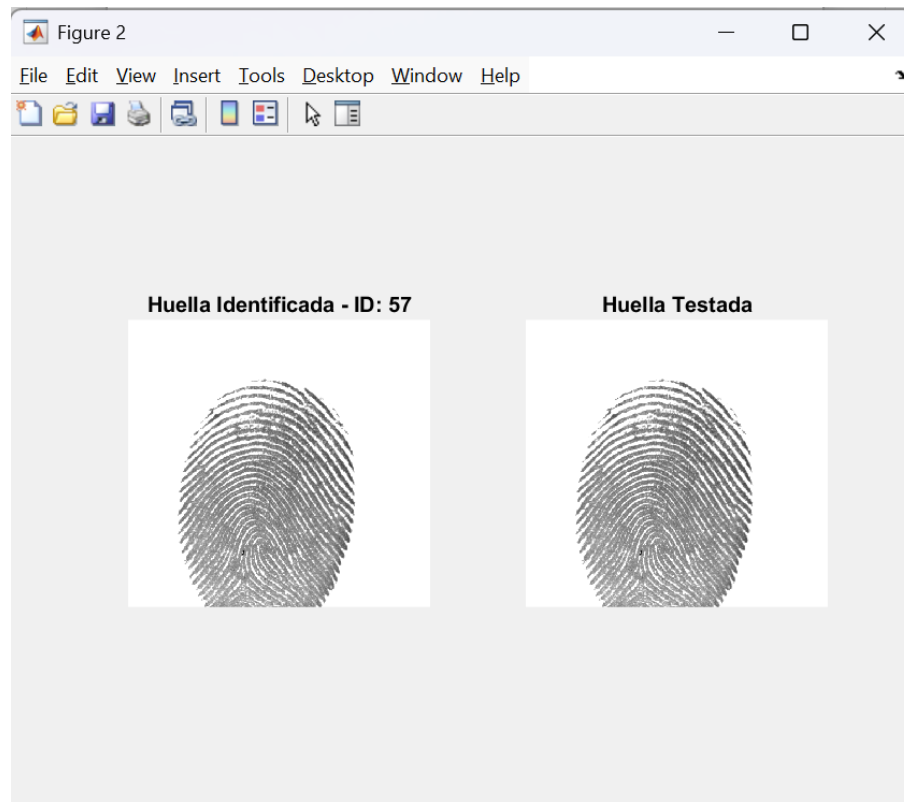


Figura 3.11: Huella identificada.

Para las pruebas de reconocimiento, se tomaron dos muestras de huellas dactilares de un individuo objetivo. Para evaluar los reconocimientos negativos, se analizaron las huellas de otros individuos. En las Tablas 3.2 y 3.3 se presentan ejemplos del porcentaje de aciertos obtenidos en el proceso de reconocimiento.

En la Figura 3.12 y la Figura 3.13 presentan gráficos adicionales que muestran el rendimiento del modelo en términos de precisión y regresión, respectivamente.

3.3.7. Modelo de clasificación multiclase.

Se decidió implementar como experimento otro modelo que se entrene para asociar las minucias (coordenadas de terminaciones y bifurcaciones) con las etiquetas correspondientes de las personas (12, 13, 17, etc.). Este es un modelo básico de clasificación multiclase. Consiste en una

Capas	Pruebas	Positivos	Falsa aceptación							
			P1 %	P2 %	P3 %	P4 %	P5 %	P6 %	P7 %	P8 %
10	30	100	67	61	11	37	60	76	62	
15	30	100	50	35	5	22	45	56	47	
20	30	90	9	9	2	15	19	18	11	
30	30	74	0	10	1	7	4	0	1	
40	30	65	0	0	1	6	3	1	1	

Tabla 3.2: Porcentaje de aciertos y falsas aceptaciones en reconocimiento de huellas dactilares al variar el número de capas del modelo.(Muestra 1)

Capas	Pruebas	Positivos	Falsa aceptación						
			P2 %	P3 %	P4 %	P5 %	P6 %	P7 %	P8 %
10	30	93	68	57	56	30	47	48	65
15	30	90	41	21	35	17	26	26	46
20	30	80	24	13	16	11	12	12	31
30	30	80	8	2	1	2	1	1	3
40	30	67	4	0.5	2	0.5	0	0	2

Tabla 3.3: Porcentaje de aciertos y falsas aceptaciones en reconocimiento de huellas dactilares con diferentes configuraciones de capas.(Muestra 2)

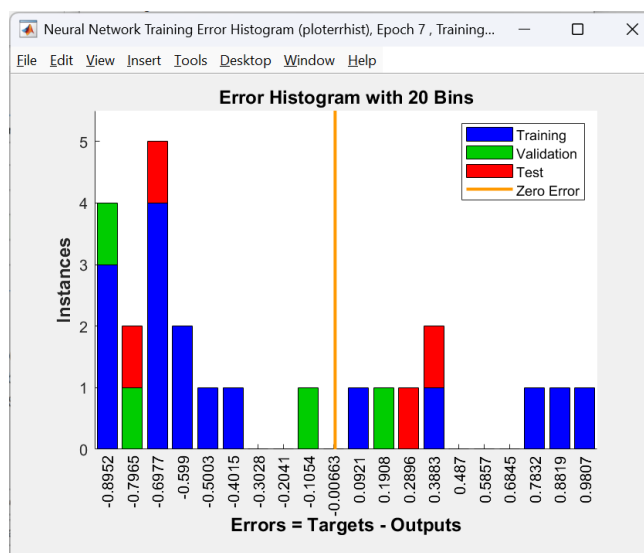


Figura 3.12: Gráfica del rendimiento del modelo en términos de precisión.

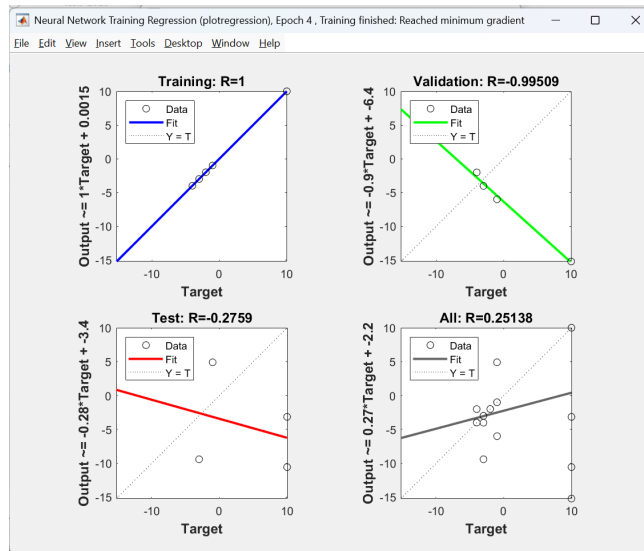


Figura 3.13: Regresión.

red neuronal feedforward (perceptrón multicapa) con una capa oculta. En seguida se describen los detalles sobre la arquitectura:

Capa de entrada:

La capa de entrada tiene tantas neuronas como características de las minucias que se están usando. En el caso de las minucias, cada fila del conjunto de datos tiene varias columnas representando las coordenadas de las terminaciones y bifurcaciones de las huellas dactilares. Se tienen 432 filas de datos de las minucias y el número de neuronas en la capa de entrada es igual al número de columnas en el archivo de minucias que se toman aleatoriamente para entrenamiento.

Capa oculta:

La red tiene una capa oculta con 120 neuronas, que está definida por la variable `hiddenLayerSize`, el número de neuronas en la capa oculta es un parámetro que puedes ajustar según el problema y el tamaño de los datos. Esta capa procesa las características de entrada y aprende representaciones intermedias para clasificarlas en una de las categorías de personas.

Capa de salida:

La capa de salida tiene tantas neuronas como clases o etiquetas posibles, que en tu caso corresponden a las personas (los 9 identificadores de personas: 12, 13, 17, 22, 27, 45, 47, 57, 76). Esto implica que la capa de salida tiene **una neurona por persona**, lo que hace que la red sea un modelo de **clasificación multiclase**. La Función de Activación en la Capa de Salida fue softmax para obtener probabilidades de pertenencia a cada clase.

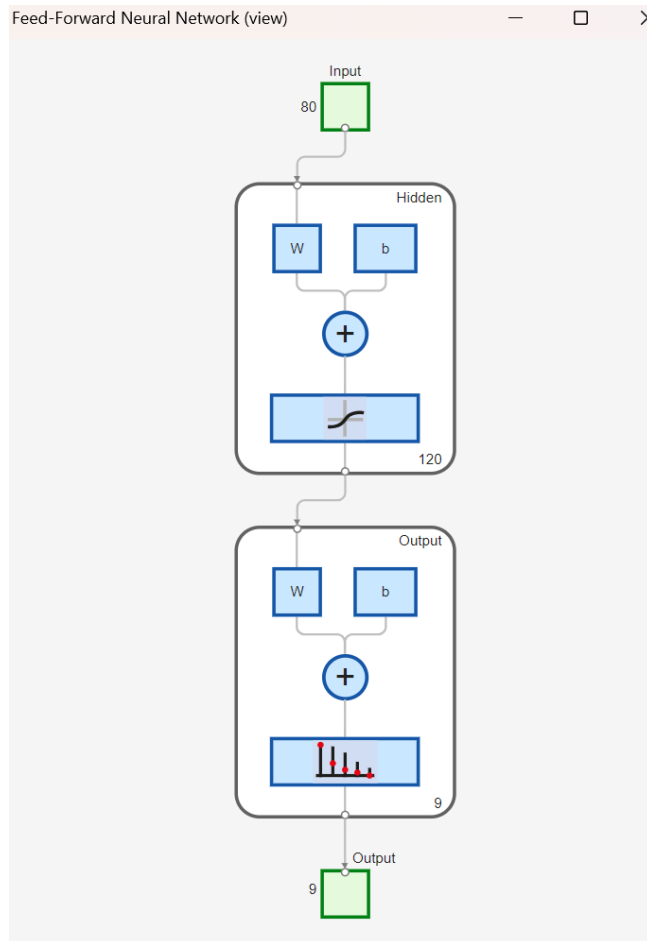


Figura 3.14: Esquema de la Red Neuronal Feed-Forward de clasificación multiclase.

