UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR
DE INGENIEROS DE TELECOMUNICACIÓN

GRADO EN INGENIERÍA DE TECNOLOGÍAS
Y SERVICIOS DE TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

DESARROLLO DE UN MODELO DE RECONOCIMIENTO DE EXPRESIONES FACIALES MEDIANTE REDES NEURONALES CONVOLUCIONALES Y APRENDIZAJE TRANSFERIDO

VADYM IVANCHUK

2018

TRABAJO FIN DE GRADO

Título .....
Desarrollo de un Modelo de Reconocimiento de Expresiones Faciales mediante Redes Neuronales Convolucionales y Aprendizaje Transferido

Autor .....
Vadym Ivanchuk

Tutor .....
Javier Rojo Lacal

Cotutor .....
Alejandro Medrano Gil

Ponente .....
María Teresa Arredondo Waldmeyer

Departamento .....
Tecnología Fotónica y Bioingeniería

TRIBUNAL

Presidente .....

Vocal .....

Secretario .....

Suplente .....

FECHA DE LECTURA |

CALIFICACIÓN |

UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR
DE INGENIEROS DE TELECOMUNICACIÓN

GRADO EN INGENIERÍA DE TECNOLOGÍAS
Y SERVICIOS DE TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

DESARROLLO DE UN MODELO DE RECONOCIMIENTO DE EXPRESIONES FACIALES MEDIANTE REDES NEURONALES CONVOLUCIONALES Y APRENDIZAJE TRANSFERIDO

VADYM IVANCHUK

2018

Resumen

El reconocimiento de expresiones faciales es un campo de estudio muy activo actualmente en las áreas de visión e inteligencia artificial, abarcando ámbitos tan diversos como los académicos, los clínicos o los comerciales. Sin embargo, este reconocimiento de emociones no es un problema sencillo para las computadoras, y en algunas ocasiones tampoco para los humanos, ya que cada individuo puede manifestar su estado afectivo de una manera distinta. En este contexto, proporcionarles a las máquinas la capacidad de identificar la condición anímica de una persona puede favorecer significativamente su desempeño en una gran variedad de tareas, incluso en algunas muy complejas que requieren de una elevada inteligencia emocional.

Con el fin de abordar estos problemas y aunque el reconocimiento de emociones puede realizarse utilizando múltiples sensores, este proyecto se centra exclusivamente en el estudio de las imágenes faciales al ser éstas uno de los principales canales de información en una comunicación interpersonal. De esta manera, este documento proporciona una breve reseña de las investigaciones en el campo de la visión e inteligencia artificial y propone un sistema de reconocimiento de expresiones faciales basado en redes neuronales convolucionales y en la técnica de transferencia de aprendizaje. A raíz de ello, son exploradas diversas arquitecturas convolucionales ampliamente extendidas (Inception-v3, Inception-ResNet-v2 y ResNet-50), así como varias bases de datos de diferente índole (ImageNet, VGGFace2 y FER-2013). Asimismo, también son llevados a cabo un análisis y una implementación de algunos métodos de aumento de datos, tanto desde el punto de vista del preprocesamiento de imágenes como desde un enfoque de generación de representaciones artificiales mediante redes generativas antagónicas. Por último, dada la complejidad del problema planteado y por consiguiente del sistema desarrollado para resolverlo, son aprovechados los recursos computacionales que proporciona la plataforma Google Cloud para disminuir el coste temporal del entrenamiento de los modelos desarrollados.

En definitiva, el planteamiento propuesto en este escrito ha demostrado ser muy efectivo, mejorando una gran parte de los resultados reportados hasta la fecha sobre el conjunto FER-2013 y además, con una inversión computacional y temporal mínima. Adicionalmente, este enfoque también ha permitido implementar el modelo del reconocimiento de expresiones faciales desarrollado en un sistema empotrado, abriéndose un amplio abanico de servicios que podrían ofrecerse en pseudo tiempo real.

Palabras Clave

Inteligencia Artificial, Visión por Computador, Aprendizaje Profundo, Reconocimiento de Expresiones Faciales, Redes Neuronales Convolucionales, Transferencia de Aprendizaje, Redes Generativas Antagónicas de Ciclo Consecuente, Google Cloud, FER-2013.

Summary

Facial expression recognition is currently a very active topic in the fields of computer vision and artificial intelligence, being exploited in areas as diverse as academic, clinical or commercial. However, the emotion recognition is not an easy problem for computers, and sometimes not for humans as well, since people can vary significantly in the way that they show their expressions. In this context, providing machines with the ability to identify mood can significantly improve their performance in a wide variety of tasks, even in some very complex that require high emotional intelligence.

To address these problems and even though the emotion recognition can be conducted using multiple sensors, this project focuses exclusively on the study of facial images as they are one of the main information channels in interpersonal communication. In this way, this document provides a brief review of researches in the field of computer vision and artificial intelligence and proposes a facial expression recognition system that uses convolutional neural networks and the transfer learning technique. As a result, various widely extended convolutional architectures are explored (Inception-v3, Inception-ResNet-v2 and ResNet-50), as well as several databases of different nature (ImageNet, VGGFace2 and FER-2013). Likewise, an analysis and implementation of some data augmentation methods are also carried out, both from the point of view of the image preprocessing and from an artificial data generation approach through generative adversarial networks. Besides, due to the complexity of the models implemented, the computational resources offered by the Google Cloud platform are used in order to reduce the training time.

Ultimately, the proposed approach has shown to be very effective, improving a large part of the results reported to date on the database FER-2013 with a minimal computational and temporal investment. Additionally, this approach has also favored the implementation of the facial expression model in an embedded system, enabling a wide range of services that could be offered in pseudo real time.

Keywords

Artificial Intelligence, Computer Vision, Deep Learning, Facial Expression Recognition, Convolutional Neural Networks, Transfer Learning, Cycle-Consistent Adversarial Networks, Google Cloud, FER-2013.

Abreviaturas

AdaGrad	Gradiente Adaptativo
ANN	Red Neuronal Artificial
API	Interfaz de Programación de Aplicaciones
Adam	Estimación del Momento Adaptativo
CNN	Red Neuronal Convolucional
CPU	Unidad de Procesamiento Central
CycleGAN	Red Generativa Antagónica de Ciclo Consecuente
FACS	Sistema de Codificación de Acción Facial
GAN	Red Generativa Antagónica
GPU	Unidad de Procesamiento Gráfico
ILSVRC	Desafío del Reconocimiento Visual a Gran Escala
MLP	Perceptrón Multicapa
ZCA	Análisis de Componentes de Fase Cero
ReLU	Unidad Lineal Rectificada
RGB	Rojo, Verde y Azul
RMSProp	Propagación del Valor Cuadrático Medio
SGD	Descenso de Gradiente Estocástico
SVM	Máquinas de Vectores de Soporte
TPU	Unidad de Procesamiento Tensorial
XML	Lenguaje de Marcas Extensible

Expresiones Matemáticas

$\mu_x = \overline{x}$	Promedio de x
$\hat{x}$	Valor normalizado de x
σ_x^2	Varianza de x
$\nabla f = f'$	Gradiente de f
$\Delta f = \nabla^2 f$	Laplaciano de f
$\|u - v\|$	Distancia entre los vectores u y v
$max(0, x)$	Función rampa o rectificador

Índice

Lista de figuras

Lista de tablas

Capítulo 1

Introducción

1.1. Motivación

Las emociones son especialmente importantes en la inteligencia humana, en la toma de decisiones racionales, en la interacción social, en la percepción, en la memoria, en el aprendizaje y en la creatividad. Además, son indispensables para un desarrollo y una gestión inteligente de las situaciones diarias. Por todo ello, la habilidad de reconocerlas de forma precisa y automatizada es una fuente extremadamente valiosa de información, tanto en el ámbito tecnológico, como social y económico. De hecho, esta capacidad de obtener un estado anímico a partir de la observación de una expresión emocional y a través de un razonamiento sobre la situación afectiva que se está dando, es uno de los grandes retos a los que se enfrenta la inteligencia artificial actualmente ya que, a pesar de que las técnicas de reconocimiento han alcanzado una madurez casi suficiente, la comprensión semántica sigue siendo un inmenso reto [4].

Esta detección de las emociones es fundamentalmente necesaria para la obtención de un mejor servicio por parte de las máquinas [52], que dotadas de cierta inteligencia afectiva en este proceso de reconocimiento, se beneficiarán de una toma de decisiones más flexible y racional, de la habilidad de determinar prominencias en el comportamiento humano, dando lugar a una atención y a una percepción más naturales, de la capacidad de abordar múltiples asuntos de una manera inteligente y eficiente, y de otras muchas más interacciones con los procesos cognitivos. Todo ello es realmente útil, por ejemplo, en las áreas en las que son utilizados dispositivos inteligentes para el cuidado de los grupos de la tercera edad, para los tratamientos de inserción social de individuos con autismo, para la rehabilitación de personas con parálisis facial o para atender a los distintos pacientes de un hospital, tareas que exigen una comprensión y un análisis profundo del entorno.

Es evidente, por lo tanto, que para conseguir una respuesta lo más natural y acertada posible por parte de una computadora ante un estímulo, es necesario que ésta logre imitar los procedimientos mediante los cuales el cerebro humano procesa la información sensorial y los razonamientos. En este sentido, por medio de la toma de una serie de datos del entorno y su posterior procesamiento mediante los algoritmos adecuados que imitan el funcionamiento de una red neuronal biológica, es posible, a día de hoy, generar una percepción artificial que iguala o excede, incluso, las capacidades humanas en algunos ámbitos [69]. Estos avances, de hecho, también han sido en gran parte gracias a las novedosas técnicas de aprendizaje profundo y automático que han sido desarrolladas en los últimos años.

En este contexto, y dado que las expresiones faciales de un individuo reflejan su estado interno, es deducible que si un ordenador es capaz de capturar una secuencia de imágenes faciales, entonces el uso de técnicas de aprendizaje profundo nos permitiría conocer el estado de ánimo de su interlocutor.

En consecuencia, es posible afirmar que estos procedimientos y métodos tienen el po-

tencial de convertirse en un factor clave en el avance de la inteligencia artificial, y por lo tanto, en el progreso y mejora de las habilidades de los ordenadores en su labor de comprensión, interacción y ofrecimiento de soluciones y servicios a los seres humanos.

1.2. Objetivos

El principal objetivo del proyecto de fin de titulación propuesto es el de desarrollar un modelo que sea capaz de reconocer, lo más fielmente posible y acercándose al estado del arte, las emociones básicas y universales (ira, asco, miedo, alegría, tristeza y sorpresa) [16], así como la ausencia de éstas (neutral), a partir de imágenes faciales estáticas y etiquetadas conforme a la expresión facial escenificada.

Por otro lado, y exclusivamente como aplicación directa del modelo desarrollado, se pretende implementar un sistema que sea capaz de reconocer las emociones de un individuo en tiempo real y que pueda ser integrado en un sistema empotrado como parte de un prototipo de espejo inteligente.

Finalmente, dado que éste es un proyecto multidisciplinar que involucra distintos ámbitos como la computación afectiva, el aprendizaje automático y profundo y la visión por computador, otro de los objetivos es el de aprender la forma en la que se relacionan estos campos y cómo su confluencia puede proporcionar soluciones a problemas complejos.

En resumen, la finalidad de este trabajo se compone de dos objetivos claramente diferenciados:

- **Objetivo Principal.** Desarrollo de un sistema de reconocimiento de expresiones faciales altamente competitivo y eficiente en lo que respecta al uso de recursos para su implementación.
- **Objetivo Secundario.** Integración de este modelo en un sistema empotrado capaz de ofrecer servicios enfocados a la monitorización de las emociones, al seguimiento de los tratamientos clínicos o a la realización de ejercicios interactivos para la rehabilitación o la mejora de la calidad de vida de determinados pacientes.

1.3. Contribuciones

Mediante el presente escrito se pretende ofrecer un nuevo enfoque dentro del ámbito del reconocimiento de las expresiones faciales, logrando unas tasas de precisión significativas con unos recursos limitados y combinando métodos estándar, como son la utilización de las redes neuronales, la transferencia de aprendizaje, el aprendizaje profundo y la visión por computador.

En este sentido, las principales contribuciones de este trabajo pueden englobarse en los siguientes puntos:

- Estudio, desarrollo y adaptación a la tarea de identificación de emociones de una serie de modelos pre-entrenados, y que a día de hoy representan el estado del arte del reconocimiento facial y de imágenes.
- Implementación y entrenamiento de estos modelos en la plataforma Google Cloud.
- Desarrollo e implementación de un tipo de red generativa antagónica, cuya capacidad de generación de imágenes de forma artificial se pretende aprovechar para extender la base de datos inicial y mejorar, de esta forma, los resultados obtenidos anteriormente.
- Desarrollo y despliegue de un sistema de reconocimiento de expresiones faciales en tiempo real como parte de un prototipo de espejo inteligente de la *Smart House Living Lab* de la E.T.S.I. de Telecomunicación.

Capítulo 2

Antecedentes

2.1. Inteligencia Emocional Artificial

El término de inteligencia emocional, popularizado gracias a la publicación de Daniel Goleman en 1995 sobre cómo los efectos de esta característica afectiva pueden influir notablemente en las decisiones cotidianas y puesto al mismo nivel que el coeficiente intelectual, se define como la capacidad de identificar, evaluar y controlar ciertas características emocionales propias y ajenas, tales como el estado de ánimo, la motivación, la frustración, la angustia, los impulsos no racionales o la empatía [21].

A pesar de ello, hasta la fecha, los investigadores que están intentando desarrollar ordenadores inteligentes se han centrado principalmente en la resolución de problemas, en el razonamiento, en el aprendizaje, en la percepción, en el lenguaje y en otras tantas tareas cognitivas de interés para este campo. De hecho, la mayoría de ellos no han tenido en cuenta la importante influencia de las emociones en estas funciones, y más cuando hay ciertas evidencias de que éstas juegan un papel fundamental en las atribuciones consideradas esenciales para la inteligencia en los humanos [18]. Por ello, este nuevo enfoque indica una clara necesidad de reconsideración del rol de las emociones en los ordenadores.

En este contexto, y en el intento de las computadoras de identificar el estado afectivo de un individuo, el sistema de reconocimiento ideal debería reunir las capacidades visuales y auditivas para capturar las expresiones faciales, los gestos y la entonación vocal con el fin de ser lo más riguroso posible. De forma adicional, otros aspectos que podrían enriquecer el resultado de la decisión final son la lectura de la temperatura corporal a través de la radiación infrarroja que emiten los cuerpos humanos, la medición de la actividad electrodérmica o la conductancia de la piel y el pulso [66].

Por lo tanto, es muy importante para la inteligencia afectiva artificial desarrollar formas de medir y aunar adecuadamente estas modulaciones, ya que pueden conducir a una mejor comprensión del estado emocional del sujeto. A pesar de ello, en este proyecto tan sólo se profundizará en el reconocimiento de las expresiones faciales, que individualmente se constituye como el módulo de mayor entropía desde el punto de vista de la identificación emocional.

2.1.1. Reconocimiento de Expresiones Faciales

Paul Ekman, la principal autoridad de las expresiones faciales en el ámbito de la psicología, ha argumentado la existencia de seis emociones básicas (ira, asco, miedo, alegría, tristeza y sorpresa) [16], mostradas en la Figura 2-1. Cada una de ellas tiene un conjunto de patrones únicos en el movimiento muscular facial. Estos patrones, así como su estudio detallado e identificación, posteriormente han sido integrados en un sistema de codificación facial (FACS) [17], el cual proporciona las técnicas necesarias para asociar los músculos

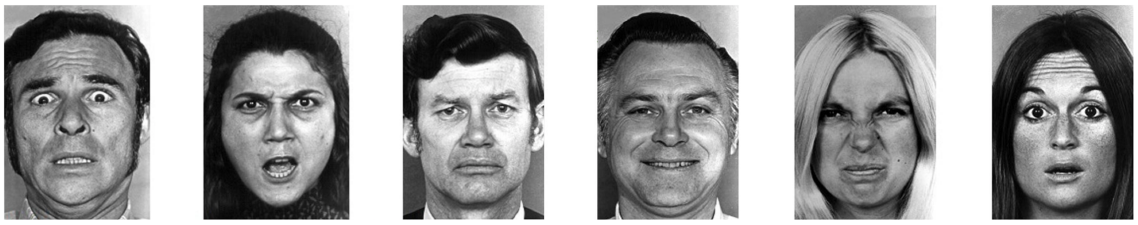


Figura 2-1: Emociones universales propuestas por Paul Ekman (de izquierda a derecha: miedo, ira, tristeza, alegría, asco y sorpresa) [16].

del rostro a su espacio emocional correspondiente. De hecho, gran parte de los intentos de automatizar el reconocimiento de las expresiones faciales, capaces de ofrecer una identificación automática y en tiempo real a partir de un análisis tridimensional de la cara del sujeto, se basan en este sistema [48].

Por otro lado, en los últimos años, y gracias principalmente a la irrupción y al enorme progreso que ha experimentado el campo del aprendizaje profundo y automático en el ámbito del reconocimiento visual [40], se han comenzado a desarrollar nuevas técnicas y métodos de identificación de las expresiones faciales. Este hecho, además, ha supuesto un nuevo punto de inflexión en este entorno dados los espectaculares resultados obtenidos gracias a la implementación de estos nuevos modelos, cuya eficacia y rendimiento se van a intentar analizar y reproducir en este proyecto.

Por último, cabe destacar que actualmente ninguno de los modelos anteriormente descritos pretende reconocer las emociones subyacentes (una sonrisa forzada, por ejemplo), centrándose la identificación tan solo en la expresión del rostro del sujeto en un instante determinado.

2.2. Aprendizaje Automático

El aprendizaje automático es una rama de la inteligencia artificial y puede definirse como un área de estudio que proporciona a las computadoras la capacidad de aprender, sin éstas estar explícitamente programadas [57]. De esta forma, mediante los variados algoritmos de aprendizaje automático es posible generar un resultado, generalmente un conjunto de predicciones o decisiones, a partir de diversos datos de entrada sin la necesidad de agregar nuevas líneas de código al sistema inicial.

En lo que respecta al proceso de aprendizaje del modelo en sí, es posible distinguir tres categorías por las que se rige este procedimiento y cuya clasificación está basada en el tipo o en la existencia de retroalimentación durante el entrenamiento:

- **Aprendizaje supervisado**. Se caracteriza por la recepción de un conjunto de entradas etiquetadas, las cuales son atribuidas a la clase de pertenencia correspondiente gracias al algoritmo de aprendizaje pertinente. En este proceso de asignación el modelo va adaptándose o modificando sus parámetros con el fin de ofrecer un mejor rendimiento.

- **Aprendizaje no supervisado**. En este caso se recibe un conjunto de entradas sin etiquetar, por lo que el modelo intenta aprender de estos datos de entrada mediante la exploración de los patrones existentes en ellos.

- **Aprendizaje por refuerzo**. Se distingue por implementar un sistema de recompensa o castigo en función de las decisiones que un agente de *software* ha tomado en un entorno dado para alcanzar un objetivo concreto.

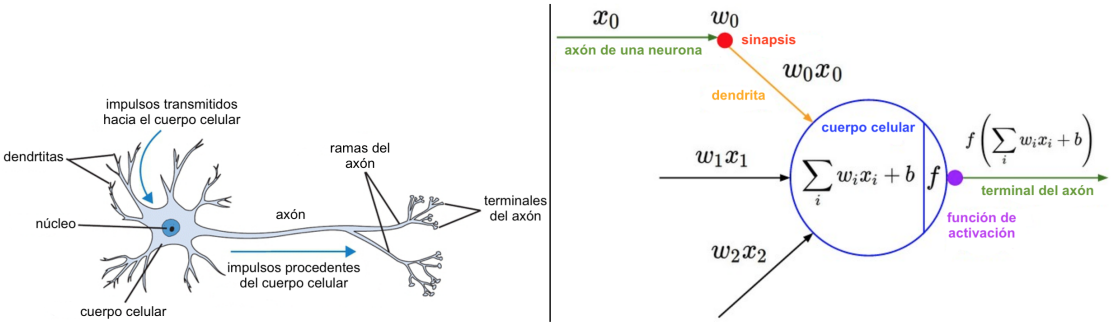


Figura 2-2: Imagen de una neurona biológica (izquierda) y el modelo matemático que intenta reproducir superficialmente su funcionamiento (derecha) [32].

Particularmente en este proyecto y tal y como se podrá comprobar durante los siguientes capítulos, el planteamiento del problema del reconocimiento de emociones desde el punto de vista de las herramientas y de los conjuntos de datos empleados en este trabajo implica un acercamiento hacia la utilización de técnicas de aprendizaje supervisado.

2.2.1. Redes Neuronales Artificiales

El aprendizaje supervisado tiene un conjunto de herramientas enfocadas a resolver problemas dentro de su dominio, siendo una de ellas, precisamente, las Redes Neuronales Artificiales (ANN).

Estas redes son un modelo computacional vagamente inspirado en las redes neuronales biológicas que constituyen los cerebros humanos [70] y cuya analogía con el modelo matemático se puede observar en la Figura 2-2. Cada una de estas neuronas recibe una serie de señales de entrada en sus dendritas y produce determinadas señales de salida que son trasladadas a lo largo del axón, el cual posteriormente se bifurca y se conecta a través de la sinapsis a las dendritas de otras neuronas. En el modelo computacional, por su parte, las señales que se desplazan a lo largo de los axones (x_i) interactúan de forma multiplicativa ($\omega_i \cdot x_i$) con las dendritas de otras neuronas en función de las fuerzas sinápticas (los pesos ω_i), que son precisamente los elementos aprehensibles de los algoritmos de aprendizaje automático. Estos pesos, por lo tanto, son los que controlan la intensidad y la dirección (sinapsis excitatoria –peso positivo– o inhibitoria –peso negativo–) de influencia de una neurona sobre otra. En el modelo biológico, las dendritas trasladan estas señales provenientes de otras neuronas al cuerpo celular, donde su suma da lugar a un potencial excitatorio postsináptico, que en caso de superar un determinado umbral causará la generación de un impulso eléctrico o potencial de acción que será enviado a largo del axón de la propia neurona. Este proceso en el modelo computacional es menos complejo al considerar que la información se encuentra tan sólo en la frecuencia de disparo del potencial de acción. En base a esta interpretación, la descarga de una neurona artificial se modela con una función de activación no lineal que representa la frecuencia de los impulsos a lo largo del axón, es decir, se toma como entrada la intensidad de la señal después de la suma ($\sum_{i=0}^{n} \omega_i \cdot x_i + b$) y se normaliza, típicamente, según el tipo de función de activación, obteniéndose una salida acotada. Cabe destacar que el parámetro b, ignorado en los próximos capítulos por razones de simplificación, es necesario para evitar interrupciones en el proceso de aprendizaje cuando las entradas inyectadas (x_i) son nulas.

Una ANN, por lo tanto, es una red estratificada de estas neuronas artificiales, pudiendo consistir en una capa de entrada, capas ocultas y una capa de salida [19]. Las estructuras más básica de una ANN, el perceptrón y el perceptrón multicapa (MLP), que permite resolver problemas que no son linealmente separables, se representan en la Figura 2-3.

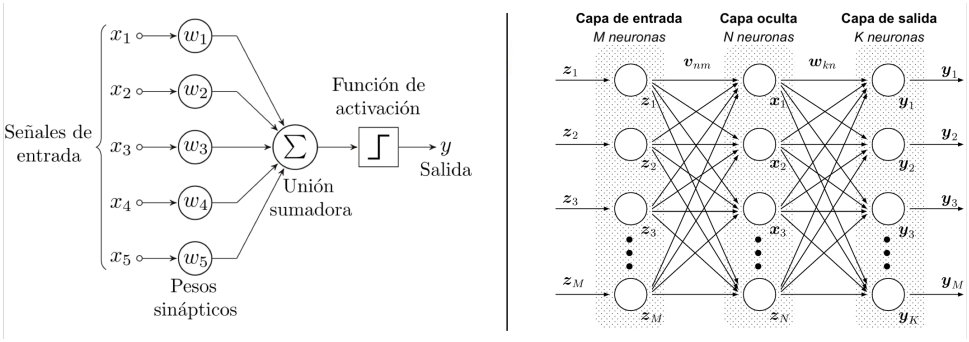


Figura 2-3: Topología de un perceptrón simple (izquierda) [73] y de un MLP con una capa oculta (derecha) [29].

En definitiva, una ANN es un conjunto de funciones que realizan una predicción que es expresada, generalmente, como una distribución de probabilidad para todas las etiquetas. Esta predicción, posteriormente, es cuantificada con respecto al valor real correspondiente, obteniéndose una función de error o de pérdidas, la cual es utilizada para actualizar los pesos de las distintas capas con el fin de mejorar la predicción global. Esto se lleva a cabo, tradicionalmente, mediante el método de propagación hacia atrás de errores o retropropagación y cuyas características se analizarán en profundidad en la Subsección 2.4.1.

En cuanto al contexto histórico, las unidades neuronales fueron reconocidas por primera vez como elementos funcionales del sistema nervioso a finales del siglo XIX a través del trabajo de Santiago Ramón y Cajal. Mediante su estudio, gracias al cual obtuvo el Premio Nobel, propuso que estas células discretas interconectadas establecían una red compleja para la transmisión de la información. Esta descripción se conoce como la doctrina de la neurona y es el modelo aceptado a día de hoy en neurofisiología.

Posteriormente, estas redes, pese a formularse como un modelo para la realización de cálculos matemáticos en la década de los cuarenta [43], no fueron implementadas como tales hasta finales de los cincuenta, cuando fue presentada la arquitectura más simple de las ANN: el perceptrón [54]. Este estaba formado por una serie de capas que contaban con un número determinado de nodos, cada uno de los cuales, excepto los de la entrada, representaban justamente a una neurona biológica. Además, fue en este artículo en el que se introdujo precisamente la técnica de aprendizaje supervisado denominada retropropagación para entrenar las redes neuronales artificiales.

A pesar de este gran paso, las limitaciones computacionales y de *hardware* de la época (también existentes hoy en día), así como un enfoque de la investigación en inteligencia artificial hacia las representaciones simbólicas de los problemas, hizo que no fuera hasta principios de la década de los noventa cuando estas redes neuronales artificiales volvieran a cobrar protagonismo. Los responsables de ello fueron diversas publicaciones de una serie de modelos que explotaban las ANN, como la NETtalk [58], que mediante el aprendizaje automático fue capaz de aprender a leer en voz alta, o la ANN de múltiples capas desarrollada por Yann LeCun [38], que permitía el reconocimiento de dígitos escritos a mano tomados de los códigos postales. Esta última publicación, de hecho, fue la pionera en integrar el reconocimiento de imágenes y el aprendizaje automático en un mismo modelo, además de introducir una serie de conceptos que posteriormente serían clave en el desarrollo de las Redes Neuronales Convolucionales (CNN), tales como la asignación de objetos según sus características a una serie de clases o la compartición de los pesos.

