ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

Desarrollo de un modelo de aprendizaje profundo para el reconocimiento y clasificación de expresiones faciales asociadas a la enfermedad de Parkinson

PROYECTO INTEGRADOR

Previo la obtención del Título de:

Ingeniero/a en Computación

Presentado por:
Galo Xavier Figueroa Villacreses
Lessette Carolina Zambrano Zurita

GUAYAQUIL - ECUADOR
Año: 2022

DEDICATORIA

Dedico este trabajo a mis padres y abuelos que han hecho todo lo posible para que yo me pueda concentrar completa y solamente en mi carrera universitaria.

Galo Xavier Figueroa Villacreses

DEDICATORIA

Dedico este trabajo a mi padre por haber sido quien me inculcó la necesidad de superación constante desde mi niñez. Sus enseñanzas aún cosechan lo mejor de mí en cada oportunidad que se presenta y esta segunda carrera es un mérito a la manera de llevar la vida conforme a sus principios y consejos; haber contado con ellos ha sido un verdadero privilegio.

Lessette Zambrano Zurita

AGRADECIMIENTOS

Agradezco a mi familia por todo el apoyo constante que me han brindado durante esta etapa. Agradezco también a todas las personas grandiosas que he conocido durante mi carrera y que tal vez sin darse cuenta me han apoyado para no rendirme y para crecer personal y profesionalmente.

Galo Xavier Figueroa Villacreses

AGRADECIMIENTOS

Agradezco primeramente a Dios por otorgarme salud y permitirme alcanzar este logro, luego a mi honorable institución Armada del Ecuador por la asignación y confianza concedida como becaria de esta segunda carrera profesional. Finalmente agradezco a mi esposo e hijos, por tolerar mis presencias ausentes durante toda la carrera, sin su paciencia no hubiera alcanzado con satisfacción este gran logro en mi vida.

Lessette Zambrano Zurita

DECLARACIÓN EXPRESA

“Los derechos de titularidad y explotación, nos corresponde conforme al reglamento de propiedad intelectual de la institución; Galo Xavier Figueroa Villacreses y Lessette Carolina Zambrano Zurita damos nuestro consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual"

Firmado electrónicamente por:
GALO XAVIER FIGUEROA VILLACRESES

Galo Xavier Figueroa Villacreses

Firmado electrónicamente por:
LESSETTE CAROLINA ZAMBRANO ZURITA

Lessette Carolina Zambrano Zurita

EVALUADORES

Firmado electrónicamente por:
RONALD RAUL CRIOLLO BONILLA

Ronald Raul Criollo Bonilla

PROFESOR DE LA MATERIA

Firmado electrónicamente por:
COLON ENRIQUE PELAEZ JARRIN

Colon Enrique Pelaez Jarrin

PROFESOR TUTOR

RESUMEN

El presente trabajo consiste en el desarrollo de un modelo de aprendizaje profundo para el reconocimiento de expresiones faciales asociadas a la enfermedad de Párkinson, con la finalidad de contar con una herramienta que aporte al análisis médico para un diagnóstico temprano y un tratamiento oportuno de la enfermedad.

La metodología de trabajo consistió primeramente en el preprocesamiento de videos de monitoreo de pacientes y personas sanas otorgados por el Centro Integral de Neurociencias en España, obteniendo múltiples imágenes que enfocan exclusivamente el rostro de las personas monitoreadas; luego se realizaron experimentos con diferentes arquitecturas de redes neuronales que tomaban como entrada las imágenes del preprocesamiento y su salida era la probabilidad de que la imagen pertenezca a una persona que padece la enfermedad de Párkinson, es necesario indicar que en esta etapa los mejores resultados provenían del modelo tipo ensamblador que concatenaba las salidas de las redes VGG-19, InceptionNet, DenseNet y Nasnet. Finalmente se desarrolló un prototipo web destinado al uso por parte de los médicos para la obtención de un pronóstico durante una consulta médica.

La precisión del modelo fue de aproximadamente un 70%, lo cual permite pueda ser utilizado como herramienta de apoyo para un diagnóstico temprano de la enfermedad.

Finalmente se concluyó que los hiper-parámetros que más influyeron en el modelo fueron los momentos del optimizador y que el modelo puede mejorar si durante el monitoreo los videos captan mayor diversidad de expresiones faciales.

Palabras Clave: Párkinson, reconocimiento facial, redes neuronales, diagnóstico temprano.

ABSTRACT

Present work consists of the development of a deep learning model for the recognition of facial expressions associated with Parkinson's disease, to have a tool that contributes to medical analysis for early diagnosis and timely treatment of the disease. Methodology consisted first in the preprocessing of monitoring videos of patients and healthy people provided by the Comprehensive Neuroscience Center in Spain, obtaining multiple images that focus exclusively on the face of the monitored people; Experiments were then carried out with different neural network architectures that took the preprocessing images as input and their output was the probability that the image belongs to a person suffering from Parkinson's disease. It is necessary to indicate that at this stage the best results came from of the assembler-type model that concatenated the outputs of the VGG-19, InceptionNet, DenseNet and Nasnet networks. Finally, a web prototype was developed for use by doctors to obtain a prognosis during a medical consultation.

The accuracy of the model was approximately 70%, which allows it to be used as a support tool for an early diagnosis of the disease.

Finally, it was concluded that the hyper-parameters that most influenced the model were the moments of the optimizer and that the model can improve if, during monitoring, the videos capture a greater diversity of facial expressions.

***Keywords**: Parkinson, facial recognition, neural networks, early diagnosis.*

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

CAPÍTULO 1

1. INTRODUCCIÓN

1.1 Descripción del problema

El reconocimiento de emociones faciales (REF) es una habilidad esencial en las interacciones sociales entre los humanos; estudios recientes muestran resultados contradictorios en el REF en enfermos de Parkinson (EP) lo que puede deberse a varias razones, entre ellas que el REF no considera el deterioro cognitivo respecto a cómo deben percibirse las emociones, la gravedad de la enfermedad, el estado de la medicación o la comorbilidad psiquiátrica, entre otras [1]. Además, la sensibilidad de los métodos de estudio varía con la dificultad del diseño de los experimentos (por ejemplo, identificación de la emoción correcta versus discriminación entre dos emociones), por lo cual, el reconocimiento de emociones es aún un problema de investigación no resuelto completamente [2].

Alonso-Recio et.al en [1] presenta un experimento en el cual los pacientes EP tienen como tarea discriminar y asignar la correspondencia de imágenes emocionales, teniendo como resultado un deterioro en el reconocimiento de todas las expresiones evaluadas. De acuerdo a M. Arroyo Menéndez [3] este deterioro cognitivo es degradante para quien lo atraviesa, ya que implica pérdida de autoestima y autonomía; conforme avanza la enfermedad aumentan las limitaciones y dependencia, cambian los roles familiares y el impacto sociológico es considerado traumático.

Las condiciones mencionadas, ponen de manifiesto los dolores del paciente más allá de un contexto clínico, sino de implicación psicológica, económica, social y familiar; lo que genera la necesidad de una herramienta que pueda complementar el diagnóstico de posibles pacientes con enfermedad de Párkinson de manera temprana.

En los últimos años, con el desarrollo de la tecnología de visión por computadora, el reconocimiento de imágenes faciales es utilizado para el diagnóstico de enfermedades; en particular aquellas en las que la enfermedad está asociada con rasgos faciales clínicamente obvios, y que suelen ser utilizados para el diagnóstico, o incluso permiten descubrir algunas enfermedades genéticas relativamente raras

comparando el registro fotográfico de pacientes con personas sanas, lo que ayuda y acelera el proceso de detección de la enfermedad permitiendo generar un tratamiento oportuno al paciente [4]. La Figura 1.1 detalla el proceso actual del diagnóstico de la enfermedad de Parkinson en un paciente, el cual comienza con sintomatología leve, clínicamente no obvia, manteniendo a las personas con atenciones ambulatorias a través de diversas especialidades por un promedio de entre 3 a 5 años, generalmente por dolencias musculares y depresión. Posterior a este tiempo, comienzan los problemas de rigidez y/o temblores musculares, que derivan al paciente hacia un neurólogo quien define si el paciente amerita o no ser diagnosticado mediante métodos invasivos; la evaluación no invasiva es la más común, pero, aunque se determina mediante indicadores cuantificables, la medición depende de la observación y percepción del médico sobre movimientos musculares del paciente en una consulta médica de evaluación. Finalmente, se determina la enfermedad de Parkinson y puede empezarse un tratamiento que produzca o suministre dopamina en el cuerpo del paciente.

