Reconocimiento de Expresiones Faciales de dolor utilizando Redes Neuronales Convolucionales

Jorge Enrique Hernaez Acha[1], Stefanía Yuliana Viterbori Ugarte[2], Diego Stalder[3]
[1−2]Autores, [3]Orientador,
Facultad de Ingeniería
Ingeniería Mecatrónica

Resumen

Las redes neuronales convolucionales son modelos computacionales diseñados para procesar y clasificar imágenes con altísima exactitud y precisión. En la medicina, técnicas de procesamiento de imágenes ya auxilian a los profesionales en el diagnóstico de enfermedades como el cáncer. En general, el malestar es medido por auto-reportes de pacientes, normalmente mediante comunicación verbal. Hoy se sabe que es posible medir el dolor a partir de cambios en las expresiones faciales. En este trabajo se describe un método de detección de dolor imagen por imagen utilizando redes neuronales convolucionales, las cuales fueron entrenadas con una base de datos que consiste en imágenes, etiquetadas por especialistas, de pacientes que experimentaban dolor de hombro. Esto se realiza de dos formas: Directamente, utilizando las puntuaciones de la escala de dolor PSPI e indirectamente, detectando las unidades de acción relacionadas al dolor. Los resultados indican que la detección de dolor a partir de las expresiones faciales puede lograr una alta exactitud.
Palabras Claves: Expresiones faciales, Redes Neuronales Convoluciones, Unidades de Acción (AUs), Escala de dolor PSPI.

1. Introducción

En los últimos años los avances en la inteligencia artificial han permitido el desarrollo de sistemas inteligentes capaces de ser utilizados en una amplia gama de aplicaciones. Estas aplicaciones no solo se han beneficiado del desarrollo, sino que implementaciones como vehículos autónomos y drones auto navegados han ayudado a este desarrollo tanto en el campo de la inteligencia artificial, como la visión computacional y el procesamiento de imágenes.

Por otro lado, en el campo de la medicina muchas empresas ya se encuentran implementando sistemas inteligentes para asistencia en diagnóstico y monitoreo de pacientes a través de visión computacional. Esto permite una mayor rapidez y precisión en la detección de enfermedades. Se espera que estas tecnologías den pie a una nueva revolución tecnológica con los hospitales inteligentes, donde se proveerá de servicios integrales a los pacientes y médicos.

En este contexto la detección automática de dolor es muy importante, ya que la detección del más simple malestar de un paciente puede ser un factor decisivo. Estudios previos indican que el rostro es una fuente de información muy confiable para detectar emociones, incomodidades, malestar, etc., y que estas características se pueden obtener mediante procesamiento de imágenes en tiempo real. Previamente se han realizado estudios para desarrollar sistemas capaces de detectar estas expresiones, pero extrayendo inicialmente una serie de características del rostro.

Muchos de estos avances han sido posibles gracias a la implementación de redes neuronales convolucionales, que son modelos computacionales de aprendizaje profundo que se especializan en el procesamiento y clasificación de imágenes. Estas redes han sido utilizadas previamente para la detección del dolor y se ha demostrado su utilidad en este campo. Es por esto que se propone un sistema computacional capaz de detectar el dolor a partir del rostro mediante el uso de redes convolucionales.

2. Objetivos

Objetivo General

Diseñar un sistema de detección automática de expresiones faciales de dolor utilizando redes neuronales convolucionales.

Objetivos Específicos

- Buscar y seleccionar una base de datos con imágenes y/o secuencias de vídeo etiquetadas y validadas por especialistas del área clínica.
- Revisar los fundamentos teóricos necesarios para procesar y clasificar imágenes digitales.
- Dominar las herramientas computacionales más utilizadas en la literatura para el procesamiento de imágenes y la implementación de sistemas inteligentes utilizando redes convolucionales.
- Aplicar un algoritmo de detección de rostros eficiente.
- Realizar pruebas comparativas con trabajos previos y diferentes arquitecturas de redes convolucionales.
- Diseñar un prototipo funcional de un identificador de expresiones de dolor con el mejor clasificador obtenido.

3. Estado del arte

3.1. Detección de rostros

Existen cientos de estudios sobre detección de rostros, Yang [1] agrupó varios métodos en cuatro categorías: *Métodos Basados en Conocimiento:* los cuales codifican el conocimiento humano de lo que constituye un rostro típico. Generalmente, las reglas miden la relación que existe entre características de la cara; *Métodos Basados en Características Invariantes:* que nos permiten encontrar características estructurales robustas que existen incluso cuando la posición, el punto de vista o las condiciones de luces varían. También son diseñados generalmente para la localización del rostro; *Métodos de Pareo de Plantillas:* Estos métodos almacenan patrones estándares de la cara para describir el rostro como un todo, son utilizados tanto para la localización como para la detección del rostro y *Métodos Basados en Apariencia:* En contraste con el método anterior, los modelos o patrones se aprenden de un conjunto de entrenamiento imágenes, las cuales representan una gran variabilidad en apariencia. Estos métodos son diseñados mayormente para la detección del rostro.

En general, los métodos basados en apariencia han demostrado mejor desempeño que los demás, gracias al rápido crecimiento de la capacidad de cálculo y el almacenamiento de datos de los computadores [2].

Ejemplos de técnicas basadas en apariencias utilizan algoritmos de AdaBoost como el de Viola y Jones [3], Support Vector Machines (SVM), Neural Networks (NN) y el clasificador de Bayes.

El trabajo de Viola y Jones [3] ha hecho que la detección de rostros sea factible en aplicaciones del mundo real como cámaras digitales y softwares de organización de fotos.

Trabajos recientes como los de Haoxiang [4] y Kalinowski [5] utilizaron Redes Neuronales Convolucionales en Cascada para la detección de rostros, los cuales dieron resultados comparables con los mejores detectores de rostros del estado del arte, además, Kalinowski [5] propone una una red compacta (*Compact Convolutional Neural Network Cascade*), la cual supera a las demás en rapidez para el problema de detección frontal de rostros y hace que sea posible obtener altas velocidades incluso para dispositivos de computo lento.