Cabe destacar que durante esta década muchos investigadores preferían utilizar las Máquinas de Vectores de Soporte (SVM) en las tareas de aprendizaje supervisado, cuya simplicidad y resultados eclipsaban a las ANN. Sin embargo, esta situación cambiaría drásticamente a partir del año 2010, cuando el desarrollo de potentes y asequibles herra-

mientas de *software* y de *hardware*, junto con la existencia de una enorme cantidad de datos a explotar, posibilitaron la realización de cálculos extremadamente grandes sobre modelos complejos, obteniéndose unos resultados sin precedentes y que marcarían el auge del aprendizaje profundo.

2.3. Aprendizaje Profundo

El último resurgimiento de las ANN se conoce como aprendizaje profundo, término cuya definición puede englobarse en los siguientes puntos estrechamente relacionados [13]:

- Es una clase de técnicas de aprendizaje automático que permite la explotación de un gran número de capas que procesan la información de forma no lineal para la extracción, transformación, análisis y clasificación de patrones.
- Es un subcampo dentro del aprendizaje automático basado en algoritmos cuya función es aprender los múltiples niveles de representación de la información de la entrada con el fin de poder modelar relaciones complejas entre los datos en una arquitectura profunda, definiéndose los conceptos del nivel superior a partir de los del nivel inferior.
- Es una nueva área de investigación del aprendizaje automático introducida con el objetivo de acercar este subcampo de las ciencias de la computación a uno de sus objetivos originales: la inteligencia artificial.

Estas técnicas y métodos que permiten aprender determinadas características por sí mismos a partir de representaciones sencillas han tenido un especial éxito en campos como la visión por computador, el procesamiento del lenguaje natural y el reconocimiento automático de voz. En el área del reconocimiento visual artificial, por ejemplo, una ANN puede alimentarse con los píxeles de una imagen, determinando posteriormente el algoritmo de aprendizaje si esta específica combinación representa cualquier característica particular que es repetida a través de una o varias imágenes.

Por otro lado, el hecho de que hoy en día ya no se emplee el término de ANN, sino el de aprendizaje profundo, se debe principalmente al impulso que recibió el campo de la inteligencia artificial gracias al trabajo de Geoffrey Hinton. Fue precisamente su publicación de 2012 sobre el reconocimiento automático de voz [25], con la colaboración de los grupos de investigación de la Universidad de Toronto, Micrososft Research, Google Research e IBM Research, la primera que contenía una implementación directa en el ámbito industrial de las técnicas de aprendizaje profundo.

En lo que respecta al ámbito de la visión por computador, el punto crítico también llegaría de la mano de Hinton dentro del contexto del Desafío del Reconocimiento Visual a Gran Escala (ILSVRC) [56]. Esta competición consiste en evaluar el desempeño de los modelos de los participantes en la clasificación de 100 000 fotografías etiquetadas a mano con la presencia o ausencia de 1 000 categorías de objetos diferentes. Su creación en 2010, así como de la base de datos ImageNet utilizada para el entrenamiento, tenían como objetivo motivar la investigación de *software* enfocado al campo del reconocimiento visual. De hecho, el avance experimentado tan sólo dos años después de la aparición del ILSVRC es considerado ampliamente como el comienzo de la revolución del aprendizaje profundo, tanto en el entorno de investigación de la inteligencia artificial como en el de la industria tecnológica [15]. Este progreso estuvo marcado por la presentación, por parte de Alex Krizhevsky, Ilya Sutskever y Geoffrey Hinton, de un modelo capaz de reducir a la mitad la tasa de error existente en ese momento [37]. El sistema desarrollado combinaba varios elementos críticos que se convertirían en los pilares fundamentales de los modelos de aprendizaje profundo: el entrenamiento mediante las Unidades de Procesamiento Gráfico (GPUs), las

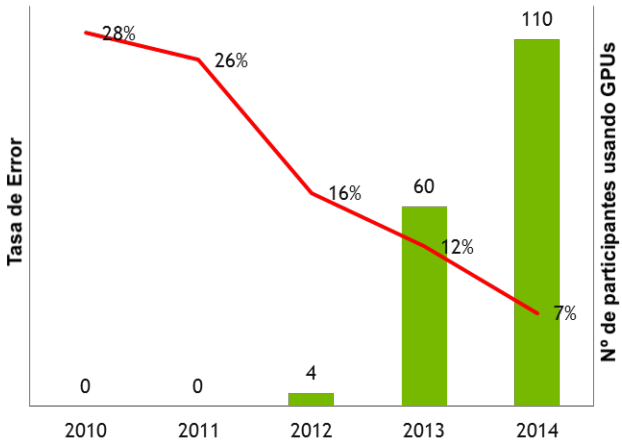


Figura 2-4: Tasa de error de la entrada ganadora del ILSVRC (línea roja) y número creciente de entradas que usan GPUs cada año (barras verdes).[6]

Redes Neuronales Convolucionales (CNN), el empleo de la Unidad Lineal Rectificada (ReLU) y del método para evitar el sobreentrenamiento, conocido como *dropout*.

2.3.1. Empleo de las Unidades de Procesamiento Gráfico

Probablemente el componente más relevante e influyente dentro del explosivo progreso del aprendizaje profundo haya sido el uso de las Unidades de Procesamiento Gráfico en el entrenamiento de los modelos diseñados.

Las GPUs son esencialmente calculadoras de coma flotante cuya arquitectura paralela de miles de núcleos ofrece la posibilidad de manejar múltiples tareas simultáneamente. Esta capacidad de paralelización, precisamente, es la que ha sido ampliamente explotada por los algoritmos de aprendizaje profundo en la realización de operaciones vectoriales y matriciales, logrando una reducción de los tiempos de entrenamiento en unas $10 - 20$ veces [40]. Esto ha posibilitado, a su vez, el entrenamiento de modelos significativamente más complejos o profundos y por lo tanto, la obtención de tasas de error cada vez más bajas.

El desarrollo experimentado y su influencia pueden apreciarse en la Figura 2-4, que muestra como los resultados del ILSVRC van mejorando a medida que se populariza el uso de las GPUs. Destaca también la caída precipitada del año 2012 propiciada por la implementación de la red AlexNet, modelo que se ha descrito brevemente en el apartado anterior y que ha revolucionado el reconocimiento e identificación visual, siendo especialmente relevante en este proyecto.

2.4. Redes Neuronales Convolucionales

Las investigaciones sobre la corteza visual animal, cuyos primeros estudios trascendentales datan de 1968 con la publicación de Hubel y Wiesel sobre los campos receptivos y la arquitectura funcional del córtex visual de los monos [26], están estrechamente relacionadas con el desarrollo de las redes neuronales convolucionales. Fue precisamente en este estudio en el que se describió por primera vez cómo las señales obtenidas por los ojos son procesadas por parcelas visuales en el neocórtex para generar detectores de bordes, de movimientos, de profundidad y de color, construyendo bloques de la escena visual.

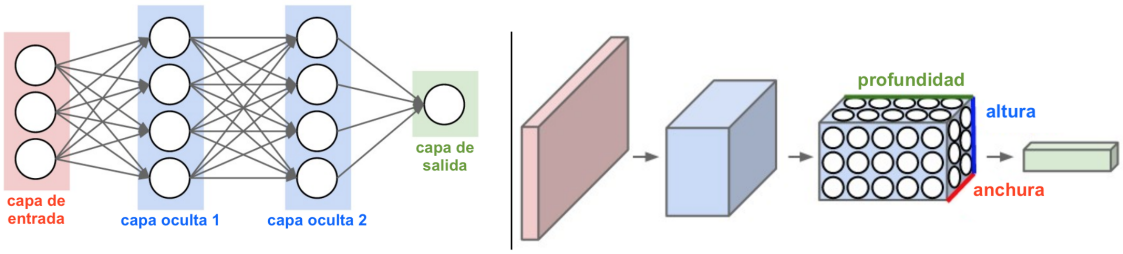


Figura 2-5: Una ANN ordinaria de 3 capas (izquierda) y una CNN (derecha) [32].

Una de las primeras implementaciones inspiradas en este estudio fue el Neocognitron [20]. Este modelo de red neuronal intentaba imitar el funcionamiento de los dos tipos de células introducidas en el anterior documento: las células simples, que responden principalmente a bordes y a líneas de orientaciones particulares y las células complejas, que presentaban campos receptivos más grande e invariancia local. A pesar de todo ello, este modelo presentaba un importante inconveniente en el proceso de aprendizaje al no haberse desarrollado, por aquella época, un método para ajustar los valores de los pesos con respecto a una medida de error para toda la red, como la retropropagación. Esta técnica de propagación hacia atrás de errores, analizada en Subsección 2.4.1, no fue implementada en el entrenamiento de las ANN hasta 1985, cuando se publicó el trabajo de Rumelhart, Hinton y Williams [55].

A partir de este momento se llevaron a cabo numerosas implementaciones con aplicaciones prácticas empleando la propagación hacia atrás de errores para entrenar las CNN, siendo Yann LeCun con su clasificador de dígitos escritos a mano (MNIST) [38] el pionero e inspirando, además, gran parte de las investigaciones futuras.

En cuanto a las redes neuronales convolucionales en sí, éstas son muy similares a las redes neuronales ordinarias tratadas en la Subsección 2.2.1: están formadas por neuronas que tienen pesos que se pueden aprender, reciben entradas a las que se aplica un producto escalar y opcionalmente una función no lineal, y obtienen, en la última capa, la puntuación de pertenencia a una determinada clase. La principal diferencia, por lo tanto, es que las arquitecturas convolucionales suponen explícitamente que las entradas son imágenes, haciendo que las operaciones sean más eficientes y reduciendo inmensamente la cantidad de parámetros en la red [32]. Además, las capas de una CNN tienen neuronas dispuestas en las 3 dimensiones: anchura, altura y profundidad, volumen que coincide, precisamente, con las dimensiones en píxeles (ancho y alto) y el modelo de color (profundidad), tradicionalmente RGB, de una imagen. Asimismo, las neuronas de una determinada capa de una CNN sólo están conectadas a una pequeña región de la capa anterior, en lugar de a todas las neuronas de una manera absolutamente conectada, como ocurría en las ANN. Es por ello que al final de la arquitectura de una CNN la imagen completa de la entrada se reduce a tan solo un vector de puntuaciones de pertenencia a una determinada clase dispuesto a lo largo largo de la dimensión de profundidad, tal y como se puede observar en la Figura 2-5.

En definitiva, una CNN simple es una secuencia de capas que se encargan de transformar un volumen de activación en otro diferente a través de una función diferenciable. Generalmente, y particularmente en este proyecto, son utilizadas principalmente las capas descritas en Sección 4.1 (capas convolucionales, de agrupación o *pooling*, íntegramente conectadas, etc.), con cuyo apilamiento sucesivo se va a pretender formar una arquitectura neuronal convolucional completa.

Algunas de estas arquitecturas, especialmente las más relevantes en el contexto de este proyecto, se enumeran a continuación:

- **LeNet** [39]. Las primeras aplicaciones exitosas de las CNN fueron desarrolladas por Yann LeCun en la década de 1990. Entre ellas, la más conocida es la arquitectura

LeNet, que se utilizó para leer los dígitos de los códigos postales.

- **AlexNet** [37]. La publicación con la que se popularizó el uso de las CNN en el campo de la visión por computador fue la ya nombrada anteriormente AlexNet. Este modelo se presentó al desafío ILSVRC de ImageNet en 2012, superando al segundo finalista con una diferencia de más del 10 % en la tasa de error. En cuanto a su arquitectura, era muy similar a LeNet, con la diferencia de que presentaba capas convolucionales más profundas, más grandes y apiladas unas sobre otras, cuando lo común era tener capas convolucionales seguidas inmediatamente de capas de agrupación.

- **GoogLeNet** [64]. El ganador del ILSVRC 2014 fue una CNN desarrollada por el grupo de investigación de inteligencia artificial de Google. Su principal contribución fue la introducción de un modelo con el que se consiguió reducir drásticamente la cantidad de parámetros en la red, pasando a contar con tan sólo 4 millones, un número más que razonable teniendo en cuenta los 60 millones de la red AlexNet. También cabe destacar las posteriores versiones de GoogLeNet desarrolladas, como Inception-v3 e Inception-v4, que, como se verá en el Capítulo 5, son fundamentales en el presente proyecto.

- **VGGNet** [59]. Obtuvo el segundo puesto en el desafío ILSVRC 2014, aportando y desarrollando la idea de que la profundidad de la red es un componente crítico para la obtención de buenos resultados. Sin embargo, a pesar de haber alcanzado una tasa de error más que razonable, este modelo presentaba una importante desventaja al contar con casi 140 millones de parámetros, haciéndolo muy costoso de entrenar y requiriendo, además, mucha memoria.

- **ResNet** [24]. Esta red residual fue la vencedora del ILSVRC 2015 al implementar una serie de conceptos clave como las conexiones directas entre capas no contiguas y la normalización por lotes, lo que además de acelerar la evaluación del modelo, favorecía la prevención del sobreaprendizaje. Actualmente, de hecho, las ResNet o sus variaciones son la opción predeterminada para implementar las CNN en la práctica [63].

2.4.1. Retropropagación

El procedimiento de la retropropagación para calcular el gradiente de una función de pérdidas con respecto a los pesos, posteriormente actualizados, de una serie de módulos multicapa no es más que una aplicación práctica y recursiva de la regla de la cadena para derivadas [40]. La idea clave es que el gradiente de la función objetivo con respecto a la entrada se calcula en la salida y se distribuye hacia atrás a través de las distintas capas de la red, tal y como puede apreciarse en la Figura 2-6. De esta manera, la ecuación de la retropropagación puede aplicarse repetidamente para propagar los gradientes a través de todos los módulos, desde la parte superior, donde la red genera una predicción determinada, hasta la parte inferior, donde es alimentada la entrada con datos externos. Esta propagación, sin embargo, no es uniforme, sino que cada una de las neuronas de las capas ocultas tan sólo recibe una fracción de la señal total de error correspondiente a la contribución relativa que aporta a la salida original. Posteriormente, una vez que estos gradientes analíticos se calculan, son utilizados para realizar una actualización de los parámetros de la red mediante un optimizador. Hay numerosos enfoques para realizar esta actualización, discutidos en la Subsección 2.4.2.

El pseudocódigo completo de este proceso se explica detalladamente en el Algoritmo 1 del Apéndice C.

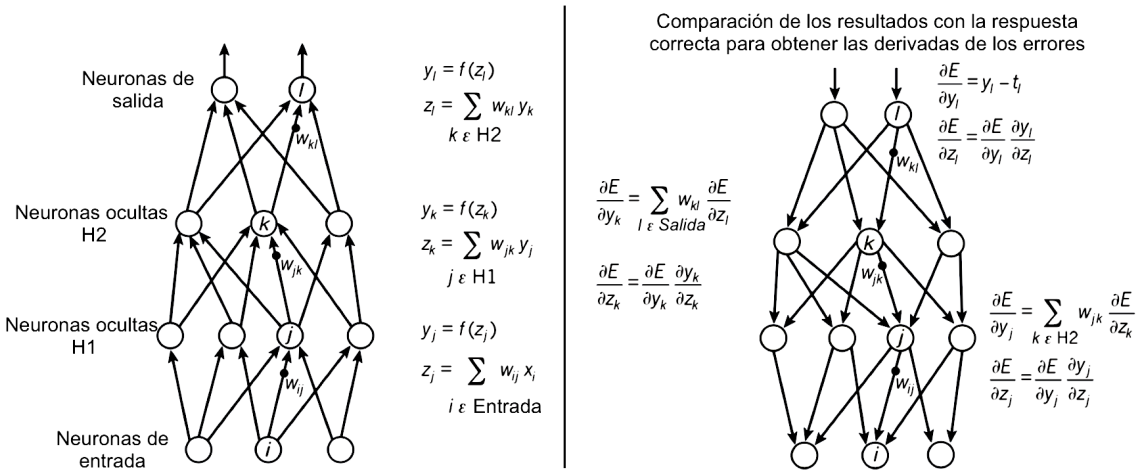


Figura 2-6: Ecuaciones utilizadas para calcular las salidas o predicciones de una red neuronal con dos capas ocultas y una capa de salida (izquierda) y las ecuaciones empleadas en la retropropagación si la función de pérdidas para cada una de las unidades neuronales es $\frac{1}{2} \cdot (y_l - t_l)^2$, donde t_l es el valor objetivo (derecha) [40].

2.4.2. Actualización de parámetros

Actualmente, los dos métodos más empleados y recomendados para realizar la actualización de parámetros, con el objetivo de encontrar una serie de pesos $\omega_{i,j}$ que minimicen la función de pérdidas, son el Descenso Estocástico del Gradiente (SGD) con Nesterov Momentum y la Estimación del Momento Adaptativo (Adam) [32].

Descenso Estocástico del Gradiente con Nesterov Momentum

La forma más simple y común de llevar a cabo la optimización es, precisamente, mediante SGD, que permite modificar los parámetros en la dirección negativa del gradiente (Ecuación 2.1), al apuntar éste en la dirección del máximo en cada uno de los puntos evaluados, con el fin de minimizar la función de pérdidas.

$$\omega_t = \omega_{t-1} - \lambda \cdot \nabla f(\omega_{t-1}) \tag{2.1}$$

donde $\omega_t \equiv$ peso en un instante determinado,

$\lambda > 0$ es la tasa de aprendizaje

Los métodos Momentum, por su parte, presentan un enfoque que ofrece mejores tasas de convergencia, ya que emplean el gradiente para actualizar los parámetros de una forma más efectiva al acumular velocidad en las direcciones que reducen continuamente la función de pérdidas, tal y como se muestra en la Ecuación 2.3.

$$v_t = \mu \cdot v_{t-1} - \lambda \cdot \nabla f(\omega_{t-1}) \tag{2.2}$$

$$\omega_t = \omega_{t-1} + v_t \tag{2.3}$$

donde $v \equiv$ vector de velocidad,

$\mu \in [0, 1]$ es el momento lineal

Nesterov Momentum, a su vez, también goza de mayores garantías teóricas de convergencia, funcionando ligeramente mejor en la práctica que el método Momentum estándar [61]. En la Figura 2-7 se puede observar la comparación, haciéndose evidente el aumento de la eficacia de este último procedimiento representado mediante la Ecuación 2.5, que da

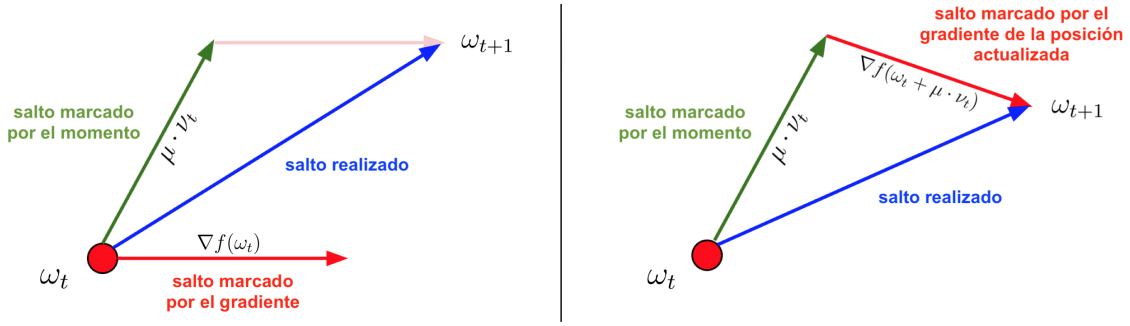


Figura 2-7: Actualización Momentum (izquierda) y actualización Nesterov Momentum (derecha) [32].

primero los saltos en la dirección del gradiente acumulado anterior para posteriormente calcular el gradiente en la posición actualizada y realizar una corrección.

$$v_t = \mu \cdot v_{t-1} - \lambda \cdot \nabla f(\overbrace{\omega_{t-1} + \mu \cdot v_{t-1}}^{w_{salto}}) \tag{2.4}$$

$$\overbrace{\omega_t}^{w_{corregido}} = \omega_{t-1} + v_t \tag{2.5}$$

Estimación del Momento Adaptativo

Al contrario que el método del descenso estocástico del gradiente, que permite manipular la tasa de aprendizaje de forma global e idéntica para toda la red, Adam adapta la tasa de aprendizaje a cada uno de los parámetros presentes en el modelo. En el artículo que describe este algoritmo, los autores exponen a Adam como la combinación de dos de las extensiones de SGD [35]:

- Algoritmo de **Gradiente Adaptativo (AdaGrad)**, el cual presenta una tasa de aprendizaje adaptativa por parámetro que mejora el rendimiento de sistemas con gradientes dispersos, como es el caso de los problemas de visión por computador.
- **Propagación del Valor Cuadrático Medio (RMSProp)**, que también explota las tasas de aprendizaje adaptativas, basadas en este caso en el promedio de las magnitudes recientes de los gradientes, lo que proporciona buenos resultados para problemas ruidosos.

De esta forma, Adam, además de almacenar un promedio del primer momento (media de medias), como en RMSProp, también utiliza el promedio de los segundos momentos de los gradientes (la varianza) para la actualización de los parámetros, proceso que se rige por la Ecuación 2.10.

$$m_t = \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot \nabla f(\omega_{t-1}) \tag{2.6}$$

$$v_t = \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot \nabla^2 f(\omega_{t-1}) \tag{2.7}$$

$$\hat{m}_t = m_t / (1 - \beta_1^t) \tag{2.8}$$

$$\hat{v}_t = v_t / (1 - \beta_2^t) \tag{2.9}$$

$$\omega_t = \omega_{t-1} - \lambda \cdot \hat{m}_t / (\sqrt{\hat{v}_t} + \epsilon) \tag{2.10}$$

donde $\lambda > 0$ es la tasa de aprendizaje,

m_t y $v_t \equiv$ estimaciones del primer y segundo momento,

$\hat{m}_t$ y $\hat{v}_t \equiv$ estimaciones del primer y segundo momento corregidas

De forma específica, este algoritmo englobado en la Ecuación 2.10 calcula en primer lugar un promedio exponencial del gradiente (m_t) y del gradiente cuadrado (v_t), cuyas tasas de descenso son controladas mediante los parámetros β_1 y β_2. Por otro lado, como m_t y v_t son inicialmente nulos, es necesario aplicarles una corrección para evitar su sesgado hacia cero, especialmente notable durante las primeras iteraciones. Tras ello, se realiza la actualización de los parámetros de manera análoga al método del descenso estocástico del gradiente.

2.5. Trabajos Relacionados

La tarea del reconocimiento de expresiones faciales puede abordarse desde múltiples puntos de vista, abarcando áreas como el procesamiento de señales, la visión por computador o el aprendizaje automático. Precisamente la combinación de estas técnicas es la que ofrece el entorno idóneo para el desarrollo e implementación de los sistemas de identificación de emociones.

En estas circunstancias, es destacable la labor del grupo de investigación comercial Affectiva, que es el líder mundial en reconocimiento de emociones. Su metodología se basa en técnicas de aprendizaje profundo aplicadas a la detección y seguimiento de rostros, a la transcripción de conversaciones, a la detección de la voz y a la clasificación de las emociones a partir de las expresiones faciales y del habla. Además, cabe destacar que los servicios ofrecidos por esta compañía son independientes de factores culturales o regionales dada la extensión de sus bases de datos [71], contando, por ejemplo, con casi 6 millones de representaciones faciales de 75 países diferentes. Este número es extraordinariamente elevado teniendo en cuenta que la mayoría de los conjuntos de entrenamiento públicos ni si quiera alcanzan las 5 0000 imágenes.

En lo que respecta a un enfoque más similar a lo desarrollado en este escrito, tanto por el uso de la misma base de datos (FER-2013 [30], analizada en profundidad en la Subsección 3.3.3) como por la metodología empleada, es necesario señalar las siguientes implementaciones:

- **Reconocimiento de Expresión Faciales utilizando Redes Neuronales Convolucionales: Estado del Arte [53]**. Este modelo, que actualmente está reconocido como el estado del arte en el ámbito del desafío FER-2013, es el resultado de la superación de una serie de cuellos de botella de los sistemas convolucionales (irregularidad de los datos, profundidad de la red, etc.). Su arquitectura está formada por un conjunto de CNNs ensambladas y basadas en modelos actuales y profundos (VGG, Inception y ResNet), así como por una serie de módulos de preprocesamiento que extraen distintos puntos de referencia de interés, igualan el histograma o realizan ajustes lineales a los datos de entrada. La precisión alcanzada por este sistema sobre el conjunto de evaluación de la base de datos FER-2013, sin otros entrenamientos auxiliares anteriores, ha sido de 75.20 %.

- **Aprendiendo los Rasgos de las Relaciones Sociales a partir de Imágenes Faciales [76]**. El modelo aquí descrito desarrolla una serie de técnicas de clasificación y extracción de características basadas en una CNN convencional y una capa adicional y paralela al modelo cuya influencia recae únicamente en la capa de decisión. Mediante esta implementación tan peculiar se pretende fusionar los datos de múltiples fuentes y conseguir, de esta forma, una aproximación espacial de las representaciones de los rostros aprendidos de clases afines. El rendimiento obtenido fue de 75.10 %.

- **Aprendizaje Profundo utilizando Máquinas de Vector de Soporte [67]**. El sistema presentado en este documento fue precisamente el ganador del desafío de reconocimiento de expresiones faciales (FER-2013) con una tasa de acierto de 71.16 %.

Se caracteriza principalmente por combinar las redes neuronales y las SVM, minimizando, por lo tanto, una función de pérdidas basada en los márgenes de decisión en lugar de la entropía cruzada.

- **Clasificación de Emociones con Aumento de Datos mediante Redes Generativas Antagónicas [78]**. La técnica propuesta en este artículo surge como resultado del estancamiento de los avances en el reconocimiento de expresiones faciales debido a las limitaciones de las bases de datos públicas existentes. Por ello, para complementar el conjunto de entrenamiento se ha propuesto una Red Generativa Antagónica de Ciclo Consecuente (CycleGAN) [77] con la que se consigue aumentar la variedad de datos, así como los márgenes entre clases semejantes. Los resultados empíricos han mostrado que es posible obtener un aumento de entre el 5 % y el 10 % en la tasa de acierto empleando estas técnicas.

- **Redes Neuronales Convolucionales para la Clasificación de Emociones y Género en Tiempo Real [2]**. En este documento se expone un marco de implementación consistente en una red neuronal convolucional que realiza las tareas de detección de rostros, clasificación de género y predicción de emociones en tiempo real. La red propuesta está inspirada en la arquitectura Xception [10], que combina módulos convolucionales separables en profundidad y módulos residuales. La precisión obtenida sobre el conjunto de evaluación de expresiones faciales FER-2013 ha sido de 66 %.

Como puede observarse, dada la estandarización de la evaluación de todos estos modelos debido a las características de la base de datos empleada, dividida en tres subconjuntos (entrenamiento, validación y evaluación), los resultados que se van a presentar a lo largo de este proyecto van a poder compararse de forma directa con las tasas anteriormente mencionadas.