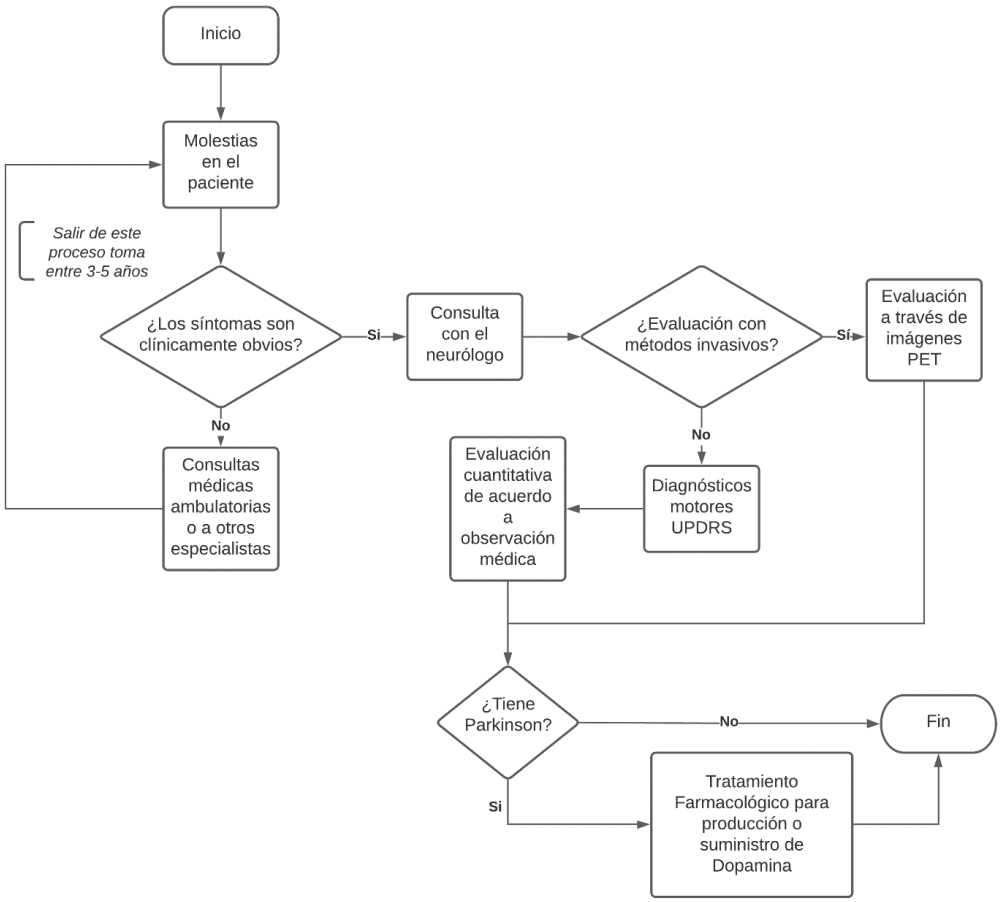


Figura 1.1. Proceso actual de diagnóstico de la EP en un paciente [Autoría propia].

1.2 Justificación del problema

Una de las grandes problemáticas respecto al diagnóstico de Parkinson en las personas, radica en el tiempo de espera desde la sintomatología inicial hasta la prescripción médica del tratamiento adecuado. La naturaleza de la enfermedad no permite una detección inmediata de la misma por medio de los métodos de diagnóstico convencionales [5], por lo que una herramienta que permita generar un diagnóstico temprano de la EP aportaría a las decisiones médicas para mitigar el daño que la enfermedad causa conforme progresa su evolución en los pacientes.

Las técnicas de aprendizaje profundo permiten abarcar problemas de gran complejidad, dentro de los cuales, la clasificación de imágenes, en general, y

específicamente aquellas vinculadas al área de la salud ha tenido un gran apogeo en el campo de la investigación.

Los diagnósticos físicos aplicados a las personas con Parkinson demuestran que los pacientes con esta enfermedad sufren un deterioro cognitivo que se refleja en una rigidez muscular, esto deriva en contracciones y cambios en las expresiones corporales [1], por lo que identificar estos patrones a partir del análisis de características faciales sería de importancia para el diagnóstico temprano y tratamiento oportuno de posibles pacientes de Parkinson de acuerdo con su nivel de criticidad.

1.3 Objetivos

1.3.1 Objetivo General

Desarrollar un modelo de aprendizaje profundo para la identificación de patrones asociados a la enfermedad de Parkinson a partir del análisis de características faciales de las personas.

1.3.2 Objetivos Específicos

1. Construir un modelo basado en técnicas de aprendizaje profundo, que permita predecir la presencia de la enfermedad de Parkinson usando videos de expresiones faciales.
2. Clasificar las secciones de la cara que el modelo de aprendizaje profundo identificó como más importantes durante la predicción de la enfermedad de Parkinson.
3. Desarrollar una aplicación web, que permita utilizar el modelo de aprendizaje construido y verificar los resultados de los pacientes.

1.4 Marco teórico

1.4.1 Antecedentes

Las anomalías en expresiones faciales suelen ser uno de los síntomas asociados a enfermedades neurológicas tanto en niños como en adultos;

la enfermedad de Parkinson afecta al 1 o 2 por mil de la población mundial, incrementa con la edad y existe en el 1% de la población sobre los 60 años [6].

Dentro de los signos principales de este desorden se encuentran la rigidez, temblor y acinesia[1] facial [2].

Así como para cualquier otro desorden neurológico, el diagnóstico de Parkinson y su posterior seguimiento, representa procedimientos costosos y muchas veces invasivos [5], lo que implica un difícil acceso y una exposición de la salud previa a un diagnóstico definitivo.

Estudios previos realizados mediante visión por computador para la identificación de Parkinson en pacientes, han planteado los siguientes experimentos:

S. Xu et.al [7] utilizaron imágenes de electroencefalogramas y redes neuronales recurrentes para detectar la presencia de la Enfermedad de Parkinson.

S. Lee et.al [8] utilizaron electroencefalogramas para detectar la Enfermedad de Parkinson usando un método innovador propuesto por los autores: una red neuronal convolucional recurrente.

U. Anusri et.al [9] utilizaron imágenes de personas sanas y personas enfermas del conjunto de datos Parkinson's Progression Markers Initiative (PPMI) para predecir de forma temprana la Enfermedad de Parkinson, usando una red neuronal convolucional con arquitectura AlexNet y otra con arquitectura VGG 16.

B. Jin et.al [10] recolectaron videos de expresiones faciales de personas enfermas y personas sanas y luego se obtienen características de la cara usando la herramienta Face++. Estas características son luego pasadas a

[1] Acinesia, ausencia pérdida o cesación de movimiento (RAE).

modelos de aprendizaje automático y aprendizaje profundo para que prediga si la persona del video está sana o tiene la Enfermedad de Parkinson.

M. R. Ali et.al [11] recolectaron videos usando la herramienta en línea www.parktest.net, en los cuales se les pidió a los participantes realizar tres expresiones faciales (sonrisa, disgusto y sorpresa). Estos videos sirvieron para entrenar modelos de aprendizaje automático que mostraron que los pacientes con la Enfermedad de Parkinson tienen menos variación en el movimiento de ciertos músculos de la cara.