Entrenamiento:

Los datos se dividieron aleatoriamente en conjuntos de entrenamiento, validación y prueba. El algoritmo de optimización utilizado para ajustar los pesos de la red neuronal es Scaled Conjugate

Gradient (trainscg). Durante el entrenamiento, la red neuronal ajusta sus pesos para minimizar la función de pérdida (en este caso, crossentropy para clasificación multiclase).

En la Figura 3.14 se resumen los resultados de entrenamiento:

- Muestra que el entrenamiento ha finalizado porque se cumplió el criterio de validación. Lo que significa que el modelo ha alcanzado un punto en el que ya no mejora significativamente en el conjunto de validación, por lo que se detiene para evitar el sobreajuste.
- La función de pérdida disminuyó de 0.0659 a 0.0363, lo que indica que el modelo mejoró durante el entrenamiento.
- El gradiente disminuyó, lo que indica que el modelo se acercó a un mínimo de la función de pérdida.
- El entrenamiento se detuvo porque el rendimiento en el conjunto de validación no mejoró durante 6 verificaciones consecutivas.

Validación y pruebas

La matriz de confusión en la Figura 3.15, muestra un ejemplo del desempeño del modelo al clasificar las huellas dactilares en diferentes clases:

- La diagonal principal (celdas azules) representa las instancias correctamente clasificadas. Por ejemplo, la clase (persona) 12 fue correctamente identificada 5 veces, al igual que la clase 13, 45, 47, 57, y 76 con 5 o 6 aciertos. Esto indica que el modelo tiene una precisión aceptable en algunas clases.
- Los valores fuera de la diagonal representan los errores de clasificación (celdas rojas). Por ejemplo:
 - o La clase 12 fue confundida con la clase 13 en 5 ocasiones.
 - o La clase 45 fue clasificada incorrectamente como 13 en 5 ocasiones.
 - o La clase 47 fue confundida con 45 en 6 ocasiones.

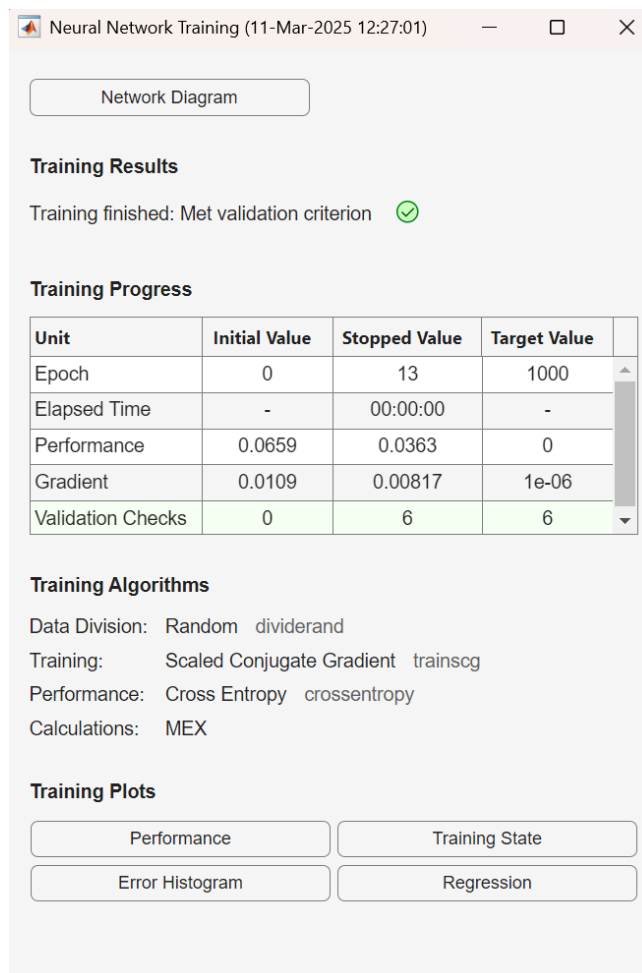


Figura 3.15: Resumen entrenamiento modelo clasificación multiclase.

o Esto sugiere que algunas clases tienen características similares y el modelo no logra diferenciarlas bien por lo que se podría ajustar la arquitectura de la red

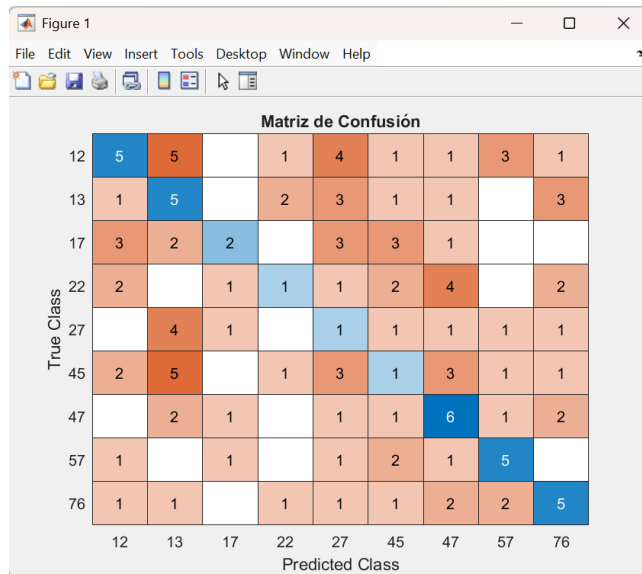


Figura 3.16: Matriz confusión modelo clasificación multiclase.

Se realizaron pruebas de identificación, con muestras aleatorias, se ejemplifica el resultado con el mayor porcentaje de precisión obtenido, el modelo predijo correctamente la clase de la muestra (etiqueta real: 27, etiqueta predicha: 27), Sin embargo, este es solo un ejemplo aislado.

```

Numero de clases: 9
Etiquetas unicas: [12;13;17;22;27;45;47;57;76]
Precision en el conjunto de validacion: 77.52%
Prueba de identificacion:
  - Etiqueta real: 27
  - Etiqueta predicha: 27
  - Resultados: Identificacion correcta.

```

Los resultados muestran que el modelo tiene un rendimiento bueno (77.52% de precisión), pero aún hay margen de mejora, lo ideal debería acercarse al 100%. La prueba de identificación correcta es un buen indicador, pero es importante seguir optimizando el modelo para aumentar la precisión general ajustando la arquitectura de la red.

3.4. Implementación con redes neuronales convolucionales.

En la metodología se siguió como primer paso, seleccionar, los conjuntos de imágenes que serán utilizados. Después, se separaron las imágenes en entrenamiento y prueba. Posteriormente, se realizó el diseño y la construcción de la red tomando como base el diseño de AlexNet y GoogleNet, finalmente se entrenó y validó la red diseñada.

3.4.1. Conjunto de datos utilizados.

El conjunto de imágenes está formado por un total de 9 clases que representan los distintos usuarios a identificar, cada clase contiene 48 imágenes sumando un total de 432 imágenes.

Se generaron carpetas que contienen las imágenes de las huellas digitales, clasificándolas por el número de usuario al que pertenece; es decir; cada carpeta representa a una clase diferente.

Al generar estas carpetas se dividió la base de datos en un conjunto de entrenamiento y un conjunto de prueba.

3.4.2. Implementación de clasificador con AlexNet.

La red AlexNet, descrita en la sección 2.2.3, fue diseñada originalmente para clasificar imágenes de dimensiones $227 \times 227 \times 3$ y es capaz de distinguir entre 1000 clases diferentes. Para adaptar esta red a nuestras necesidades, fue necesario realizar ajustes en los datos y en la arquitectura de la red.