Capítulo 3

Marcos de Implementación

Actualmente hay una gran cantidad de entornos de trabajo desarrollados exclusivamente para el aprendizaje profundo. Algunos de los más populares son TensorFlow, Theano, Caffe, Keras o PyTorch. En el caso particular de este proyecto y tal como se expondrá a lo largo de este capítulo son usados básicamente dos: Keras y Tensorflow, dado que son los más versátiles al ofrecer la mayor cantidad de herramientas que simplifican notablemente las implementaciones y los que permiten una integración más directa con otras plataformas como Google Cloud u OpenCV.

También cabe mencionar que cada uno de estos entornos emplea el lenguaje de programación interpretado Python como interfaz de programación *front end*, lo que lo convierte en el escenario mas extendido para el desarrollo de aplicaciones de aprendizaje automático. Además, Python es capaz de combinarse con lenguajes de programación de bajo nivel como C o C++, que actúan generalmente como *back end*.

3.1. Keras y TensorFlow

Keras es una Interfaz de Programación de Aplicaciones (API) de redes neuronales de alto nivel, escrita en Python y capaz de ejecutarse sobre TensorFlow. Esta última, por su parte, es una biblioteca de software de código abierto desarrollada por el departamento de inteligencia artificial de Google para la computación numérica de alto rendimiento con la capacidad de implementarse y ejecutarse utilizando múltiples Unidades de Procesamiento Central (CPUs), Unidades de Procesamiento Gráfico (GPUs) y Unidades de Procesamiento Tensorial (TPUs).

La decisión del empleo de estas dos herramientas, concretamente de Keras sobre TensorFlow, radica en que su combinación proporciona un entorno modular, extensible y fácil de usar, ofreciendo un gran número de utensilios de aprendizaje profundo. En este contexto, Keras facilita tanto las herramientas para desarrollar las arquitecturas neuronales, como el medio y los métodos para entrenar los modelos. Además, en lo que respecta al campo del reconocimiento visual, pone a disposición de la comunidad una gran variedad de modelos pre-entrenados con distintas bases de datos y que pueden emplearse para la predicción, la extracción de características y la afinación de otros sistemas que planteen problemas similares. De hecho, la existencia de estos modelos, que son reconocidos como los estados del arte dentro de su ámbito, simplifica notablemente la implementación de técnicas como la transferencia de aprendizaje, tratada en el Sección 4.2.

3.2. Google Cloud Plataform

Google Cloud Platform es una plataforma que ofrece un conjunto de servicios modulares de computación en la nube, entre los que destacan el almacenamiento de datos, el

	Memoria	Núcleos	B/W Memoria	Memoria disponible
NVIDIA Quadro K4000	3 GB GDDR5	768	134 GB/s	–
NVIDIA Tesla K80	12 GB GDDR5	2496	240 GB/s	1 – 52 GB
NVIDIA Tesla P100	16 GB HBM2	3584	732 GB/s	1 – 104 GB

Tabla 3.1: Comparación entre las GPUs empleadas durante el desarrollo del proyecto.

análisis de datos y el aprendizaje automático. Dentro de este último punto, se van a utilizar básicamente tres herramientas:

- **Cloud Machine Learning Engine**. Es un servicio que proporciona un entorno de implementación de modelos complejos de aprendizaje automático, permitiendo ejecutar aplicación desarrolladas sobre TensorFlow con ciertas alteraciones. Concretamente, mediante este servicio es posible realizar tareas previas de procesado de datos, entrenar modelos de aprendizaje profundo y evaluar sistemas existentes.

  La GPU empleada por este servicio y la única ofrecida actualmente es la NVIDIA Tesla K80.

- **Compute Engine – Cloud Datalab**. Este producto permite desarrollar proyectos en máquinas virtuales escalables de alto rendimiento y ejecutarlos en las infraestructuras de Google. El medio específico que se utiliza es Cloud Datalab, que es una herramienta interactiva que posibilita la exploración, el análisis y la visualización de datos, así como el desarrollo de modelos de aprendizaje automático en *notebooks* con Python y TensorFlow.

  En este caso se dispone de una GPU más potente, la NVIDIA Tesla P100, que permite superar las limitaciones de memoria impuestas por el servicio Cloud Machine Learning Engine.

- **Cloud Storage**. Es el sistema de almacenamiento de la plataforma Google Cloud. Abarca tareas relacionadas tanto con el suministro, el archivado y el análisis de datos, como con el aprendizaje automático. En este proyecto su utilización se centra en el guardado de los puntos de control de los modelos, así como en el almacenamiento de los registros y métricas que permiten analizar el estado del entrenamiento o del aprendizaje en tiempo real.

El uso de esta plataforma para la afinación de los modelos propuestos en el Capítulo 5 tiene como finalidad la reducción de forma abrupta del tiempo de aprendizaje. De hecho, empleando las herramientas anteriormente descritas se ha corroborado personalmente un incremento de la velocidad de entrenamiento de entre 6 y 7 veces con respecto a la utilización de un computador con una GPU NVIDIA Quadro K4000 (gama alta del año 2013) y de hasta 30 veces en comparación con el desempeño obtenido por un computador portátil personal que carece de unidades de procesamiento gráfico. Las comparaciones entre estos equipos puede verse en la Tabla 3.1.

Por último, hace falta mencionar que la elección de Google Cloud Plataform en favor de otros servicio de computación en la nube, como Amazon Web Services o Microsoft Azure, ha estado motivada principalmente por razones económicas y de soporte ya que Google es el proveedor que ofrece la mayor variedad de servicios dentro de su periodo de prueba gratuito.

Figura 3-1: Imágenes extraídas de la base de datos FER-2013 [53].

3.3. Bases de Datos

En el proceso de desarrollo del sistema de reconocimiento de emociones de este proyecto se emplean, directa o indirectamente, tres tipos de bases de datos distintas. Por un lado se hace uso de los conjuntos de datos ImageNet [12] y VGGFace2 [9], que son precisamente los aprendidos por los modelos pre-entrenados ofrecidos por Keras. Por el otro destaca FER-2013 [30], que es la base de datos mediante la cual se va a realizar el afinamiento de los sistemas iniciales para dotarlos de la capacidad de reconocer expresiones faciales.

3.3.1. ImageNet

ImageNet es una base de datos visual enfocada en la investigación de *software* para el reconocimiento de objetos visuales y compuesta por tres subconjuntos de datos: entrenamiento (1.2 millones de imágenes), validación (50 000 imágenes) y evaluación (100 000 imágenes). Estas fotografías, recopiladas de Flickr y otros motores de búsqueda, están etiquetadas a mano indicando la presencia o ausencia de 1 000 categorías de objetos, los cuales abarcan ámbitos tan diversos como razas de perro, especies de plantas u hongos, distintos objetos cotidianos o personas.

Esta amplitud, heterogeneidad y no superposición de clases es la que ha convertido precisamente a ImageNet en el punto de referencia para los algoritmos de clasificación de imágenes, dominados por las redes neuronales convolucionales y las técnicas de aprendizaje profundo desde 2012. Por esta razón, de hecho, gran parte de los modelos que mejor rendimiento han obtenido con esta base de datos se han incluido en la biblioteca de Keras.

3.3.2. VGGFace2

Esta base de datos contiene 3.31 millones de imágenes faciales de 9 131 sujetos célebres, con un promedio de 362.6 fotografías por individuo y con una gran diversidad de poses, de edades, de iluminación, de etnias y de profesiones. Todo este conjunto de datos se divide en un grupo de entrenamiento de 8 631 individuos y en uno de pruebas de 500 individuos. Además, VGGFace2 proporciona anotaciones que permiten la evaluación de los distintos sujetos en dos escenarios diferentes: coincidencia de rostros de diferentes posturas y coincidencia de rostros de diferentes edades.

3.3.3. FER-2013

FER-2013 es la base de datos estáticos de expresiones faciales de mayor relevancia disponible públicamente. Consta de 35 887 representaciones en escala de grises adquiri-

	Ira	Asco	Miedo	Alegría	Tristeza	Sorpresa	Neutral
Núm. de imágenes	4 953	547	5 121	8 989	6 077	4 002	6 198

Tabla 3.2: Número de imágenes por cada expresión facial de la base de datos FER-2013.

das de entornos no acondicionados expresamente con una resolución de 48×48 píxeles. Este conjunto reúne rostros de distinta naturaleza en términos de edad, orientación de la cara, etnia y género, tal y como puede observase en la Figura 3-1 y está dividido en tres subgrupos: entrenamiento, validación y evaluación con 28 709, 3 589 y 3 589 muestras respectivamente. Asimismo, cada una de las imágenes está etiquetada con respecto a la expresión manifestada por el sujeto en la fotografía.

En la Tabla 3.2 se muestra la cantidad de imágenes en esta base de datos de las seis expresiones básicas y la expresión neutral. Como puede advertirse, hay un desequilibrio de clases notable, hecho que va a influir negativamente en el desempeño del modelo, especialmente para las emociones con menor representación. Como solución parcial a este problema se plantean en la Sección 4.3 una serie de técnicas de aumento de datos.

Por último, cabe destacar que la precisión humana sobre este conjunto es de aproximadamente $65 \pm 5\,\%$ [22].

3.4. OpenCV

OpenCV es una librería de código abierto de visión por computador escrita en C/C++ y con un fuerte enfoque en aplicaciones en tiempo real. Por este motivo y teniendo en cuenta su extendido uso, es la que se va a utilizar para el desarrollo del software encargado de detectar y capturar el rostro de forma periódica y en pseudo tiempo real para la posterior identificación de la expresión facial o, dicho de otra forma, para la confección del espejo inteligente. Las características y la descripción detallada de los algoritmos empleados para tal fin se exponen en la Sección 4.4.

Por su parte, la implementación de este sistema se va a llevar a cabo mediante Python y, dado que OpenCV fue diseñada para ser efectiva computacionalmente, en un sistema operativo GNU/Linux optimizado para el hardware de la Raspberry PI.

Capítulo 4

Metodología

A continuación se van a desarrollar los conceptos y técnicas empleadas para la implementación de los sistemas de reconocimiento de expresiones faciales desarrollados en la Capítulo 5, así como sus interacciones con los entornos descritos en el Capítulo 3.

4.1. Capas Empleadas en las Arquitecturas Propuestas

La forma más común de una arquitectura convolucional consiste en el apilamiento de algunas capas convolucionales y capas ReLU seguidas por capas de agrupación, repitiéndose este patrón hasta que la imagen de entrada disminuya espacialmente a un tamaño más reducido. En algunos puntos es común hacer transiciones a capas totalmente conectadas, de *dropout* o capas de normalización. Sin embargo, en la práctica y en el 90 % de las aplicaciones el tipo de estructura es prácticamente intrascendente ya que las CNN son diseñadas o entrenadas desde cero tan solo en ocasiones excepcionales [32].

En otros términos, la disposición de la arquitectura convolucional más habitual, cuyas partes se describirán más detalladamente en los siguientes apartados, sigue el siguiente patrón:

$$\boxed{\text{Entrada}} \rightarrow \boxed{\boxed{\text{Capa Convolucional} \rightarrow \text{Capa ReLU}} \times \mathbf{N} \rightarrow \text{Capa } \textit{Pooling}} \times \mathbf{M} \rightarrow$$
$$\boxed{\text{Capa Totalmente Conectada} \rightarrow \text{Capa ReLU}} \times \mathbf{K} \rightarrow \boxed{\text{Capa Totalmente Conectada}}$$

También cabe destacar que cada una de las capas que a continuación se detallan son ofrecidas como funciones particulares por la API de Keras.

Figura 4-1: Las dos primeras iteraciones del proceso de convolución (izquierda) [33] y resultado de la convolución de una imagen con una matriz detectora de bordes (derecha) [11].

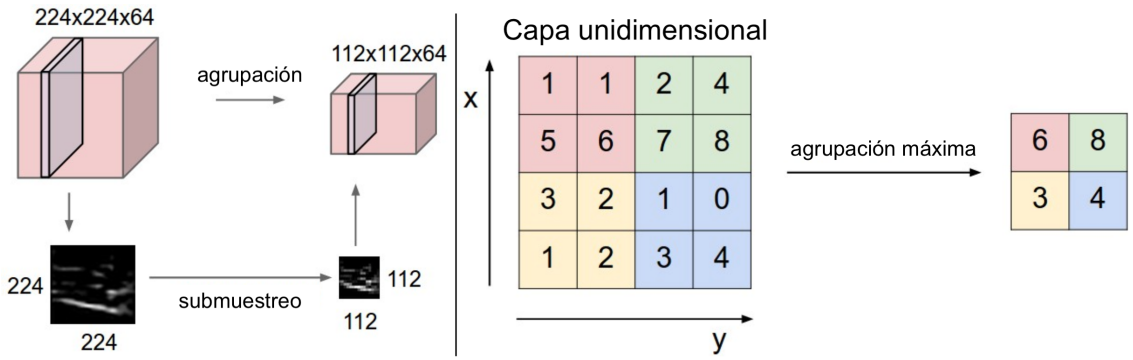


Figura 4-2: Operación de *max pooling* o de reducción de muestreo con un filtro de dimensión 2×2 y paso de 2 unidades de píxel [32].

4.1.1. Capa Convolucional

La capa convolucional es el componente básico de una CNN y sobre la que recae la mayor parte de las operaciones computacionales de la red. Su funcionamiento es análogo al de un filtro que discrimina toda la información que no sea relevante para el mapa de características o de activación. Durante el aprendizaje, cada uno de estos filtros se desliza a través del ancho y alto del volumen de entrada, calculándose la convolución entre el filtro y un área determinada del dato inyectado. De esta forma, se va generando un mapa de activación bidimensional que representa las respuestas del filtro en cada posición espacial, tal y como se puede advertir en la Figura 4-1 y mediante la Ecuación 4.1.

$$(I * K)_{x,y} = \sum_{i=0}^{h-1} \sum_{j=0}^{w-1} K_{i,j} \cdot I_{x+i,y+j} \equiv \text{Mapa de Activación} \tag{4.1}$$

donde $I \equiv$ imagen bidimensional de entrada,

$K \equiv$ matriz de convolución o *kernel* de dimensión $h \times w$

Esta compartición de pesos a lo largo de todo el campo visual es precisamente la que permita que todas las neuronas de una capa convolucional respondan de forma uniforme a una característica concreta, independientemente de su posición. Consecuentemente, la red aprenderá los filtros o pesos que se activan con algún tipo de singularidad visual, como bordes o manchas de algún color específico en las primeras capas o patrones enteros en las capas superiores. Finalmente, el volumen de salida se obtiene apilando los mapas de activación a lo largo de la dimensión de profundidad.

Por último, es necesario remarcar que la agrupación sucesiva de estas capas convolucionales tiene como fin la imposición de una arquitectura compuesta de filtros no lineales que, conforme aumenta la profundidad de la red, se van haciendo más globales y por lo tanto más sensibles a regiones más amplias del espacio de píxeles.

4.1.2. Capa de Agrupación

La función principal de la capa de agrupación o de *pooling* es la de reducir de forma progresiva el tamaño espacial (anchura y altura) de los mapas de activación para disminuir la cantidad de parámetros y la carga computacional de la red, controlando, de esta manera, el sobreaprendizaje. Esta operación también es conocida como reducción de muestreo ya que la síntesis de las características de una región implica la pérdida de información.

Existen varios esquemas posibles para realizar esta agrupación mediante la cual la imagen de entrada se divide en un conjunto de áreas que son reducidas a un único valor.

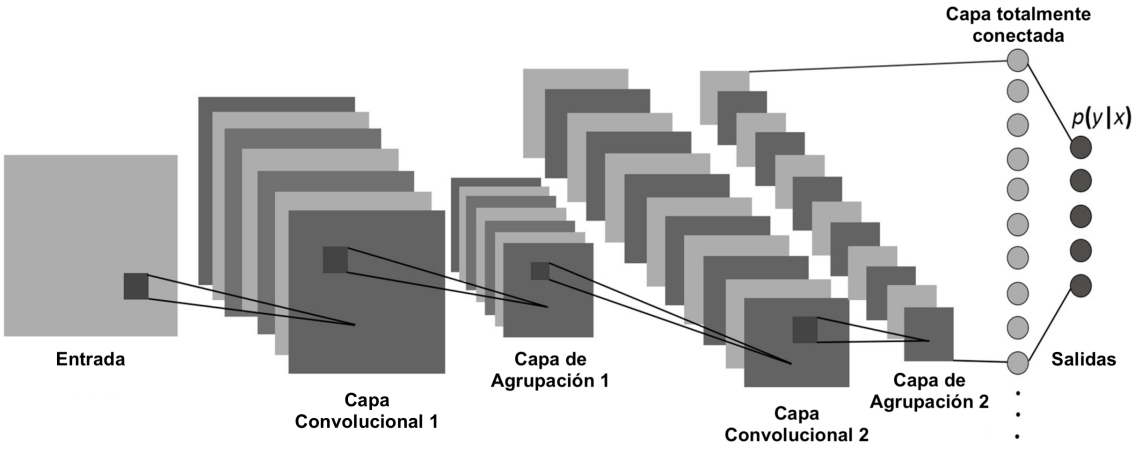


Figura 4-3: Estructura simple de una CNN consistente en capas convolucionales, de agrupación y una capa totalmente conectada [1].

Destacan los siguientes:

- **Agrupación máxima (*Max pooling*)**. Toma el valor máximo de los píxeles de un bloque determinado.
- **Agrupación promedio (*Average pooling*)**. Calcula y toma el valor medio de un área de píxeles definida.
- **Agrupación promedio global (*Global Average pooling*)**. Presenta el mismo funcionamiento que la capa de agrupación promedio con la diferencia de que reduce los mapas de activación de cada dimensión a un único valor.

Tal y como se puede observar en la Figura 4-2, los métodos de agrupación máxima y promedio dan lugar a una reducción del tamaño de los datos por un factor igual a la dimensión de la ventana sobre la cual se opera o de destino, por lo que es común insertarlas periódicamente entre las sucesivas capas convolucionales en una arquitectura CNN [32].

En la práctica y a pesar de que la agrupación promedio ha sido ampliamente explotada históricamente, la operación de agrupación máxima presenta un mejor funcionamiento al conservar de forma más eficaz las características más importantes de la imagen, haciendo que la red sea invariante a pequeñas transformaciones y distorsiones [7].

En lo que respecta a la agrupación promedio global, esta es vista como un regularizador estructural que explícitamente exige una correspondencia entre los mapas de activación y las clases de correspondencia [41]. Es por este motivo por el cual es común incorporarlas en las etapas finales de los modelos y, al ser menos propensa al sobreaprendizaje, como alternativa a las capas totalmente conectadas.

4.1.3. Capa Totalmente Conectada

Una capa totalmente conectada es aquella cuyas neuronas, que no comparten conexiones dentro del mismo nivel, presentan enlaces absolutamente a cada una de las unidades de activación de la capa anterior, tal y como se ha visto en las redes neuronales convencionales (ANN) de la Subsección 2.2.1.

Dentro del contexto de las CNN, su principal función es la de realizar la clasificación en la última parte del modelo. Esto se consigue mediante una asignación de cada una de las dimensión de los datos de entrada a las clases de salida, obteniéndose de esta forma una decisión basada en la imagen completa insertada.

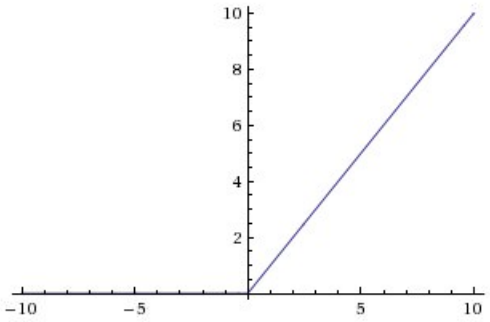


Figura 4-4: Unidad Lineal Rectificada (ReLU) [31].

En la Figura 4-3 se puede observar tanto el empleo de la capa totalmente conectada en calidad de elemento clasificador, como una estructura típica simplificada de una CNN.

4.1.4. Funciones de Activación

La función de activación es una parte esencial de las arquitecturas de las ANN dado que tiene la capacidad de favorecer o perjudicar una determinada región del espacio de entrada de una unidad neuronal al controlar su umbral de activación.

En la actualidad, la función no lineal más popular es la Unidad Lineal Rectificada (ReLU) [40], representada en la Figura 4-4 y a través de la Ecuación 4.2 y cuya tarea es análoga a la de un rectificador de media onda.

$$f(x) = max(0, x) \tag{4.2}$$

Anteriormente, en las redes neuronales se solían emplear funciones de activación más suaves (sigmoide, tangente hiperbólica, etc.), sin embargo, las ReLU generalmente permiten un aprendizaje mucho más rápido en redes profundas, acelerando el proceso incluso por un factor de 6 gracias a su forma lineal no saturante [37]. Además, presentan un menor coste computacional ya que su implementación tan solo requiere la aplicación de un umbral a una matriz de activación en cero, evitando, de esta manera, las costosas operaciones exponenciales de otras funciones.

A pesar de todo ello, mediante la función de activación ReLU no es posible computar las probabilidades de pertenencia de una entrada a unas determinadas clases de salida al ser los valores calculados por esta expresión difíciles de interpretar. En consecuencia, es necesario introducir e implementar una nueva función de activación, típicamente en la última capa de la arquitectura, para medir la compatibilidad de un conjunto de parámetros con respecto a las distintas categorías posibles. Esta habilidad, precisamente, es reunida por la función exponencial normalizada o SoftMax.

Tal y como puede observarse en la Ecuación 4.3, esta función toma un vector N-dimensional y evalúa cada uno de sus elementos dentro del rango $[0, 1]$, cuya suma, además, pasa a ser la unidad, lo que proporciona un entorno especialmente adecuado para la interpretación probabilística en las tareas de clasificación multiclase.

$$p_i = \frac{e^{f_{y_i}}}{\sum_{j=1}^{N} e^{f_j}} \qquad \forall\ i \in 1...N \tag{4.3}$$

donde $f_j \equiv$ elemento j-ésimo del vector de probabilidades

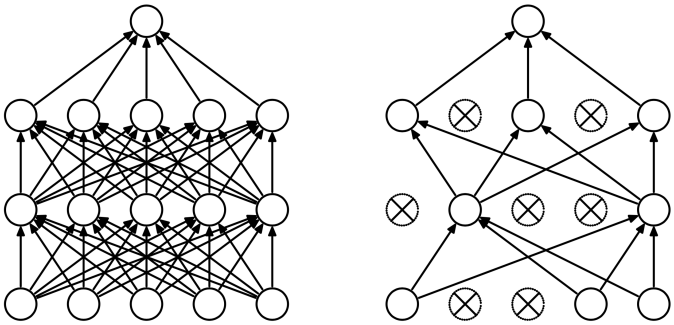


Figura 4-5: Red neuronal estándar con 2 capas ocultas (izquierda) y red análoga a la anterior a la que se le ha aplicado la operación de *dropout* (derecha) [60].

4.1.5. Capa de Normalización por Lotes

La normalización por lotes es una técnica desarrollada para mitigar el efecto de una inicialización desacertada de los pesos de las redes neuronales. Se consigue al forzar explícitamente a las activaciones de la capa previa a asumir una distribución unitaria gaussiana en cada lote al comienzo del entrenamiento.

En comparación con los modelos convencionales de reconocimiento de imágenes, la inclusión de este procedimiento da lugar a que se alcancen los mismos resultados con un número de iteraciones de entrenamiento 14 veces menor [28]. Esto es gracias a que la introducción de esta capa posibilita el empleo de tasas de aprendizaje mucho más altas, a la vez que permite eliminar la compleja labor de diseñar un inicializador adecuado.

En la práctica, este tipo de módulos son insertados inmediatamente después de las capas totalmente conectadas o convolucionales y antes de las funciones no lineales. Además, es una técnica muy utilizada actualmente ya que las redes neuronales que las explotan son significativamente más robustas a una inicialización incorrecta [32].

Por último, es necesario indicar que el pseudocódigo de este proceso de normalización viene descrito más detalladamente en el Algoritmo 2 del Apéndice C.

4.1.6. Capa de *Dropout*

La operación de *droput* es una técnica que permite evitar el sobreaprendizaje, proporcionando una forma de combinar, aproximadamente y de forma exponencial, distintas arquitecturas neuronales de un mismo modelo de manera eficiente [60].

Su funcionamiento es bastante simple, ya que consiste en el apagado aleatorio, en cada una de las iteraciones del proceso de aprendizaje, de una serie de unidades neuronales con una probabilidad de $1 - p$. Cabe destacar que esta red reducida mantiene los pesos originales de las neuronas desactivadas, eliminándose, tan solo, las conexiones entrantes y salientes, tal y como se muestra en la Figura 4-5.

Esta técnica principalmente minimiza el impacto de las neuronas con activaciones dominantes, lo que proporciona a su vez un entorno de mayor independencia entre las distintas unidades durante el entrenamiento. Asimismo, su empleo también consigue mejorar significativamente la velocidad de entrenamiento al reducirse el número de interacciones entre los nodos, lo que además da lugar a una respuesta más generalizada por parte de la arquitectura ante la inserción de datos nuevos.

En definitiva, es un procedimiento muy básico que presenta un funcionamiento excelente para combatir el sobreaprendizaje sin la necesidad de introducir más regularizadores [32].

4.2. Transferencia de Aprendizaje

El aprendizaje por transferencia consiste en la reutilización del conocimiento adquirido durante la resolución de un problema concreto y su posterior aplicación como parte de la solución a uno nuevo, de diferente índole, pero estrechamente relacionado.

Esta forma de abordar los problemas de aprendizaje profundo se debe a que para conseguir unos resultados meramente aceptables es necesario un modelo complejo, el cual, para un entrenamiento desde cero, requiere tanto de considerables recursos (GPUs), como de tiempo (días e incluso semanas de entrenamiento, dependiendo del modelo y del número de GPUs utilizadas). De hecho, en la práctica, el entrenamiento de una red reuronal convolucional completa con una inicialización aleatorio o pseudoaleatoria es realizada por un número reducido de personas, ya que es relativamente raro tener un conjunto de datos de tamaño suficiente con los recursos requeridos para manejarlos [32].

Existen dos escenarios principales del aprendizaje transferido:

- **Extracción de características fijas**. Se emplea una CNN entrenada previamente con una base de datos determinada y a la que se le elimina la última capa totalmente conectada, introduciéndose en su lugar un clasificador instruido con un nuevo conjunto de datos. De esta forma, la red original tan solo funciona como un extractor de características fijo, mientras que el predictor insertado posibilita una clasificación particularizada.

- **Afinación de modelos**. En esta ocasión no sólo se remplaza y reentrena el clasificador de la parte superior de la CNN con el nuevo conjunto de datos, sino que también son ajustados los pesos de la red inicial mediante la continuación del proceso de retropropagación. Asimismo, es posible afinar un número determinado de capas, generalmente las superiores, manteniendo fijos los pesos de las demás. Esto es especialmente útil para evitar el sobreaprendizaje al ser las CNN sensibles a características más genéricas en los niveles iniciales y más específicas en los finales.