1.4.2 Redes Neuronales

Las redes neuronales son una familia de las técnicas de procesamiento de información basado en el sistema nervioso biológico y que tratan de reproducir el comportamiento inteligente a partir de su unidad de procesamiento (neurona) [6]. De tal manera que toda señal que ingresa a la red de neuronas interconectadas representa un estímulo que al ser procesado genera una respuesta de acuerdo con la configuración que se le haya otorgado.

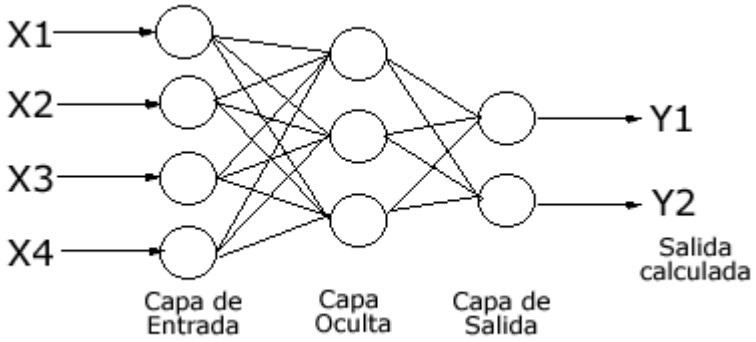


Figura 1.2. Esquema básico de funcionamiento de las redes neuronales [12].

La Figura 1.2 se muestra la configuración básica de toda red neuronal, las entradas se reciben como un vector de datos, luego el procesamiento que define la red se da en la capa oculta para posteriormente enviarlo a la capa de salida.

A lo largo del tiempo, se han definido distintos tipos de arquitecturas de redes neuronales dependiendo del tipo de dato que se quiere modelar. Un ejemplo aplicado al procesamiento de imágenes o multimedia son las redes convolucionales CNN, cuyos inicios fueron en 1982 por Kunihiko Fukushima [13], quien replicó el procesamiento hacia atrás desde la información recibida por el córtex visual. Fue en el año 2012 a partir de la red ***AlexNet*** implementada por Geoffrey Hinton, Ilya Sutskerver y Alex Krishevsky en donde se pudo visualizar la potencia de esta tipología de red para la clasificación de imágenes [13].

Las redes CNN se basan en la variación de información que pasa a través de las capas de convolución; a partir de técnicas de submuestreo reducen el volumen de datos durante la clasificación, dado que tienen su especialización en datos de 2 o más dimensiones. La Figura 1.3 muestra con mayor detalle la arquitectura de una red convolucional básica.

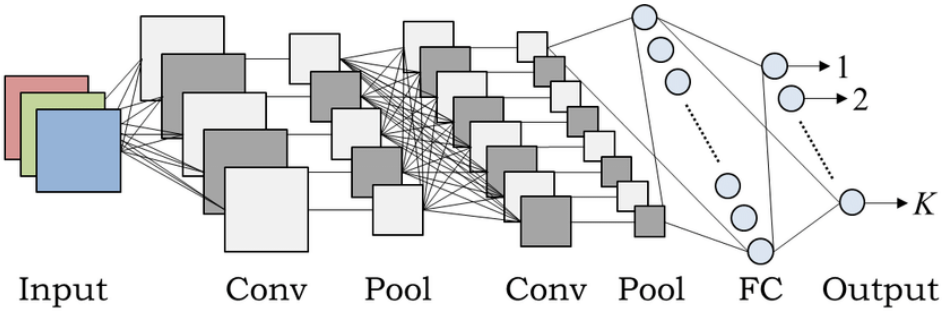


Figura 1.3. Configuración básica de una red neuronal convolucional (CNN) [14]

Los parámetros básicos que se configuran en una red de acuerdo con [15] son:

Learning Rate, influye en la intensidad con la que se dará el aprendizaje, es decir la medida en que la nueva información adquirida reemplazará a la anterior.

Batch Size, cantidad de ejemplos que se procesan en cada iteración del aprendizaje.

Funciones de activación, este componente modifica la combinación lineal de las salidas de la capa anterior y parámetros w, aplicando una función que usualmente es *ReLU*, sigmoide o *tanh*, aunque puede ser cualquier otra.

Cantidad de épocas, cada época corresponde a un ciclo completo es decir evaluación de todo el conjunto de entradas.

Por otra parte, las capas que pueden formar parte de una red CNN de acuerdo a [16] son:

Convolucional, en la cual se realiza la operación de convolución; esta operación se realiza sobre matrices y deben definirse parámetros de tamaño de filtro, *stride* o desplazamiento sobre la matriz y *padding* o relleno de ceros para incrementar tamaño de la matriz de ser necesario.

Filtro, la operación convolucional consiste en sumar la multiplicación elemento a elemento de una sección de la matriz de entrada y otra matriz definida por el diseñador de la red; esta última matriz es llamada filtro. Cada capa convolucional puede contar con varios filtros para capturar características distintas de la entrada.

Pooling, es la capa sobre la cual se realiza una reducción de dimensionalidad de la matriz devuelta por la capa anterior, que generalmente era una capa de convolución.

Dense o Fully-Connected, cada neurona en esta capa se conecta a todas las de la capa anterior, la salida de esta capa está representada por $f(z)$, donde z corresponde a $\sum_{i=1}^{n} w_i x_i$, siendo x las entradas que llegan a cada neurona de la capa dense y w los pesos asignados; f finalmente corresponde a una función de activación aplicada a z, que puede ser ReLU, Softmax, Sigmoide, etc.

Droput, esta capa ayuda a prevenir problemas de entrenamiento de la red como el *overfitting* o memorización de las salidas, consiste en apagar y

prender neuronas durante el procesamiento de acuerdo con una tasa específica.

Batch normalization, consiste en normalizar las salidas z con el fin de mejorar el rendimiento de la red.

Los parámetros y componentes mencionados juegan un rol importante en los resultados que una red convolucional puede presentar. Sin embargo, tan o más importante lo es la forma en que estos componentes se agrupan y ordenan, es decir, la arquitectura de la red.

A lo largo del tiempo, se han propuesto diversas arquitecturas de redes convolucionales, cada una con sus respectivas ventajas o mejoras frente a propuestas previas. Entre las más destacadas, podemos mencionar:

VGG-19, tiene 16 capas convolucionales y 3 capas completamente conectadas. En esta arquitectura, se utiliza a ReLU como función de activación y los filtros son de tamaño 2x2 y 3x3 [17]. La Figura 1.4 muestra la arquitectura de una red VGG-19.

Esta arquitectura, por su simplicidad, aporta en nuestro entrenamiento a la extracción de características básicas de las imágenes faciales.