3.4.2.1. Preprocesamiento de los datos.

En primer lugar, se redimensionaron todas las imágenes del conjunto de datos al tamaño requerido por AlexNet ($227 \times 227 \times 3$). Esto se logró utilizando la función `imresize` de la toolbox

de procesamiento de imágenes de MATLAB. Posteriormente, se organizaron las imágenes en un almacén de datos mediante la función `imageDatastore`, que permite gestionar grandes volúmenes de imágenes y sus etiquetas correspondientes.

```
allImages = imageDatastore('TrainingData', IncludeSubfolders', true, ...  
    'LabelSource', 'foldernames');
```

3.4.2.2. División del conjunto de datos

El conjunto de imágenes se dividió en dos categorías: entrenamiento y validación. Se asignó el 70% de las imágenes para entrenamiento y el 30% restante para validación. Para garantizar que la selección de imágenes fuera aleatoria y evitar sesgos, se utilizó la función `randomize`, que asegura que no se repita el mismo conjunto de imágenes en cada iteración del entrenamiento.

```
[trainingImages, testImages] = splitEachLabel(allImages, 0.7, 'randomize');
```

3.4.2.3. Modificación de la arquitectura de AlexNet

Dado que las últimas capas de AlexNet están configuradas para clasificar 1000 clases, fue necesario modificar estas capas para adaptarlas a nuestro problema específico, que involucra solo 9 clases. Se eliminaron las últimas capas de la red (capa de activación, softmax y clasificación) y se reemplazaron con una nueva capa completamente conectada (`fullyConnectedLayer`) que especifica el número de salidas deseadas (en este caso, 9). Esta modificación permitió que la red pudiera clasificar correctamente las huellas digitales de los usuarios.

3.4.2.4. Entrenamiento de la red

El entrenamiento de la red se configuró utilizando la función `trainingOptions` de MATLAB. Se especificaron varios parámetros clave, como el entorno de ejecución, el tamaño del lote, la tasa de aprendizaje y el número máximo de épocas. Durante el entrenamiento, la red iteró sobre los

1	'data'	Image Input	227x227x3 images with 'zerocenter' normalization
2	'conv1'	Convolution	96 11x11x3 convolutions with stride [4 4] and padding [0 0 0 0]
3	'relu1'	ReLU	ReLU
4	'norm1'	Cross Channel Normalization	cross channel normalization with 5 channels per element
5	'pool1'	Max Pooling	3x3 max pooling with stride [2 2] and padding [0 0 0 0]
6	'conv2'	Grouped Convolution	2 groups of 128 5x5x48 convolutions with stride [1 1] and padding [2 2 2 2]
7	'relu2'	ReLU	ReLU
8	'norm2'	Cross Channel Normalization	cross channel normalization with 5 channels per element
9	'pool2'	Max Pooling	3x3 max pooling with stride [2 2] and padding [0 0 0 0]
10	'conv3'	Convolution	384 3x3x256 convolutions with stride [1 1] and padding [1 1 1 1]
11	'relu3'	ReLU	ReLU
12	'conv4'	Grouped Convolution	2 groups of 192 3x3x192 convolutions with stride [1 1] and padding [1 1 1 1]
13	'relu4'	ReLU	ReLU
14	'conv5'	Grouped Convolution	2 groups of 128 3x3x192 convolutions with stride [1 1] and padding [1 1 1 1]
15	'relu5'	ReLU	ReLU
16	'pool5'	Max Pooling	3x3 max pooling with stride [2 2] and padding [0 0 0 0]
17	'fc6'	Fully Connected	4096 fully connected layer
18	'relu6'	ReLU	ReLU
19	'drop6'	Dropout	50 % dropout
20	'fc7'	Fully Connected	4096 fully connected layer
21	'relu7'	ReLU	ReLU
22	'drop7'	Dropout	50 % dropout
23	"	Fully Connected	9 fully connected layer
24	'prob'	Softmax	softmax
25	"	Classification Output	crossentropyex

Tabla 3.4: Arquitectura de la red AlexNet modificada

lotes de imágenes, ajustando los pesos de las neuronas para minimizar el error de clasificación. El proceso de entrenamiento se llevó a cabo utilizando el conjunto de entrenamiento, mientras que el conjunto de validación se utilizó para monitorear el rendimiento de la red y evitar el sobreajuste.

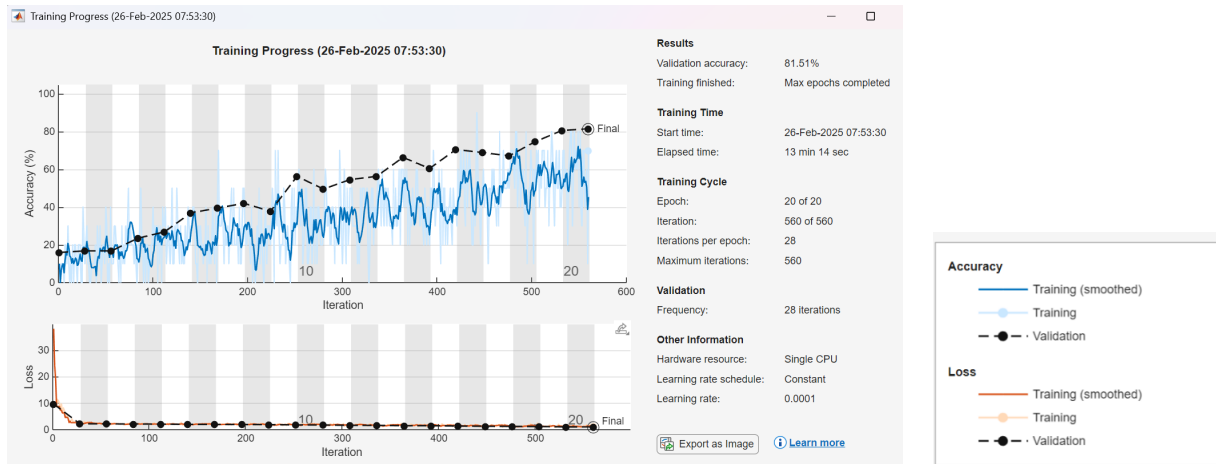
```

opcionesEntrenamiento = trainingOptions('sgdm', ...
    'InitialLearnRate', 1e-4, ...
    'MaxEpochs', 20, ...
    'MiniBatchSize', miniBatchSize, ...
    'Momentum', 0.8, ...
    'VerboseFrequency', frecuenciaValidacion, ...
    'ValidationData', imagenesPrueba, ...
    'ValidationFrequency', frecuenciaValidacion, ...
    'ValidationPatience', 10, ...
    'Plots', 'training-progress';

```

En la imagen, se muestra el progreso del entrenamiento del modelo, incluyendo la precisión en validación, el tiempo de entrenamiento, y otros detalles relevantes como la tasa de aprendizaje, el hardware utilizado e información sobre el rendimiento final después de completar el

entrenamiento.



(a) Nombre 1.

(b) Nombre 2.

Figura 3.17: Progreso y resultados del entrenamiento del modelo

3.4.2.5. Evaluación del modelo

Una vez finalizado el entrenamiento, se evaluó el rendimiento de la red utilizando el conjunto de prueba. La evaluación consistió en tres tareas principales: **predicción de clases, conversión de etiquetas y medición del tiempo de clasificación**. La precisión de la red se calculó como el porcentaje de predicciones correctas sobre el total de imágenes de prueba.

```

inicioClasificacion = tic;
probabilidadesPredichas = predict(redEntrenada, imagenesPrueba, 'MiniBatchSize',
    miniBatchSize);

[~, indicesClasesPredichas] = max(probabilidadesPredichas, [], 2);

etiquetasPredichas = categories(imagenesPrueba.Labels);
etiquetasPredichas = etiquetasPredichas(indicesClasesPredichas);

tiempoClasificacion = toc(inicioClasificacion);
disp(['Tiempo de clasificacion: ', num2str(tiempoClasificacion), ' segundos'])

```

Para visualizar los resultados, se generó una matriz de confusión de dimensiones NxN, donde N es el número de clases (en este caso, 9). La diagonal principal de la matriz muestra la cantidad de imágenes clasificadas correctamente, mientras que las demás celdas indican las clasificaciones erróneas, especificando con qué clase se confundió cada imagen. Esta matriz permitió identificar patrones de error y evaluar la precisión general de la red.

matrizConfusion =

11	3	0	0	0	0	0	0	0
2	12	0	0	0	0	0	0	0
0	3	10	0	0	0	0	1	0
0	0	0	13	0	0	1	0	0
0	0	0	0	14	0	0	0	0
0	0	0	0	0	14	0	0	0
0	0	0	0	0	0	11	0	2
0	0	1	2	0	0	0	3	0
2	2	0	0	1	0	0	0	9

matrizConfusionPorcentual =

78.5714	21.4286	0	0	0	0	0	0	0
14.2857	85.7143	0	0	0	0	0	0	0
0	21.4286	71.4286	0	0	0	0	7.1429	0
0	0	0	92.8571	0	0	7.1429	0	0
0	0	0	0	100.0000	0	0	0	0
0	0	0	0	0	100.0000	0	0	0
0	0	0	0	0	7.1429	11	0	14.2857
0	0	14.2857	28.5714	0	0	0	14.8571	0
14.2857	14.2857	0	0	7.1429	0	0	0	64.2857

```
Tiempo de entrenamiento: 808.9158 segundos  
Tiempo de calificación: 3.4903 segundos  
Precisión: 0.81513
```

3.4.2.6. Prueba de clasificación

Finalmente, se realizaron pruebas adicionales con imágenes de huellas digitales no incluidas en el conjunto de entrenamiento o validación. Estas pruebas permitieron verificar la capacidad de generalización de la red y su precisión en la identificación de personas. Los resultados mostraron que la red AlexNet, con las modificaciones realizadas, fue capaz de clasificar la mayoría de las imágenes, demostrando su eficacia en la tarea de reconocimiento de huellas digitales.

el contenido de la imagen, muestra las predicciones realizadas por el modelo, la confianza asociada a cada predicción y las etiquetas reales de los usuarios. Además muestra la evaluación del rendimiento de la red implementada.