En el contexto de este proyecto, se va a explotar principalmente la afinación de arquitecturas convolucionales dada la disponibilidad de una cantidad aceptable de imágenes de expresiones faciales, así como de una serie de modelos pre-entrenados y estrechamente relacionados con el ámbito del reconocimiento de emociones. De hecho, transferir modelos y luego afinarlos da como resultado redes que ofrecen una mejor generalización en comparación con aquellas que son entrenadas directamente con el conjunto de datos disponible [75].

4.3. Aumento de datos

En el campo del aprendizaje profundo, donde el tamaño de las bases de datos tiene una gran influencia en el resultado final, el aumento de datos se usa a menudo para expandir tanto los resultados como la versatilidad del entrenamiento. En cuanto a las técnicas existentes, éstas pueden agruparse en tres tipos principales:

- **Aumento a través de Transformaciones Geométricas**. Se generan nuevas imágenes mediante transformaciones lineales (rotación, traslación, escalado, etc.) que preservan la clase o etiqueta de la representación original.

- **Aumento Guiado por Atributos (AGA)** [14]. El conjunto de entrenamiento es aumentado mediante descriptores de características en lugar de imágenes. Específicamente, esta técnica aprende a sintetizar ciertas propiedades guiada por los valores estimados de un conjunto de atributos, como la profundidad o la pose.

- **Aumento mediante Redes Generativas Antagónicas (GAN)** [23]. A partir de una distribución de datos inicial, estas redes son capaces de producir nuevas representaciones de forma artificial gracias a la imitación de ciertas características de alto nivel extraídas de las imágenes originales.

De esta forma y teniendo en cuenta las limitaciones de la mayoría de las bases de datos, se hace evidente que el empleo de estas técnicas puede dar lugar a mejoras considerables de los resultados. Asimismo, en el contexto de la clasificación de imágenes de expresiones faciales, donde los datos, en la mayoría de los casos, son inadecuados o insuficientes y la distribución de clases desequilibrada, parece resultar especialmente adecuado el empleo de las redes generativas y de las transformaciones geométricas para mejorar el conjunto de entrenamiento.

4.3.1. Transformaciones Geométricas

Con el objetivo de aprovechar lo máximo posible el conjunto de entrenamiento, las transformaciones geométricas pretenden extender la heterogeneidad de los datos de entrada de modo que durante el aprendizaje el modelo entrenado no procese más de una vez la misma imagen. De esta forma se consigue prevenir el sobreaprendizaje y obtener una mejor generalización mediante la alimentación del sistema con unos datos que presentan características diferentes en cada iteración del proceso de entrenamiento.

En este contexto, Keras facilita una serie de clases que permiten, entre otras tantas tareas, la configuración de transformaciones aleatorias, la normalización de los datos de entrada o la aplicación de técnicas como el Análisis de Componentes de Fase Cero (ZCA).

A pesar de todo ello, estos métodos no son suficientes para eliminar la alta correlación existente entre las distintas muestras [36], lo que convierte a estas transformaciones en una solución incompleta, aunque parcialmente efectiva, al problema que supone el uso de un número de imágenes limitado.

4.3.2. Redes Generativas Antagónicas

Las redes generativas antagónicas son una clase de algoritmos de inteligencia artificial desarrolladas por Ian Goodfellow en 2014 y utilizadas en el aprendizaje automático no supervisado [23]. Están formadas por un sistema de dos redes neuronales que compiten entre ellas y cuyo funcionamiento es análogo al algoritmo recursivo o método de decisión *minimax* restringido para dos individuos, el cual consiste en elegir el mejor movimiento para ti mismo suponiendo que tu contrincante escogerá el peor para ti.

En la Figura 4-6 se muestra la estructura de este tipo de redes caracterizadas por integrar dos modelos diferentes, uno generativo y otro discriminativo, que además, son entrenados simultáneamente. La representación matemática del funcionamiento de esta arquitectura es reproducido mediante la Ecuación 4.4. En esta fórmula, el primer término representa la expectación de que el discriminador $D(x)$ reconozca siempre los datos de la distribución real ($p_{datos}(x)$), mientras que el segundo, por su parte, constituye la expectación de que el generador $G(z)$ consiga engañar al discriminador al transformar, mediante la red generativa, los datos de la entrada aleatoria ($p_z(z)$) en una muestra artificial lo suficientemente similar a los datos de la distribución real. En este contexto, el discriminador intentará incrementar la probabilidad de asignar la etiqueta correcta a ambas fuentes de datos, lo que se traduce en una maximización de la función $\mathcal{V}$. Por el contrario, la tarea del generador es exactamente la opuesta, ya que tratará de minimizar la función $\mathcal{V}$ de modo

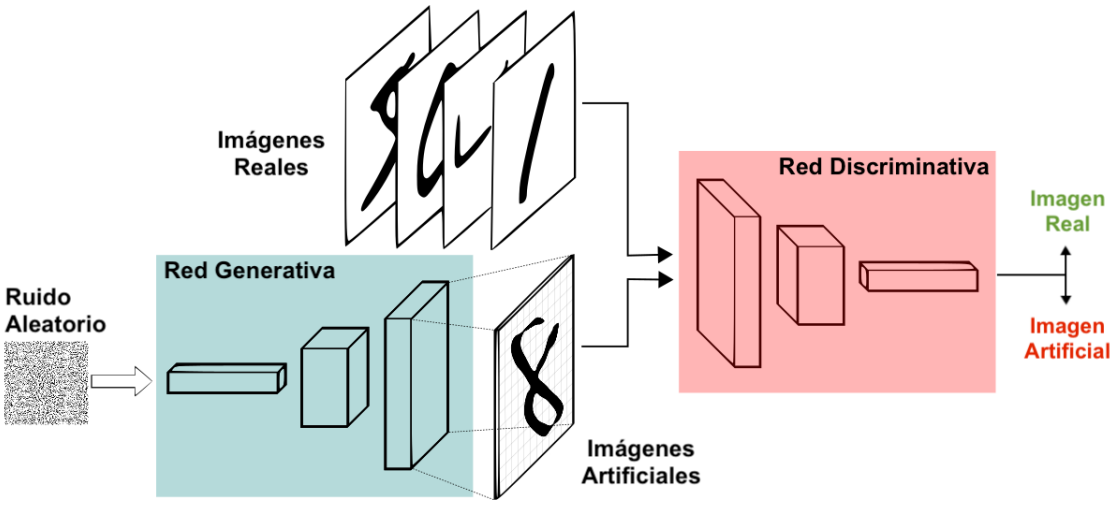


Figura 4-6: Arquitectura de una Red Generativa Antagónica (GAN) [50].

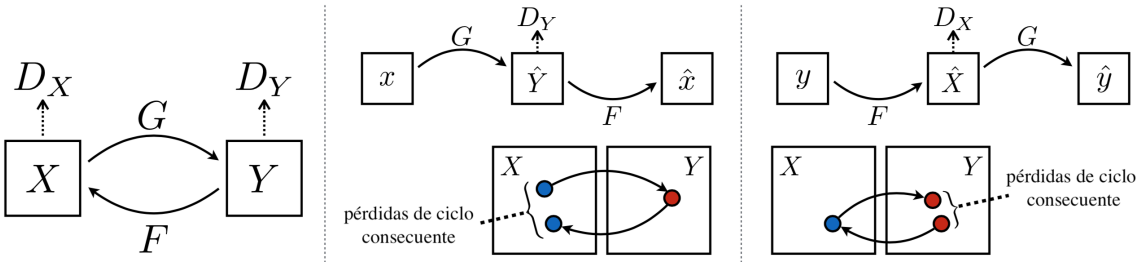


Figura 4-7: Estructura de los procesos de correspondencia directa ($G : X \rightarrow Y$) e inversa ($F : Y \rightarrow X$) en las Redes Generativas Antagónicas de Ciclo Consecuente [77].

que la diferencia entre los datos reales y los artificiales sea la mínima.

$$\min_G \max_D \mathcal{V}(D,G) = \underbrace{\mathbb{E}_{x\sim p_{datos}(x)}\big[\log D(x)\big]}_{1^{er}\ término} + \underbrace{\mathbb{E}_{z\sim p_z(z)}\big[\log(1-D(G(z)))\big]}_{2^{o}\ término} \tag{4.4}$$

En definitiva, esta técnica permite generar imágenes que parecen, al menos superficialmente, auténticas para los observadores humanos a partir de otras que están parcialmente relacionadas con el resultado que se quiere obtener.

Redes Generativas Antagónicas de Ciclo Consecuente

Es un método recientemente introducido [77] para la generación de imágenes que es particularmente útil al permitir utilizar datos de entrenamiento no emparejados, es decir, para entrenar este tipo de arquitecturas no es necesario establecer correspondencias directas entre las imágenes de los dominios inicial y final. Su funcionamiento está basado en las redes GAN convencionales a las que se le añade una función adicional que monitoriza las pérdidas de ciclo consecuente, hecho que permite al modelo aprender tanto las correspondencias directas como las inversas.

En la Figura 4-7 se puede observar la estructura simplificada de esta red, mientras que

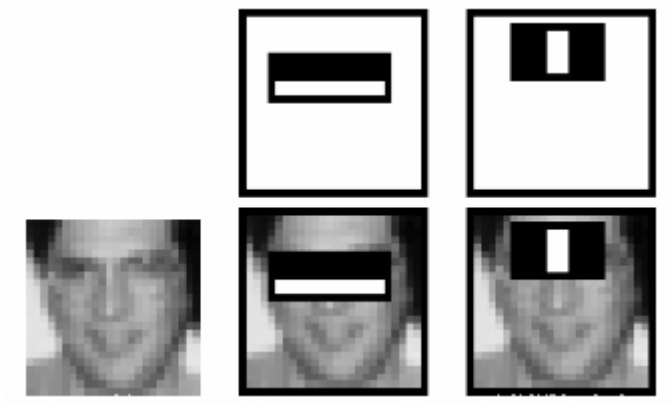


Figura 4-8: Características seleccionadas por AdaBoost. La primera característica mide la diferencia de intensidad entre la región de los ojos y la región superior de las mejillas, mientras que la segunda compara las intensidades de las regiones oculares con la del puente de la nariz [72].

la expresión por la que se rige el aprendizaje es la de la Ecuación 4.5.

$$\begin{aligned} \min_{G,F} \max_{D_X,D_Y} \mathcal{V}(G,F,D_X,D_Y) &= \mathcal{L}_{GAN}(D_X,F) + \mathcal{L}_{GAN}(D_Y,G) \\ &\quad + \lambda \cdot \mathcal{L}_{cycleGAN}(G,F) \qquad (4.5) \\ \text{donde } \mathcal{L}_{GAN} &\equiv \text{Ecuación 4.4} \\ \mathcal{L}_{cycleGAN}(G,F) &= \mathbb{E}_{x\sim p_{datos}(x)}\big[\big\|F(G(x)) - x\big\|\big] + \\ &\quad \mathbb{E}_{y\sim p_{datos}(y)}\big[\big\|G(F(y)) - y\big\|\big] \qquad (4.6) \\ \lambda &\equiv \text{peso de la función de pérdidas de ciclo consecuente} \end{aligned}$$

En esta ocasión no sólo se pretende hacer que las imágenes generadas se perciban como las imágenes objetivo, sino que también las reconstruidas sean identificadas como las originales, garantizándose así la consistencia del ciclo. Es por ello que se introduce el término de la Ecuación 4.6, el cual tiene en consideración el error existente entre las representaciones de entrada y sus reconstrucciones obtenidas al pasar a través de las dos funciones de mapeo, G y F.

Resumidamente, en el ámbito particular de este proyecto, la utilización de estas redes tiene como fin el aumento del número de imágenes que tienen menor representación de la base de datos de expresiones faciales FER-2013 para obtener un mejor comportamiento del sistema final.

4.4. Detección de rostros en tiempo real

Para conseguir una integración efectiva del módulo que reconoce las expresiones faciales y del sistema empotrado, encargado de proporcionarle al sistema principal imágenes capturadas en tiempo real y a partir de las cuales se extrae tan solo el rostro del individuo concreto, se emplea, tal y como se ha mencionado en la Sección 3.4, la librería de visión artificial OpenCV. Este entorno es especialmente adecuado para las tareas aquí planteadas ya que proporciona diversos tipos de clasificadores o detectores ya entrenados y encapsulados en ficheros XML, que pueden ser aprovechados para una gran variedad de tareas.

En lo que respecta al procedimiento en sí de la detección del rostro, este se va a realizar mediante clasificadores en casada basados en el efectivo método para detectar objetos propuesto por Paul Viola y Michael Jones en 2001 [72]. Este método de aprendizaje automático

consiste básicamente en entrenar una función cascada sobre numerosas imágenes positivas (imágenes con caras) y negativas (imágenes sin caras), extrayéndose los distintos atributos mediante las características Haar, tal y como se muestra en la Figura 4-8. Su funcionamiento es similar a los filtros convolucionales vistos anteriormente, obteniéndose un único valor a partir de cada característica al restar la suma de los píxeles debajo del rectángulo blanco a la suma de los píxeles debajo del rectángulo negro. Sin embargo, la aplicación de estas operaciones a imágenes de altas resoluciones resulta muy costoso computacionalmente, por lo que se hace necesario utilizar métodos para disminuir el número de operaciones. Destacan el meta-algoritmo de aprendizaje automático Adaboost, que consigue optimizar la selección de características, y el método de la cascada de clasificadores, que aprovecha la hipótesis de que la mayor parte de los píxeles de una imagen son irrelevantes para agrupar la aplicación de las distintas características Haar en diferentes etapas de clasificación.

En resumen, esta detección de rostros mediante OpenCV logra ser especialmente eficaz, tanto computacionalmente como con respecto al rendimiento ofrecido.

Capítulo 5

Modelos Propuestos y Resultados

Tal y como se ha comentado en la Sección 4.2, la explotación de las técnicas de transferencia de aprendizaje y por lo tanto el empleo de unos modelos con una serie de pesos ya definidos con respecto a un conjunto de datos determinado, constituyen la base de los sistemas propuestos en este capítulo.

Por consiguiente, son utilizados en primera instancia los modelos Inception-v3 [65] e Inception-ResNet-v2 [62] con los pesos entrenados previamente sobre el conjunto de datos ImageNet descrito en la Subsección 3.3.1. Sin embargo, dado que las propiedades de las imágenes de esta base de datos difieren en gran medida de las características faciales que se intentan aprender y reconocer, se ha optado por explorar el uso de modelos menos eficaces pero enfocados al reconocimiento facial. Es por ello que en última instancia se emplea la arquitectura ResNet-50 [24] preetrenada con la base de datos VGGFace2 expuesta en la Subsección 3.3.2. La comparación de estos modelos en el desempeño del desafío ILSVRC puede observarse en la Figura 5-1. Esta representación, además, escenifica la principal razón por la cual se han elegido estos sistemas concretos: son los que mejores tasas obtienen con respecto al coste computacional y al número de parámetros.

En definitiva, a lo largo del proceso de desarrollo de un sistema de reconocimiento de expresiones faciales válido, se han ido explorando numerosas arquitecturas (Inception-v3, Inception-ResNet-v2 y ResNet-50) y técnicas (aumento de datos) con el objetivo de ir obteniendo cada vez mejores resultados sobre la base de datos FER-2013. Asimismo, a fin de permitir una comparación justa con respecto a los resultados de la Sección 2.5, son utilizados los protocolos de uso estipulados inicialmente por este desafío y que establecen la división de esta base de datos en tres conjuntos diferentes: entrenamiento, validación y evaluación.

5.1. Arquitecturas afinadas

5.1.1. Inception-v3

Inception-v3 es el resultado de las investigaciones llevadas a cabo por el equipo de Google para conseguir un modelo con una arquitectura cada vez más profunda e inteligente y capaz de desenvolverse de forma eficiente, tanto computacionalmente como cualitativamente, en el desafío ILSVR.

En la Figura 5-2 se muestra la estructura simplificada y adaptada al problema del reconocimiento de expresiones faciales del modelo Inception-v3. Esta arquitectura propuesta es básicamente una sucesión de tramos convolucionales y no linealidades, empleándose la función de activación ReLU y la normalización por lotes en cada una de las etapas según lo descrito en Sección 4.1, aunque esto último no se muestre explícitamente en la representación anteriormente mencionada. Como se ha podido advertir, la característica principal

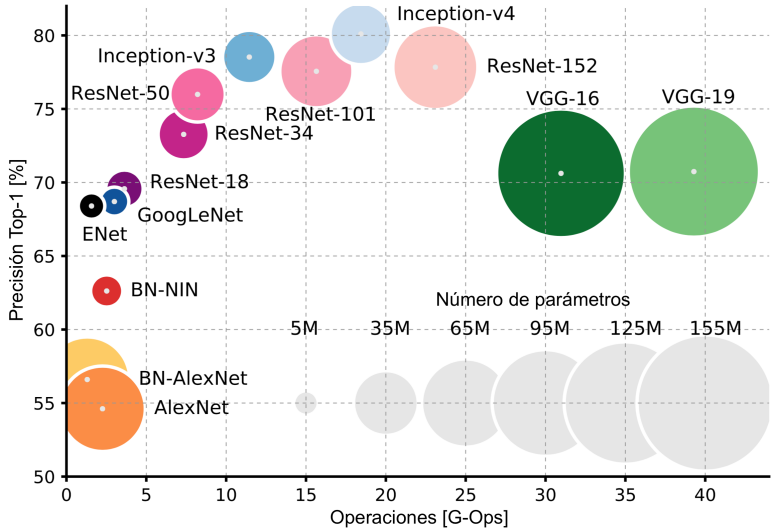


Figura 5-1: Precisión Top-1 frente al coste computacional de una iteración del proceso de aprendizaje y el número de parámetros de la red [8]. Cabe destacar que aunque el modelo Inception-ResNet-v2 no se incluya en la figura, presenta características muy similares a Inception-v4 [62].

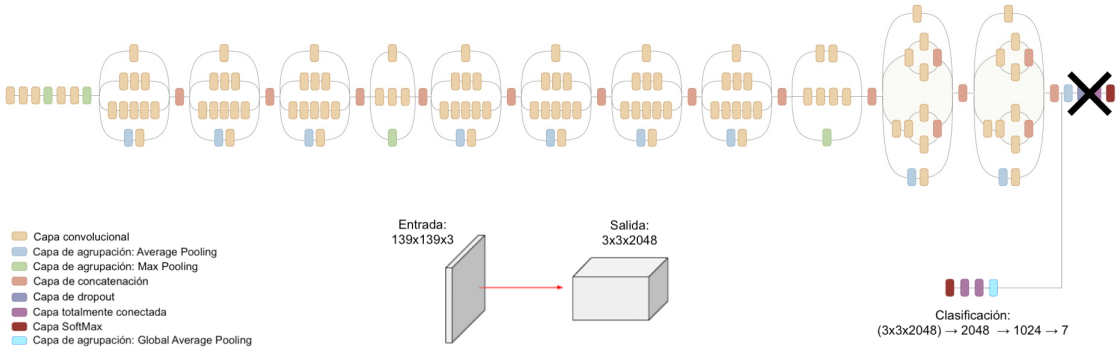


Figura 5-2: Arquitectura del modelo Inception-v3 adaptada al problema del reconocimiento de expresiones faciales [45].

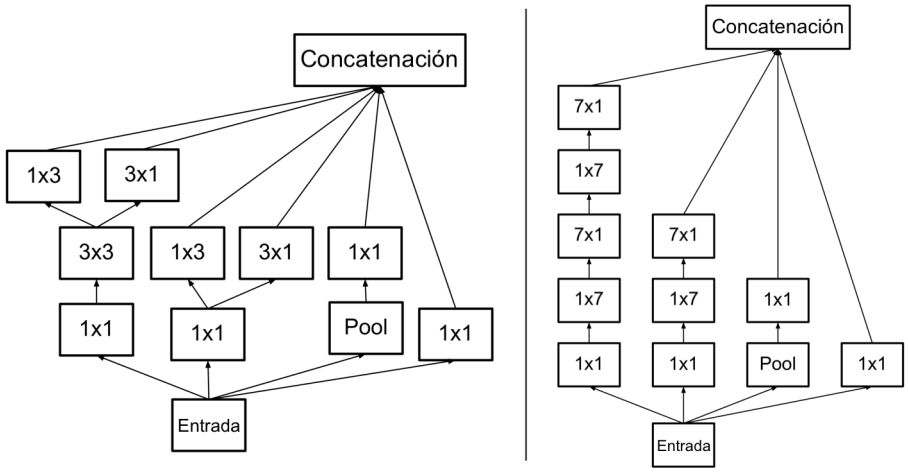


Figura 5-3: Módulos Inception empleados en la arquitectura Inception-v3. Estos bloques son utilizados para promover las representaciones de alta dimensión (izquierda) y la factorización de las convoluciones de dimensión $n \times n$ (derecha) [65].

de este modelo es el hecho de que el flujo de datos a lo largo de algunas secciones de esta red no es secuencial, sino que se realiza en paralelo. Estas agrupaciones son conocidas como módulos Inception y representan la solución a los problemas de eficiencia de las redes estado del arte predecesoras, cumpliendo, además, la misma función y obteniendo los mismos resultados que una capa convolucional estándar. Las diferentes estructuras de estos módulos empleados por el modelo Inception-v3 pueden observarse en la Figura 5-3. La idea detrás de estas disposiciones paralelas es que dada la misma entrada para varias capas convolucionales o de agrupación de distinto tamaño se generen características únicas para cada una de ellas que posteriormente procederán a concatenarse. Este enfoque, sin embargo, da lugar a una salida con una profundidad extremadamente grande que es solucionada mediante la utilización de filtros convolucionales 1×1, especialmente efectivos para reducir la dimensionalidad [41] e incorporados justo antes de las capas convolucionales de mayor tamaño. Otro de los puntos que es explotado por los módulos Inception es la sustitución de los filtros tradicionales de tamaño $n \times n$ por una secuencia de capas convolucionales de dimensiones $1 \times n$ y $n \times 1$. Mediante esta técnica se consigue disminuir drásticamente los costes computacionales a medida que n aumenta. En la práctica y tal como se ha visto en la Figura 5-3, son utilizados básicamente filtros con $n = 7$ y $n = 3$.

Por otro lado, dado que esta red ha sido diseñada para competir en el reto ImageNet, es necesario modificar la etapa de clasificación del modelo Inception-v3 original, tal y como se ha indicado en la Figura 5-2. De esta forma, en primer lugar se ha procedido a introducir una capa basada en la agrupación promedio global, que impone la correspondencia entre los mapas de activación y las clases, reduce el número de parámetros y además es menos propensa al sobreaprendizaje que las capas convencionales totalmente conectadas [41]. Posteriormente y con la finalidad de reducir la dimensionalidad de la red de una manera suave a las 7 clases correspondientes a las expresiones faciales que se pretenden clasificar, son insertadas dos capas totalmente conectadas de 2 048 y 1 024 neuronas respectivamente.

En cuanto a la entrada y puesto que las imágenes del conjunto de datos FER-2013 presentan una resolución bastante baja (48×48 píxeles) en comparación con las representaciones de ImageNet (299×299 píxeles) y las mínimas aceptadas por el modelo Inception-v3 (139×139 píxeles), es requerida una modificación parcial del comportamiento de la primera etapa de la arquitectura utilizada. Concretamente es necesario modificar las dos primeras capas de agrupación para evitar la omisión de parte de los datos inyectados en la entrada. Sin embargo, esto ya es realizado por Keras de forma automática.

5.1.2. Inception-ResNet-v2

Inception-ResNet-v2 es una ampliación, llevada a cabo por el grupo de inteligencia artificial de Google, de los conceptos planteados en la arquitectura Inception-v3 mediante dos metodologías que tienen como fin aumentar el rendimiento y el número de módulos Inception. La primera consiste simplemente en el empleo de una estructura ligeramente distinta dentro de los propios bloques en función de la posición de éstos en la red, mientras que la segunda, por su parte, plantea la utilización de conexiones residuales, similares a las desarrolladas por el equipo de Microsoft en las redes ResNet [24], para acelerar el proceso de entrenamiento y aumentar la profundidad de la arquitectura. Estas conexiones, además, favorecen la simplificación de los bloques Inception. En definitiva, estas dos ideas permiten alcanzar e incluso superar el rendimiento de los modelos puros de Inception con un número significativamente menor de iteraciones de entrenamiento.

La arquitectura sintetizada de este modelo se expone en la Figura 5-4. Como puede observarse, a parte de las estructuras de entrada y de clasificación, cuyo funcionamiento es análogo al descrito en la Subsección 5.1.1, esta red está formada principalmente por 5 módulos Inception distintos y denominados como Inception-ResNet-A, Reduction-A, Inception-ResNet-B, Reduction-B e Inception-ResNet-C. Estos tipos de bloques, represen-

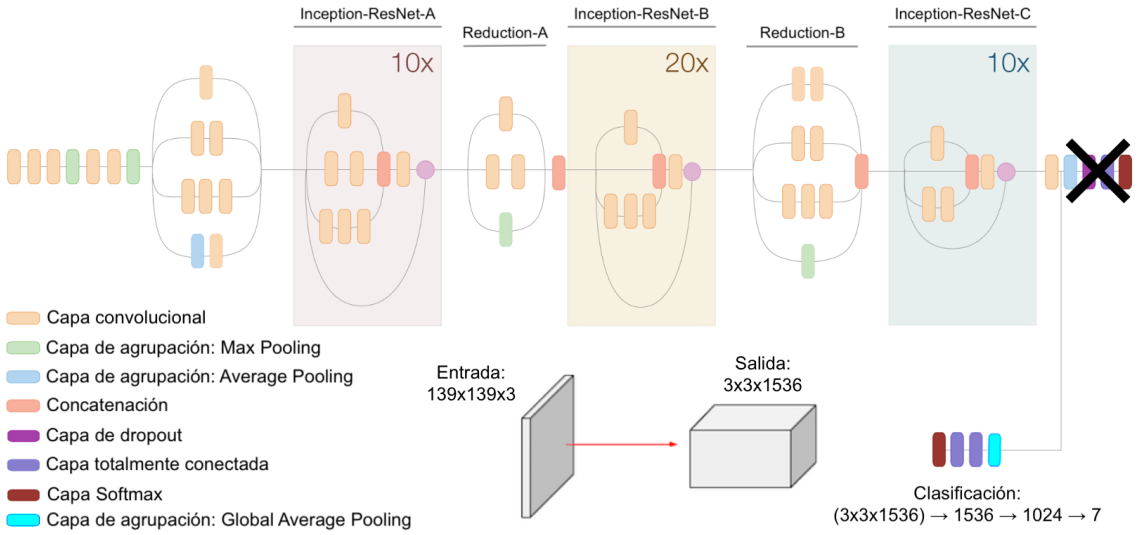


Figura 5-4: Arquitectura comprimida y adaptada al problema del reconocimiento de expresiones del modelo Inception-ResNet-v2 [5].

tados en la Figura D-1 del Apéndice D, tienen dos propósitos diferentes, encargándose esencialmente de la construcción de los mapas de características de los datos de entrada y de la reducción de estos espacios de activación con el fin de obtener unas representaciones homólogas de menor tamaño.