3.2. Reconocimiento de expresiones de dolor

Aunque existen muchas investigaciones en el reconocimiento automático de expresiones faciales afectivas, como puede encontrarse en [6], hasta no hace mucho tiempo habían pocos trabajos enfocados en la detección automática de expresiones de dolor. Gracias al rápido avance en visión computacional y también a recientes publicaciones de bases de datos como la UNBC-McMaster [7], el análisis de las expresiones faciales de dolor ha tenido un avance significativo. A continuación describiremos algunos trabajos publicados.

Existen varios avances esta área, los más destacados se presentan en las investigaciones de Ashraf [8], Lucey [7], Kaltwang [9] y Hammal [10], donde desde distintas perspectivas, se han analizado los posibles métodos de detección e identificación de gestos faciales.

Inicialmente se utilizaban algoritmos y métodos iterativos de alta precisión, pero con un costo computacional que se elevaba exponencialmente a medida que aumentaba la complejidad de los gestos, el número de personas, y los distintos rasgos a procesar, además eran necesarias máquinas de soporte vectorial (SVM) [11], cuyo costo hasta se podía describir como privativo, por lo que quedaba reservado su uso solo a las más grandes universidades y centros de investigación.

Así también, una de las principales limitaciones operativas era la utilización de modelos de perspectiva rígidos, lo que no permitía transformaciones de los rostros para eliminar rasgos innecesarios o erróneos.

En sus trabajos en 2009, y subsecuentemente en 2010, Ashraf [8] y Lucey [7] respectivamente, nos presentan un enfoque distinto a la resolución de este problema, pero aún utilizando los mismos marcadores o *flags* para la detección de los gestos, los AUs o unidades de acción [12]. Utilizando los modelos activos de apariencia (AAM *Active Appearance Model*), los cuales nos indican los puntos de referencia de un rostro humano, o sea, las características principales como ojos, nariz, boca, etc, un rostro es fácilmente transformable, y por lo tanto, normalizable, solucionando así el problema de rigidez de estudios anteriores.

Los resultados de estos estudios, aunque se encuentran lejos de ser perfectos, ideales o siquiera aceptables, muestran que aunque el campo aún no se encuentra muy avanzado, los métodos utilizados tienen potencial aún explotable.

En 2011, Kaltwang [9], teniendo como antecedentes los trabajos de Ashraf [8] y Lucey [7], propuso un método de detección de dolor completamente distinto a los métodos anteriores, en lugar del método estándar de 2 pasos, en el cual inicialmente se extraían los AUs de una imagen, para luego realizar el cálculo del dolor mediante SVM, su propuesta fue la de separar este proceso en 3 pasos: El primero la extracción de los rasgos de forma y apariencia del rostro, el segundo de separar los datos en distintos conjuntos de entrenamiento, en los cuales se entrenan redes neuronales mediante modelos distintos de regresión y finalmente calcular el valor final del dolor mediante la media de los valores obtenidos como predicción por la red, al mismo tiempo que esos valores se vuelven a introducir como entrada de una nueva red, que estima un valor final a partir de los anteriores.

Finalmente el trabajo más reciente en la detección de dolor a nivel de imagen, fue realizado por Hammal [10] en 2012, en este caso, también con un enfoque completamente distinto, utilizando como base los trabajos anteriores, en lugar de extraer las características principales de forma y apariencia, Hammal procesó cada imagen mediante filtros y transformaciones de orientación, para obtener 9216 características, que serían luego procesador por dos redes neuronales entrenadas de distinta forma: La primera, mediante la utilización de toda la base de datos, utilizando una proporción de 80 % para entrenamiento, y 20 % para validación. Este proceso de validación cruzada se realizó 5 veces por ca-

da entrenamiento. En la segunda, se procedió a dejar fuera a uno de los sujetos de prueba, para realizar la validación del entrenamiento con el mismo.

4. Escala de dolor de Prkachin and Solomon

Se han hecho muchos estudios para identificar indicadores faciales válidos de dolor. En estos estudios se utiliza un sistema se codificación facial llamado FACS (Facial Action Coding Systems), diseñado por Paul Ekman [12], en el que cada movimiento muscular es catalogado como una Unidad de Acción (AU). A inicios del 1992, Prkachin [13] encontró que cuatro unidades de acción -fruncimiento de ceja (AU4), mejillas estiradas (AU6 y AU7), labio superior contraído (AU9 y AU10) y ojos cerrados (AU43)- contienen información a cerca del dolor. En un trabajo posterior, Prkachin y Solomon [14] confirmaron que estas cuatro acciones contenían la mayor cantidad de información acerca del dolor. Ellos definieron al dolor como la suma de las intensidades de estas cuatro acciones. La métrica de intensidad de dolor de Prkachin y Solomon (PSPI) se define como:

$$PSPI = AU4 + max(AU6; AU7) + max(AU9; AU10) + AU43 \quad (1)$$

Esto es, la suma de $AU4$, $AU6$ o $AU7$ (cual sea mayor en intensidad), $AU9$ o $AU10$ (cual sea mayor en intensidad) y $AU43$, lo que da una escala de 0 a 16. Por ejemplo en la figura 1 tenemos $AU4b + AU6e + AU7e + AU9e + AU43$, esto dará un PSPI de $2+5+5+1 = 13$.

La escala PSPI es la única actualmente, que puede medir el dolor en cada imagen.

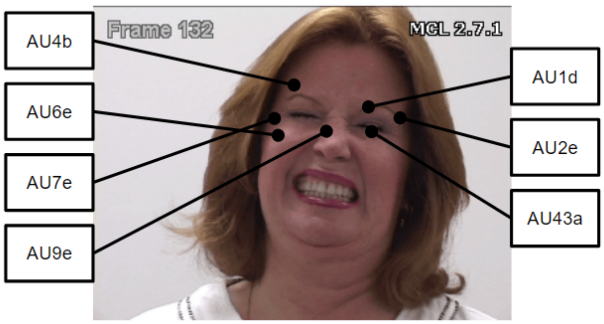

Figura 1: Ejemplo del cálculo del PSPI. Se indican los AUs presentes en la imagen con sus respectivas intensidades, los relacionados a la formula del PSPI son: $AU4b + AU6e + AU7e + AU9e + AU43$, esto dará un PSPI de $2+5+5+1 = 13$. **Fuente:** Elaboración propia utilizando la base de datos [7].