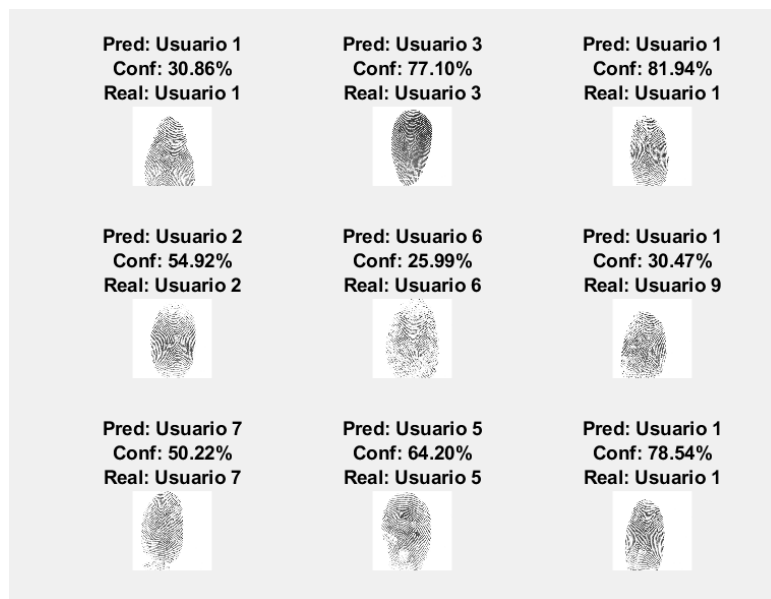


Figura 3.18: Resultados de Clasificación: Predicciones y Confianza en la Identificación de usuarios.

3.4.2.7. Interpretación de resultados

La matriz de confusión reveló que la red tuvo un buen porcentaje de aciertos en la clasificación de las huellas digitales, lo que indica que el modelo generaliza bien para el conjunto de datos utilizado. Sin embargo, se observaron algunos errores de clasificación, principalmente en casos donde las huellas presentaban características similares entre usuarios. Estos errores podrían reducirse en futuras iteraciones del modelo mediante la incorporación de más datos de entrenamiento o la aplicación de técnicas de aumento de datos para mejorar la variabilidad del conjunto.

3.4.2.8. Ajustes para mejorar el rendimiento de la red Alexnet

Aunque el modelo alcanzó una precisión del 81.51 % en el conjunto de validación, con una precisión media del 79.37 %, y mostró un tiempo de clasificación relativamente bajo (3.49 segundos), es posible realizar ajustes adicionales para optimizar aún más su rendimiento. El tiempo de entrenamiento fue de 808.92 segundos, lo que indica que el proceso es eficiente, pero existen áreas de mejora que podrían incrementar la precisión y reducir el tiempo de ejecución.

Se realizaron los siguientes ajustes:

- Uso de un optimizador diferente. Se cambió la técnica de optimización sgd (Stochastic Gradient Descent with Momentum) por Adam.
- Ajuste de la tasa de aprendizaje (InitialLearnRate). Se probó una reducción progresiva de la tasa de aprendizaje usando learningRateSchedule = 'piecewise', bajando la tasa después de ciertas épocas para mejorar la estabilidad.
- Aumento de datos (Data Augmentation). Se aplican transformaciones a las imágenes existentes (rotaciones, traslaciones, volteos, cambios de brillo, etc.) para generar más datos de entrenamiento.
- Modificación en la división de datos. 80
- Aumento del número de épocas.

- Cambios en la arquitectura de la red. Se agregó Batch Normalization después de cada capa fullyConnectedLayer para mejorar la estabilidad del entrenamiento.
- Reemplazar dropoutLayer(0.5) con valores más pequeños, como 0.3, para mejorar el balance entre regularización y retención de información.
- Ajuste en el tamaño del mini-lote (MiniBatchSize). Se probó con valores más altos, como 32 o 64

Con estos ajustes se obtuvieron los siguientes resultados de entrenamiento y evaluación.

Entrenamiento

- Tiempo total de entrenamiento: del 1163.08 segundos a 1246.7998 segundos (aproximadamente 19 minutos y 23 segundos a 22 minutos y 27 segundos).
- Precisión en validación: Se obtuvo de 92.94 % al 100 %, lo que indica un buen ajuste del modelo durante el proceso de aprendizaje.

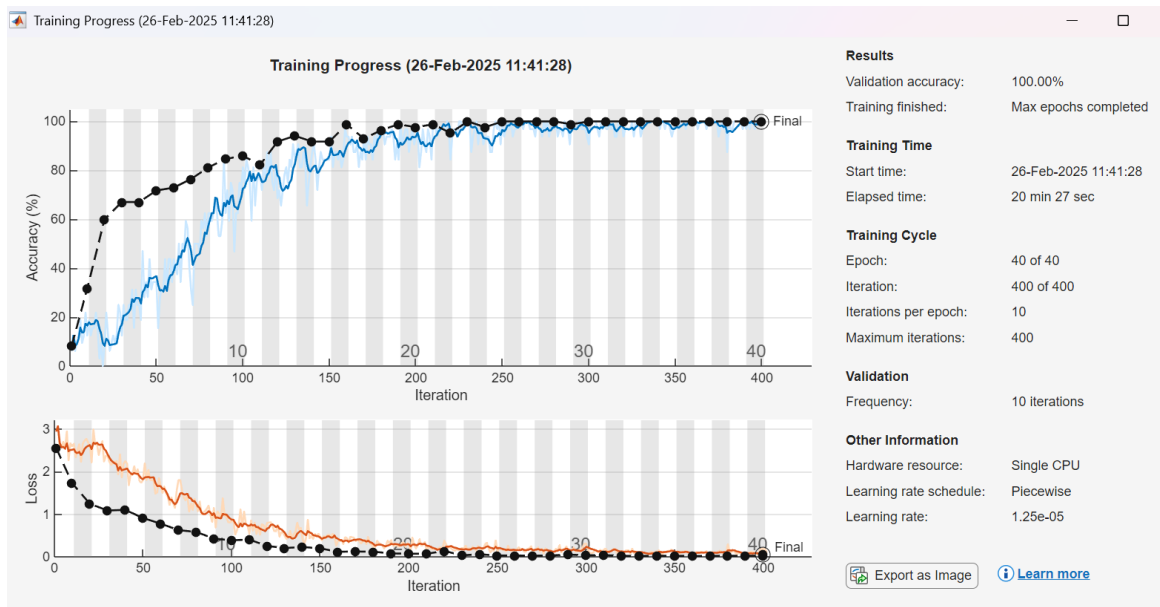


Figura 3.19: Progreso del Entrenamiento: Precisión del 100 % en Validación y Detalles del Ciclo de Entrenamiento.

matrizConfusion =

100	0	0	0	0	0	0	0	0
0	100	0	0	0	0	0	0	0
0	0	100	0	0	0	0	0	0
0	0	0	100	0	0	0	0	0
0	0	0	0	100	0	0	0	0
0	0	0	0	0	100	0	0	0
0	0	0	0	0	0	100	0	0
0	0	0	0	0	0	0	100	0
0	0	0	0	0	0	0	0	100

3.4.2.10. Análisis visual

Se realizaron pruebas con imágenes de huellas dactilares para verificar la precisión del modelo. En la mayoría de los casos, la predicción coincidió con la etiqueta real con altos niveles de confianza. Sin embargo, hubo algunos casos con menor confianza, lo que podría indicar posibles mejoras en el modelo o en la calidad de los datos de entrada.

El modelo muestra un rendimiento sólido con una precisión superior al 90 %, tiempos de procesamiento adecuados y un buen ajuste a los datos de prueba, lo que sugiere que es apto para su implementación en tareas de reconocimiento biométrico.

En conclusión, la implementación de AlexNet, con las adaptaciones realizadas, demostró ser una herramienta efectiva para la clasificación de huellas digitales, alcanzando una precisión satisfactoria en la identificación de usuarios.

3.4.3. Implementación de clasificador con GoogleNet

Las imágenes son redimensionadas a un tamaño de 224x224x3 ajustándola al tamaño requerido por la capa de entrada de la CNN GoogleNet. Se utilizó la función googlenet para cargar la red pre entrenada, como se muestra en el fragmento de Código.