De forma concreta y tras procesarse inicialmente las imágenes en la entrada, esta información es inyectada en los bloques Inception-ResNet-A de la Figura D-1a, que mediante el uso de filtros de dimensiones reducidas (1×1 y 3×3) son capaces de detectar características muy básicas y disminuir el tamaño de las fotografías. Tras ello, estos datos son introducidos en el módulo Reducción-A expuesto en la Figura D-1d, que condensa aún más las dimensiones de las representaciones a fin de acelerar el entrenamiento de los posteriores bloques Inception-ResNet-B de la Figura D-1b. Estos últimos, de hecho, contienen un número menor de capas que sus predecesores Inception-ResNet-A, aunque utilizan filtros de mayor tamaño (1×1, 1×7 y 7×1) con el propósito de detectar características más complejas. Posteriormente, los resultados de estos últimos módulos son transferidos a la estructura Reduction-B que al contar con más capas que Reduction-A, como puede observarse en la Figura D-1e, es capaz de hacer frente a una mayor cantidad de datos que son la consecuencia directa del empleo de filtros de mayores dimensiones en los bloques Inception-ResNet-B. Toda esta información reducida es transferida seguidamente a los módulos Inception-ResNet-C, mostrados en la Figura D-1c, y que debido al aumento de la complejidad de la arquitectura emplean en esta ocasión un número reducido de filtros para aminorar el coste temporal del entrenamiento. Finalmente, la conexión secuencial de todos estos bloques según la Figura 5-4 da lugar a una red con una profundidad considerable, que tras extraer las características de las imágenes inyectadas en la entrada procederá a categorizarlas mediante el módulo clasificador adaptado al contexto del reconocimiento de expresiones faciales y con una arquitectura idéntica a la del modelo Inception-v3.

5.1.3. ResNet-50

La arquitectura ResNet se concibió por primera vez en 2015 [24] como un intento de crear un modelo de clasificación de imágenes que solucionara los problemas de las redes convencionales cuya respuesta, al contrario de lo que se podía intuir, se degradaba significativamente al aumentarse la profundidad. De esta forma, esta investigación demostró

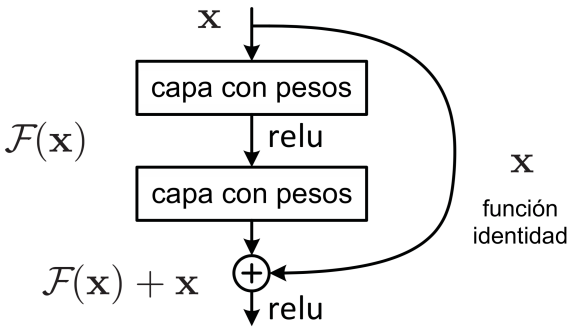


Figura 5-5: Bloque residual básico [24].

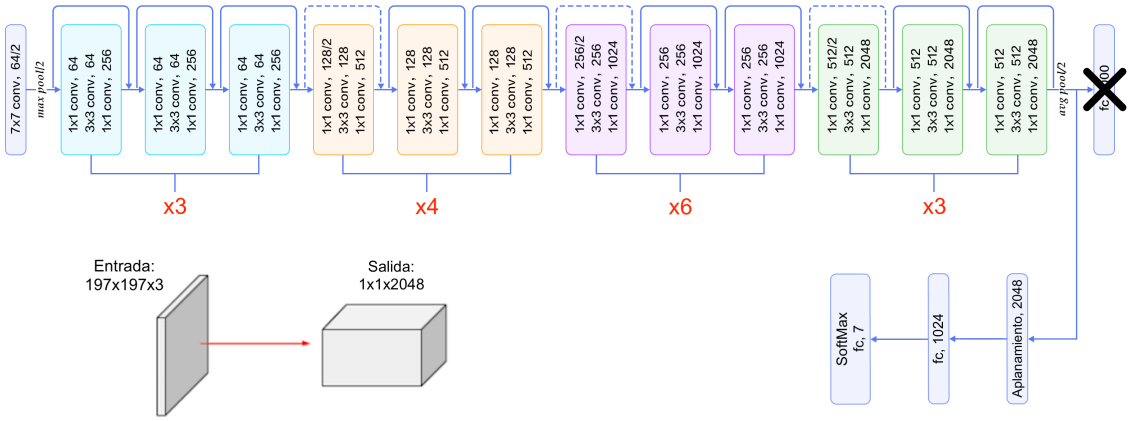


Figura 5-6: Arquitectura del modelo ResNet-50 adaptado al problema del reconocimiento de expresiones faciales [51].

que la agregación de ciertas conexiones adicionales que pasaban por alto determinadas capas a las redes tradicionales permitía seguir aumentando el número de niveles sin que el rendimiento decayera. La unidad básica que aprovecha esta idea y de la que se componen este tipo de redes se expone en la Figura 5-5. Como se puede advertir, estos módulos constan básicamente de dos capas convolucionales apiladas y una conexión que agrega los datos de la entrada directamente a la salida, tal y como se muestra en la Ecuación 5.1.

$$H(x) = F(x) + x \tag{5.1}$$

donde $x \equiv$ entrada del bloque

$F(x) \equiv$ función residual aprendida por las dos capas convolucionales

La hipótesis de esta implementación expone que es más fácil optimizar los coeficientes de la asignación original que se encuentren cerca de la función de identidad forzando la función residual a cero, que realizar un mapeo de identidad mediante una sucesión de capas no lineales [24]. Esta característica es precisamente la que se explota a lo largo de la profundidad de este tipo de redes permitiendo evitar y aliviar cualquier complejidad que vaya apareciendo durante el entrenamiento simplemente mediante la omisión de partes de la red.

Hay numerosas variaciones de las arquitecturas ResNet, empleándose con una gran variedad de niveles de profundidad (34, 50, 101 e incluso 152 capas). Sin embargo, en el presente proyecto se usa principalmente la ResNet de 50 capas de la Figura 5-6, ya que es precisamente la utilizada por el Grupo de Geometría Visual de la universidad de Oxford en el entrenamiento de la base de datos VGGFace2 [9] y a partir de cuyos resultados se va a desarrollar la red particular enfocada al problema del reconocimiento de emociones.

Como puede observarse, esta arquitectura adaptada no es más que una sucesión de bloques residuales con la novedad, con respecto a los modelos anteriormente descritos, de que, dada la naturaleza de la red, se ha empleado una capa auxiliar de aplanamiento que vincula la etapa de clasificación y la de extracción de características. Asimismo y aunque no aparezcan por simplicidad en la Figura 5-6, cabe destacar que son empleadas la capa de activación ReLU y la capa de normalización por lotes en cada una de las etapas convolucionales, tal y como se describió más detalladamente en la Sección 4.1.

5.2. Entrenamiento

Por un lado, el método de aprendizaje descrito en este apartado trata de aprovechar lo máximo posible las capacidades de la base de datos FER-2013 mediante un preprocesamiento meticuloso y por otro, beneficiarse de los recursos computacionales ofrecidos por la plataforma de Google Cloud.

5.2.1. Preprocesamiento de los Datos

En este proyecto, los objetivos principales del preprocesamiento del conjunto de entrenamiento de la base de datos FER-2013 son el aumento de la heterogeneidad o diversidad de estas imágenes y su adaptación a las características de los distintos modelos empleados, así como a las especificaciones impuestas por ciertas herramientas de la API de Keras. Por todo ello, son aplicadas las siguientes transformaciones a los datos antes de su procesamiento por las arquitecturas planteadas en el apartado anterior:

- **Redimensionamiento**. En primer lugar es necesario modificar la resolución original de 48×48 píxeles de las imágenes del conjunto de datos FER-2013 a las dimensiones aceptadas por los modelos Inception-v3, Inception-ResNet-v2 y ResNet-50 cargados en la librería de Keras. En los tres casos, y dadas las limitaciones de memorias observadas durante el entrenamiento (tratadas más detenidamente en las secciones correspondientes), la conversión se realiza a las proporciones mínimas aceptadas por estas arquitecturas:

  - **Inception-v3**: 139×139 píxeles.
  - **Inception-ResNet-v2**: 139×139 píxeles.
  - **ResNet-50**: 197×197 píxeles.

  Este proceso se realiza mediante el método considerado como estándar para este tipo de tareas de manipulación de imágenes y es conocido como interpolación bicúbica. A pesar de que su tiempo de procesamiento es más lento en comparación con las alternativas existentes (interpolación bilineal o interpolación por el vecino más cercano) es el que mejor resultados permite obtener en términos de calidad. Por ello y para evitar aumentar más el ruido de las imágenes empleadas, cuya calidad de por sí es bastante pobre, se ha optado por esta opción.

  Por otro lado, las tres redes originales presuponen en la entrada imágenes con un modelo de color RGB, por lo que también se hace necesario hacer una conversión de las representaciones de expresiones faciales en escala de grises a esta especificación. Esto es realizado simplemente mediante una reproducción de las imágenes originales a lo largo de las tres dimensiones correspondientes a los tres espacios de color de destino.

- **Normalización**. Dado que tras el redimensionado las imágenes se componen de una serie de coeficientes RGB en el intervalo $[0, 255]$, es indispensable realizar una normalización con el fin de evitar el manejo de valores demasiado altos que generalmente

dan lugar a una ralentización del proceso de aprendizaje. Concretamente, y con el objetivo de no eliminar en primera instancia los valores de los pesos iniciales, se han seguido los mismos procedimientos de regularización que los propuestos en las redes originales preentrenadas:

- **Inception-v3** y **Inception-ResNet-v2**. Los valores RGB del intervalo $[0, 255]$ son adaptados al espacio $[-1, 1]$ mediante la expresión matemática de la Ecuación 5.2.

$$\hat{x} = \frac{x}{127.5} - 1 \qquad \forall\, x \in [0, 255] \tag{5.2}$$

- **ResNet-50**: Tanto en el documento original de este modelo [24] como en el artículo que hace uso de la base de datos VGGFace2 [9], la normalización es llevada a cabo mediante la sustracción a cada uno de los píxeles del valor promedio del espacio de color al que corresponden. Mediante este proceso se consigue el centrado de los datos en cero. Sin embargo, dado que el conjunto FER-2013 redimensionado según el punto anterior presenta los mismos coeficientes RGB en los 3 canales, se ha considerado conveniente realizar la sustracción del mismo promedio a cada una de las dimensiones según la Ecuación 5.3.

$$\hat{x} = x - \overline{x} = x - 128.8006 \qquad \forall\, x \in [0, 255] \tag{5.3}$$

  Asimismo, el valor de esta media se ha calculado tan solo sobre los datos de entrenamiento del conjunto FER-2013, aplicándose posteriormente a los grupos de validación y evaluación.

- **Transformaciones geométricas**. Tal y como se ha visto en la Subsección 4.3.1, la aplicación de una serie de transformaciones geométricas a las imágenes del conjunto de entrenamiento puede enriquecer la base de datos empleada. Por ello, en el caso particular de este proyecto se realizan las siguientes conversiones lineales de forma pseudoaleatoria en cada iteración del entrenamiento:

  - **Rotación**. Es aplicada una rotación de sentido aleatorio de entre 0 y 10 grados sexagesimales a cada una de las imágenes. Para la elección de este valor se han tenido en cuenta el rango máximo de flexión lateral del cuello, que varía entre 20 y 45 grados [74],y el hecho de que ciertas imágenes del conjunto de entrenamiento FER-2013 ya presentan rostros ligeramente girados.
  - **Transformación de cizallamiento**. Esta transformación también es conocida como inclinación y es efectuada en las direcciones del eje de abscisas con una intensidad comprendida entre los 0 y los 10 grados sexagesimales.
  - **Volteo**. Se fuerza a un volteo horizontal aleatorio de las imágenes inyectadas.
  - ***Zoom***. A pesar de que los datos de entrenamiento ya se componen de imágenes con rostros centrados y enfocados, se ha recurrido a una ampliación o reducción de carácter aleatorio de las representaciones del 10 % con el objetivo de lograr una mayor generalización.
  - **Relleno**. Dado que algunas de las transformaciones anteriormente nombradas implican la introducción de puntos externos a las imágenes en los campos receptivos originales, se hace necesaria la aplicación de un método que evite la inclusión de este ruido. En este contexto se han explorado distintos procedimientos de relleno, como la propagación constante de píxeles o el método de los vecinos más próximos. Sin embargo, los mejores resultados observados experimentalmente se han obtenido mediante la técnica de reflexión de las imágenes transformadas.

5.2.2. Proceso de Aprendizaje

El procedimiento llevado a cabo para conseguir un aprendizaje efectivo en cada uno de los modelos ha consistido, en primer lugar, en un entrenamiento tan solo de las capas de la etapa de clasificación particularizada al problema del reconocimiento de expresiones faciales. Tras esta técnica para evitar una inicialización aleatoria de los niveles superiores se ha seguido con un entrenamiento íntegro de los modelos completos, que es precisamente sobre el cual se van a reportar los resultados finales.

En este contexto, se procede a describir a continuación algunas de las características compartidas del proceso de aprendizaje de los tres modelos descritos anteriormente (Inception-v3, Inception-ResNet-v2 y ResNet-50):

- **Tamaño de los lotes**. Para aprovechar lo máximo posible las capacidades de computo que ofrece la plataforma Google Cloud y agilizar el aprendizaje se consideró oportuno en un primer momento inyectar a los modelos lotes con un número de ejemplos de entrenamiento considerable. Sin embargo, dado que el alimentador de datos provisto por la API de Keras es ejecutado en paralelo al modelo, se ha comprobado experimentalmente que, incluso la GPU más potente ofrecida por Google Cloud (NVIDIA Tesla P100, cuyas características se encuentran en la Tabla 3.1), es incapaz de asignar la memoria necesaria que requieren los lotes con más de 128 imágenes. Este problema también es favorecido por la complejidad de los modelos usados, que además presentan una gran cantidad de parámetros como se puede comprobar en la Tabla 5.1, así como por la realización de un intenso preprocesamiento a cada una de las imágenes en cada iteración. Si bien, este último procedimiento está optimizado para que sea realizado exclusivamente por la CPU. Teniendo en cuenta estas consideraciones, se ha llegado a la conclusión de que utilizar lotes de menor tamaño es la manera más simple y efectiva de reducir la memoria demandada. Por todo ello, el tamaño de los lotes se ha estipulado en 128 imágenes. También cabe remarcar que esta dimensión es múltiplo de 2 debido a las evidencias existentes de que de esta manera la GPU es capaz de distribuir de una forma más óptima la carga de trabajo debido a la forma en la que se codifican los datos [27].

- **Función de pérdidas**. Hay una gran cantidad de maneras de cuantificar la función de pérdidas que determina la calidad de la predicción realizada con respecto a unos determinados datos de entrada. En el caso particular de este proyecto y considerando la naturaleza de los datos del conjunto FER-2013, categorizados en 7 clases, es empleada la función de la entropía cruzada categórica. Su uso para determinar el desempeño de un modelo concreto en cada iteración está ampliamente extendido y es, de hecho, la elección predeterminada en los problemas de aprendizaje profundo multiclase debido a las mejoras de velocidad que permite obtener con respecto a otras implementaciones tradicionales como el coste cuadrático o el exponencial [47]. Su expresión matemática puede observarse en la Ecuación 5.4. De esta última también se puede deducir que en el proceso de aprendizaje se intentará minimizar la entropía cruzada entre las probabilidades estimadas y la distribución verdadera, para posteriormente actualizar los pesos correspondientemente.

$$H(p,q) = \sum_{x} p(x) \log q(x) \tag{5.4}$$

  donde $p(x) \equiv$ distribución verdadera donde toda la masa de probabilidad está en la clase correcta, es decir, $p = [0, ...1, ..., 0]$ contiene un único 1 en la posición iésima

  $q(x) \equiv$ distribución estimada

- **Iteraciones de entrenamiento de las capas superiores**. Dado que el objetivo

del entrenamiento de la etapa de clasificación introducida es la obtención de unos resultados ligeramente superiores a los obtenidos con la inicialización aleatoria, que ajusta los pesos iniciales con una distribución uniforme en el intervalo $[-0.05, 0.05]$, se ha estipulado un número de tan solo 5 iteraciones, que ha permitido alcanzar tasas de acierto en esta subred de entre el 20 % y el 30 %.

- **Optimizador empleado en el entrenamiento de las capas superiores**. En la práctica, el optimizador Adam es recomendado como el algoritmo predeterminado para las tareas de aprendizaje profundo ya que su uso requiere un menor número de iteraciones para converger que la alternativa SGD y sus variantes [32]. Sin embargo, este optimizador es propenso en algunas ocasiones a no alcanzar la convergencia más óptima [34], aunque dado que la etapa de clasificación introducida tan solo se va a entrenar durante 5 iteraciones en este punto, este hecho carece de importancia. Los parámetros utilizados son los especificados por defecto en el artículo que describe este algoritmo [35] y cuya notación está vinculada a lo descrito en la Sección 2.4.2 y concretamente a la Ecuación 2.10: $\lambda = 0.001, \beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-8}$.

- **Iteraciones de entrenamiento del modelo completo**. Los tres modelos tienen especificado un entrenamiento de 100 iteraciones sobre la arquitectura completa, aunque tal y como se verá a continuación se ha considerado conveniente interrumpir el proceso de aprendizaje en el momento en el que vayan apareciendo indicios de convergencia con el propósito de no consumir recursos de forma innecesaria.

En lo que respecta al entrenamiento de la red completa, a lo largo del entrenamiento de los sucesivos modelos se han ido explorando y ajustando ciertos parámetros que se procederán a describir a continuación.

Inception-v3 e Inception-ResNet-v2

Dado que estos dos modeles han sido pre-entrenados con la base de datos ImageNet, que como se vio en la Subsección 3.3.1 difiere en gran medida del conjunto FER-2013 empleado en este proyecto, se procederá a afinar los pesos de todas las capas originales ya que es probable que incluso en los niveles inferiores la red extraiga características no relevantes para la tarea del reconocimiento de expresiones faciales.

En lo que respecta al proceso de aprendizaje en sí, para la actualización de los pesos de la red completa del modelo Inception-v3 se ha considerado oportuno utilizar el optimizador SGD con Nesterov Momentum puesto que Adam suele lograr una generalización peor para los conjuntos de validación y evaluación [34]. De igual manera, es usada una tasa de aprendizaje más pequeña que la especificada por defecto con la intención de no distorsionar los pesos iniciales demasiado rápido y más cuando el clasificador introducido está siendo entrenado a partir de una inicialización incierta. En consecuencia, son elegidos una tasa de aprendizaje de $\lambda = 10^{-4}$ y un momento lineal de $\mu = 0.9$ [61] siguiendo la notación de la Ecuación 2.4.

A pesar de todo ello, al ser los resultados del modelo Inception-v3 no del todo satisfactorios, tal y como se comprobará en la Sección 5.3, se ha tomado la decisión de entrenar un tipo de red más compleja, como es Inception-ResNet-v2, y con una tasa de aprendizaje inicial más agresiva de $\lambda = 0.001$ para optimizar los tiempos de entrenamiento y de convergencia. Sin embargo, dado que el empleo de tasas de aprendizaje altas conlleva el riesgo de estancamiento en peores valores de pérdidas, también se ha considerado conveniente ir disminuyendo la tasa de aprendizaje a la mitad cada 5 iteraciones.

Asimismo, en las dos implementaciones y para tener un control adicional sobre el proceso de entrenamiento en caso de que el modelo alcance la convergencia o se quede estancado en algún punto, se ha introducido una función que disminuye la tasa de aprendizaje a la

mitad si no se obtiene ninguna mejora del valor de pérdidas durante 5 iteraciones seguidas en el caso de Inception-v3 y de 3 en el caso de Inception-ResNet-v2. De igual manera, se ha estipulado que se detenga completamente el entrenamiento y se guarde el contexto si las pérdidas no son reducida durante 20 iteraciones seguidas durante el proceso de aprendizaje del modelo Inception-v3 y durante 10 iteraciones en el caso de la red Inception-ResNet-v2.

ResNet-50

En esta ocasión la base de datos VGGFace2 expuesta en la Subsección 3.3.2 y empleada por esta red sí que presenta una gran similitud con el conjunto FER-2013, por lo que es posible suponer que al menos los primeros niveles van a extraer características de interés para el problema del reconocimiento de emociones. Por este motivo y teniendo en cuenta las limitaciones de tamaño que presenta el conjunto FER-2013, se ha decidido dejar los pesos de los primeros 9 módulos residuales (los módulos de color azul de la Figura 5-6) sin afinar para evitar el sobreaprendizaje del modelo. El resto de bloques ha sido entrenado con normalidad y empleando, al igual que en los modelos anteriores, un optimizador SGD con Nesterov Momentum con los siguientes parámetros: $\lambda = 10^{-4}$ y $\mu = 0.9$. También es necesario señalar que se ha estipulado que esta tasa de aprendizaje, λ, se reduzca a la mitad cada vez que el desempeño del modelo, en lo que respecta a las pérdidas, no mejore durante 3 iteraciones seguidas, interrumpiéndose el entrenamiento completamente a las 10 iteraciones de no presentar avances.

Finalmente cabe destacar que tanto en este modelo como en los anteriores se hace un seguimiento del proceso de aprendizaje en tiempo real mediante la utilización de la herramienta Tensorboard ofrecida por TensorFlow y que permite exportar y representar gráficamente desde Keras ciertas métricas cuantitativas. En el caso particular de este proyecto se monitorizan las tasas de pérdidas y de acierto sobre los datos de entrenamiento y validación y la tasa de aprendizaje. Del mismo modo, también se van guardando los puntos de control del mejor modelo obtenido hasta la fecha en cada iteración del entrenamiento.

5.2.3. Despliegue en la Plataforma Google Cloud

Para realizar el entrenamiento de los modelos Inception-v3, Inception-ResNet-v2 y ResNet-50 se ha empleado la herramienta descrita en la Sección 3.2 y conocida como Cloud Datalab. El motivo detrás de esta decisión es que este servicio es precisamente el que permite un despliegue más inmediato de los modelos al emular el funcionamiento de los *notebooks* de Python y la que ofrece la GPU con las mejores características en términos de desempeño y memoria (NVIDIA Tesla P100). De hecho, el empleo de la herramienta alternativa (Cloud Machine Learning Engine) implica una reducción excesiva del tamaño de los lotes, lo que no favorece para nada el tiempo de entrenamiento, al ofrecer tan solo los servicios de la GPU NVIDIA Tesla K80, con un tamaño de memoria que es la mitad de la disponible en la NVIDIA Tesla P100.

5.3. Resultados

Los resultados de los entrenamientos, junto con las características principales de los modelos empleados y su comparación, se encuentran sintetizados en la Tabla 5.1. De igual manera, ciertas métricas extraídas a lo largo del entrenamiento se pueden encontrar en el Apéndice D. Concretamente, han sido monitorizadas las tasas de acierto y las pérdidas, resultando sus valores particularmente útiles a lo largo del proceso de aprendizaje ya que han permitido disponer de una realimentación directa del desempeño de los modelos sobre el conjunto de validación de la base de datos FER-2013, y por lo tanto de su convergencia, en cada instante del entrenamiento.

	Tasa de acierto	Duración del entrenamiento	Número de parámetros	Tamaño del modelo
Inception-v3	0.6386	5h 9m 37s (80 iteraciones)	23 873 703	192.1 MB
Inception-ResNet-v2	0.6500	1h 4m 26s (27 iteraciones)	55 857 255	449.3 MB
ResNet-50	0.7125	2h 29m 45s (40 iteraciones)	25 613 383	308.3 MB

Tabla 5.1: Comparación entre las características de los distintos modelos empleados y su desempeño sobre el conjunto de evaluación de la base de datos FER-2013.

En lo que respecta a los resultados finales, los modelos han sido examinados sobre el conjunto de evaluación, usándose este grupo exclusivamente para este propósito con el fin de obtener una serie de rendimientos que puedan ser comparados directamente con los planteados en la Sección 2.5. Asimismo y con el objetivo de evitar el sobreaprendizaje, los sistemas que han sido elegidos para la evaluación son los que corresponden precisamente con las iteraciones de menor pérdida del proceso de entrenamiento completo.

5.3.1. Inception-v3

Esta primera aproximación al problema del reconocimiento de expresiones faciales ha permitido alcanzar una tasa de acierto de 63.86 % con el modelo de la iteración 60, posterior a la cual no se ha conseguido mejorar la pérdida de 1.0664 sobre el conjunto de validación e interrumpiéndose el entrenamiento tras observarse una convergencia tras las 80 primeras iteraciones, tal y como puede observarse en la Figura D-2a del Apéndice D.

En cuanto al desempeño del modelo sobre el conjunto de evaluación particularizado para cada clase, éste se expone mediante las matrices de confusión de la Figura D-3. De estas representaciones se puede concluir que, tal y como se intuyó en un primer momento, los mejores resultados son obtenidos en las categorías con mayor representación (alegría). Por otro lado, también se puede observar que hay ciertas clases cuyas diferencias entre sí en lo que respecta a la expresión facial son más sutiles, lo cual, aunado al hecho de tener unos datos bastante limitados, da lugar a numerosas asignaciones incorrectas (entre las clases de asco e ira, entre la expresión de tristeza y la neutral, etc.).

Finalmente, este resultado obtenido con tan solo 5 horas de entrenamiento puede parecer aceptable en un primer momento dado que hasta los modelos más simples entrenados desde cero, y contando con solamente 3 módulos Inception, requieren de un tiempo de aprendizaje de más de 20 horas para alcanzar resultados semejantes [44]. Sin embargo, objetivamente y con respecto a los trabajos similares de la Sección 2.5, este desempeño es bastante pobre.

5.3.2. Inception-ResNet-v2

La implementación de este modelo surge como un intento de optimizar los resultados obtenidos inicialmente mediante el sistema Inception-v3. En esta ocasión, al emplearse unos parámetros más agresivos, el tiempo de convergencia se reduce bastante, obteniéndose una tasa de acierto de 65.00 % con el modelo entrenado con tan solo 17 iteraciones, a partir de las cuales se produce un estancamiento de las pérdidas en torno a 1.0353, lo que puede comprobarse en la Figura D-4a. A pesar de ir disminuyendo periódicamente la tasa de aprendizaje, es posible que ésta sea demasiado alta en un primer momento y que no se haya alcanzado la región de convergencia óptima, sin embargo, la diferencia en este caso sería intrascendente. Además, mediante el desarrollo de este modelo se buscaba obtener

una mejora sustancial que al no ser alcanzada ha propiciado la llegada a la conclusión de que la base de datos ImageNet sobre la que están pre-entrenados los modelos Inception-v3 e Inception-ResNet-v2, al diferir en gran medida con el objetivo tratado en este proyecto, no es la adecuada para una transferencia de aprendizaje basada en expresiones faciales. Por todo ello, y a pesar de volver a obtener unos resultados aceptables y mejorar ligeramente el desempeño de las clases menos representadas (Figura D-5) con un tiempo de entrenamiento de apenas 1 hora, se ha optado por explorar los modelos pre-entrenados con bases de datos exclusivamente de rostros.