5. Base de datos

La base de datos utilizada es la *UNBC-McMaster Shoulder Pain Expression Archive Database*, la misma fue desarrollada por el Laboratorio de Investigación de la Universidad de Pittsburgh, en donde se realizaron pruebas con voluntarios que padecían dolor de hombro crónico, y mediante investigadores y psicólogos, se realizó una medida del dolor a nivel de imagen [7].

La base incluye una documentación con una medida de cada AU presente o no presente en cada una, y el posterior cálculo del dolor o PSPI para cada imagen, además cuenta con los reportes hechos por las personas en prueba para cada secuencia.

En resumen, la base de datos contiene:

- Expresiones espontáneas temporales: 200 secuencias de vídeo con expresiones faciales espontáneas relacionadas con el dolor,
- Códigos FACS: 48.398 imágenes codificadas mediante el sistema FACS manualmente,
- Valoraciones de auto-reportes y de observadores: para cada secuencia de vídeo

Utilizando la ecuación 1 el PSPI puede variar del 0 al 16, en la tabla 1 se muestra el inventario de la UNBC-McMaster Shoulder Pain Archive de acuerdo a esta escala de dolor.

Cada AU es medido en una escala de intensidad de la A (menos intenso) hasta la E (más intenso), el inventario de la base de datos UNBC-McMaster Shoulder Pain Archive se muestra la tabla 2.

Cuadro 1: Distribución del nivel de dolor(PSPI) en la Base de datos utilizada para el entrenamiento.

PSPI	Frecuecia
0	40.029
1-2	5.260
3-4	2.214
5-6	512
7-8	132
9-10	99
11-12	124
13-14	23
15-16	5

Fuente: [7].

Cuadro 2: Distribución de la unidades de acción (AU) en la Base de datos utilizada para el entrenamiento. Note que para AU43, la única intensidad es A (los ojos solo pueden estar abiertos o cerrados).

AU	A	B	C	D	E	Total
4	202	509	225	74	64	1074
6	1776	1663	1327	681	110	5557
7	1362	991	608	305	100	3366
9	93	151	68	36	75	423
10	171	208	63	61	22	525
12	2145	1799	2158	736	49	6887
20	286	282	118	0	20	706
25	767	803	611	138	88	2407
26	431	918	265	478	1	2093
43	2434	–	–	-	-	2434

Fuente: [7].

6. Redes Neuronales Convolucionales

Las redes neuronales, vienen de la idea de imitar el funcionamiento del cerebro de los organismos vivos: un conjunto de neuronas conectadas entre sí y que trabajan en conjunto, sin que haya una tarea concreta para cada una. Con la experiencia, las neuronas van creando y reforzando ciertas conexiones para "aprender" algo que se queda fijo en el tejido.

En su libro, Damián Matich [15] define a las redes neuronales como un sistema de computación compuesto por un gran número de elementos simples, elementos de procesos muy interconectados, los cuales procesan información por medio de su estado dinámico como respuesta a entradas externas.

El funcionamiento de una red neuronal se basa en la combinación de determinados conceptos matemáticos y computacionales, el primero de ellos es el perceptrón, el cual es un elemento con varias entradas y una salida, y un peso para cada entrada, el cual determina la ponderación de la misma.

La complejidad del sistema aumenta a medida que el número de perceptrones, que forman un conjunto o capa, como llamaremos de ahora en más al conjunto de perceptrones. Así como puede aumentar el número de perceptrones, también se pueden agregar nuevas capas al modelo, como se observa en la figura, creando una red multicapa.

Las redes convencionales escalan exponencialmente en número de parámetros a medida que se agregan nuevas capas, lo que dificultaría el procesamiento de imágenes. Es por esto que dentro de las redes neuronales, existen redes especializadas en el procesamiento de imágenes, que son conocidas como redes neuronales convolucionales.

Las redes convolucionales aprovechan el hecho de que sus entradas consisten en solamente imágenes, y por lo tanto las arquitecturas están limitadas de una forma más sensible. En particular, a diferencia de las redes convencionales, las capas de una red ConvNet tiene sus neuronas ordenadas en 3 dimensiones: ancho, alto y profundidad. Por esta característica, una neurona no necesita conectarse a todas las neuronas de una capa anterior, sino solamente a una pequeña región de esta.

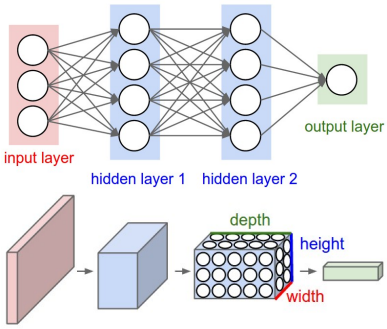

Figura 2: Arriba: Una red neuronal regular de 3 capas. Abajo: Las neuronas de un arreglo ConvNet en tres dimensiones (ancho, alto, profundidad), como se visualiza en una capa. Cada capa de una ConvNet transforma un volumen de entrada 3D a un volumen de salida 3D. **Fuente:** [16].

Una red convolucional es un clasificador formado por una serie de capas, en las cuales se procesa y reduce la imagen en la entrada a sus características más resaltantes. Las capas que conforman una red convolucional son:

- **Capa de convolución:** es el núcleo de los bloques en una red convolucional. Consta de una serie de filtros de parámetros autoajustables. Cada filtro es pequeño en ancho y alto y se extiende a la profundidad de la entrada. Estos filtros se convolucionan individualmente con cada sección de la imagen, moviendose a lo largo y ancho con un paso predefinido.