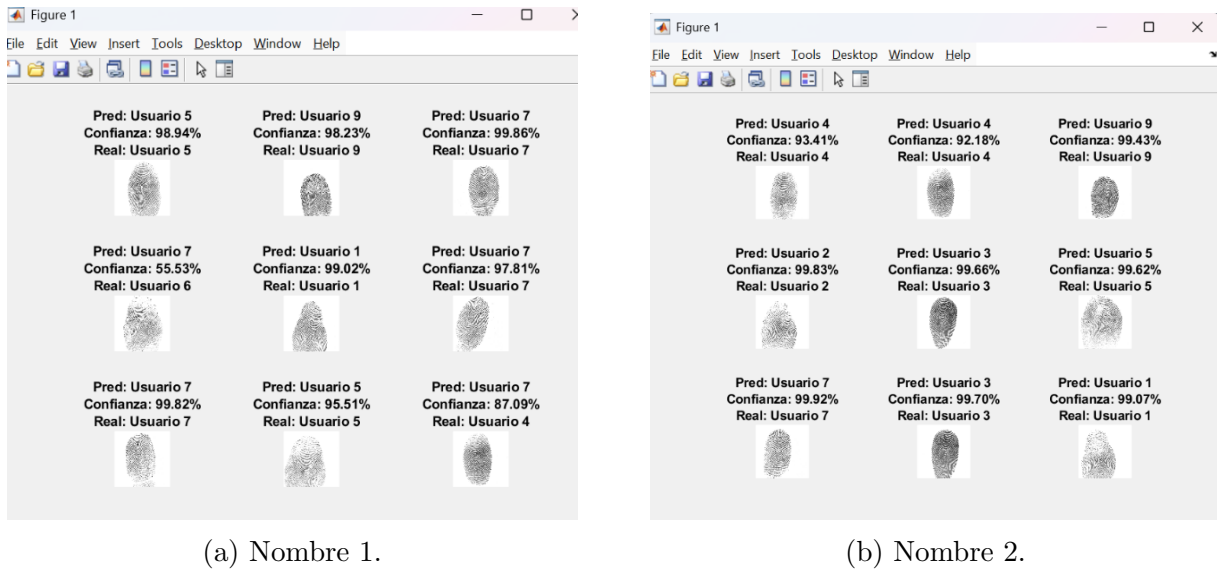


Figura 3.20: Resultados de clasificación: Predicción y niveles de confianza en la identificación de usuarios.

```
net = googlenet; %carga de la preentrada
```

Al igual que en la implementación de AlexNet, el conjunto de imágenes de test está dado por un almacén de datos, que incluye principalmente, los archivos de imagen y las etiquetas de cada una de las mismas. Se dividen las dos categorías, una para entrenamiento (289 imágenes) y otro para validación (119) y se seleccionan las imágenes de forma aleatoria.

```
allImages = imageDatastore('TrainingData', 'IncludeSubfolders', true, ...
    'LabelSource', 'foldernames');
[trainingImages, testImages] = splitEachLabel(allImages, 0.7, 'randomize');
```

3.4.3.1. Entrenamiento de la red en GoogleNet.

Para el entrenamiento de la CNN GoogleNet, se configuran diversos parámetros mediante la función `trainingOptions`. Entre estos parámetros se incluyen:

- **Algoritmo de optimización:** Se utiliza Stochastic Gradient Descent with Momentum (SGDM), un método ampliamente empleado en el entrenamiento de redes neuronales debido a su eficacia.

- **Conjunto de imágenes de validación:** Se define el conjunto de datos de validación (testImages) para evaluar el rendimiento del modelo durante el entrenamiento.
- **Tamaño del lote (MiniBatchSize):** Se procesan 15 imágenes por iteración, lo que permite un equilibrio entre la eficiencia computacional y la precisión del entrenamiento.
- **Número máximo de épocas (MaxEpochs):** Se establece un límite de 20 épocas, donde una época representa un pase completo sobre el conjunto de entrenamiento. Este valor controla cuántas veces la red verá todo el conjunto de datos durante el entrenamiento.
- **Tasa de aprendizaje inicial (InitialLearnRate):** Se fija en 1e-4, un valor común para problemas de clasificación de imágenes. La tasa de aprendizaje determina la magnitud de los ajustes en los pesos de la red durante cada iteración, influyendo directamente en la velocidad y estabilidad del entrenamiento.
- **Frecuencia de validación (ValidationFrequency):** La red se valida cada 10 iteraciones para monitorear su rendimiento y evitar el sobreajuste.
- **Entorno de ejecución (ExecutionEnvironment):** En este caso, se utiliza la CPU ('cpu') para llevar a cabo el entrenamiento.

```
%opciones de entrenamiento
oprions = trainingOptions('sgdm', ...
    'MiniBatchSize', 15, ...
    'MaxEpochs', 20, ...
    'InitialLearnRate', 1e-4, ...
    'ValidationData', testImages, ...
    'ValidationFrequency', 10, ...
    'Verbose', 1, ...
    'ExecutionEnvironment', 'cpu', ...
    'Plots', 'training-progress');
```

Durante el proceso de entrenamiento, las imágenes de huellas digitales se cargan y preprocesan utilizando la función personalizada leerImágenes. Posteriormente, los datos se dividen en dos conjuntos: uno para entrenamiento (trainingImages) y otro para validación (testImages). Una vez configurados los parámetros, la red se entrena utilizando el conjunto de entrenamiento

y se valida periódicamente con el conjunto de validación. Esto permite ajustar los pesos de la red y garantizar que el modelo generalice correctamente. Al finalizar el entrenamiento, se inspecciona la última capa de la red para verificar el rendimiento del modelo y asegurar que ha aprendido adecuadamente las características de las huellas digitales.

```
%entrenamiento
testImages.ReadFcn = @leerImagenes;
trainingImages.ReadFcn = @leerImagenes;
trainedGN = trainNetwork(trainingImages, lgraph, options);
trainedGN.Layers(end) %inspeccion de la ultima capa de la red entrenada
```

En la siguiente imagen, se muestra el progreso del entrenamiento, incluyendo métricas como la pérdida y la precisión en los conjuntos de entrenamiento y validación.

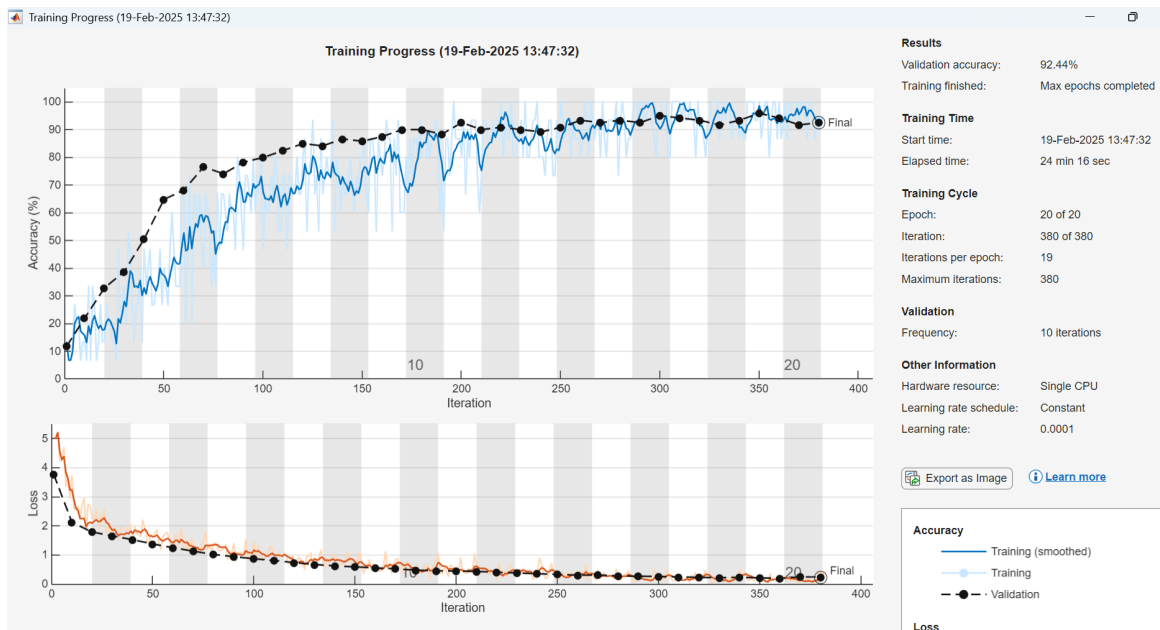


Figura 3.21: Progreso del Entrenamiento de Modelo GoogleNet.