5.3.3. ResNet-50

Esta red, cuyos pesos iniciales han sido transferidos a la API de Keras desde el modelo ResNet-50 entrenado mediante el *framework* Caffe sobre el conjunto VGGFace2, es la que con diferencia ha permitido obtener los mejores resultados sobre la fracción de evaluación de la base de datos FER-2013 con respecto a los modelos anteriores. De hecho, la tasa de acierto final alcanzada, de 71.25 %, ha superado tanto a la mejor marca obtenida en el desafío original en el que se propuso el conjunto FER-2013 [30], como a la gran parte de las publicaciones académicas de los últimos años, quedando a tan solo 4 puntos por debajo de los complejos modelos estado del arte que son el resultado de reunir y aunar múltiples redes profundas [53]. Además, cabe destacar que el desempeño humano en este contexto es del 65 ± 5 % [22], lo que evidencia la ambigüedad de esta base de datos y la dificultad de obtener sobre ella resultados aceptables.

A pesar de todo ello, lo más destacable de la tasa aquí alcanzada es que este valor es consecuencia de un entrenamiento sobre la GPU NVIDIA Tesla P100 de apenas 2 horas y media. Concretamente, el modelo utilizado para la evaluación corresponde a la iteración 30 del aprendizaje, en el que se alcanzó un mínimo de pérdidas de 0.9049, tal y como se puede comprobar en la Figura D-6a. Esta reducción de pérdidas en relación a los modelos Inception también ha permitido aumentar notablemente el número de clasificaciones correctas de las clases con menor representación, alcanzándose unas tasas superiores al 60 % en prácticamente todas las categorías, tal y como lo reflejan las matrices de confusión de la Figura D-7.

Por último y como puede observarse en la Tabla 5.1, esta arquitectura también permite disminuir ligeramente el tamaño del modelo con respecto a la red Inception-ResNet-v2, que es un parámetro que puede resultar crítico en caso de que se pretenda desplegar este sistema en un sistema empotrado o dispositivo móvil.

Capítulo 6

Extensión de la Base de Datos FER-2013

Obtenidos unos resultados más que aceptables y eficientes en lo que respecta al tiempo de entrenamiento, se ha intentado dar un paso más allá con la intención de mejorar las tasas de acierto y la calidad de la base de datos inicial mediante la implementación de una Red Generativa Antagónica de Ciclo Consecuente (CycleGAN) [77] que permita la producción artificial de imágenes y, por lo tanto, la eliminación del desequilibrio de la distribución de clases del conjunto FER-2013. En un principio tan sólo se pretende aumentar el número de datos etiquetados con la expresión facial de asco, que como se vio en la Tabla 3.2 es con diferencia el gesto con menor representación. Las imágenes a partir de las cuales se pretenderá generar esta última emoción corresponderán, dada la adaptabilidad de su naturaleza, a la clase de la expresión neutral.

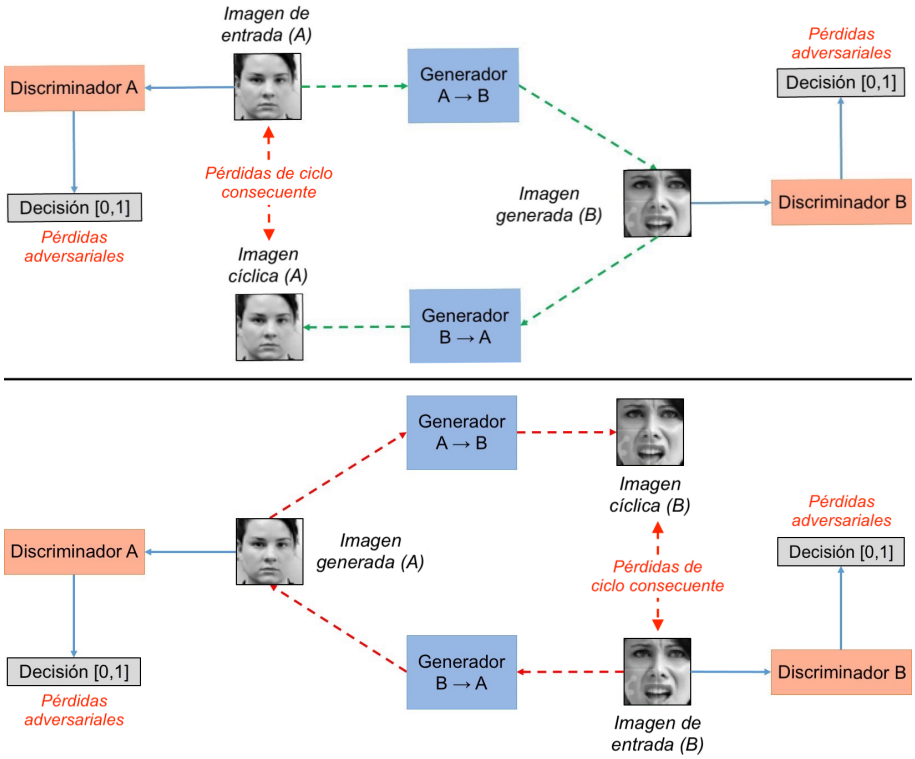
6.1. Arquitectura propuesta

La arquitectura aquí desarrollada y esquematizada en la Figura 6-1 es una adaptación de la estructura propuesta en el documento original de las redes CycleGAN y que ha demostrado unos resultados más que razonables en el área de las redes generativas de base neuronal. Asimismo, al igual que los modelos descritos anteriormente, el acondicionamiento se ha llevada a cabo íntegramente en la API de Keras y sobre TensorFlow.

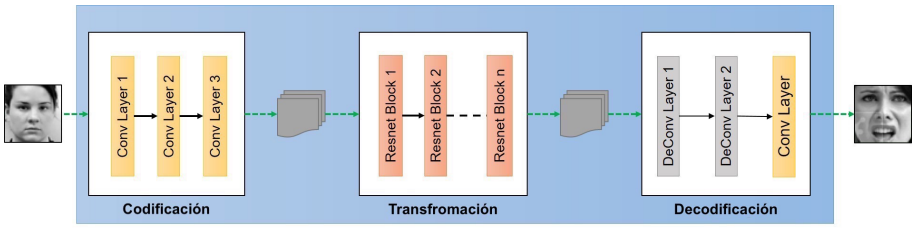
6.1.1. Red generadora

Desde un enfoque de alto nivel es posible dividir la red del generador en 3 módulos distintos dispuestos según la Figura 6-1b:

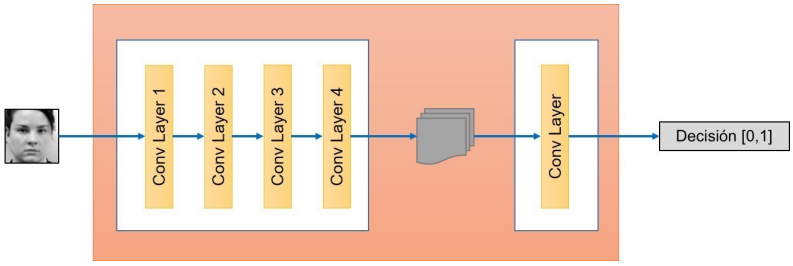
- **Módulo de codificación**. Está formado por 3 capas convolucionales que extraen las características superficiales y disminuyen el mapa de activación de la imagen inyectada.
- **Módulo de transformación**. Dado que el objetivo de los procesos de aumento de datos es la retención de ciertas características de la entrada original, como el tamaño y la forma del rostro en este caso, se hace evidente que para este tipo de transformaciones puede resultar especialmente efectivo utilizar arquitecturas residuales, que además favorecen la estabilización de las respuestas de las redes profundas. Es por todo ello que en el contexto particular y en el documento original son empleados básicamente 6 módulos residuales (Figura 5-5) constituidos por filtros convolucionales de tamaño reducido (3×3).



(a) Estructura completa de la red CycleGAN.



(b) Arquitectura simplificada del generador de la red CycleGAN.



(c) Arquitectura simplificada del discriminador de la red CycleGAN.

Figura 6-1: Arquitectura esquematizada y simplificada de la Red Generativa Antagónica de Ciclo Consecuente [3].

- **Módulo de decodificación**. Su función es exactamente la opuesta al primer módulo ya que es precisamente el que trata de reconstruir y trasladar la imagen original etiquetada con la emoción neutral al dominio objetivo que corresponde a la expresión de asco. Para tal fin se emplean dos capas deconvolucionales que recomponen el ancho y el alto de la imagen y una última capa convolucional que restaura los canales RGB de la representación inicial.

Asimismo, cabe destacar que son utilizadas en cada una de las capas la función de activación ReLU con fugas y la normalización de instancias. Esta última es una sustitución de la técnica de normalización por lotes y es empleada en este proyecto por las mejoras drásticas y demostrables que permite obtener a las redes neuronales profundas en las tareas de generación de datos [68]. Intuitivamente, este tipo de normalización consigue que la distribución de cada una de las imágenes de un determinado lote parezca gaussiana, lo que permite eliminar información específica y por lo tanto simplificar y optimizar la generación.

En cuanto a la ReLU con fugas, ésta ha reportado mejores resultados en las redes generativas que la ReLU tradicional ya que parece cubrir el espacio de color de una forma más optimizada [23]. Igualmente, esta función de activación es el intento de resolver el problema de la muerte de las ReLU convencionales, que para valores menores que cero desactivan la neurona, y por lo tanto del desvanecimiento del gradiente en esas unidades que paralizan el aprendizaje. Su expresión matemática se expone en la Ecuación 6.1.

$$f(x) = max(x, \alpha x) \qquad \forall \alpha \leq 1 \qquad (6.1)$$
$$\text{donde } \alpha \equiv \text{constante}$$

En el documento original y en el caso particular de este proyecto se toma $\alpha = 0.2$.

6.1.2. Red discriminativa

La función de esta red es básicamente tratar de predecir si la imagen introducida en su entrada es real o procede de la salida del generador. En este caso su arquitectura, representada en la Figura 6-1c, presenta una estructura simple de 5 capas convolucionales de tamaño 4×4 apiladas con una mínima alteración en el último nivel para permitir el cálculo del error de mínimos cuadrados del discriminador [42], necesario para realizar la actualización periódica de la función de pérdidas según lo especificado en la Ecuación 4.4. Esta ligera modificación consiste en la supresión de la función de activación y de la unidad de normalización de instancias.

6.2. Entrenamiento

6.2.1. Preprocesamiento de los datos

Teniendo en cuenta la diversidad de las imágenes del conjunto FER-2013, en esta ocasión tan solo se va a proceder a realizar un preprocesamiento sutil, que además permitirá disminuir el coste computacional de la red. Por ello, se realizan únicamente las siguientes transformaciones:

- **Redimensionamiento**. Dado que la entrada de la arquitectura CycleGAN ha sido ajustada para aceptar las imágenes de la base de datos FER-2013 con la resolución por defecto (48×48), tan solo se realiza una replicación de los datos en escala de gris a los tres espacios de color RGB.

- **Normalización**. El modelo de color RGB de las imágenes de la entrada se adapta al intervalo $[-1, 1]$ mediante la Ecuación 5.2 ya vista anteriormente.

- **Volteo horizontal**. Esta transformación geométrica de carácter aleatorio es la única que se va a aplicar en vista de su eficacia computacional y de enriquecimiento del conjunto de entrada.

6.2.2. Proceso de Aprendizaje

El proceso de entrenamiento de esta arquitectura ha seguido los puntos descritos en la Sección 4.3.2 y ha tenido como objetivo la reducción de las funciones de pérdidas detalladas en la Ecuación 4.5 y mostradas gráficamente en la Figura 6-1a. Asimismo, cabe destacar que los parámetros de este modelo han sido escogidos con respecto al artículo original que describe este tipo de redes generativas [77]:

- **Inicialización**. En primer lugar y dado que esta red ha sido entrenada desde cero se ha visto necesario realizar una inicialización de los pesos a partir de una distribución gaussiana con una media de 0 y una desviación estándar de 0.02.
- **Tamaño de los lotes**. Las imágenes se han alimentado en la red de una en una, lo que ha implicado también actualizar las funciones de pérdidas y los pesos en cada pasada.
- **Optimizador**. Se ha empleado el optimizador Adam con una tasa de aprendizaje de $\lambda = 0.0002$ (Ecuación 2.10), $\beta_1 = 0.5$ (Ecuación 2.6) y $\beta_2 = 0.999$ (Ecuación 2.7).
- **Peso de la función de pérdidas de ciclo consecuente**. Se le ha otrogado un valor de $\lambda = 10$ (Ecuación 4.5).
- **Iteraciones**. La red ha sido entrenada durante 7 300 iteraciones sobre los datos categorizados con las expresiones faciales de asco y neutral.

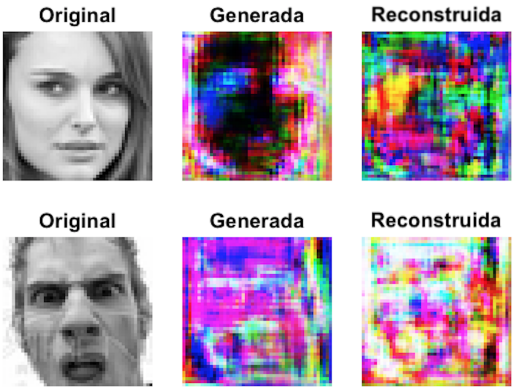
6.2.3. Despliegue en la Plataforma Google Cloud

En esta ocasión el entrenamiento en la plataforma de Google Cloud se ha realizado mediante la herramienta Cloud Machine Learning Engine descrita en la Sección 3.2 y empleando la GPU NVIDIA Tesla K80. Esto es debido a que ya no se presentan las limitaciones de memoria anteriormente existentes al utilizarse esta vez un número bastante inferior de imágenes que reúnen solo las emociones de la clases neutral y asco.

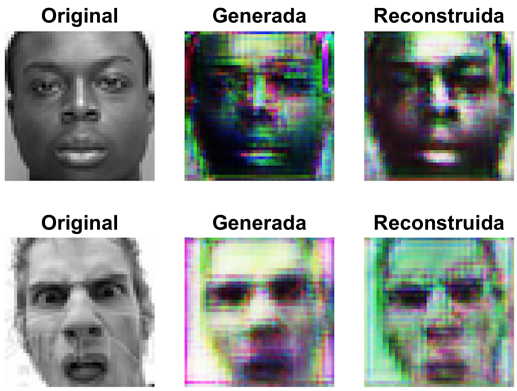
6.3. Resultados

En un primer momento y dado que en cada iteración el número de datos inyectados al modelo está limitado por la cantidad de imágenes de la clase con menor representación, en este caso la etiquetada con la expresión de asco, se estipuló un entrenamiento de 30 000 iteraciones considerando, además, los entornos comunes de implementación [46]. Sin embargo y como se puede intuir, estos modelos son muy costosos de entrenar y más en situaciones como la aquí propuesta donde se intenta establecer una correlación entre una categoría con 4 965 imágenes (neural) y una con 436 (asco). Por ello, debido a las limitaciones económicas de la prueba gratuita de Google Cloud, esta red únicamente ha podido ser entrenada durante 7 300 iteraciones, empleando para ello unas 57 horas aproximadamente.

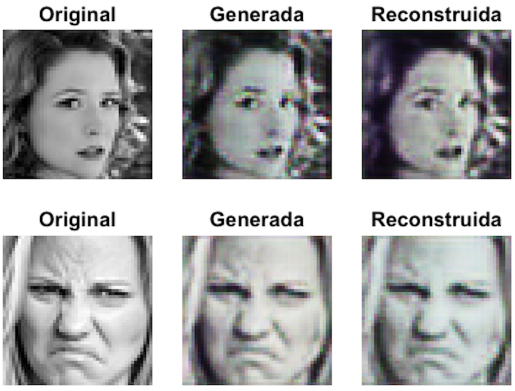
En la Figura D-8 del Apéndice D se pueden consultar las pérdidas del generador y del discriminador de este entrenamiento parcial. Desafortunadamente, éstas son muy poco intuitivas ya que la mejora de las métricas de uno de estos dos módulos implica la degradación de la respuesta del otro. El funcionamiento esperado, por lo tanto, es que tanto las pérdidas del generador como las del discriminador converjan hacia valores constantes.



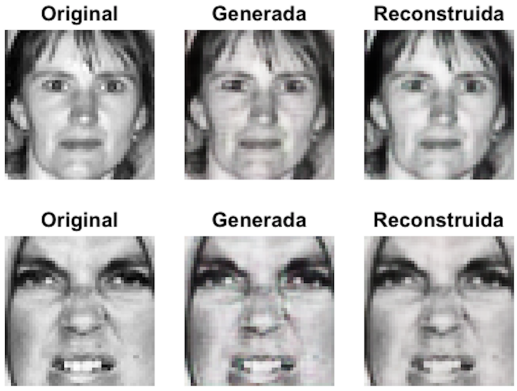
(a) Imágenes extraídas en la iteración 0.



(b) Imágenes extraídas en la iteración 500.



(c) Imágenes extraídas en la iteración 4 000.



(d) Imágenes extraídas en la iteración 7 300.

Figura 6-2: Imágenes extraídas de la red CycleGAN durante el entrenamiento.

Por otro lado, en la Figura 6-2 se puede observar la progresión de unas determinadas imágenes a lo largo de este entrenamiento incompleto. Son visibles los indicios que muestran que la red progresa convenientemente, logrando apreciarse incluso en las últimas iteraciones (Figura 6-2d) la intensificación de la expresión neutral y un cierto suavizado en la expresión de asco.

Capítulo 7

Integración del Modelo en un Sistema Empotrado

Teniendo en cuenta la calidad del modelo desarrollado en última instancia, se ha tomado la decisión de darle un uso directo mediante su integración en un sistema empotrado, y por consiguiente en un prototipo de espejo inteligente, de tal manera que pueda ser empleado para proporcionar una gran variedad de servicios.

En lo que respecta a este proceso automatizado de reconocimiento, y dado que los ordenadores captan la información de su entorno gracias a los distintos sensores que incorporan, se establecen los siguientes criterios de diseño para una identificación eficiente [52]:

- **Entrada**. Extracción de imágenes del rostro del individuo identificado mediante una cámara web en tiempo real.
- **Preprocesamiento**. Adaptación de las fotografías tomadas a los estándares del modelo empleado.
- **Razonamiento**. Estimación de la emoción por parte del sistema reconocedor a partir de las imágenes capturadas y procesadas periódicamente.
- **Salida**. Presentación de la expresión facial reconocida o toma de decisiones a partir de ésta.

Siguiendo estas pautas, se ha procedido a desarrollar una interfaz básica que al reconocer a un individuo concreto (módulo ya implementado previamente en el prototipo del espejo) le proponga una serie de actividades en las que éste tenga que imitar una emoción determinada, cuantificándose el acierto o el fallo tras un periodo estipulado, así como el tiempo de respuesta del usuario. No obstante, dada la complejidad y el tamaño del modelo ResNet-50 entrenado, el sistema empotrado empleado (Raspberry Pi 3 Modelo B), de 1 GB de RAM, no fue capaz de asignar la memoria requerida por este módulo con la configuración por defecto. Para solucionar este problema, por lo tanto, se procedió a aumentar la memoria virtual de la Raspberry mediante el espacio de intercambio *swap*, que permite encaminar y tratar la memoria desbordada en una región de almacenamiento secundario. Sin embargo, tal y como se intuía, estos accesos al disco duro ralentizan el funcionamiento del sistema y por lo tanto limitan la variedad de los servicios que puedan ofrecerse.

Por otro lado, se ha realizado una entrevista de evaluación como una acción educativa dentro del marco del proyecto europeo *mHealth tOol for parkinsOn's disease training and rehabilitation at Patient's home* (HOOP) del programa EIT Health [49]. Este proyecto estudia el uso de dispositivos móviles para complementar las terapias del rehabilitador, permitiendo al paciente la realización de ejercicios de un modo continuado, independiente y ubicuo al tiempo. Además, estas actividades son monitorizados por medio de una serie de

sensores cuyos datos son recolectados para su posterior valoración por parte del profesional correspondiente a través de una plataforma web. Por lo tanto, dentro del marco de este proyecto y a fin de identificar posibles aplicaciones del presente trabajo, se ha realizado un encuentro con la logopeda de la Asociación de Familiares de Enfermos de Alzheimer (AFA Parla). La interfaz que le ha sido presentada consistía en una interacción persona-computadora básica en la que al usuario se le va solicitando la escenificación de unas determinadas expresiones faciales. Asimismo, es ofrecida al individuo una realimentación en tiempo real de la intensidad de la emoción que está representando, que se computa como válida tras la superación de un determinado umbral, así como una representación de su propio rostro. Las ideas reportadas en esta entrevista, así como la valoración del producto por parte de esta profesional que puede consultarse en la Figura E-2 del Apéndice E, son las enumeradas a continuación:

- Dado que los ejercicios empleados con los pacientes con Parkinson comprenden comúnmente praxias y técnicas de relajación, respiración y entonación, parece más interesante enfocar los ejercicios de imitación de emociones a trastornos autistas.
- Hay que tener en gran consideración el umbral de intensidad de la expresión facial a partir del cual se produce el acierto para evitar frustrar al usuario. También se ha reportado que sería muy interesante ir ajustando el umbral con respecto a la mejoría o degradación del estado de los pacientes.
- Proposición de otras técnicas cuya implementación podría ser interesante en un futuro, como el reconocimiento de las emociones a partir del habla o la entonación.
- Reportada la utilidad del espejo inteligente y alusión a técnicas cuya implementación podría ser de gran interés: ejercicios enfocados a praxias (especialmente a movimientos faciales concretos) y ejercicios que favorezcan la coordinación fonorespiratoria (soplado de una vela artificial o realización de actividades enfocados al tono y a la intensidad de la voz).
- Es preferible desarrollar ejercicios variados, de corta duración y con un enfoque lúdico.
- El producto debería estar encaminado a ser un soporte para el profesional durante el transcurso de la terapia.

Capítulo 8

Conclusiones

8.1. Conclusiones

A lo largo de este documento se han investigado y combinado algunas técnicas de visión por computador e inteligencia artificial para clasificar las expresiones faciales. Como se ha podido comprobar, este es un problema complejo que ha requerido de un análisis profundo para obtener unos resultados, que aunque no han sido los mejores, han alcanzado unos valores significativamente competitivos teniendo en cuenta el tiempo y los medios que se han invertido para afinar el modelo ResNet-50 pre-entrenado con la base de datos VGGFace2. En este contexto, la solución aquí planteada al problema del reconocimiento de expresiones faciales puede suponer una gran oportunidad para todos aquellos que deseen implementar un identificador de emociones con una respuesta más que aceptable y dispongan de recursos computacionales limitados.

De forma concreta, en un principio en este trabajo se propusieron dos puntos de vista desde los cuales abordar la tarea de identificaciones de emociones. Por un lado se han utilizado los modelos Inception-v3 e Inception-ResNet-v2 pre-entrenados sobre el conjunto ImageNet, mientras que por otro es empleada la red ResNet-50 preentrenada con la base de datos VGGFace2. A pesar de que objetivamente los dos primeros modelos han reportado mejores resultados en amplias tareas de aprendizaje profundo, el hecho de que ResNet-50 tenga los pesos adaptados al conjunto VGGFace2 marca la diferencia en favor de este último modelo. De hecho, las desigualdades reportadas son significativas, alcanzando el sistema ResNet-50 una precisión de 71.25 % sobre el conjunto de evaluación de la base de datos FER-2013, una tasa bastante superior a los 65.00 % y 63.86 % de las arquitecturas Inception-ResNet-v2 e Inception-v3 respectivamente. Esto evidencia la inmensa importancia que tiene la naturaleza de las imágenes empleadas en los modelos pre-entrenados. En realidad, el número y la calidad de los datos son fundamentales para obtener un buen desempeño, más incluso que el diseño de la propia arquitectura en algunas ocasiones, tal y como se ha podido comprobar a lo largo de este proyecto. Precisamente por este motivo es por el cual se ha decidido explorar la generación artificial de imágenes mediante las redes generativas antagónicas. Sin embargo, puesto que éstas son muy costosas de entrenar, finalmente no se ha podido llegar a unos resultados concluyentes al realizarse tan solo un entrenamiento parcial. A pesar de ello, hay numerosas evidencias empíricas que muestran que la aplicación de éstas técnicas de generación de datos pueden aumentar entre el 5 % y el 10 % el desempeño de los modelos iniciales [78], lo que en nuestro caso supondría la superación del estado del arte actual.

En última instancia también se ha logrado implementar un modelo de reconocimiento de expresiones faciales en tiempo real en un sistema empotrado, explorándose los servicios que este desarrollo sería capaz de ofrecer desde una perspectiva biomédica.

Finalmente, hace falta añadir que el código fuente de este trabajo está disponible,

junto con los modelos entrenados, los resultados y las instrucciones para replicarlos, en el siguiente repositorio: `https://github.com/ivadym/FER`.

8.2. Líneas Futuras

Dada la multidisciplinariedad de este proyecto, hay una gran variedad de vías por las que seguir las investigaciones aquí plasmadas. De esta forma, por un lado sería interesante continuar con el objetivo inicial de diseño de un sistema de reconocimiento de expresiones faciales óptimo. En este contexto, podría resultar beneficioso realizar un entrenamiento de la red ResNet-50 completa y con una tasa de aprendizaje aún menor con el objetivo de obtener una mejor convergencia. Asimismo, también valdría la pena intentar terminar el entrenamiento incompleto de las redes CycleGAN, generar nuevas imágenes para la categoría correspondiente a la expresión de asco y comprobar si las mejoras que se obtienen son significativas. En caso de que lo fueran, es probable que la generación de más datos de las clases menos representadas pudiera dar lugar a que se superase el estado del arte actual.

Por otro lado y más en un contexto práctico, se podría enfocar la continuación de este proyecto a la obtención de un sistema de reconocimiento más heterogéneo, combinando varias bases de datos de expresiones faciales distintas. Esto tendría como objetivo mejorar el desempeño del reconocimiento en situaciones reales, en lugar de centrarse en obtener las mayores tasas sobre una base de datos determinada.

Esta última perspectiva es precisamente la que permitiría obtener un mejor funcionamiento del sistema implantado en el prototipo de espejo inteligente. Por ello, ésta es otra de las alternativas en las que se pueden enfocar los trabajos futuros: optimizar y añadir funcionalidades al espejo inteligente dentro del proyecto HOOP anteriormente descrito.

Apéndice A

Impactos del Trabajo Fin de Grado

El empleo de las computadoras para la realización de múltiples y variadas tareas aumenta constantemente en la sociedad, por lo que proporcionar a estas máquinas la capacidad de una percepción emocional puede suponer un nuevo punto de inflexión en el campo de inteligencia artificial. De hecho, diseñar un sistema que sea capaz de reconocer fielmente la expresión facial o el estado de ánimo de una persona podría permitir que las computadoras realizaran una gran variedad de nuevas y complejas tareas que en un principio implicaban la necesidad de disponer de una gran comprensión del entorno y de la situación. Algunos ejemplos de estas labores podrían ser la ocupación del cuidado de las personas de la tercera edad, la participación en diversos ejercicios de rehabilitación o simplemente la monitorización de los estados anímicos.