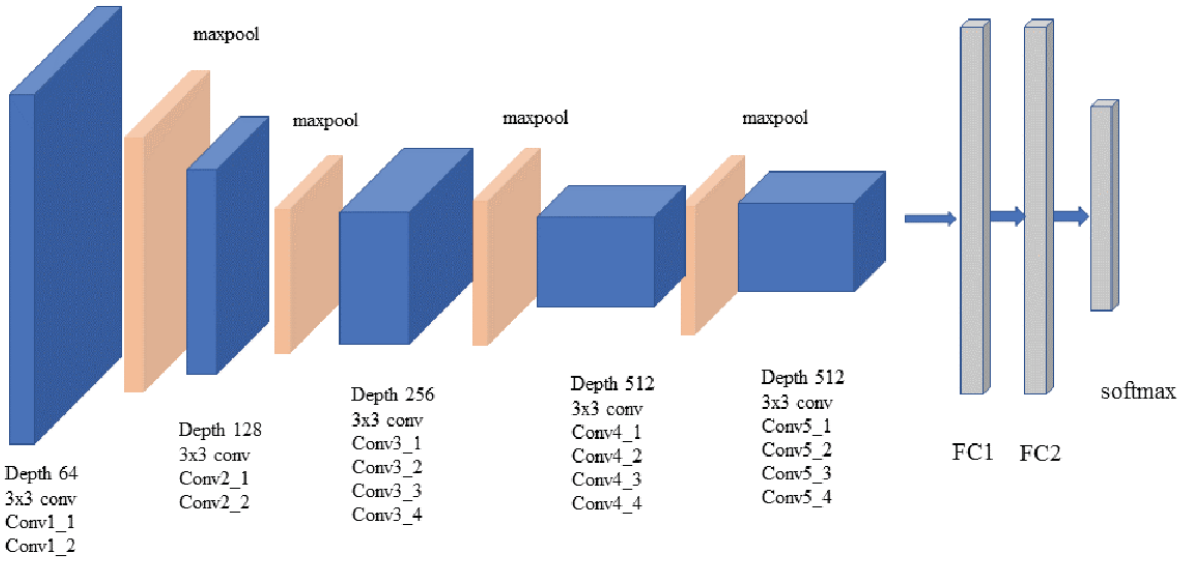
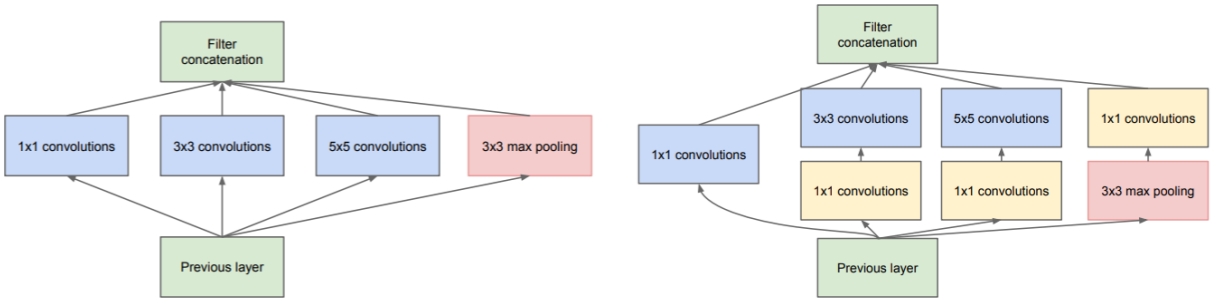


Figura 1.4: VGG-19 architecture. [18]

Inception-v1, introduce el concepto de módulos *Inception*, los cuales agrupan en paralelo varias capas convolucionales con filtros de distinto tamaño (1x1, 3x3 y 5x5), para luego concatenar su resultado y enviarla al siguiente bloque en la red. Además, esta arquitectura cuenta con dos redes auxiliares de capas completamente conectadas en etapas tempranas de la red [19]. La Figura 1.5 muestra los módulos Inception de la red Inception-V1.

Esta red fue elegida dentro de nuestra solución porque intenta extraer características desde diferentes niveles de convolución y concatenarlas antes de pasar a la siguiente capa de red, esto mejora el *accuracy* significativamente, ya que al utilizar imágenes muy similares debe procurarse la discriminación efectiva de características faciales.



(a) Inception module, naïve version (b) Inception module with dimension reductions

Figura 1.5. Módulos Inception que son parte de la red Inception-V1. [19]

DenseNet, es una arquitectura en la que cada capa de red está conectada con todas las demás capas, lo que permite un conocimiento colectivo que ajusta una menor cantidad de parámetros que otras arquitecturas CNN [20]. La Figura 1.6 muestra la arquitectura de una red DenseNet.

Elegimos esta arquitectura dentro de nuestra solución porque permite que se utilice el conocimiento de todas las capas anteriores, fortaleciendo el flujo del gradiente y por lo tanto mejorando el aprendizaje y la precisión obtenida capa a capa.

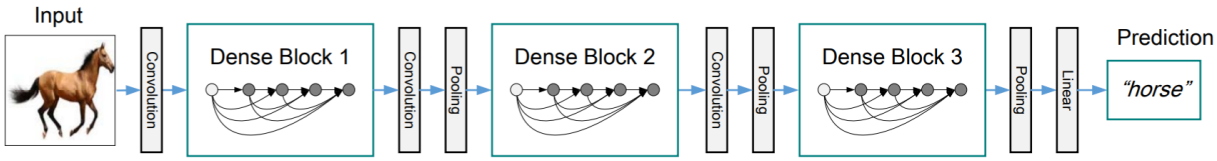


Figura 1.6: Arquitectura de una red DenseNet con tres bloques Dense. [20]

NASNetMobile, es la red resultante de aplicar la técnica *NAS* sobre los conjuntos de datos *ImageNet* y *CIFAR10*. *Neural Architecture Search* (*NAS*) es una técnica desarrollada por Google que busca la mejor configuración de capa convolucional (o "celda") en un espacio de varias configuraciones de ellas. Luego, estas celdas se combinan para formar una arquitectura completa llamada *NASNet*. La red formada por versión reducida de la mejor celda encontrada es llamada NASNetMobile [21]. La

Figura 1.7 muestra la arquitectura de la mejor celda encontrada y la celda reducida.

Esta red fue seleccionada para nuestra solución por tener un ajuste eficiente a partir de *transfer learning* y por optimizar el manejo de la sobrecarga computacional que los modelos CNN demandan.

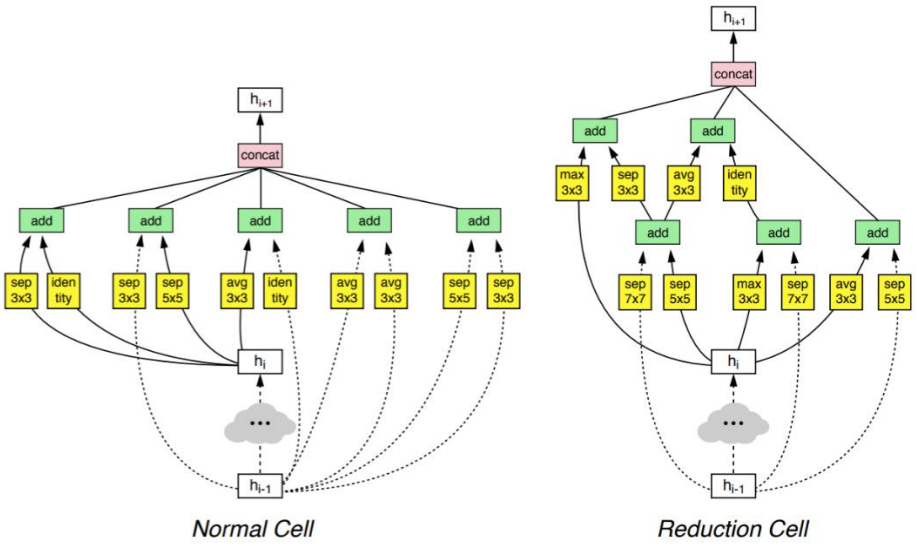


Figura 1.7. Arquitectura de los dos tipos de celdas de las arquitecturas NASNet. [21]

1.4.3 Herramientas para implementación

Dentro del ámbito de la programación, existen herramientas que permiten abstraer procedimientos complejos para agilizar la implementación de los sistemas o productos que se deseen obtener, tanto para el procesamiento interno de los componentes, como para su diseño y muestra a los usuarios finales. Los conceptos a mencionar facilitan la comprensión de las herramientas seleccionadas:

Software Libre, tendencia actual que proclama el acceso al código fuente de herramientas informáticas para su libre uso, ejecución distribución y modificación.

Python, es un lenguaje de programación interpretado que cuenta con licencia propia y está certificado por el movimiento *OpenSource* y que fue diseñado bajo una cultura de simplicidad y legibilidad al usuario. Cuenta con múltiples librerías y documentación que puede llamarse en tiempo de ejecución, además de que su sitio oficial permite la publicación de librerías previa revisión para abstraer cualquier tipo de tarea. [22]

OpenCV, es biblioteca de código abierto que facilita la manipulación de imágenes y videos, así como la aplicación de cientos de modelos de visión por computador en el lenguaje de programación Python. [23]

TensorFlow, biblioteca de código abierto flexible con recursos, funciones, métodos y configuraciones disponibles para implementar aplicaciones de aprendizaje automático. [24]

Keras, es una interfaz de programación creada sobre TensorFlow que facilita y agilita la creación de modelos de aprendizaje automático. [25]

Django, es un *framework* web nacido en el año 2003, basado en el lenguaje de programación Python y que tiene como finalidad la creación rápida de sitios web dinámicos. [26]

Matplotlib, es una biblioteca de código abierto para crear visualizaciones de todo tipo (estáticas, animadas e interactivas) en Python. [27]

CAPÍTULO 2

2. METODOLOGÍA

2.1 Análisis

En esta sección se detallan los requerimientos necesarios para el desarrollo del modelo de aprendizaje profundo implementado. La Figura 2.1 muestra un diagrama de actividades generales de la solución planteada, desde la recolección de los datos, pasando por el entrenamiento del modelo y llegando a la evaluación de este.