3.4.3.2. Evaluación de la red con datos de validación.

Una vez finalizado el entrenamiento de la red GoogleNet, se procede a evaluar su rendimiento utilizando el conjunto de datos de validación. Este proceso consta de los siguientes pasos:

```

Cantidad de imágenes de entrenamiento: 289
Cantidad de imágenes de validación : 119
Initializing input data normalization.
=====
| Epoch | Iteration | Time Elapsed | Mini-batch | Validation | Mini-batch | Validation | Base Learning |
|       |          | (hh:mm:ss)  | Accuracy   | Accuracy   | Loss       | Loss       | Rate          |
|=====|=====|=====|=====|=====|=====|=====|=====|
| 1 | 1 | 00:00:13 | 13.33% | 11.76% | 4.9957 | 3.7615 | 1.0000e-04 |
| 1 | 10 | 00:00:48 | 13.33% | 21.85% | 3.7160 | 2.1132 | 1.0000e-04 |
| 2 | 20 | 00:01:24 | 13.33% | 32.77% | 2.2373 | 1.7825 | 1.0000e-04 |
| 2 | 30 | 00:02:01 | 33.33% | 38.66% | 1.7146 | 1.6480 | 1.0000e-04 |
| 3 | 40 | 00:02:37 | 46.67% | 50.42% | 1.4749 | 1.5198 | 1.0000e-04 |
| 3 | 50 | 00:03:12 | 40.00% | 64.71% | 1.6139 | 1.3715 | 1.0000e-04 |
| 4 | 60 | 00:03:48 | 66.67% | 68.07% | 1.1912 | 1.2442 | 1.0000e-04 |
| 4 | 70 | 00:04:24 | 40.00% | 76.47% | 1.2767 | 1.1238 | 1.0000e-04 |
| 5 | 80 | 00:05:00 | 66.67% | 73.95% | 1.3764 | 1.0116 | 1.0000e-04 |
| 5 | 90 | 00:05:36 | 46.67% | 78.15% | 1.3860 | 0.9372 | 1.0000e-04 |
| 6 | 100 | 00:06:12 | 46.67% | 79.83% | 1.4228 | 0.8632 | 1.0000e-04 |
| 6 | 110 | 00:06:48 | 73.33% | 82.35% | 0.7842 | 0.8036 | 1.0000e-04 |
| 7 | 120 | 00:07:25 | 80.00% | 84.87% | 0.6253 | 0.7301 | 1.0000e-04 |
| 7 | 130 | 00:08:02 | 66.67% | 84.03% | 0.9035 | 0.6667 | 1.0000e-04 |
| 8 | 140 | 00:08:38 | 73.33% | 86.55% | 0.7849 | 0.6183 | 1.0000e-04 |
| 8 | 150 | 00:09:14 | 80.00% | 85.71% | 0.6044 | 0.5852 | 1.0000e-04 |
| 9 | 160 | 00:09:49 | 73.33% | 87.39% | 0.5192 | 0.5490 | 1.0000e-04 |
| 9 | 170 | 00:10:26 | 60.00% | 89.92% | 0.8407 | 0.5206 | 1.0000e-04 |
| 10 | 180 | 00:11:01 | 80.00% | 89.92% | 0.6547 | 0.4708 | 1.0000e-04 |
| 10 | 190 | 00:11:37 | 73.33% | 88.24% | 0.8101 | 0.4519 | 1.0000e-04 |
| 11 | 200 | 00:12:13 | 80.00% | 92.44% | 0.5311 | 0.4411 | 1.0000e-04 |
| 12 | 210 | 00:12:49 | 73.33% | 89.92% | 0.5595 | 0.4278 | 1.0000e-04 |
| 12 | 220 | 00:13:26 | 100.00% | 90.76% | 0.1201 | 0.4002 | 1.0000e-04 |
| 13 | 230 | 00:14:02 | 93.33% | 89.92% | 0.3806 | 0.3759 | 1.0000e-04 |
| 13 | 240 | 00:14:40 | 80.00% | 89.08% | 0.4239 | 0.3499 | 1.0000e-04 |
| 14 | 250 | 00:15:24 | 100.00% | 90.76% | 0.1910 | 0.3250 | 1.0000e-04 |
| 14 | 260 | 00:16:06 | 80.00% | 93.28% | 0.4386 | 0.3001 | 1.0000e-04 |
| 15 | 270 | 00:16:47 | 80.00% | 92.44% | 0.4703 | 0.3050 | 1.0000e-04 |
| 15 | 280 | 00:17:27 | 93.33% | 93.28% | 0.2341 | 0.2792 | 1.0000e-04 |
| 16 | 290 | 00:18:08 | 100.00% | 92.44% | 0.2488 | 0.2622 | 1.0000e-04 |
| 16 | 300 | 00:18:49 | 93.33% | 94.96% | 0.2191 | 0.2582 | 1.0000e-04 |
| 17 | 310 | 00:19:30 | 100.00% | 94.12% | 0.0937 | 0.2385 | 1.0000e-04 |
| 17 | 320 | 00:20:10 | 93.33% | 93.28% | 0.2179 | 0.2231 | 1.0000e-04 |
| 18 | 330 | 00:20:50 | 86.67% | 91.60% | 0.1848 | 0.2147 | 1.0000e-04 |
| 18 | 340 | 00:21:29 | 86.67% | 93.28% | 0.4928 | 0.2249 | 1.0000e-04 |
| 19 | 350 | 00:22:08 | 100.00% | 95.80% | 0.1274 | 0.2084 | 1.0000e-04 |
| 19 | 360 | 00:22:48 | 100.00% | 94.12% | 0.1514 | 0.1906 | 1.0000e-04 |
| 20 | 370 | 00:23:29 | 93.33% | 91.60% | 0.1110 | 0.2394 | 1.0000e-04 |
| 20 | 380 | 00:24:09 | 93.33% | 92.44% | 0.3368 | 0.2333 | 1.0000e-04 |
|=====|=====|=====|=====|=====|=====|=====|=====|
Training finished: Max epochs completed.

```

Figura 3.22: Progreso del Entrenamiento del Modelo con Exactitud y Pérdida por Época e Iteración.

$confMat =$

1.000	0	0	0	0	0	0	0	0
0	1.000	0	0	0	0	0	0	0
0	0	1.000	0	0	0	0	0	0
0	0	0	0.9286	0	0	0.0714	0	0
0	0	0	0	0.7857	0.2143	0	0	0
0.0714	0	0	0	0	0.9286	0	0	0
0	0	0	0	0	0.0714	0.9286	0	0
0	0	0.4286	0	0	0	0	0.5714	0
0	0	0	0	0	0	0	0	1.000

Resultados obtenidos. El valor de accuracy indica la precisión del modelo de la red en el conjunto de validación, significa que el 92.437 % de las imágenes fueron clasificadas correctamente (exactitud), lo que indica que es efectivo para la mayoría de las clases, Sin embargo, hay cierta confusión entre algunas clases, especialmente entre Usuario 5 y Usuario 6, y entre Usuario 8 y Usuario 3. Esto podría deberse a similitudes en las características de las huellas digitales de estos usuarios o a un desbalance en el conjunto de datos.

La Figura 19 muestra ejemplos concretos de cómo el modelo GoogleNet clasifica imágenes de huellas digitales, porcentajes asociados a cada predicción. Como se puede observar, el modelo tiene un alto nivel de confianza en la mayoría de las clasificaciones, como en el caso de "Usuario 6, 100 %" y "Usuario 5, 99.4 %". Sin embargo, también se identifican casos donde la confianza es ligeramente menor, como en "Usuario 3, 91.1 %", lo que sugiere que el modelo tiene cierta incertidumbre en estas predicciones. Estos resultados refuerzan la precisión general del modelo (92.437 %) y proporcionan hallazgos valiosos para futuras mejoras.

Para calcular la precisión se analiza la matriz de confusión en la cual cada fila representa la clase real, y cada columna representa la clase predicha. Para calcular la precisión por clase, seguimos estos pasos:

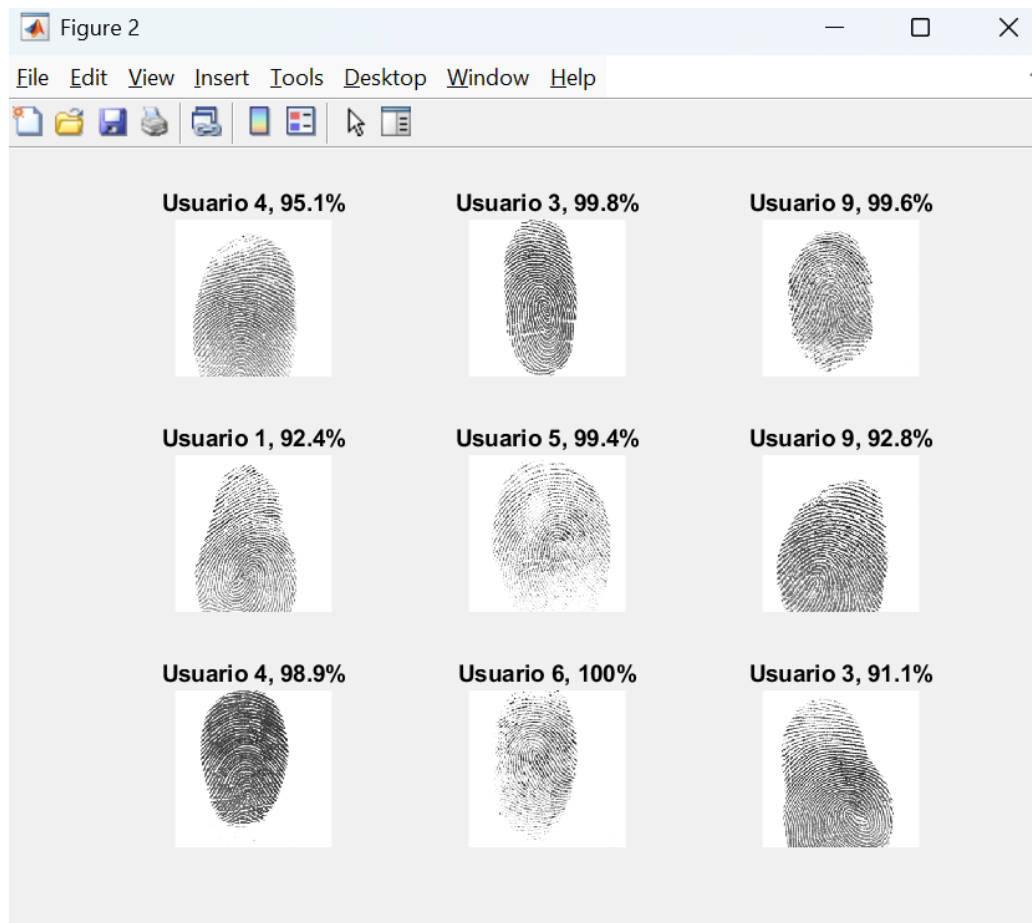


Figura 3.23: Ejemplos de clasificación del modelo GoogleNet

1 Identificar TP y FP para cada clase:

- o **TP:** Valor en la diagonal principal para cada clase.
- o **FP:** Suma de los valores en la columna correspondiente a la clase, excluyendo el TP.