  Los parámetros que definen a la capa de convolución son:

  - **Profundidad:** La profundidad, que no debe ser confundida con la profundidad de la imagen, es el número de filtros que se utilizarán.
  - **Paso:** es el número de pixeles que se moverá el filtro cada vez que pase a una nueva sección de la imagen.
  - **Zero-Padding:** añadir ceros alrededor de la imagen permite controlar las dimensiones de la salida.

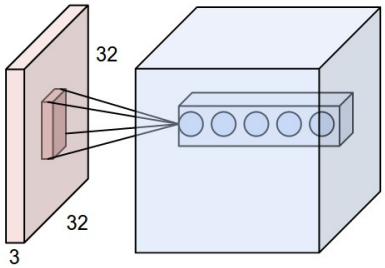

Figura 3: Ejemplo de un volumen de entrada (32×32×3) y el filtro con la misma profundidad. **Fuente:** [16].

- **Capa de reducción o pooling:** La capa de Pooling es la encargada de reducir el tamaño espacial de la imagen. Esto reduce el número de parámetros y facilita el proceso computacional de la red, además de reducir a la imagen a sus parámetros más significativos en cada sector.

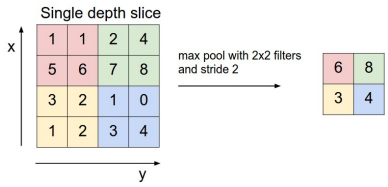

Figura 4: Ejemplo de max-pooling de 2×2 con paso 2 aplicado a una entrada de (4×4) y el filtro con la misma profundidad. **Fuente:** [16].

- **Capa de clasificación:** Las capas de clasificación son capas completamente conectadas que tienen todas sus neuronas conectadas a todas las neuronas de la capa anterior. La única diferencia entre estas capas y las de convolución es que las de convolución están conectadas localmente.

7. Detección automática de expresiones de dolor

Para resolver el problema de la detección de expresiones de dolor se han implementado dos metodologías que se describen a continuación:

1. Detección directa del PSPI
2. Detección indirecta del dolor utilizando AUs para cálculo del PSPI

Una aplicación se muestra en la figura 5, donde la camara que capta la imagen de entrada puede apuntar a la camilla de hospital por ejemplo. Luego se realiza la detección del rostro y el módulo de detección de predicción será una red neuronal entrenada, nuestro tranajo consiste en seleccionar esta red y entrenarla para conseguir los mejores resultados posibles. En la siguiente sección se habla del entrenamiento de dicha red.

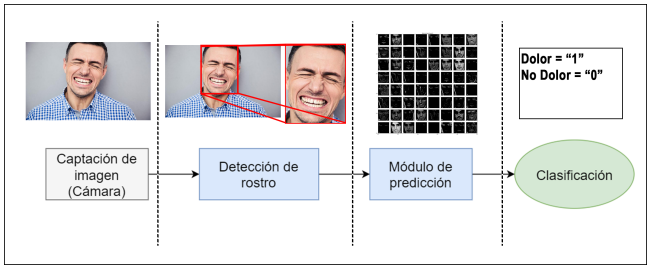

Figura 5: Sistema propuesto.

8. Entrenamiento de la red

El ciclo de entrenamiento esta dividido en cuatro partes, como muestra la figura 6. En cada caso se han probado diversos algoritmos de detección y en cada uno de estos las variaciones se pueden dividir en tres grupos principales, como son el pre-procesamiento, la extracción de características y la clasificación, lo cual se explicará detalladamente a lo largo de la sección.

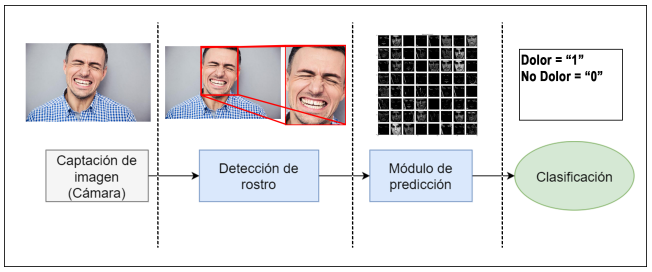

Figura 6: Sistema propuesto.

8.1. Preprocesamiento

Dado que una red convolucional de esta magnitud deberá procesar decenas de miles de imágenes, se debe intentar reducir la carga a la GPU procesando previamente las imágenes, además, solo alrededor del 30 % de cada imagen contiene información útil, esto es, un rostro.

Por lo tanto en esta etapa vamos a extraer toda la información útil de la base de datos UNBC-McMaster Shoulder Pain Expression Archive Database descrita en la sección **??**.

Las fases del preprocesamiento se dividen en:

- Identificación y extracción del rostro.
- Aumento de datos mediante transformaciones geométricas y matemáticas.
- Redimensionamieto de las imágenes.
- Creación de conjuntos de datos.

Identificación y extracción del rostro

En este proceso se utiliza un algoritmo de detección de rostros, el cual retorna las coordenadas de los puntos del rostro y luego se recorta la imagen en estos puntos como puede verse en la figura 7.

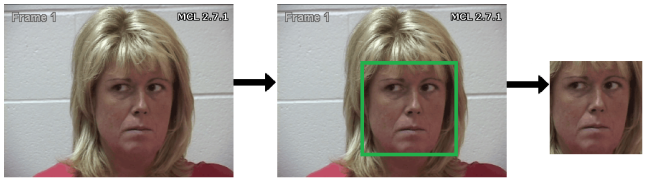

Figura 7: Detección de rostro: Imagen original, imagen con rostro detectado, imagen recortada. **Fuente:** Elaboración propia utilizando la base de datos de [7].

Para la detección de rostros se han probado los siguientes dos métodos por lo estudiado en la sección 3.1:

1. **Algoritmo de Viola - Jones:** Extracción de características utilizando filtros Haar en cascada entrenado para detectar caras [3].

2. **Algoritmo basado en Redes Neuronales:** Utilizando una red pre-entrenada existente para detección de rostros.

Para el primer caso se hemos utilizado el algoritmo que ya viene implementado en la librería para Python OpenCV [17] que cuenta con la función *faceCascade.detectMultiScale*.

En el segundo caso se ha utilizado la librería Dlib [18] que cuenta con una función llamada *face_ detector*, la podemos encontrar en el enlace dlib.net.