Desde otro punto de vista, al ser las expresiones faciales un reflejo del estado interno de una persona, esta información resulta especialmente valiosa para la obtención de una realimentación directa e instantánea ante un estímulo. Esta cuestión es precisamente la que es cada vez más explotada por las distintas empresas comerciales que buscan mejorar las ventas y los productos que ofrecen.

De hecho, esta aspiración de reunir cada vez una cantidad mayor de datos a partir de los cuales extraer conclusiones es lo que ha detonado y favorecido el espectacular avance de las técnicas de aprendizaje automático y profundo. Sin embargo, estos métodos requieren de una inmensa cantidad de energía, por lo que, a pesar de que el consumo íntegro de Google proviene de fuentes renovables, los impactos medioambientales son evidentes.

Tal y como puede observarse, el reconocimiento de emociones es un área que tiene un gran potencial para ser explotado por múltiples campos heterogéneos, como son los académicos, los sociales, los cínicos o los económicos y comerciales. En el caso particular de este proyecto el área en el que se tiene un mayor impacto es la académica o investigadora, ya que el objetivo principal de este escrito ha sido el diseño de un sistema que sea especialmente competitivo en las tareas del reconocimiento de emociones sobre una base de datos normalizada. En menor medida y en el contexto del estudio de los servicios que podría ofrecer el sistema desarrollado, el módulo, de implantarse en el espejo inteligente final, podría desempeñar una labor de soporte a los profesionales de los sectores médicos. Esto último, por curso natural, también implicaría un impacto económico para el proyecto europeo dentro del cual se engloba la parte de implementación de este trabajo.

Apéndice B

Presupuesto económico del Trabajo Fin de Grado

MANO DE OBRA	Horas/año	Salario anual	Dedicación (horas)	TOTAL
Ingeniero recién graduado	1470	22.000,00 €	450	6.734,69 €
Doctor en Ingeniería	1470	35.000,00 €	50	1.190,48 €
				7.925,17 €

RECURSOS MATERIALES	Precio de compra	Uso (meses)	Amortización (años)	TOTAL
Ordenador personal	1.500,00 €	7	5	175,00 €
Raspberyy PI 3 B	36,19 €	7	2	10,56 €
Accesorios Raspberry PI 3 B	17,85 €	7	2	5,21 €
				190,76 €

MATERIAL FUNGIBLE	TOTAL
Material de oficina	55,00 €
	55,00 €

RECURSOS SOFTWARE	Modelo	Horas de entrenamiento	Precio/hora	TOTAL
Google Cloud Plataform	Inception-v3	5	1,07 €	5,35 €
	Inception-ResNet-v2	1	1,07 €	1,07 €
	ResNet-50	2,5	1,07 €	2,68 €
	CycleGAN	57	0,98 €	55,86 €
				64,96 €

CONTRATOS ADMINISTRACIONES PÚBLICAS		TOTAL
Gastos generales	15%	1227,13301
Beneficio industrial	6%	564,48 €
		1.791,61 €

SUBTOTAL	10.027,50 €
IVA	21%
PRESUPUESTO TOTAL	**12.133,28 €**

Apéndice C

Algoritmos

Algoritmo 1 Algoritmo de la retropropagación

function BACKPROPAGATION(datos, red)
  variables: pesos ($\omega_{i,j}$), tasa de aprendizaje (λ)
  for each $\omega_{i,j}$ **in** red **do** ▷Inicialización de parámetros
    $\omega_{i,j} \leftarrow$ pequeño valor aleatorio
  repeat
    for each x **in** datos **do** ▷Propagación de entradas/Obtención de salidas
      for each neurona j **in** capa i = 0 **do**
        $\alpha_{0,j} \leftarrow x$ ▷Asignación de las entradas
      for capa i = 1 **to** capa i = k **do**
        for each neurona j **in** capa i **do**
          $s_{i,j} \leftarrow \sum_j \omega_{i,j} \cdot \alpha_{i-1,j}$
          $\alpha_{i,j} \leftarrow f(s_{i,j})$ ▷Aplicación de la función de activación
      for each neurona j **in** capa i = k **do** ▷Propagación hacia atrás
        $\Delta_{k,j} \leftarrow f'(s_{k,j}) \cdot (y_{k,j} - \alpha_{k,j})$ [1]
      for capa i = k-1 **to** capa i = 0 **do**
        for each neurona j **in** capa i **do**
          $\Delta_{i,j} \leftarrow f'(s_{i,j}) \cdot \sum_j \omega_{i,j} \cdot \Delta_{i+1,j}$ [1]
      for each $\omega_{i,j}$ **in** red **do** ▷Actualización de los pesos
        $\omega_{i,j} \leftarrow \omega_{i,j} - \lambda \cdot \alpha_{i,j} \cdot \Delta_{i,j}$ ▷Descenso Estocástico del Gradiente
  until todos los datos se clasifican correctamente **or** satisfacción de otro criterio
  return red

[1] Conversión de la derivada de error con respecto a la salida en la derivada de error con respecto a la entrada multiplicándola por el gradiente de $f(s_{i,j})$.

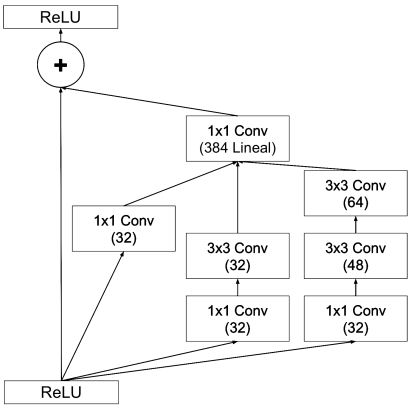
Algoritmo 2 Algoritmo de la técnica de normalización por lotes [28]

function BatchNormalization(datos, red)
  variables: γ, β [1] ▷Parámetros por aprender
  repeat
    for each lote $\mathfrak{B} = \{x_{1...m}\}$ **in** datos **do**
      $\mu_{\mathfrak{B}} \leftarrow \frac{1}{m} \cdot \sum_{i=1}^{m} x_i$ ▷Promedio del lote
      $\sigma^2_{\mathfrak{B}} \leftarrow \frac{1}{m} \cdot \sum_{i=1}^{m} (x_i - \mu_{\mathfrak{B}})^2$ ▷Varianza del lote
      $\hat{x}_i \leftarrow \frac{x_i - \mu_{\mathfrak{B}}}{\sqrt{\sigma^2_{\mathfrak{B}} + \epsilon}}$ ▷Normalización
      $y_i = \gamma \cdot \hat{x}_i + \beta$ ▷Escalado y desplazamiento
  until todos los lotes de datos normalizados
  return y

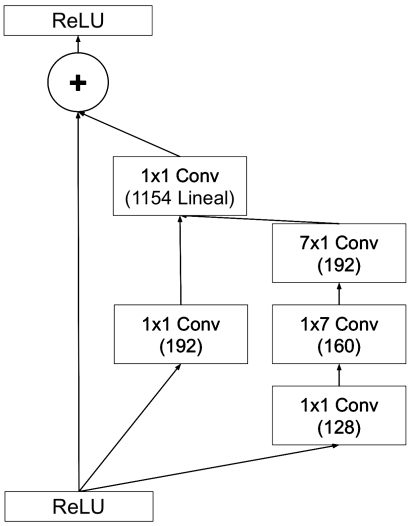
[1] La tarea de los parámetros γ y β es la de mantener la media y la varianza a unos valores preestablecidos.

Apéndice D

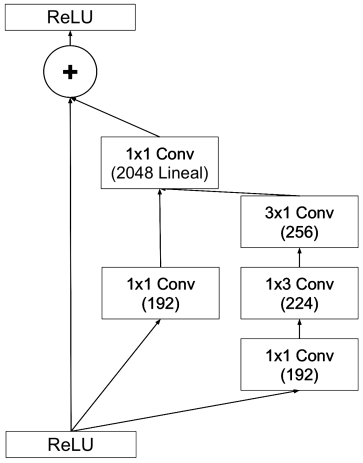
Figuras



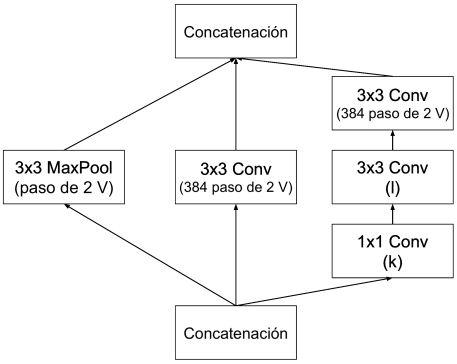
(a) Bloque Inception-ResNet-A.



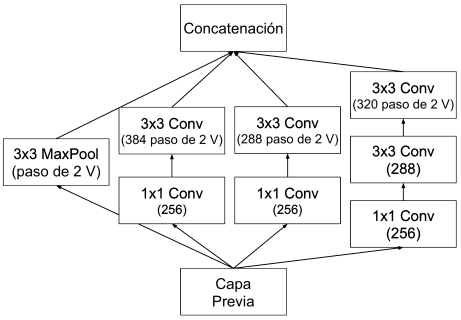
(b) Bloque Inception-ResNet-B.



(c) Bloque Inception-ResNet-C.

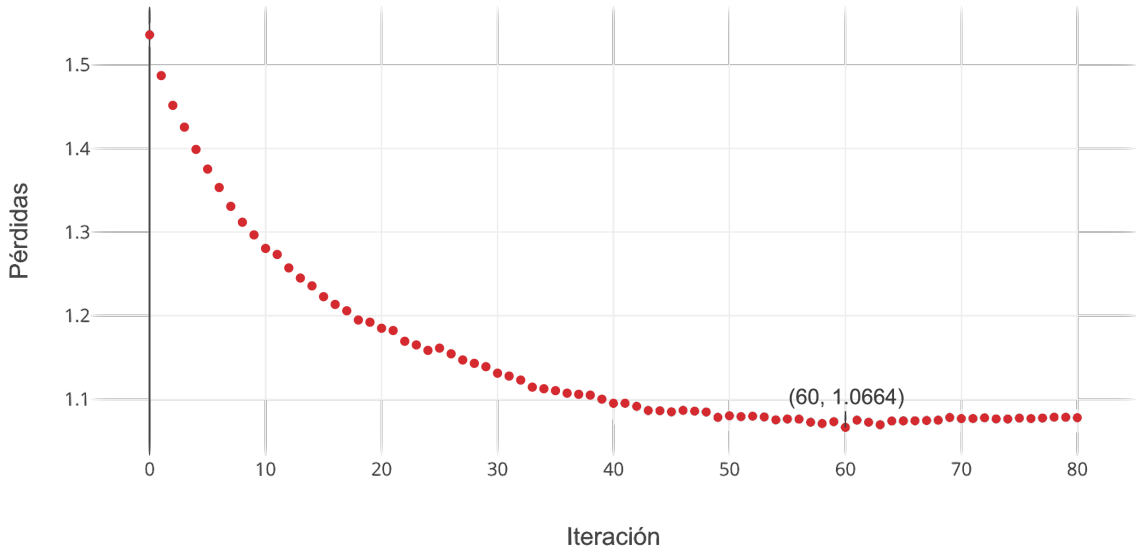


(d) Bloque Reduction-A.

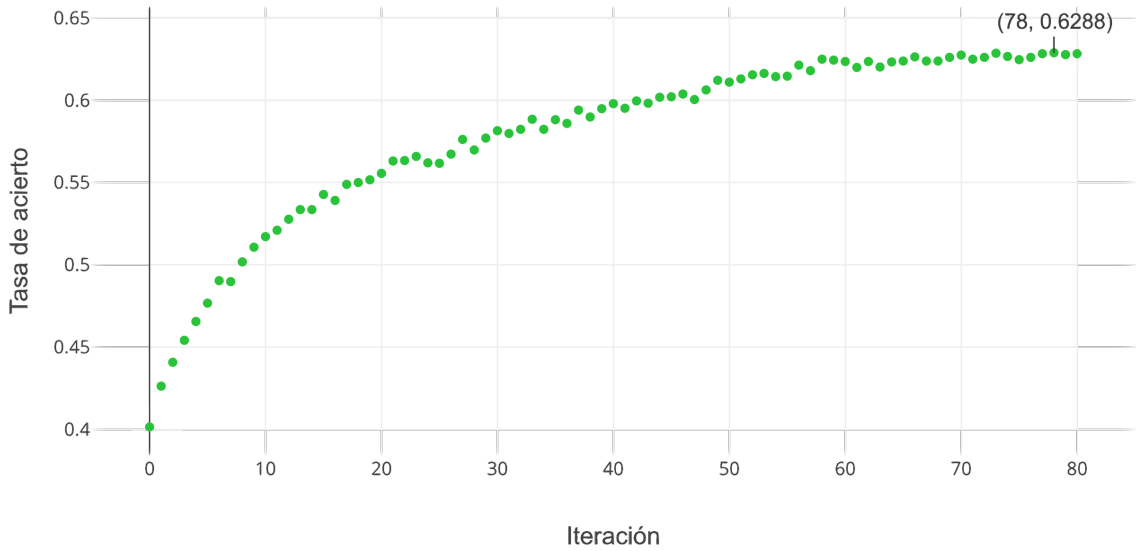


(e) Bloque Reduction-B.

Figura D-1: Módulos empleados en la arquitectura Inception-ResNet-v2 [62].

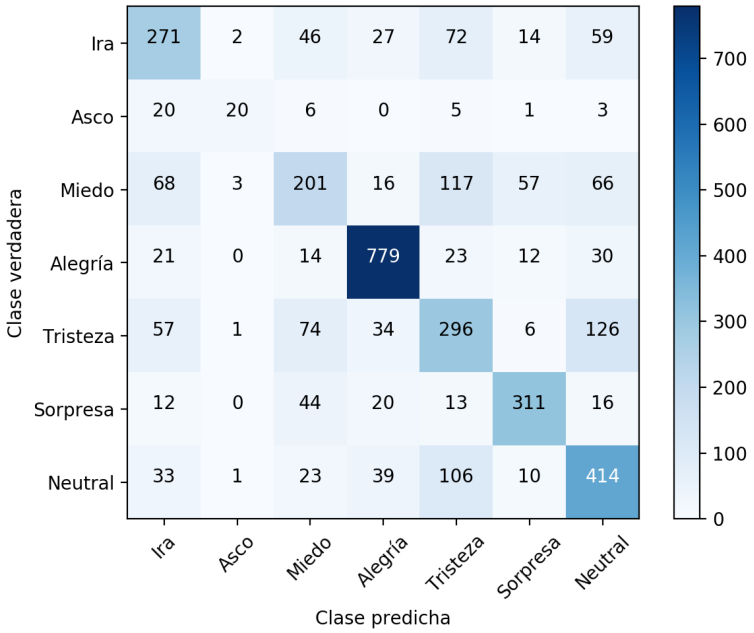


(a) Pérdidas calculadas a lo largo del entrenamiento del modelo Inception-v3.

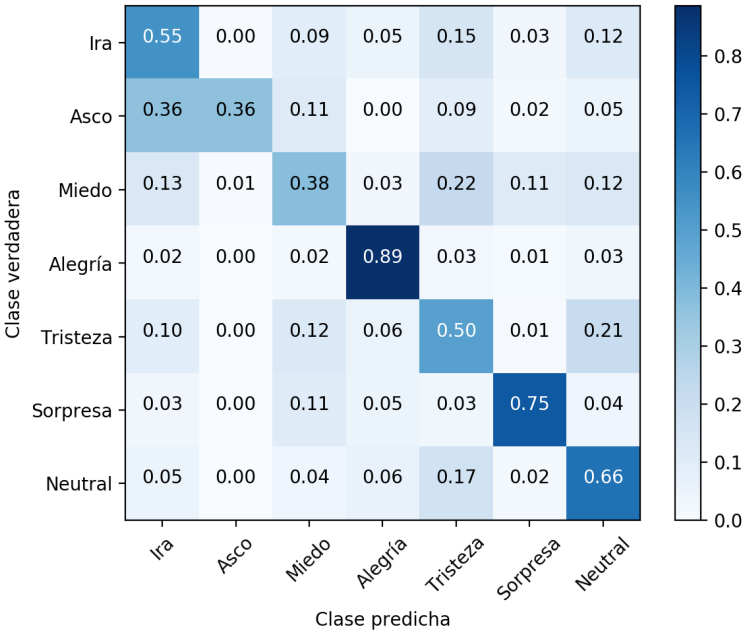


(b) Tasas de acierto calculadas a lo largo del entrenamiento del modelo Inception-v3.

Figura D-2: Métricas calculadas a lo largo del entrenamiento del modelo Inception-v3 sobre el conjunto de validación de la base de datos FER-2013.

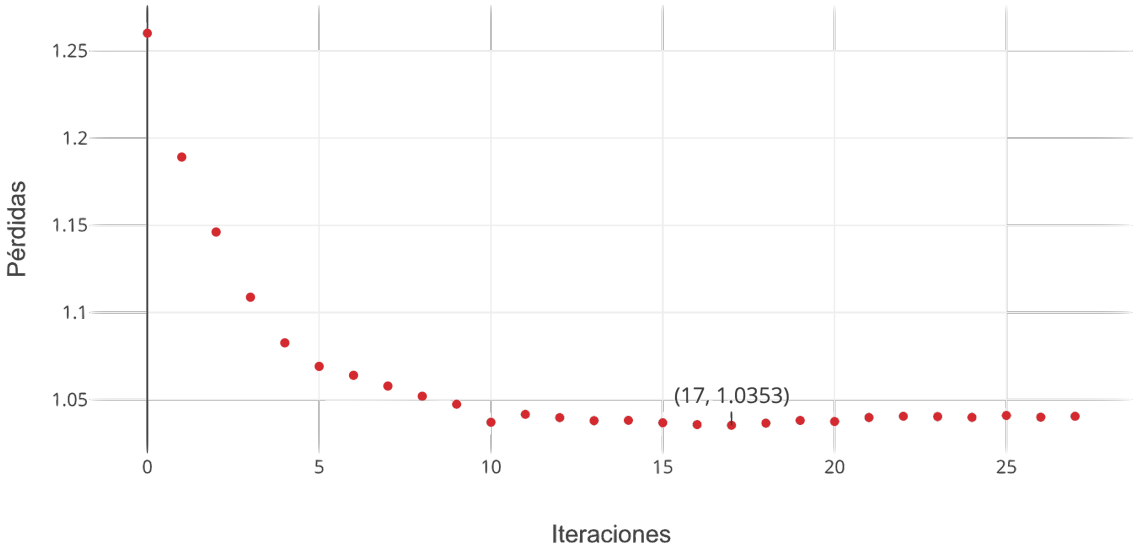


(a) Matriz de confusión del modelo Inception-v3.

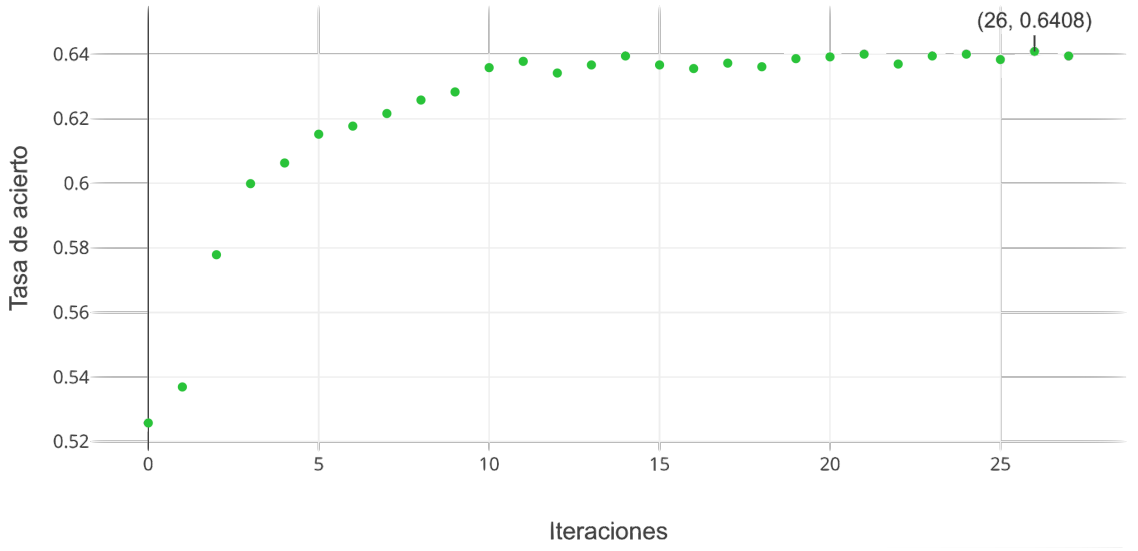


(b) Matriz de confusión normalizada del modelo Inception-v3.

Figura D-3: Matrices de confusión del conjunto de evaluación de la base de datos FER-2013 estimadas sobre el modelo entrenado Inception-v3.

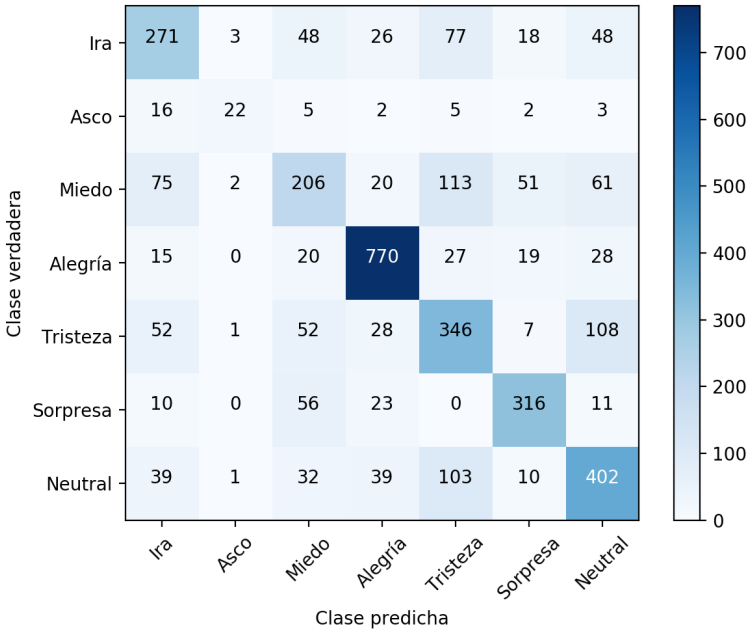


(a) Pérdidas calculadas a lo largo del entrenamiento del modelo Inception-ResNet-v2.

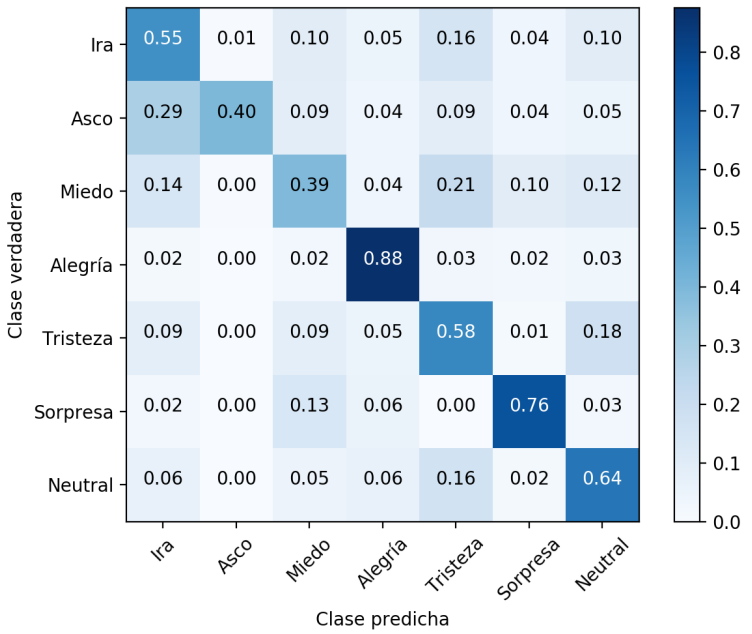


(b) Tasas de acierto calculadas a lo largo del entrenamiento del modelo Inception-ResNet-v2.

Figura D-4: Métricas calculadas a lo largo del entrenamiento del modelo Inception-ResNet-v2 sobre el conjunto de validación de la base de datos FER-2013.

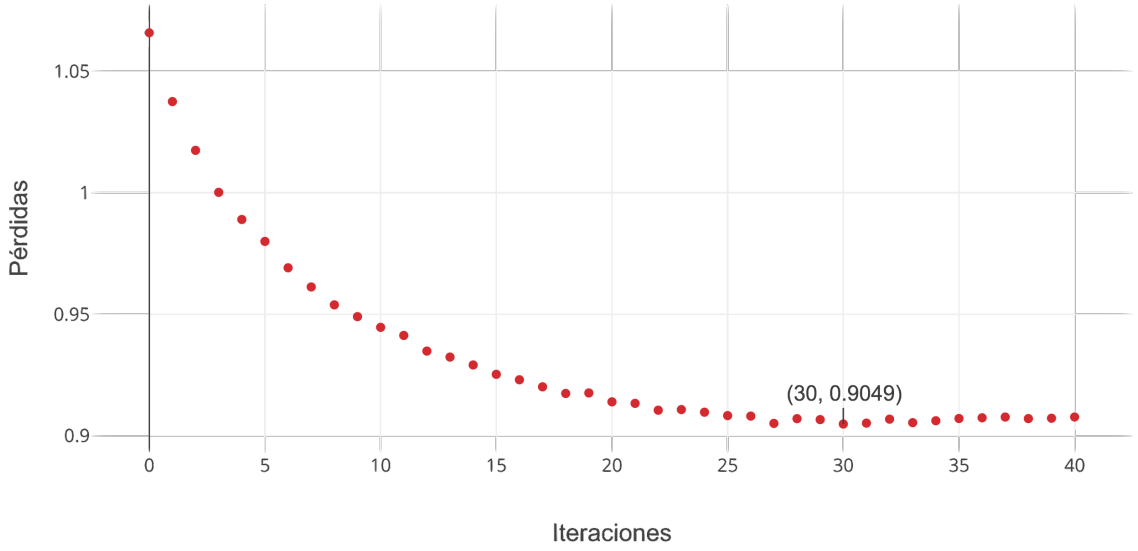


(a) Matriz de confusión del modelo Inception-ResNet-v2.