2 **Calcular la precisión para cada clase:**

- o Usamos la fórmula de precisión $= \frac{TP}{TP+FP}$.

Por ejemplo el cálculo para la clase usuario 4 ;

- **TP (Usuario 4):** 13 (el modelo clasificó correctamente 13 imágenes como Usuario 4).
- **FP (Usuario 4):** 1 (el modelo clasificó incorrectamente 1 imagen como Usuario 4).

Precisión (Usuario 4) = $13/13+1 \times 100 = 92.86\%$

A continuación, se muestra la precisión para cada clase:

Clase	TP	FP	Precisión (%)
Usuario 1	14	1	93.33
Usuario 2	14	0	100.00
Usuario 3	14	3	82.35
Usuario 4	13	1	92.86
Usuario 5	11	0	100.00
Usuario 6	13	4	76.47
Usuario 7	13	0	100.00
Usuario 8	4	0	100.00
Usuario 9	14	0	100.00

Capítulo 4

Resultados

4.1. Resultados obtenidos

En este capítulo se presentan los resultados experimentales obtenidos a partir de la implementación de los algoritmos propuestos basados en redes neuronales artificiales (RNA) y redes neuronales convolucionales (CNN). Se analizan tanto las tasas de error como los tiempos de ejecución, con el objetivo de evaluar la eficacia y eficiencia de los clasificadores desarrollados. Además, se comparan los resultados entre las diferentes arquitecturas para determinar cuál de ellas ofrece un mejor rendimiento en la tarea de clasificación de huellas digitales.

4.1.1. Base de Datos

La base de datos utilizada para validar los algoritmos está compuesta por imágenes de huellas dactilares correspondientes a 9 usuarios diferentes. Cada usuario cuenta con 48 muestras de huella, tomadas en diferentes momentos y condiciones, lo que suma un total de 432 imágenes. Las imágenes tienen un tamaño de 256 x 256 píxeles y se dividieron en conjuntos de entrenamiento y prueba para evaluar el rendimiento de los clasificadores.

4.1.2. Métricas de Evaluación

Para evaluar el rendimiento de los algoritmos de reconocimiento biométrico, se utilizaron las siguientes métricas:

- **FRR (False Rejection Rate):** Tasa de falso rechazo, que representa la probabilidad de que un usuario autorizado sea incorrectamente rechazado por el sistema.
- **FAR (False Acceptance Rate):** Tasa de falsa aceptación, que indica la probabilidad de que un usuario no autorizado sea incorrectamente aceptado por el sistema.
- **EER (Equal Error Rate):** Punto en el que las tasas FAR y FRR se cruzan, proporcionando un umbral óptimo para el sistema.
- **Precisión en validación:** Porcentaje de predicciones correctas sobre el total de imágenes evaluadas.
- **Matriz de Confusión:** Herramienta que permite visualizar el rendimiento del clasificador, mostrando cuántas predicciones fueron correctas y cuántas se confundieron entre clases.
- **Tiempo de Entrenamiento:** Tiempo total requerido para entrenar el modelo.
- **Tiempo de Clasificación:** Tiempo requerido para clasificar una nueva imagen.

4.1.3. Resultados Experimentales del Algoritmo Basado en Redes Neuronales Artificiales

Se presentan los resultados obtenidos a partir de la implementación del algoritmo propuesto basado en redes neuronales artificiales (RNA, Se analizan los tiempos de ejecución de cada etapa del proceso, así como el rendimiento del clasificador en términos de precisión y tasas de error. Además, se incluyen tablas y gráficas que permiten visualizar de manera clara el desempeño del modelo.

El algoritmo basado en RNA fue implementado utilizando un perceptrón multicapa con algoritmo de retropropagación. Se realizaron pruebas con diferentes configuraciones de capas ocultas (10, 15, 20, 30 y 40) para determinar cuál ofrecía el mejor rendimiento.

4.1.3.1. Tiempos de Ejecución

Los tiempos de ejecución de cada una de las etapas que conforman el algoritmo completo se obtuvieron utilizando la función tic-toc de MATLAB en una computadora personal. A continuación, se presenta una tabla con los tiempos promedio de cada etapa:

Etapas	Tiempo (segundos)
Mejoramiento de la huella	
Segmentación	2.63
Normalización	2.97
Orientación	3.30
Filtrado	72.18
Binarización	65.70
Adelgazamiento	57.05
Tiempo total	203.83
Extracción de características	
Identificación de minucias	3.36
Eliminación de falsas minucias	0.64
Tiempo total	4.00
Tiempo total de procesamiento	207.83
Tiempo de reconocimiento	0.01443103

Tabla 4.1: Tiempos de ejecución

Como se observa, la etapa de mejoramiento de la huella es la más costosa en términos de tiempo, seguida por la binarización y el adelgazamiento. Sin embargo, el tiempo de reconocimiento es extremadamente bajo, lo que indica que el sistema es eficiente una vez que las características de la huella han sido extraídas.

4.1.3.2. Desempeño de la Red Neuronal

Se realizaron pruebas con diferentes configuraciones de capas ocultas (10, 15, 20, 30, 40 y 50) para evaluar el impacto en el tiempo de entrenamiento y reconocimiento. Los resultados se muestran en la siguiente tabla:

Como se puede observar, a medida que aumenta el número de capas ocultas, el tiempo de entrenamiento se incrementa significativamente. Sin embargo, el tiempo de reconocimiento se mantiene prácticamente constante, lo que sugiere que el modelo es eficiente en la fase de

Capas ocultas	Tiempo promedio de entrenamieto (minutos)	Tiempo promedio de reconocimiento (segundos)
10	10.14	0.0209
15	23.31	0.0156
20	44.48	0.0148
30	108.21	0.0144
40	228.53	0.0144

Tabla 4.2: Relación entre profundidad del modelo y eficiencia computacional: Tiempos de entrenamiento vs. reconocimiento

clasificación, independientemente de la complejidad de la red.

4.1.3.3. Tasas de Reconocimiento y Errores de Clasificación

El algoritmo fue evaluado utilizando un conjunto de 130 huellas digitales correspondientes a 9 individuos, con 15 muestras por cada uno . A continuación, se presentan los resultados en términos de tasas de reconocimiento y errores de clasificación.

Reconocimiento del Individuo

se evaluó la capacidad del modelo para reconocer correctamente al individuo P1 (positivos) y determinar falsas aceptaciones con otros individuos. Los resultados se muestran en la Tabla 4.3.

Capas ocultas	Pruebas	Positivos (P1)	Falsa aceptación
10	30	100 %	52.42 %
15	30	100 %	37.14 %
20	30	90 %	11.85 %
30	30	74 %	3.28 %
40	30	65 %	6.00 %

Tabla 4.3: Relación entre complejidad del modelo (Capas) y exactitud en identificación

Los resultados muestran que, a medida que se aumenta el número de capas ocultas, la tasa de falsa aceptación disminuye significativamente. Sin embargo, el porcentaje de reconocimiento positivo también tiende a disminuir, lo que sugiere un equilibrio entre la precisión y la capacidad

de generalización del modelo.

En una segunda prueba, se evaluó la capacidad del modelo para reconocer correctamente al individuo **P2**. Los resultados se muestran en la Tabla.

Capas ocultas	Pruebas	Positivos (P2)	Falsa aceptación
10	30	93 %	53.00 %
15	30	90 %	31.40 %
20	30	80 %	17.00 %
30	30	80 %	2.57 %
40	30	67 %	1.28 %

Tabla 4.4: Relación entre complejidad del modelo (Capas) y Exactitud en pruebas de Identificación de individuo P2

4.1.3.4. Tasas Promedio de Reconocimiento

Finalmente, se calculó la tasa promedio de reconocimiento para cada configuración de capas ocultas. Los resultados se muestran en la Tabla 4.5.

Capas ocultas	Porcentaje de reconocimiento
10	88 %
15	85 %
20	81 %
30	80 %
40	67 %

Tabla 4.5: Porcentaje de éxito en reconocimiento según profundidad del modelo

4.1.3.5. Análisis de Resultados

Los resultados muestran que el modelo basado en RNA es capaz de alcanzar una alta precisión en la clasificación de huellas digitales, especialmente con configuraciones de 10 a 30 capas ocultas. Sin embargo, a medida que se incrementa el número de capas, el tiempo de entrenamiento aumenta significativamente, mientras que la precisión tiende a disminuir. Esto sugiere

que existe un punto óptimo en la complejidad de la red, donde se logra un equilibrio entre precisión y eficiencia computacional.

En términos de tasas de error, se observa que el modelo tiene una mayor tendencia a cometer falsas aceptaciones con configuraciones de menor complejidad (10 y 15 capas ocultas). Sin embargo, con configuraciones más complejas (30 capas ocultas o más), las tasas de falsa aceptación disminuyen considerablemente, lo que indica una mayor robustez del modelo.