Aumento de datos

Uno de los problemas de la base de datos utilizada es que la cantidad de imágenes que contienen algún AU o PSPI mayor distinto de cero, es muy baja. Esto puede ocasionar sobreajuste (*Overfitting*). Una de las soluciones a este problema es el aumento de datos.

Existen varios métodos de aumento de datos, en este caso no podemos usar todos ellos como por ejemplo el escalamiento o la translación ya que nuestra entrada de datos debe ser el rostro de la persona. Aún así hemos utilizado algunos de ellos como son:

- Rotaciones: Se hicieron rotaciones pequeñas (≤ 5 grados) en el plano en ambos sentidos.
- Simetría: Como los rosotros humanos son simétricos de derecha a izquierza, se hicieron simetrías en este sentido.
- Cambios en las condiciones de iluminación: Esto se hizo añadiendo ruido gaussiano a las imágenes.

Además se hicieron combinaciones de los métodos citados arriba, pudiéndose obtener un total de hasta 12 imágenes, como muestra la figura 8.

Figura 8: Aumento de datos de imágenes de dolor. Arriba se ve la imágen original, la primera columan de la izquierda se obtiene a partir de la original mediante rotaciones de $\pm 5^o$, la segunda columna contiene las simetrías y las dos últimas se obtienen sumando ruído gaussiano a las anteriores. **Fuente:** Elaboración propia utilizando la base de datos de [7].

Redimensionamiento

La cantidad de parámetros de entrada a la red debe ser fija, por lo tanto el tamaño de las imágenes de entrada a la red debe ser el mismo para todos. Los tamaños originales varían entre 71×71 y 186×186 píxeles, entonces se han redimensionado todas las imágenes a un tamaño fijo en cada prueba utilizando la función *resize()* de *OpenCV*, la cual utiliza el método de interpolación bilineal.

Creación de conjuntos de datos

Se cargan todos los datos, esto es, las imágenes y las respectivas etiquetas, dependiendo de la clase a la que corresponda. Además los datos son separados en 3 subconjuntos: Entrenamiento, test y validación. Para la validación se toma una persona de todo el conjunto de datos. Los datos restantes se mezclan aleatoriamente y se dividen en 80 % para el entrenamiento (*training set*) y 20 % para el teste (*test set*).

9. Comparación de arquitecturas de redes neuronales convolucionales

Existen varias arquitecturas de redes neuronales utilizadas para la clasificación de imágenes. Entre estas, se han seleccionado las arquitecturas Xception, VGG16 y ResNet50 para realizar pruebas comparativas de desempeño. Así también, se ha desarrollado una arquitectura convolucional simple para contrastar sus resultados con las más utilizadas.

Cuadro 3: Comparación de arquitecturas utilizadas para clasificación de imágenes.

Modelo	Tamaño	Profundidad	Parámetros	Tamaño entrada	Pesos pre-entrenados
Modelo Simple	790 KB	9	1.704.162	90×90	no
Xception	88 MB	126	22.910.480	299×299	imagenet
VGG16	528 MB	23	138.357.544	224×224	imagenet
ResNet50	99 MB	168	25.636.712	224×224	imagenet

Fuente: Elaboración propia.

9.1. Arquitectura simple

Una arquitectura simple basada en capas de convolución, pooling y dropout es propuesta como primera opción en este trabajo, y está representada en la figura 9.

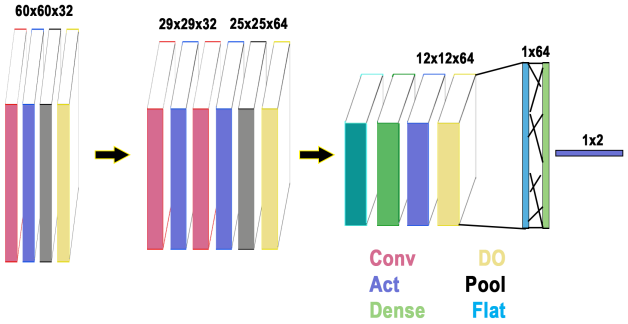

Figura 9: Diagrama Modelo simple

9.1.1. Arquitectura VGG16

La arquitectura VGG es una de las ConvNet más simples. Está formada por capas de convolución de tamaño 3×3 en sucesión, hasta reducir la imagen a un solo vector.

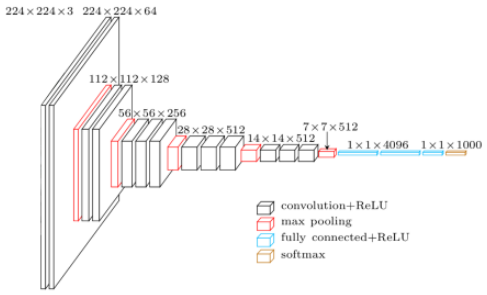

Figura 10

9.1.2. Arquitectura Xception

Esta arquitectura desarrollada por Google, es una red de muy alta profundidad pero de poca interconexión entre capas. La figura 11 muestra el esquema de la red.

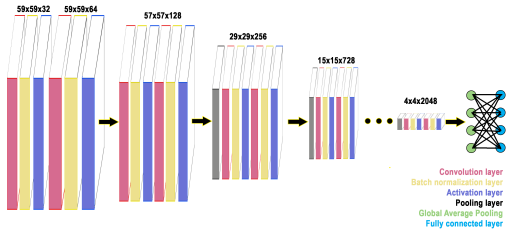

Figura 11

9.1.3. Arquitectura ResNet

La arquitectura ResNet es un modelo de aprendizaje residual. Basa su funcionamiento en la idea de que las redes neuronales no se ajustan a todos los conjuntos de datos, sino que existen ciertos conjuntos que no son entrenables. Con estos conjuntos, mientras más capas agregamos, la precisión decae. Dado que las redes neuronales están estructuradas de forma que trabajen con conjuntos de datos no específicos, la arquitectura ResNet propone un sistema basado en una función de residuo, que se adapta al conjunto de datos.