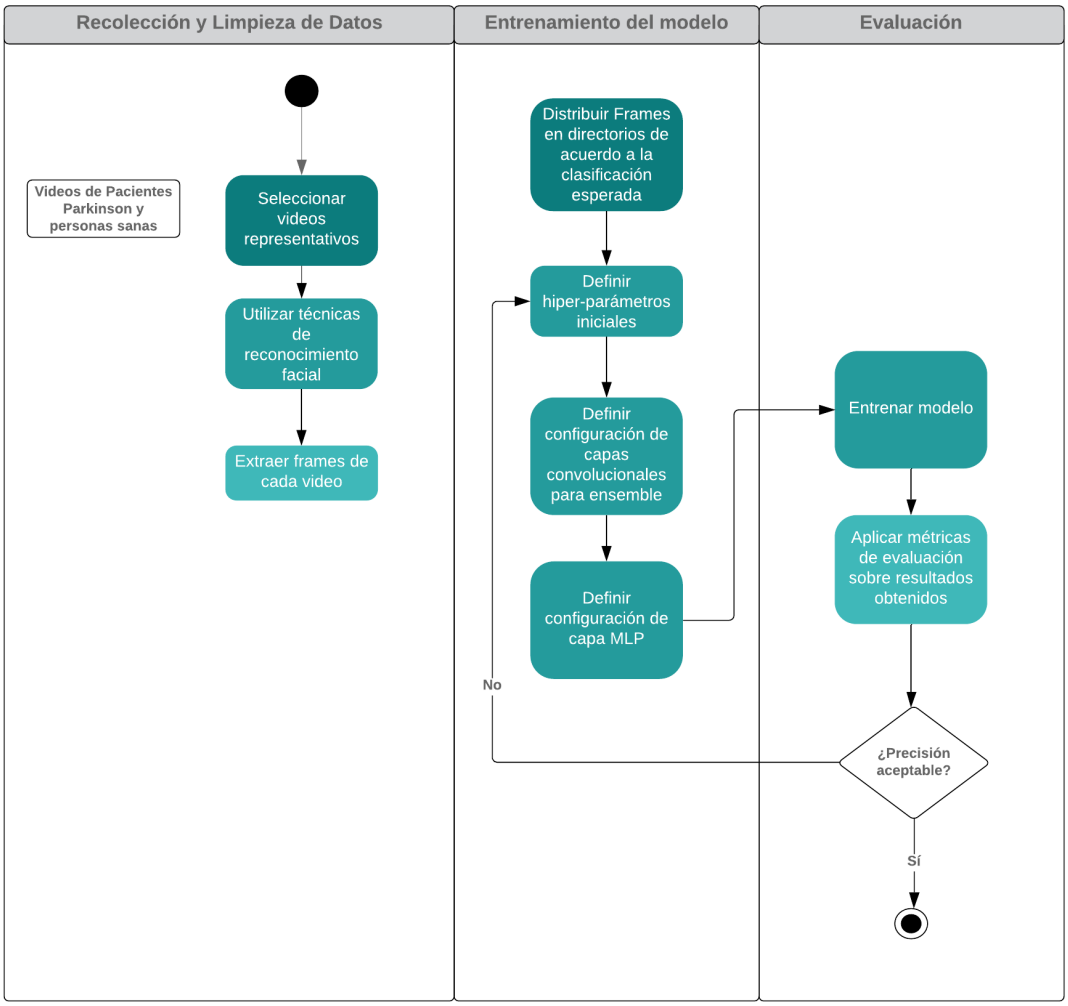


Figura 2.1. Diagrama de actividades de la solución propuesta [Autoría propia].

2.1.1 Requerimientos

2.1.1.1 Requerimientos Funcionales

Tabla 2.1. Descripción de los requerimientos funcionales.

MODELO DE APRENDIZAJE PROFUNDO	
Componente	**Requerimientos**
Datos	• Videos deben cortarse a una duración de 30 segundos. • La entrada del modelo serán frames obtenidos a partir de los videos luego de haber sido sometidos a algoritmos de detección de rostro.

Capas Convolucionales	• Aplicar metodología ensemble entre los modelos de redes neuronales siguientes:
	  ○ NASNetMobile
	  ○ InceptionV3
	  ○ DenseNet201
	  ○ VGG19
	• Los experimentos durante el procesamiento del modelo procurarán mitigar el overfitting a partir de las siguientes técnicas en orden de prioridad:
	  1. Adición de capas de Dropout
	  2. Operaciones de aumento de datos

Capa MLP	• La entrada será la salida de las capas convolucionales, y la salida serán 2 neuronas correspondientes a la existencia o no de la enfermedad de Parkinson.
Evaluación del Modelo	• Utilizar al menos 2 métricas de evaluación para tener referencia de ajuste de hiper-parámetros.

Interfaz Web	• Roles de la interfaz:
	  ○ Médico
	  ○ Administrador
	• El rol médico deberá contar con las siguientes funcionalidades, utilizando solo 1 pantalla:
	  ○ Carga de video
	  ○ Visualización de video cargado

	○ Clasificación de Paciente ○ Espacio para observaciones ○ Guardar registro • El rol administrador debe contar con las siguientes funcionalidades: ○ Visualización, búsqueda y filtrado de todos los registros realizados en el sistema.

2.1.1.2 Requerimientos No Funcionales

- Procesamiento y clasificación debe ser rápida.
- Modelo desarrollado debe integrarse serializado y tener parámetros livianos de procesamiento.
- La interfaz web debe ser sencilla, de pocas pantallas y de pocos clics para interacción del usuario.
- La precisión del diagnóstico debe ser superior a un 90% por tratarse de diagnósticos en el área de la salud.
- Disponibilidad del sistema 24 horas del día.

2.1.2 Alcance

Diseño e implementación de un modelo de aprendizaje profundo, como una herramienta que contribuya al diagnóstico temprano de la enfermedad de Parkinson, accesible mediante la web y enfocada en características faciales de objetos multimedia.

Para el entrenamiento del modelo implementado se utilizó un conjunto de imágenes faciales obtenidas desde videos de monitoreo de pacientes Parkinson y personas sanas, otorgados por el Laboratorio de Bioingeniería y Neuroimagen LNB de la Facultad de Ingeniería Mecánica y Ciencias de la Producción de ESPOL.

Los módulos desarrollados para interacción del usuario fueron:

- Clasificación Paciente por video, a partir del modelo de aprendizaje profundo implementado.

- Registro del monitoreo de un paciente.
- Para el usuario administrador la visualización de todos los registros existentes.

2.1.3 Limitaciones

- La ausencia de instrucciones otorgadas al paciente durante la grabación, lo que generaliza su interacción independientemente de los estados de ánimo o “emociones” que pueda expresar.
- Los datos nuevos que ingresen al sistema no aportarán a la mejora del modelo inmediatamente, ya que su re-entramiento se debe realizar como un proceso aislado y luego integrarse nuevamente a la plataforma web.

2.1.4 Riesgos y beneficios

El principal beneficio de la solución es el apoyo al diagnóstico temprano, que puede ser complementado con la información obtenida sin procesos invasivos en una consulta médica.

El riesgo mayor está representado por la imprecisión del modelo, ya que, aunque sea un pequeño porcentaje, al tratarse de diagnósticos médicos, no puede utilizarse como resultado absoluto, sino como apoyo al criterio del médico que está evaluando a los pacientes.