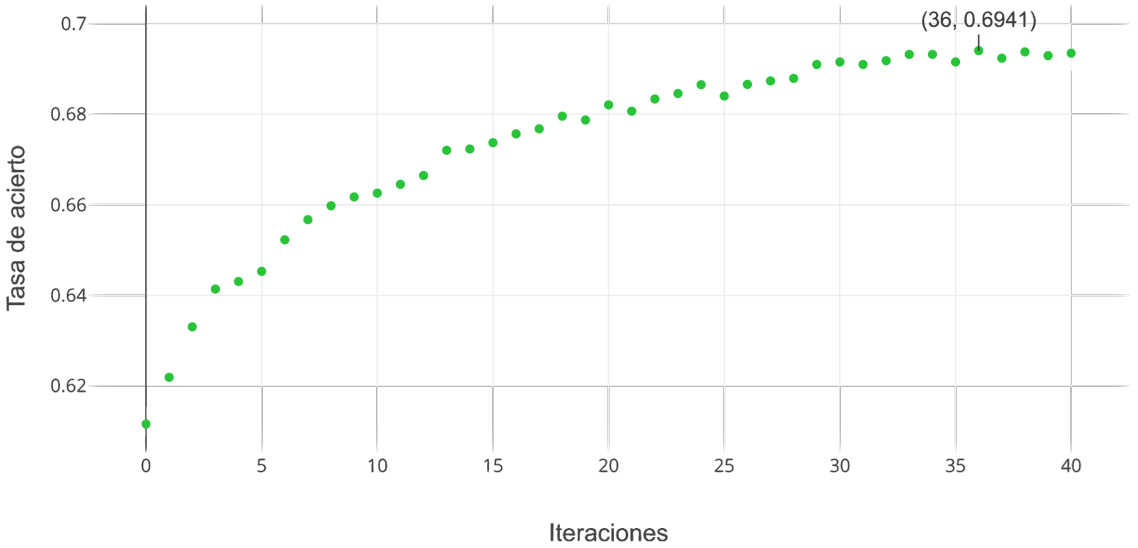


(b) Matriz de confusión normalizada del modelo Inception-ResNet-v2.

Figura D-5: Matrices de confusión del conjunto de evaluación de la base de datos FER-2013 estimadas sobre el modelo entrenado Inception-ResNet-v2.

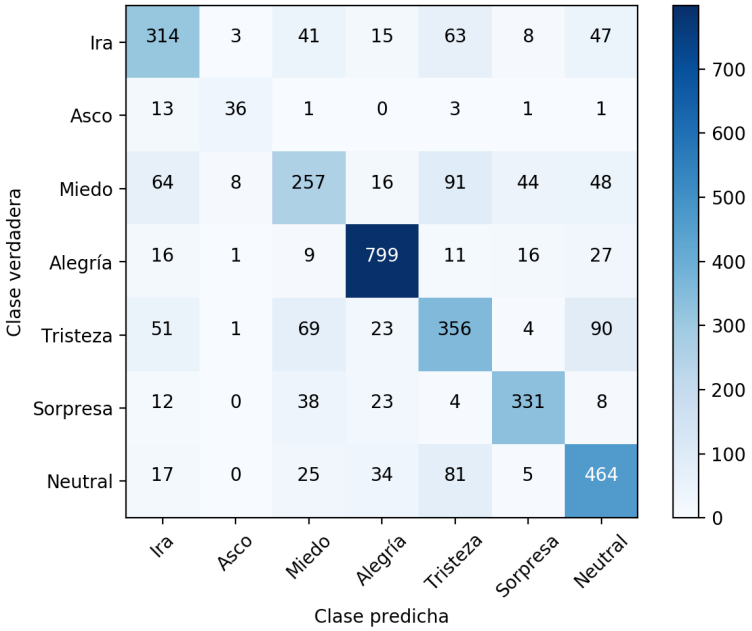


(a) Pérdidas calculadas a lo largo del entrenamiento del modelo ResNet-50.

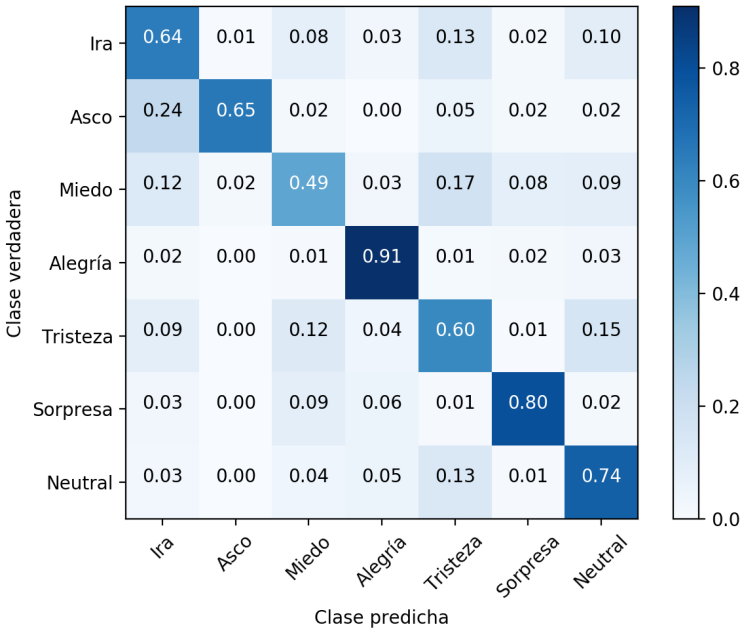


(b) Tasas de acierto calculadas a lo largo del entrenamiento del modelo ResNet-50.

Figura D-6: Métricas calculadas a lo largo del entrenamiento del modelo ResNet-50 sobre el conjunto de validación de la base de datos FER-2013.



(a) Matriz de confusión del modelo ResNet-50.



(b) Matriz de confusión normalizada del modelo ResNet-50.

Figura D-7: Matrices de confusión del conjunto de evaluación de la base de datos FER-2013 estimadas sobre el modelo entrenado ResNet-50.

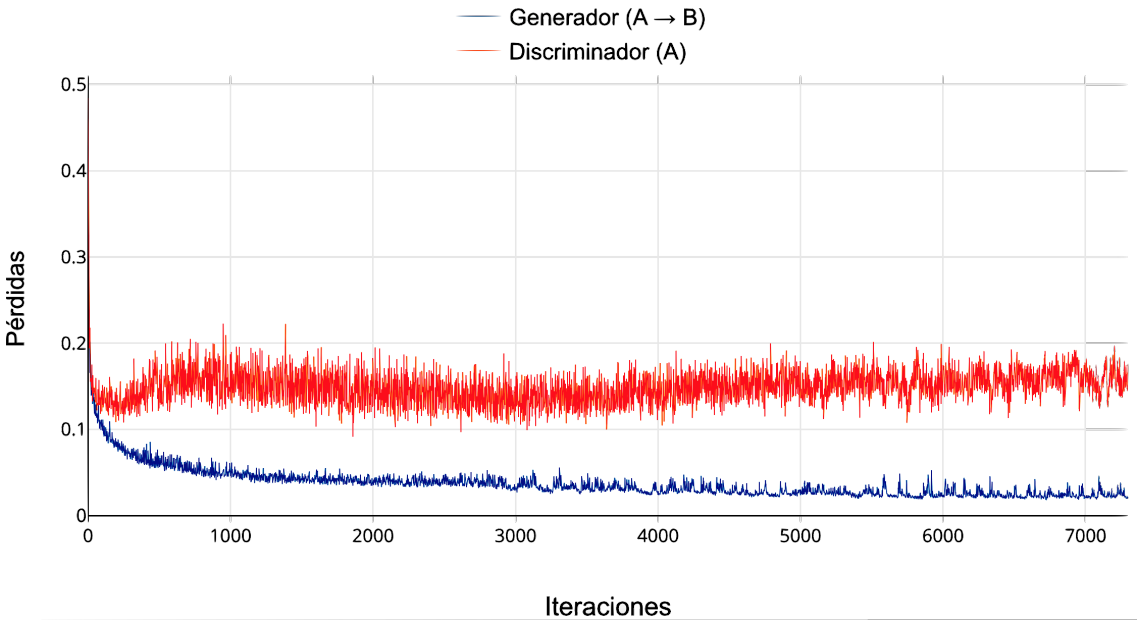
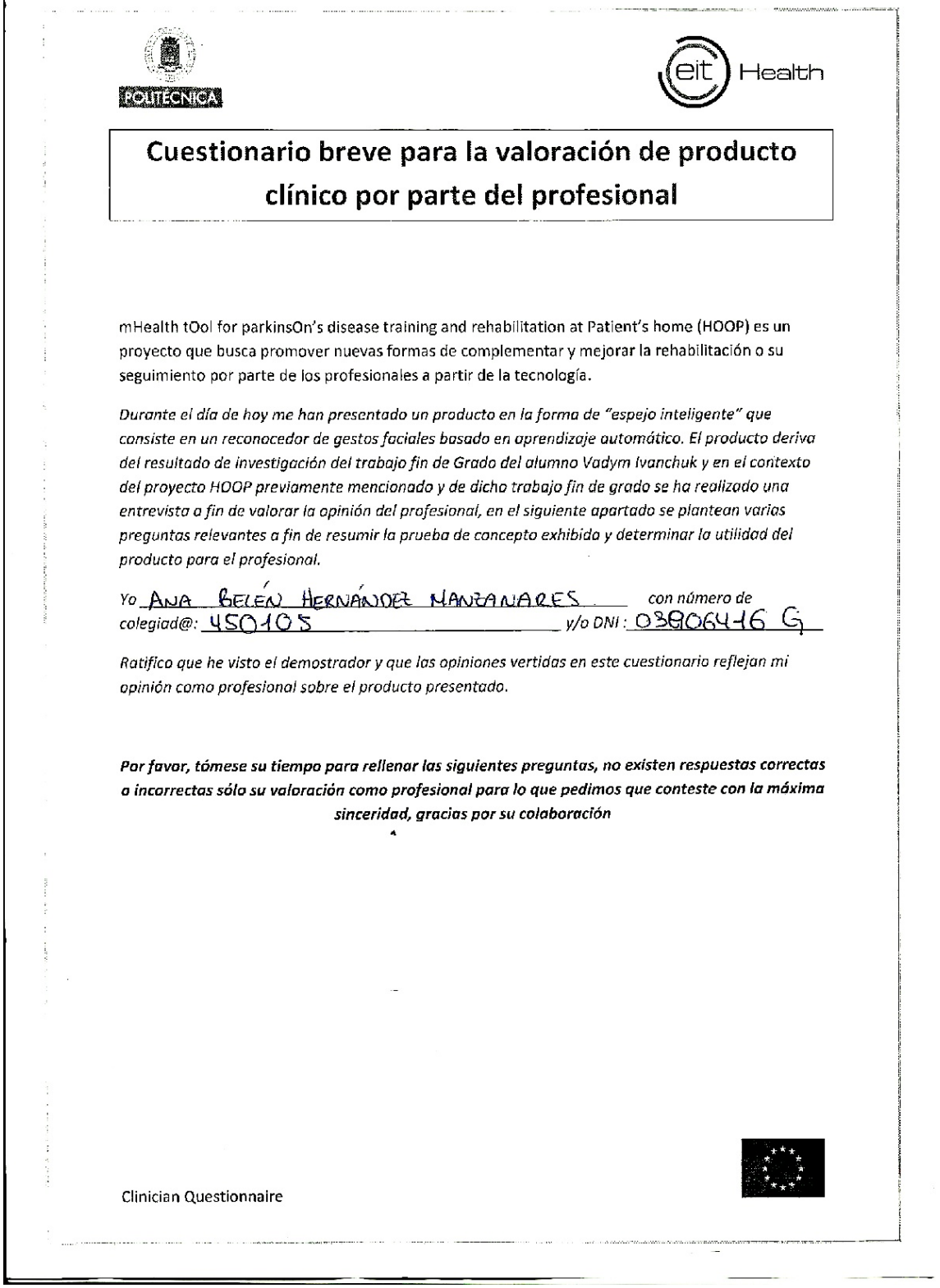


Figura D-8: Pérdidas del generador $A \rightarrow B$ y del discriminador A de la red CycleGAN (Figura 6-1) a lo largo de su entrenamiento.

Apéndice E

Miscelánea



POLITÉCNICA

eit Health

Cuestionario breve para la valoración de producto clínico por parte del profesional

mHealth tOol for parkinsOn's disease training and rehabilitation at Patient's home (HOOP) es un proyecto que busca promover nuevas formas de complementar y mejorar la rehabilitación o su seguimiento por parte de los profesionales a partir de la tecnología.

Durante el día de hoy me han presentado un producto en la forma de "espejo inteligente" que consiste en un reconocedor de gestos faciales basado en aprendizaje automático. El producto deriva del resultado de investigación del trabajo fin de Grado del alumno Vadym Ivanchuk y en el contexto del proyecto HOOP previamente mencionado y de dicho trabajo fin de grado se ha realizado una entrevista a fin de valorar la opinión del profesional, en el siguiente apartado se plantean varias preguntas relevantes a fin de resumir la prueba de concepto exhibida y determinar la utilidad del producto para el profesional.

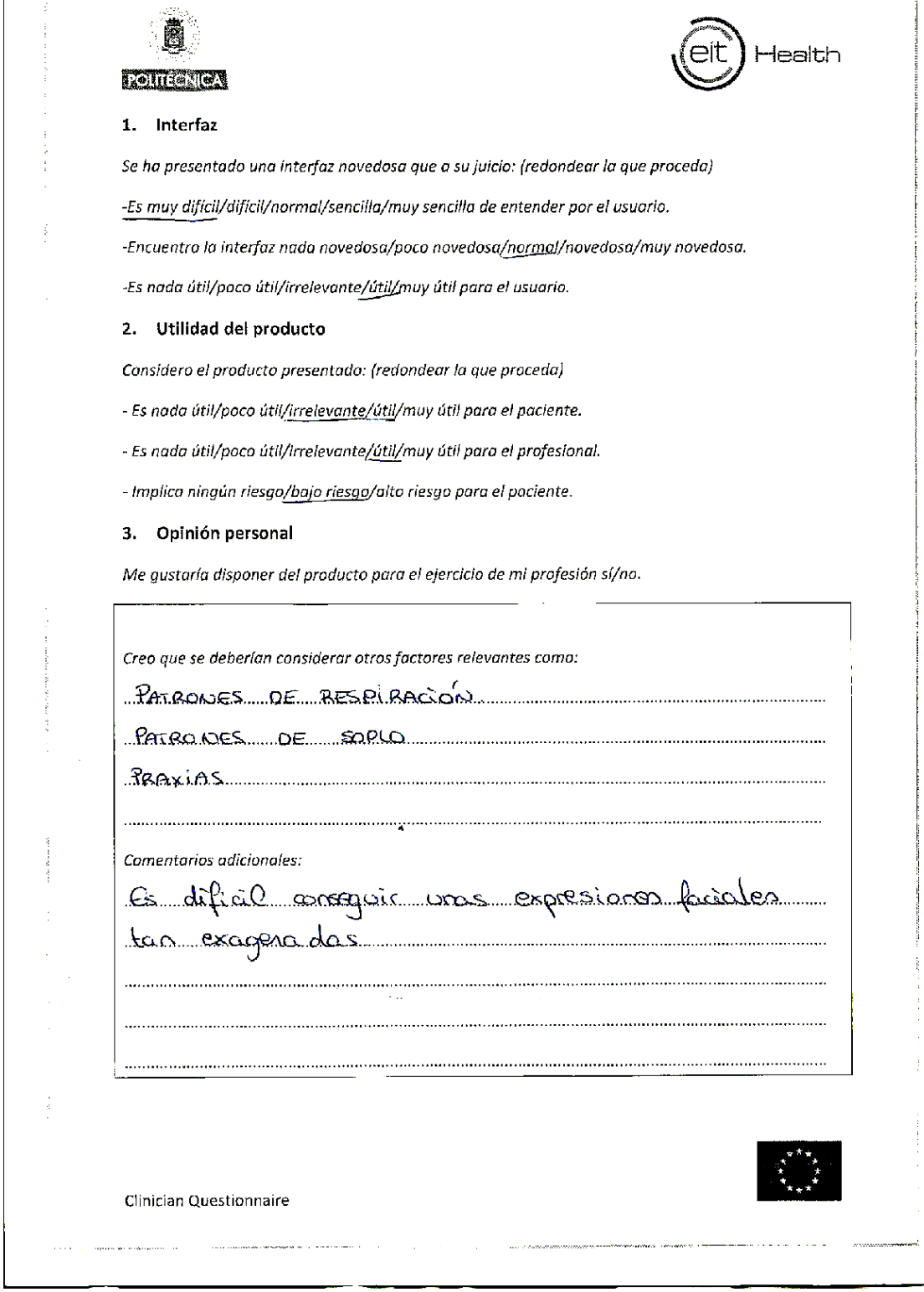
Yo ANA BELÉN HERNÁNDEZ MANZANARES *con número de colegiad@:* 450105 *y/o DNI :* 03906416 G

Ratifico que he visto el demostrador y que las opiniones vertidas en este cuestionario reflejan mi opinión como profesional sobre el producto presentado.

Por favor, tómese su tiempo para rellenar las siguientes preguntas, no existen respuestas correctas o incorrectas sólo su valoración como profesional para lo que pedimos que conteste con la máxima sinceridad, gracias por su colaboración

Clinician Questionnaire

Figura E-1: Informe reportado por la profesional de la AFA Parla.



POLITÉCNICA

eit Health

1. Interfaz

Se ha presentado una interfaz novedosa que a su juicio: (redondear la que proceda)

-Es muy difícil/difícil/normal/sencilla/muy sencilla de entender por el usuario.

-Encuentro la interfaz nada novedosa/poco novedosa/normal/novedosa/muy novedosa.

-Es nada útil/poco útil/irrelevante/útil/muy útil para el usuario.

2. Utilidad del producto

Considero el producto presentado: (redondear la que proceda)

- Es nada útil/poco útil/irrelevante/útil/muy útil para el paciente.

- Es nada útil/poco útil/irrelevante/útil/muy útil para el profesional.

- Implica ningún riesgo/bajo riesgo/alto riesgo para el paciente.

3. Opinión personal

Me gustaría disponer del producto para el ejercicio de mi profesión sí/no.

Creo que se deberían considerar otros factores relevantes como:

PATRONES DE RESPIRACIÓN

PATRONES DE SOPLO

PRAXIAS

Comentarios adicionales:

Es difícil conseguir unas expresiones faciales tan exageradas

Clinician Questionnaire

Figura E-2: Informe reportado por la profesional de la AFA Parla.

Bibliografía

[1] Saleh Albelwi and Ausif Mahmood. A framework for designing the architectures of deep convolutional neural networks. *Entropy*, 2017.

[2] Octavio Arriaga, Matias Valdenegro-Toro, and Paul Plöger. Real-time convolutional neural networks for emotion and gender classification. *CoRR*, 2017.

[3] Hardik Bansal and Archit Rathore. Understanding and implementing cyclegan in tensorflow. `https://hardikbansal.github.io/CycleGANBlog/`, 2017.

[4] Christoph Bartneck, Michael J. Lyons, and Martin Saerbeck. The relationship between emotion models and artificial intelligence. *CoRR*, 2017.

[5] Google AI Blog. Improving inception and image classification in tensorflow. `https://ai.googleblog.com/2016/08/improving-inception-and-image.html`, 2016.

[6] NVIDIA Developer Blog. Nvidia and ibm cloud support imagenet large scale visual recognition challenge. `https://devblogs.nvidia.com/nvidia-ibm-cloud-support-imagenet-large-scale-visual-recognition-challenge/`, 2015.

[7] Y-Lan Boureau, Jean Ponce, and Yann Lecun. A theoretical analysis of feature pooling in visual recognition. *ICML 2010 - Proceedings, 27th International Conference on Machine Learning*, 2010.

[8] Alfredo Canziani, Adam Paszke, and Eugenio Culurciello. An analysis of deep neural network models for practical applications. *CoRR*, 2016.

[9] Qiong Cao, Li Shen, Weidi Xie, Omkar M. Parkhi, and Andrew Zisserma. Vggface2: A dataset for recognising faces across pose and age. *CoRR*, 2017.

[10] François Chollet. Xception: Deep learning with depthwise separable convolutions. *CoRR*, 2016.

[11] Wikipedia contributors. Kernel (image processing) — Wikipedia, the free encyclopedia. `https://en.wikipedia.org/w/index.php?title=Kernel_(image_processing)&oldid=831652789`, 2018.

[12] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. `http://www.image-net.org/`, 2009.

[13] Li Deng and Dong Yu. *Deep Learning: Methods and Applications*. NOW Publishers, 2014.

[14] Mandar Dixit, Roland Kwitt, Marc Niethammer, and Nuno Vasconcelos. AGA: attribute guided augmentation. *CoRR*, 2016.

[15] The Economist. From not working to neural networking. https://www.economist.com/news/special-report/21700756-artificial-intelligence-boom-based-old-idea-modern-twist-not, 2016.

[16] Paul Ekman, Wallace V. Friesen, Phoebe Ellsworth, Arnold P. Goldstein, and Leonard Krasner. *Emotion in the Human Face*. Pergamon Press, 1972.

[17] Paul Ekman, Wallace V. Friesen, and Joseph C. Hager. *Facial Action Coding System: The Manual on CD ROM*. A Human Face, 2002.

[18] Robert J. Emmerling, Vinod K. Shanwal, and Manas K. Mandal. *Emotional Intelligence: Theoretical and Cultural Perspectives*. Nova Science Publishers, 2008.

[19] Andries P Engelbrecht. *Computational Intelligence: An Introduction*. John Wiley & Sons, 2007.

[20] Kunihiko Fukushima. A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 1980.

[21] Daniel Goleman. *Emotional intelligence: Why it can matter more than IQ*. Bantam Books, 1995.

[22] Ian J. Goodfellow et al. Challenges in representation learning: A report on three machine learning contests. *ArXiv e-prints*, 2013.

[23] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *CoRR*, 2014.

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[25] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 2012.

[26] David Hubel and Torsten Wiesel. Receptive fields and functional architecture of monkey striate cortex. *The Journal of Physiology*, 1968.

[27] Rishabh Kumar Jain (Intel), Rajeswari Ponnuru (Intel), Ajit Kumar P. (Intel), and Ravi Keron N. (Intel). Cifar-10 classification using intel optimization for tensorflow. https://software.intel.com/en-us/articles/cifar-10-classification-using-intel-optimization-for-tensorflow, 2017.

[28] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, 2017.

[29] Teijiro Isokawa, Haruhiko Nishimura, and Nobuyuki Matsui. Quaternionic multilayer perceptron with local analyticity. *Information*, 2012.

[30] Kaggle. Challenges in representation learning: Facial expression recognition challenge. https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge, 2013.

[31] Kaggle. Rectified linear units (relu) in deep learning. https://www.kaggle.com/dansbecker/rectified-linear-units-relu-in-deep-learning/notebook, 2018.

[32] Andrej Karpathy. Stanford university cs231n: Convolutional neural networks for visual recognition. http://cs231n.github.io/, 2018.

[33] KDnuggets. Understanding deep convolutional neural networks with a practical use-case in tensorflow and keras. https://www.kdnuggets.com/2017/11/understanding-deep-convolutional-neural-networks-tensorflow-keras.html, 2017.

[34] Nitish Shirish Keskar and Richard Socher. Improving generalization performance by switching from adam to SGD. *CoRR*, 2017.

[35] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, 2014.

[36] Alex Krizhevsky. Learning multiple layers of features from tiny images. *Technical report, University of Toronto*, 2012.

[37] Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 2012.

[38] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1989.

[39] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998.

[40] Yann Lecun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 2015.

[41] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *CoRR*, 2013.

[42] Xudong Mao, Qing Li, Haoran Xie, Raymond Y.K. Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. *CoRR*, 2016.

[43] Warren S. McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 1943.

[44] Ali Mollahosseini, David Chan, and Mohammad H. Mahoor. Going deeper in facial expression recognition using deep neural networks. *CoRR*, 2015.

[45] Tommy Mulc. Inception modules: explained and implemented. https://hackathonprojects.files.wordpress.com/2016/09/74911-image03.png, 2016.

[46] Ryota Natsume, Tatsuya Yatagawa, and Shigeo Morishima. RSGAN: face swapping and editing using face and hair representation in latent spaces. *CoRR*, 2018.

[47] Michael A. Nielsen. *Neural Networks and Deep Learning*. Determination Press, 2015.

[48] Noldus. Facereader. http://www.noldus.com/human-behavior-research/products/facereader, 2018.

[49] European Institute of Innovation & Technology (EIT). Eit health. https://www.eithealth.eu/.

[50] O'REILLY. Build a neural network that learns to generate handwritten digits. https://www.oreilly.com/learning/generative-adversarial-networks-for-beginners, 2017.

[51] PaddlePaddle. Image classification. http://paddlepaddle.org/docs/develop/book/03.image_classification/index.html, 2017.

[52] Rosalind W. Picard. *Affective Computing*. First MIT Press paperback edition, 2000.

[53] Christopher Pramerdorfer and Martin Kampel. Facial expression recognition using convolutional neural networks: State of the art. *CoRR*, 2016.

[54] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 1958.

[55] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. *Bradford Books/MIT Press*, 1985.

[56] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision (IJCV)*, 2015.

[57] A. L. Samuel. Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, 1959.

[58] Terrence J. Sejnowski and Charles R. Rosenberg. Nettalk: A parallel network that learns to read aloud. *MIT Press*, 1988.

[59] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *3rd International Conference on Learning Representations (ICLR2015)*, 2014.

[60] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 2014.

[61] Ilya Sutskever. *Training Recurrent Neural Networks*. PhD thesis, University of Toronto, 2013.

[62] Christian Szegedy, Sergey Ioffe, and Vincent Vanhoucke. Inception-v4, inception-resnet and the impact of residual connections on learning. *CoRR*, 2016.

[63] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. *AAAI*, 2017.

[64] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *Proceedings of the IEEE*, 2014.

[65] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. *CoRR*, 2015.

[66] K. Takahashi. Remarks on emotion recognition from multi-modal bio-potential signals. *2nd International Conference on Autonomous Robots and Agents*, 2004.

[67] Yichuan Tang. Deep learning using support vector machines. *CoRR*, 2013.

[68] Dmitry Ulyanov, Andrea Vedaldi, and Victor S. Lempitsky. Instance normalization: The missing ingredient for fast stylization. *CoRR*, 2016.

[69] Yoav Shoham (Stanford University), Raymond Perrault (SRI International), Erik Brynjolfsson (MIT), Jack Clark (OpenAI), and Calvin LeGassick. Artificial intelligence index – 2017 annual report. `https://aiindex.org/2017-report.pdf`, 2017.

[70] Marcel van Gerven and Sander Bohte. Artificial neural networks as models of neural information processing. *Frontiers in Computational Neuroscience*, 2017.

[71] Thomas Vandal, Daniel McDuff, and Rana El Kaliouby. Event detection: Ultra large-scale clustering of facial expressions. *2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, 2015.

[72] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2001.

[73] Wikipedia. Perceptrón — wikipedia, la enciclopedia libre. `https://es.wikipedia.org/w/index.php?title=Perceptr%C3%B3n&oldid=107276002`, 2018.

[74] W.F. Windle. *The Spinal cord and its reaction to traumatic injury: anatomy, physiology, pharmacology, therapeutics*. M. Dekker, 1980.

[75] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? *CoRR*, 2014.

[76] Zhanpeng Zhang, Ping Luo, Chen Change Loy, and Xiaoou Tang. Learning social relation traits from face images. *CoRR*, 2015.

[77] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *CoRR*, 2017.

[78] Xinyue Zhu, Yifan Liu, Zengchang Qin, and Jiahong Li. Data augmentation in emotion classification using generative adversarial networks. *CoRR*, 2017.