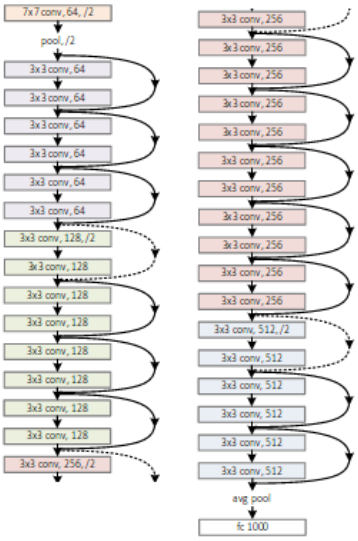

Figura 12: Diagrama de arquitectura ResNet50

10. Métricas utilizadas

10.1. Matriz de confusión

Para una mejor compresión de las pruebas y resultados, se presenta previamente una descripción de las métricas utilizadas para analizar el rendimiento de los algoritmos de clasificación utilizados.

		PREDICHO		
		Clase A	Clase B	Clase C
REAL	Clase A	5	2	0
	Clase B	3	3	2
	Clase C	0	1	11

Figura 13: Ejemplo de matriz de confusión.

10.1.1. Exactitud

La exactitud (ACC del inglés *Accuracy*) es una métrica que nos indica el porcentaje de predicciones correctas sobre el total de datos.

$$\text{ACC} = \frac{\sum \text{Predicciones Verdaderas}}{\sum \text{Predicciones totales}} \tag{2}$$

En el ejemplo de la figura 13 se tiene:

$$\text{AC} = \frac{5+3+11}{5+2+0+3+3+2+0+1+11} = 70{,}37\,\%.$$

10.1.2. Sensibilidad

La sensibilidad o razón de verdadero positivo (TPR del inglés *True positive rate*) es la proporción de casos positivos que se identificaron correctamente en una clase sobre el total de verdaderos de esa clase.

$$\text{TPR} = \frac{\text{Predicciones Verdaderas}}{\text{Total verdaderos}} \tag{3}$$

En el ejemplo de la figura 13 se tiene para la clase A:

$$\text{TPR} = \frac{5}{5+2+0} = \frac{5}{7} = 71{,}43\,\%.$$

10.1.3. Precisión

La precisión (P) es la proporción de los casos positivos que se identificaron correctamente sobre el total de predicciones de esa clase.

$$\mathrm{P} = \frac{\text{Predicciones Verdaderas}}{\text{Predicciones totales}} \tag{4}$$

En el ejemplo de la figura 13 se tiene para la clase A:

$$\mathrm{P} = \frac{5}{5+3+0} = \frac{5}{8} = 62{,}5\,\%.$$

10.1.4. Valor F1

El Valor-F1 (*F1-Score*) es una medida de la precisión de un test, se emplea para obtener un valor único en el que estén ponderados la precisión y la sensibilidad. Está definido como la media armónica de ambos valores.

$$\mathrm{F1} = \frac{2}{\frac{1}{\mathrm{TPR}} + \frac{1}{\mathrm{P}}} \tag{5}$$

En el ejemplo de la figura 13 se tiene para la clase A:

$$\mathrm{F1} = \frac{2}{\frac{7}{5} + \frac{8}{5}} = \frac{2}{3} = 66{,}67\,\%.$$

11. Resultados

11.1. Detección de rostros

En el caso de las cascadas de Haar se han detectado 48.297 imágenes de las 48.398 con que contaba la base de datos. Obteniéndose una **precisión de 99.79 %** y se han detectado varios casos en los cuales el rostro esta muy desviado del centro de la imagen cuando la persona no mira directamente de frente. Luego en las pruebas de la red convolucional pre-entrenada se han detectado 48.199 imágenes de las 48.398 con que contaba la base de datos. Obteniéndose una **precisión de 99.59 %** y no se han detectado casos en los que detecta malas coordenadas para los rostros de alguna imágen.

Por otro lado, el tiempo promedio de detección con redes neuronales es de 90ms, mientras que el detector con cascada Haar tarda en promedio 16ms por imagen.

Finalmente hemos hecho la detección utilizando redes neuronales y en caso de no encontrar el rostro con este algoritmo, se activa el algoritmo que utiliza los filtros Haar. Utilizando la combinación de ambos algoritmos hemos obtenido una precisión del 99.98 % (sólo 11 rostros no detectados). En la figura 14 podemos ver algunas imágenes en la que ninguno de los dos algoritmos detectó rostros, estas imágenes no han sido utilizadas por no estar en buenas condiciones.

Figura 14: Rostros no detectados. **Fuente:** Base de datos UNBC-McMaster [7].

11.2. Clasificación mediante redes convolucionales

Cuadro 4: Comparación de modelos.

Detección del AU43	**Test interno**			**Test Externo**		
Extracción de características	**ACC**	**F1 No AU**	**F1 AU43**	**ACC**	**F1 No AU**	**F1 AU43**
Modelo Simple	98.10	98.47	97.51	96.05	97.25	94.37
Modelo Simple	96.21	96.87	95.20	91.65	93.00	89.64
VGG16	99.75	99.80	99.67	96.38	97.11	95.16
VGG16	99.92	99.93	99.89	96.10	96.68	95.15
VGG16	99.76	99.80	99.68	95.95	96.71	94.71
ResNet50	99.55	99.64	99.12	93.78	94.23	95.20
Xception	99.45	99.56	99.27	95.70	96.63	94.04
Xception	99.86	99.16	98.62	95.18	96.25	93.25
Xception	**99.92**	**99.93**	**99.89**	93.86	95.36	90.93
Xception	99.86	99.89	99.82	**96.80**	**97.51**	**95.52**

En la tabla 4 se muestran algunos modelos implementados para la detección del AU43.

En primer lugar se probó el modelo más sencillo de todos, obteniéndose una precisión de hasta **98 %**.

Seguidamente se probaron las arquitecturas VGG16 y Xception, ya que las mismas tienen pesos preentrenados, lo cual es muy útil cuando se tiene desbalanceo de datos, como ocurre con la base de datos utilizada. El modelo que tuvo mejores resultados en el test interno fue el Xception con las dos útimas capas totalmente conectadas, estos resultados no difieren demasiado del resto, excepto en el caso del modelo simple.