2.1.5 Usuarios de la solución

- Personal médico especializado en el área de Neurología.
- Pacientes sanos con probabilidad o sintomatología de enfermedad de Parkinson.

2.2 Diseño de la solución

2.2.1 Dataset

Las imágenes de entrenamiento fueron extraídas y procesadas a partir de videos de monitoreo de pacientes Párkinson y personas sanas, cuya edad era de entre 40 y 60 años. El conjunto de datos mencionado fue otorgado por el Centro Integral de Neurociencias (HM CINAC) en España y consiste

en 55 videos de 1 minuto, de los cuales 31 corresponden a pacientes y 24 a personas sanas.

2.2.2 Preprocesamiento de los datos

A continuación, se detalla las distintas etapas de preprocesamiento que fueron realizadas sobre los datos para prepararlos para el entrenamiento de los modelos:

1. **Reducción de la duración de los videos** de 1 minuto a 30 segundos, con el fin de reducir el ruido y acelerar el entrenamiento.
2. **Extracción del rostro de la persona** de cada imagen que compone cada video, con el fin de reducir el ruido visual provocado por el entorno en el que se encuentra cada uno de ellos.
3. **Estandarización de los colores de cada video**, con el fin de que el algoritmo de aprendizaje y optimización converja más rápido.
4. **Aplicación de técnicas de aumento de datos**, con el fin de incrementar la varianza en ellos y mejorar el rendimiento del modelo. Estas técnicas se describen en la Tabla 2. de la sección 2.2.5.

2.2.3 Prototipado de la solución

La Figura 2.2 muestra la arquitectura de la solución propuesta. Esta se basó en redes de tipo CNN, diseñadas para procesar las imágenes obtenidas del procesamiento de los videos. Posteriormente, las salidas de las capas CNN ingresan a una capa tipo MLP, cuya salida es la clasificación esperada del modelo, respecto a la existencia o no de la enfermedad de Parkinson en el paciente.

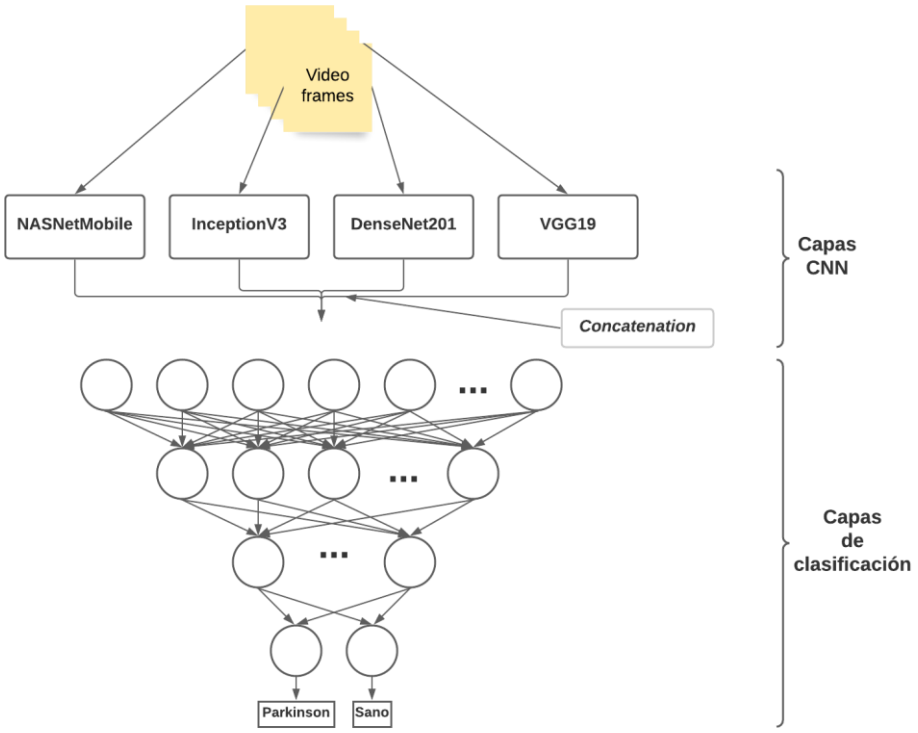


Figura 2.2. Diagrama de la arquitectura "Ensemble" del modelo de la solución. [Autoría propia].

2.2.4 Hiper-parámetros básicos

La Tabla 2.2 muestra los hiper-parámetros básicos del modelo utilizado y los valores asignados en los experimentos. La nomenclatura ha sido colocada por los autores para facilidad de la descripción de los experimentos en secciones posteriores.

Tabla 2.2. Listado de hiper-parámetros.

Nomenclatura	Nombre	Valores utilizados
BS	Batch Size	32
EP	Epochs	1000, 2000
OP	Optimizer	Adam
LR	Learning Rate	0.0001, 0.00001
LOSS	Loss Function	mae

Las arquitecturas de las redes CNN utilizadas fueron:

- NASNetMobile
- InceptionV3
- DenseNet201
- VGG19

2.2.5 Experimentos realizados

La Tabla 2.3, a continuación, muestra las configuraciones hechas en el modelo para mejorar su rendimiento y evitar problemas de *overfitting*. Las configuraciones descritas reflejan los experimentos sobre las capas convolucionales de la red; en la arquitectura MLP se mantuvieron en todos los experimentos 256 neuronas de entrada, 2 neuronas de salida y una capa de *dropout* intermedia.

Tabla 2.3. Descripción de experimentos.

No.	Arquitectura de Red	Dataset	Operaciones Aumento de Datos	Dropout
1	NASNetMobile InceptionV3 DenseNet201 VGG19	Sin punto blanco en rostro	zoom_range: 0.2 shear_range: 0.2	0.5
2				No
3			zoom_range: 0.2 shear_range: 0.2 width_shift_range: 0.5 height_shift_range: 0.5 horizontal_flip: True rotation_range: 20 brightness_range:(0.5, 1.5)	0.5
4				No
5		Sin punto y sin imágenes duplicadas de años diferentes	zoom_range: 0.2 shear_range: 0.2	0.5
6				No
7			zoom_range: 0.2 shear_range: 0.2 width_shift_range: 0.5 height_shift_range: 0.5	0.5
8				No

			horizontal_flip: True rotation_range: 20 brightness_range:(0.5, 1.5)	

Como puede observarse se realizaron 8 experimentos por cada una de las arquitecturas de redes convolucionales utilizadas; es decir, en total se desarrollaron 32 experimentos. Además, es necesario indicar que también se realizaron experimentos con capas de *dropout*, las cuales se ubicaron al final de cada bloque convolucional.

Finalmente, las operaciones de aumento de datos aplicadas, previo al entrenamiento de cada arquitectura, corresponde a cambios en el ancho, alto, brillo, acercamiento y rotación del set de datos ya existente.

2.2.6 Métodos de evaluación

Se utilizaron cinco métricas que son comúnmente usadas para evaluar modelos de clasificación y para definir los mejores parámetros en los experimentos realizados. [28]

Para comprender las fórmulas respectivas es necesario entender los campos correspondientes a la matriz de confusión como se muestran en la Tabla 2.4.

Tabla 2.4. Campos de matriz de confusión.

		REALIDAD	
		0	1
PREDICCIÓN	0	TN	FP
	1	FN	TP

2.2.6.1 *Exactitud o Accuracy*

Es la métrica más común utilizada en problemas de clasificación. Consiste en determinar la cantidad de predicciones acertadas sobre el total de predicciones hechas en el conjunto de pruebas.

A continuación, se muestra la Ecuación (1) para calcular el *accuracy*:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

Equivalente al total de predicciones acertadas sobre el total de predicciones realizadas.

2.2.6.2 *Precision*

Precision es la fracción de predicciones acertadas sobre el total de predicciones obtenidas para una clase dada en el conjunto de pruebas. Se puede interpretar como la probabilidad de que un ejemplo tomado aleatoriamente de entre los ejemplos clasificados como una clase dada realmente pertenezca a dicha clase.