En conclusión, el algoritmo basado en RNA demostró ser efectivo para la clasificación de huellas digitales, alcanzando tasas de reconocimiento superiores al 80 % en la mayoría de las configuraciones. Sin embargo, es importante considerar el equilibrio entre la precisión y el tiempo de entrenamiento al seleccionar la configuración óptima de la red. En futuras líneas de trabajo, se podrían explorar técnicas de optimización para reducir el tiempo de entrenamiento y mejorar aún más la precisión del modelo.

4.1.4. Resultados del algoritmo Basado en Redes Neuronales Convolucionales

Se implementaron dos arquitecturas de CNN: AlexNet y GoogleNet. Ambas redes fueron entrenadas utilizando transferencia de aprendizaje, lo que permitió adaptar los modelos preentrenados a la tarea específica de clasificación de huellas digitales. 5.4.1 Resultados de AlexNet AlexNet fue entrenada con el conjunto de imágenes divididas en 80 % para entrenamiento y 20 % para validación. Los resultados obtenidos fueron los siguientes:

- **Precisión en Validación:** 100 %
- **Precisión Media:** 100 %
- **Tiempo de Entrenamiento:** 1246.7998 segundos (22 minutos y 27 segundos)
- **Tiempo de Clasificación:** 2.1921 segundos

La matriz de confusión mostró que la red tuvo un buen porcentaje de aciertos en la clasificación de las huellas digitales, aunque se observaron algunos errores de clasificación, principalmente

en casos donde las huellas presentaban características similares entre usuarios.

4.1.4.1. Resultados de GoogleNet

GoogleNet fue entrenada con el mismo conjunto de datos que AlexNet. Los resultados obtenidos fueron los siguientes:

- **Precisión en Validación:** 92.44 %
- **Precisión Media:** 93.89 %
- **Tiempo de Entrenamiento:** 22 minutos y 19 segundos
- **Tiempo de Clasificación:** 2.19 segundos

La matriz de confusión de GoogleNet mostró un buen porcentaje de aciertos, aunque se observaron algunas confusiones entre clases, especialmente entre usuarios con huellas digitales similares. Estos errores podrían reducirse mediante la incorporación de más datos de entrenamiento o la aplicación de técnicas de aumento de datos.

GoogleNet mostró un buen rendimiento en términos de precisión, alcanzando un 92.44 % en validación. Sin embargo, el tiempo de entrenamiento fue significativamente menor, lo que sugiere que GoogleNet es menos compleja computacionalmente que incluye menos capas y parámetros.

4.1.4.2. Comparativa entre AlexNet y GoogleNet

En la siguiente tabla se presenta una comparativa entre las dos arquitecturas de CNN:

Métrica	AlexaNet	GoogleNet
Precisión en validación	100 %	92.44 %
Precisión en prueba	100 %	92.44 %
Precisión media	100 %	93.89 %
Tiempo de entrenamiento	22 min 22 s	22 min 19 s
Timepo de clasificaión	2.19 s	2.19 s

Tabla 4.6: Análisis comparativo entre AlexNet y GoogleNet: métricas de precisión y eficiencia

4.1.4.3. Conclusiones Parciales

Ambas redes mostraron una alta estabilidad, con precisiones medias superiores al 90 %. Sin embargo, AlexNet con los ajustes antes descritos demostró una mayor consistencia, alcanzando precisiones del 100 % en algunos casos. Sin embargo, GoogleNet sigue siendo una opción viable, especialmente en aplicaciones donde se requiere una arquitectura más profunda y compleja.

Los resultados obtenidos demuestran que las redes neuronales convolucionales, en particular AlexNet, ofrecen un mejor rendimiento en términos de precisión y tiempo de ejecución en comparación con las redes neuronales artificiales tradicionales. Además, las CNN eliminan la necesidad de un preprocesamiento manual de las imágenes, lo que simplifica el proceso de clasificación.

4.1.5. Conclusiones Generales y Futuras Líneas de Trabajo

Tras la implementación y evaluación de los clasificadores basados en RNA y CNN, se puede concluir que las redes convolucionales son más eficientes y precisas para el reconocimiento de huellas dactilares. Sin embargo, es importante destacar que el rendimiento de las RNA puede mejorarse optimizando el número de capas ocultas y ajustando los parámetros de entrenamiento.

Como futuras líneas de trabajo se podrían explorar técnicas de aumento de datos para mejorar la precisión de los modelos, así como la implementación de optimizadores más eficientes para reducir el tiempo de entrenamiento, mejores hiperparámetros para las RNA, con el fin de mejorar su rendimiento. Además, se podría considerar la integración de otras arquitecturas de CNN, como ResNet, para evaluar su rendimiento en la tarea de clasificación de huellas digitales y la Implementación en Hardware Especializado como GPUs o TPUs, para reducir los tiempos de entrenamiento y ejecución.

En conclusión, la implementación de AlexNet demostró ser una herramienta efectiva para la clasificación de huellas digitales, alcanzando una precisión superior al 92 % y, en algunos casos, del 100 %, lo que la convierte en una opción robusta y eficiente para aplicaciones de identificación biométrica.

Bibliografía

- [1] P. R. P. B. Carrasco Miguel A., “RECONOCIMIENTO BIOMETRICO DE AUDIO Y ROSTRO: UN SISTEMA VIABLE DE IDENTIFICACIÓN,” Chile, 2003.
- [2] Taipei, “Sensing Fingerprints,” *A & S International*, p. 102, 2003.
- [3] M. Election, “Automatic Fingerprint Identification,” *IEEE Spectrum*, p. 36, 1973.
- [4] M. P. e. Wikipedia, “https://es.wikipedia.org/wiki/Huella_dactilar,” [Último acceso: Abril 2025].
- [5] M. E. T. Martín, “ESTUDIO DE LA INTEROPERABILIDAD DE DISPOSITIVOS DE HUELLA DACTILAR – ANÁLISIS DE RENDIMIENTO DE DIFERENTES TECNOLOGÍAS,” 2015.
- [6] D. S. Zorita, “Reconocimiento automático mediante patrones biométricos de huella dactilar,” 2003.
- [7] M. K. y. A. Tojo, *Fingerprint Pattern Classification*, 1984.
- [8] A. L. Castro, “Tratamiento digital de huellas dactilares.”
- [9] J. L. García, “Imagen extraída de: Tesis ” Algoritmo para la identificación de personas basado en huellas dactilares” .”
- [10] “MathWorks,” [En línea]. Disponible en: <https://la.mathworks.com/discovery/deep-learning.html>. [Último acceso: Abril 2025].

- [11] Z. F. Victor, “Algoritmo de Clasificación de Huellas Dactilares Basado en Redes Neuronales Función Base Radial,” *REVISTA PGI - INVESTIGACIÓN, CIENCIA Y TECNOLOGÍA*, pp. 103-108.
- [12] L. G. Juan, “Algoritmo para la identificación de personas basado en huellas dactilares.”
- [13] S. J. J. Kandel E. R., *Principles of Neural Science*, New York: McGraw-Hill, 2000.
- [14] “Imagen obtenida de: CONCEPTOS BÁSICOS SOBRE REDES NEURONALES,” [En línea]. Disponible en: <http://grupo.us.es/gtocoma/pid/pid10/RedesNeuronales.htm#redesbiologicas> [Último acceso: Abril 2025].
- [15] D. J. Match, “Redes Neuronales: Conceptos Básicos y Aplicaciones,” 2001.
- [16] G. d. T. C. y. M. Aplicada, “Conceptos Básicos sobre Redes Neuronales,” [En línea]. Disponible en: <http://grupo.us.es/gtocoma/pid/pid10/RedesNeuronales.htm#estructuraANS>. [Último acceso: Abril 2025].
- [17] P. P. Elia, “DISEÑO DE UNA METODOLOGÍA PARA EL PROCESAMIENTO DE IMÁGENES MAMOGRÁFICAS BASADA EN TÉCNICAS DE APRENDIZAJE PROFUNDO,” 2017.
- [18] U. o. Standford, “Convolutional Neural Networks for Visual Recognition,” [En línea]. Disponible en: <https://cs231n.github.io/convolutional-networks/>. [Último acceso: Abril 2025].
- [19] “Imagen obtenida de: <https://cs231n.github.io/convolutional-networks/> [Último acceso: Abril 2025].”
- [20] “Imagen extraída de: <https://la.mathworks.com/discovery/deep-learning.html> [Último acceso: Abril 2025].”
- [21] “Imagen recuperada de: <https://republicacientifica.files.wordpress.com/2012/07/huellas-digitales.jpg> [Último acceso: Abril 2025].”

- [22] “Imagen recuperada de: http://www.biometricos.cl/equipos_biometria/images/fotos_docs/image4377.gif [Último acceso: Abril 2025].”
- [23] V. Ramkumar, “Fast Enhancement Algorithm with Improved Clarity of Ridge and Valley Structures for Fingerprint Images,” 2011.
- [24] J. R. M. Antonio, “Imagen obtenida de: ”Diseño de un módulo de propiedad intelectual basado en FPGA para reconocimiento de huellas digitales”,” 2010.