Por otro lado, el modelo con mejores resultados en el test externo es el Xception con una capa Global Average Pooling y otra totalmente conectada.

Si bien no existen diferencias significativas en cuanto a resultados entre las arquitecturas VGG y Xception, como se ha visto en la tabla 3, la arquitectura VGG tiene una mayor cantidad de parámetros y por lo tanto se ha utilizado la arquitectura Xception para los demás tests.

Por otro lado, la arquitectura simple a pesar de no tener tan buenos resultados, tiene un costo computacional mucho menor. Así hemos comparado esta arquitectura con la Xception en la detección de otras Unidades de acción como puede verse en la tabla 5. La diferencia en los resultados del test externo es mayor que con el AU43.

Cuadro 5: Comparación de los modelos Xception y el Modelo simple considerando los AUs 4 y 7

		Test Interno			**Test externo**		
AU	**Arquitectura**	**ACC**	**F1 No AU**	**F1 AU**	**ACC**	**F1 No AU**	**F1 AU**
4	M. Simple	99.58	99.25	99.01	89.95	95.26	29.44
4	Xception	99.89	99.75	99.77	93.42	96.95	57.84
7	M. Simple	99.32	99.21	99.12	77.52	78.95	75.15
7	Xception	99.91	99.86	99.89	87.85	88.46	86.27

11.3. Comparación de resultados con el Estado del Arte

La primera comparación con resultados previos de otros autores es la de la medición detección directa del PSPI en las clases Dolor (PSPI≥1) y No Dolor (PSPI=0). Los resultados se muestran en las tablas. En el trabajo de Ashraf [8] utilizaron Máquinas de Soporte Vectorial (SVM *Suppor Vector Machine*) para la clasificación de cada imagen en Dolor y No dolor. La validación implementada fue dejando una persona fuera del conjunto de entrenamiento y test y evaluando a la misma.

Cuadro 6: Comparación de los resultados de [8] con los propios.

Resultados Ashraf [8]			
	Dolor	**No Dolor**	**Sensibilidad**
Dolor	2957	632	82 %
No Dolor	3671	8501	70 %
Precisión	45 %	93 %	**73 %**

Resutados propios			
	Dolor	**No Dolor**	**Sensibilidad**
Dolor	956	241	80 %
No Dolor	187	453	71 %
Precisión	84 %	65 %	**77 %**

La siguiente comparación esta hecho con el trabajo de Lucey descrita en [19] en donde se realiza la detección para cada AU relacionado al dolor en las clases No AU y AU (intensidad ≥1), además de la detección directa del dolor. Los resultados de la exactitud se muestran en la tabla 7.

Cuadro 7: Comparación de los resultados de Lucey [19] con los propios. Se muestra la exactitud de la detección de cada AU en las clases no AU y AU (intensidad≥ 1), asi como la detección directa del dolor en las clases No dolor (PSPI=0) Y dolor (PSPI≥1).

Unidad de Acción	**AU**	**R. Previos**	**R. Propios**
Fruncimiento de cejas	4	72.5	93.0
Mejilla levantada	6	85.4	86.4
Parpados estirados	7	82.6	87.4
Nariz Arrugada	9	85.3	53.2
Levantamiento labial	10	89.2	66.1
Labios estirados	12	85.7	91.6
Labios acanalados	20	77.9	84.9
Labios separados	25	78.8	88.1
Mandíbula caída	26	73.5	74.8
Ojos cerrados	43	87.5	95.5

Por último, se comparan los resultados de Hammal, descritos en [10] en la que se realizó la detección del dolor para 4 niveles: No Dolor (PSPI=0), Dolor Diminuto (PSPI=1), Dolor Débil (PSPI=2) y Dolor fuerte (PSPI≥3). En la tabla 8 se muestran la precisión y el valor F1 para cada nivel, tanto en la validación interna (test con sujetos dentro del conjunto de entrenamiento) como en la validación externa (test con sujetos fuera del conjunto de entrenamiento).

Cuadro 8: Comparación de resultados de Hammal con los obtenidos en el trabajo

	Resultados de Hammal				**Resultados Propios**			
Validación	**Interna**		**Externa**		**Interna**		**Externa**	
Intensidad	**PR**	**F1**	**PR**	**F1**	**PR**	**F1**	**PR**	**F1**
0	95	96	65	57	100	99	25	37
1	97	92	37	67	97	98	77	48
2	97	91	35	40	98	98	26	19
≥3	98	95	70	60	99	99	22	35

11.4. Implementación

Una dificultad grande en visión computacional es la ejecución en tiempo real, lo que implica que el proceso de un solo cuadro de vídeo debe ser completado dentro de 30-40 milisegundos aproximadamente. Esto lleva a que en la implementación final del modelo se deban realizar ciertos compromisos.

El sistema de detección de rostros implementado finalmente fue el de cascada de Haar por ser el más eficiente en costo computacional y tomar la quinta parte del tiempo en detectar rostros que el método de red pre-entrenada.

Así también el método de clasificación seleccionado para la implementación fue la red convolucional de modelo simple, la cual tiene un costo computacional mucho menor que las demás arquitecturas.

El sistema final tiene un tiempo total de procesamiento de cada imagen de 54ms en promedio. Siendo el tiempo medio de detección de rostros de 45ms, y el tiempo medio de clasificación del mismo de 9ms.