La Ecuación (2) para calcular la precisión de un modelo se muestra a continuación:

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

Equivalente a las predicciones acertadas sobre el total de predicciones para una clase dada.

2.2.6.3 *Recall*

Recall es la fracción de predicciones acertadas sobre el total de ejemplos de una clase dada en el conjunto de pruebas. Se puede interpretar como la probabilidad de que un ejemplo tomado

aleatoriamente de entre todos aquellos pertenecientes a una clase dada sea clasificado como tal.

La Ecuación (3) para calcular el *recall* de un modelo se muestra a continuación:

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

Equivalente a las predicciones acertadas sobre el total de valores esperados para una clase dada.

2.2.6.4 F1

La intención de F1 es integrar a las métricas *precision* y *recall* en una sola. Se describe como la media armónica entre *precision* y *recall*.

La Ecuación (4) para calcular el f1 de un modelo se muestra a continuación:

$$f1 = 2 * \frac{precision * recall}{precision + recall} \quad (4)$$

Es necesario indicar que los valores de *precision* y *recall* corresponden a la misma clase.

2.2.6.5 AUC ROC (Área bajo la curva ROC)

Es la métrica utilizada para determinar la exactitud de un modelo basado en probabilidades, identifica las tasas de verdaderos positivos y falsos positivos. La Figura 2.3 muestra la curva ROC marcada en azul y su área sombreada en gris.

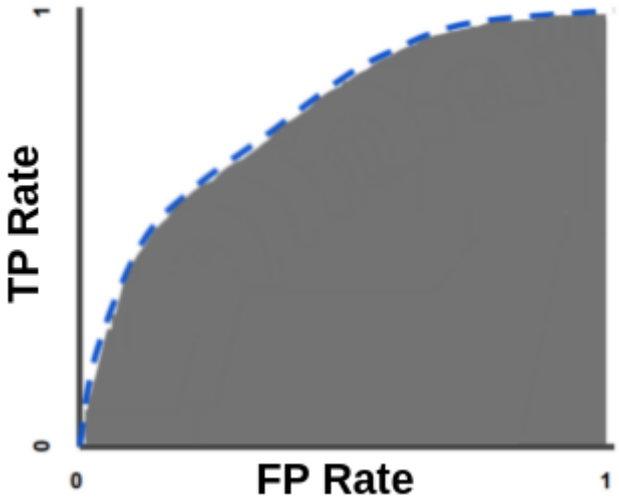


Figura 2.3. Métrica de evaluación AUC para clasificación binaria. [15]

La Ecuación (5) para calcular la tasa de verdaderos positivos de un modelo se muestra a continuación:

$$TPR = \frac{TP}{TP + FN} \tag{5}$$

La Ecuación (6) para calcular la tasa de falsos positivos de un modelo se muestra a continuación:

$$FPR = \frac{FP}{TN + FP} \tag{6}$$

La Ecuación (7) para calcular el AUC ROC de un modelo se muestra a continuación:

$$AUC = \int TPR\, d(FPR) \tag{7}$$

CAPÍTULO 3

3. RESULTADOS Y ANÁLISIS

3.1 Modelo Desarrollado

Los datos que se obtuvieron en la etapa del preprocesamiento carecían de expresiones y movimientos que diversificaban las expresiones faciales; por lo tanto, se generaron imágenes repetidas por cada video y el aumento de datos le dio mayor significancia a este problema. Debido a esto, fue necesario eliminar complejidad o regularizar las redes de aprendizaje profundo ya existentes, ubicando capas de Dropout internas que reduzcan el overfitting en la etapa de entrenamiento y validación; esto cerraba la brecha entre la predicción en la etapa de testing.

La Tabla 3.1, a continuación, muestra los resultados de los experimentos iniciales, descritos en el capítulo 2, utilizando cada red de manera particular con un conjunto de datos correspondiente al análisis de 49500 *frames*, con el optimizador de SGD[2]. Los experimentos se realizaron a partir de una grilla de valores para el momento del optimizador, taza de aprendizaje y épocas.

Tabla 3.1. Resultados de experimentos por red.

Red	momentum	learningrate	epochs	accuracy	precision	recall	auc	f1
Inception	0.5	0.0001	100	0.5856	0.4933	0.5820	0.5851	0.5340
Inception	0.5	0.0001	1000	0.5701	0.4784	0.5959	0.5741	0.5307
Inception	0.5	0.00001	100	0.5875	0.4953	0.5913	0.5881	0.5391
Inception	0.5	0.00001	1000	0.6068	0.5173	0.5403	0.5965	0.5286
Inception	0.9	0.0001	100	0.6261	0.5443	0.5125	0.6084	0.5279
Inception	0.9	0.0001	1000	0.5921	0.5000	0.4995	0.5777	0.4998
Inception	0.9	0.00001	100	0.6174	0.5407	0.4124	0.5855	0.4679
Inception	0.9	0.00001	1000	0.6306	0.5671	0.3994	0.5947	0.4687
Mobile	0.5	0.0001	100	0.5735	0.4815	0.5913	0.5763	0.5308
Mobile	0.5	0.0001	1000	0.5713	0.4757	0.4986	0.5600	0.4869

[2] SGD.- Optimizador de descenso de gradiente estocástico.

Mobile	0.5	0.00001	100	0.5834	0.4906	0.5589	0.5796	0.5225
Mobile	0.5	0.00001	1000	0.5928	0.5009	0.4958	0.5777	0.4984
Mobile	0.9	0.0001	100	0.5319	0.4360	0.5023	0.5273	0.4668
Mobile	0.9	0.0001	1000	0.4949	0.3970	0.4588	0.4893	0.4256
Mobile	0.9	0.00001	100	0.5403	0.4471	0.5366	0.5397	0.4878
Mobile	0.9	0.00001	1000	0.5701	0.4709	0.4347	0.5491	0.4520
VGG19	0.5	0.0001	1000	0.5001	0.4079	0.4516	0.5123	0.4565
VGG19	0.5	0.00001	1000	0.5123	0.4738	0.4034	0.5205	0.4352
VGG19	0.9	0.0001	1000	0.4895	0.4355	0.3876	0.5135	0.4223
VGG19	0.9	0.00001	1000	0.4916	0.4233	0.3967	0.5002	0.4332
Nasnet	0.5	0.0001	1000	0.5225	0.4839	0.5332	0.5271	0.4778
Nasnet	0.5	0.00001	1000	0.6048	0.5693	0.4323	0.5736	0.4698
Nasnet	0.9	0.0001	1000	0.5123	0.4987	0.5003	0.5184	0.4475
Nasnet	0.9	0.00001	1000	0.5278	0.4537	0.4892	0.4932	0.4179

Los resultados obtenidos muestran que a pesar de que el mejor experimento obtiene un *accuracy* de 0.63 el *recall* es muy bajo, lo que quiere decir que hay una probabilidad muy grande de obtener falsos positivos; lo cual es muy perjudicial en el campo de la medicina.

La Tabla 3.2 muestra los resultados obtenidos en los experimentos utilizando la metodología ensamblador con los resultados concatenados de las 4 redes descritas anteriormente. Es necesario indicar que en estos experimentos se utilizaron 8250 imágenes correspondientes a 5 por cada segundo de video, lo que corresponde a tan solo la sexta parte del set de datos inicial; la grilla alternó valores de taza de aprendizaje, tamaño del lote, optimizadores y sus respectivos momentos; las épocas fueron 10000 para todos los experimentos.

Tabla 3.2. Resultados de experimentos metodología ensemble.