12. Conclusiones

Este trabajo final de grado aborda el problema de la detección de expresiones faciales de dolor mediante el uso de redes neuronales convolucionales. Las principales conclusiones se resumen a continuación:

- Se seleccionó una base de datos de entrenamiento creada por especialistas del área clínica.
- Se utilizaron las herramientas más eficaces según el estado del arte para realizar el procesamiento de imágenes, entrenamiento e implementación del sistema de detección de dolor.
- Para implementar la etapa de detección de rostro, se realizaron pruebas comparativas de desempeño y exactitud de dos de los métodos más utilizados en la literatura. Los experimentos mostraron que tanto el filtro Haar como el detector de rostro basado en redes convolucionales pre-entrenadas se logran precisiones mayores al 99 %. Un análisis cualitativo de los rostros detectados mostró que la redes CNN son mejores, pero el detector de cascada de Haar resultó ser el más eficiente en términos de costo computacional (5 veces más rápido que el método de basado en redes convolucionales pre-entrenadas)
- En fase de entrenamiento, se mostró que es necesario utilizar la técnica de aumento de datos para balancear el conjunto de entrenamiento y lograr una buena exactitud y precisión en la detección de dolor directa e indirecta.
- Se comparó detalladamente la exactitud y el desempeño de las arquitectura mas competitivas según el estado de arte. Los resultados muestran la detección de la unidad de acción 43(cerrar los ojos) tienen una exactitud superior al 99 % y 90 % para el conjunto de teste y validación respectivamente. Esto indica que la detección indirecta de

dolor a partir del AU43 tiene una alta confiabilidad. El modelo simple se destacó por su baja complejidad y tiempo de entrenamiento, lo cual lo hace ideal para su implementación un hardware con recursos limitados. Pero el modelo Xception mostró mejores resultados que el modelo simple en cuanto a exactitud cuando se entrenaron sistemas para detectar los AUs 4 y 7.

- La resultados indican que la detección directa del dolor utilizando la arquitectura Xception puede lograr exactitud del 99 % y 77 % para el conjunto de test y validación respectivamente.
- La resultados indican que la detección directa del dolor considerando varios niveles de dolor no consiguió una buena capacidad de generalización a pesar que el conjunto de teste indica una exactitud de hasta 98 %.
- Comparaciones con trabajos previos muestran que las redes neuronales convolucionales superan a los métodos de detección previamente utilizados en 2 % y 9 % para la detección directa e indirecta(mediante el AU43).
- Se implementó un prototipo demostrativo utilizando un computador de escritorio de gama media(sin placas de procesamiento gráfico) para mostrar el funcionamiento del sistema de detección de rostros y los principales módulos de predicción entrenados. Los experimentos realizados previamente indicaron que el detector de rostro Haar y la red neuronal convolucional con arquitectura simple demandarían menores recursos computacionales. Esto permitió capturar y evaluar imágenes con un tasa de muestreo de hasta 200 milisegundos.

13. Trabajos futuros

- Aumentar la base de datos
- Implementar el sistema de detección en un hardware como por ejemplo el Jetson TK1.
- Clasificar una expresión de dolor considerando la información de una secuencia de imágenes y no sola de una imagen.
- Investigar la detección de otras expresiones faciales.
- Implementar el método de test K-fold para mayor generalización del entrenamiento.

Referencias

[1] M.-H. Yang, D. J. Kriegman, and N. Ahuja, "Detecting faces in images: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, pp. 34–58, Jan. 2002.

[2] C. Zhang and Z. Zhang, "A survey of recent advances in face detection," June 2010.

[3] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," 2001.

[4] H. Li, Z. Lin, X. Shen, J. Brandt, and G. Hua, "A convolutional neural network cascade for face detection," June 2015.

[5] I. Kalinowski and V. Spitsyn, "Compact convolutional neural network cascade for face detection," *CoRR*, vol. abs/1508.01292, 2015.

[6] Z. Zeng, M. Pantic, G. I. Roisman, and T. S. Huang, "A survey of affect recognition methods: Audio, visual, and spontaneous expressions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, pp. 39–58, Jan 2009.

[7] P. Lucey, J. F. Cohn, K. M. Prkachin, P. E. Solomon, S. Chew, and I. Matthews, "Painful data: The unbc-mcmaster shoulder pain expression archive database," *IEEE International Conference on Automatic Face and Gesture Recognition and Workshops (FG 2011)*, pp. 57–64, 2011.

[8] A. B. Ashraf, S. Lucey, J. F. Cohn, T. Chen, Z. Ambadar, K. M. Prkachin, and P. E. Solomon, "The painful face – pain expression recognition using active appearance models," *Image and Vision Computing*, vol. 27, no. 12, pp. 1788 – 1796, 2009. Visual and multimodal analysis of human spontaneous behaviour.

[9] S. Kaltwang, O. Rudovic, and M. Pantic, *Continuous Pain Intensity Estimation from Facial Expressions*, pp. 368–377. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012.

[10] Z. Hammal and J. F. Cohn, "Automatic detection of pain intensity," in *Proceedings of the 14th ACM International Conference on Multimodal Interaction*, ICMI '12, (New York, NY, USA), pp. 47–52, ACM, 2012.

[11] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and Computing*, vol. 14, pp. 199–222, Aug 2004.

[12] P. Ekman, "Facial signs, facts, fantasies and posibilities," *Sight, Sound and Sense*, vol. 1, 1978.

[13] K. M. Prkachin, "The consistency of facial expressions of pain: a comparison across modalities," vol. 51, 1992.

[14] P. E. S. Kenneth M. Prkachin, "The structure, reliability and validity of pain expression: Evidence from patients with shoulder pain," *Department of Psychology, University of Northern British*, vol. 139, 2008.

[15] D. J. Matich, *Redes Neuronales: Conceptos Básicos y Aplicaciones*, pp. 1–. Rosario, Argentina: Universidad Tecnológica Nacional, 2001.

[16] F.-F. Li, A. Karpathy, and J. Johnson, "Cs231n: Convolutional neural networks for visual recognition," 2016.

[17] G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision in C++ with the OpenCV Library*. O'Reilly Media, Inc., 2nd ed., 2013.

[18] D. E. King, "Dlib-ml: A machine learning toolkit," *Journal of Machine Learning Research*, vol. 10, pp. 1755–1758, 2009.

[19] P. Lucey, J. Cohn, S. Lucey, I. Matthews, S. Sridharan, and K. M. Prkachin, "Automatically detecting pain using facial actions," in *2009 3rd International Conference on Affective Computing and Intelligent Interaction and Workshops*, pp. 1–8, Sept 2009.