Mom.	L_rate	batch	Opt.	beta1	beta2	Acc.	Prec.	recall	auc	f1
	0.00001	128	Adam	0.5	0.995	0.466	0.579	0.547	0.438	0.563
0	0.00001	64	SGD			0.481	0.588	0.573	0.449	0.581
	0.000001	32	Adam	0.5	0.995	0.505	0.605	0.606	0.470	0.606

	0.00001	128	Adam	0.5	0.995	0.524	0.621	0.620	0.492	0.621
0	0.00001	64	SGD			0.514	0.610	0.625	0.476	0.618
	0.00001	64	Adam	0.5	0.995	0.527	0.620	0.637	0.489	0.628
0	0.00001	64	SGD			0.528	0.619	0.644	0.488	0.631
0.5	0.00001	64	SGD			0.539	0.625	0.662	0.497	0.643
	0.00001	64	Adam	0.7	0.995	0.553	0.633	0.685	0.507	0.658
0.5	0.00001	64	SGD			0.554	0.633	0.688	0.508	0.659
	0.0001	32	Adam	0.5	0.995	0.651	0.736	0.691	0.637	0.713
	0.00001	64	Adam	0.5	0.9	0.562	0.637	0.702	0.514	0.668
0.9	0.00001	64	SGD			0.567	0.641	0.707	0.520	0.672
	0.00001	64	Adam	0.7	0.995	0.568	0.640	0.711	0.519	0.674
0.5	0.00001	64	SGD			0.580	0.646	0.731	0.529	0.686
0.9	0.00001	64	SGD			0.581	0.647	0.732	0.530	0.687
	0.00001	64	Adam	0.5	0.995	0.596	0.658	0.741	0.546	0.697
	0.00001	32	Adam	0.5	0.995	0.653	0.711	0.754	0.619	0.732
	0.00001	64	Adam	0.5	0.995	0.598	0.655	0.759	0.543	0.703
	0.00001	64	Adam	0.7	0.995	0.607	0.659	0.773	0.550	0.712
	0.00001	64	Adam	0.3	0.995	0.610	0.662	0.773	0.555	0.713
	0.00001	64	Adam	0.5	0.999	0.639	0.689	0.774	0.593	0.729
	0.00001	64	Adam	0.5	0.995	0.687	0.732	0.789	0.652	0.760
	0.00001	64	Adam	0.5	0.7	0.662	0.703	0.798	0.615	0.747
0.9	0.00001	64	SGD			0.633	0.671	0.814	0.571	0.735

La reducción de datos fue significativamente importante para el rendimiento del modelo, debido a la similitud entre las imágenes extraídas de cada video. Asimismo, puede se puede notar que los momentos de los optimizadores fueron los parámetros que más influyeron en el proceso de aprendizaje.

Tabla 3.3. Métricas del modelo final.

Accuracy	Precision	Recall	AUC	f1
0.686695279	0.7322335025	0.7893296854	0.6515772851	0.7597103357

Los resultados del mejor modelo obtenido se muestran en la Tabla 3.3, en el cual se observa un *recall* alto, es decir que existe poca probabilidad de falsos positivos en el diagnóstico.

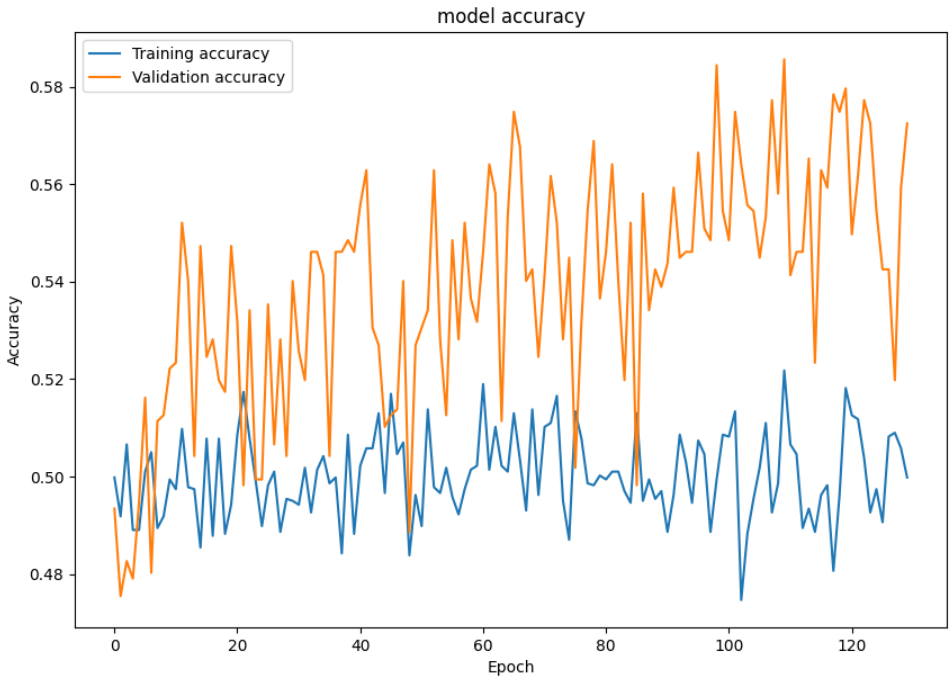


Figura 3.1. Precisión inicial de modelo desarrollado [Autoría propia].

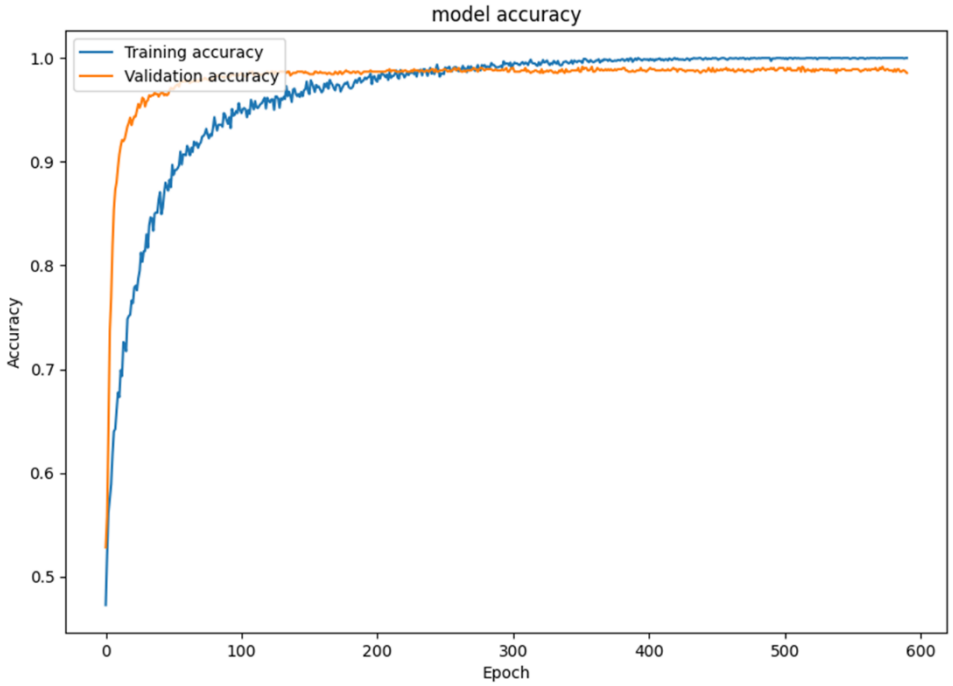


Figura 3.2. Precisión final de modelo desarrollado [Autoría propia].

La Figura 3.1 y la Figura 3.2 muestran el progreso de la precisión alcanzada en el modelo. Por un lado, la Figura 3.1 muestra los resultados de la metodología aplicada a los modelos preexistentes de las redes convolucionales VGG19, DenseNet, InceptionNet y NasnetMobile; mientras que la Figura 3.2 muestra los resultados de las redes personalizadas con su nivel de complejidad y conjunto de datos reducido.

En ambas figuras se muestra un mejor desempeño durante la validación, sin embargo, existe una brecha muy grande respecto a la tabla de resultados de las métricas, por lo que se pudo verificar que el conjunto de datos a utilizarse para mejorar el desempeño de la red debió constituirse con mayor diversidad de expresiones faciales; es decir los videos de los pacientes no debían ser solo estáticos sino más bien deberían grabarse siguiendo patrones de movimiento y/o gesticulares para que no existan demasiadas similitudes entre los frames del grupo de entrenamiento.

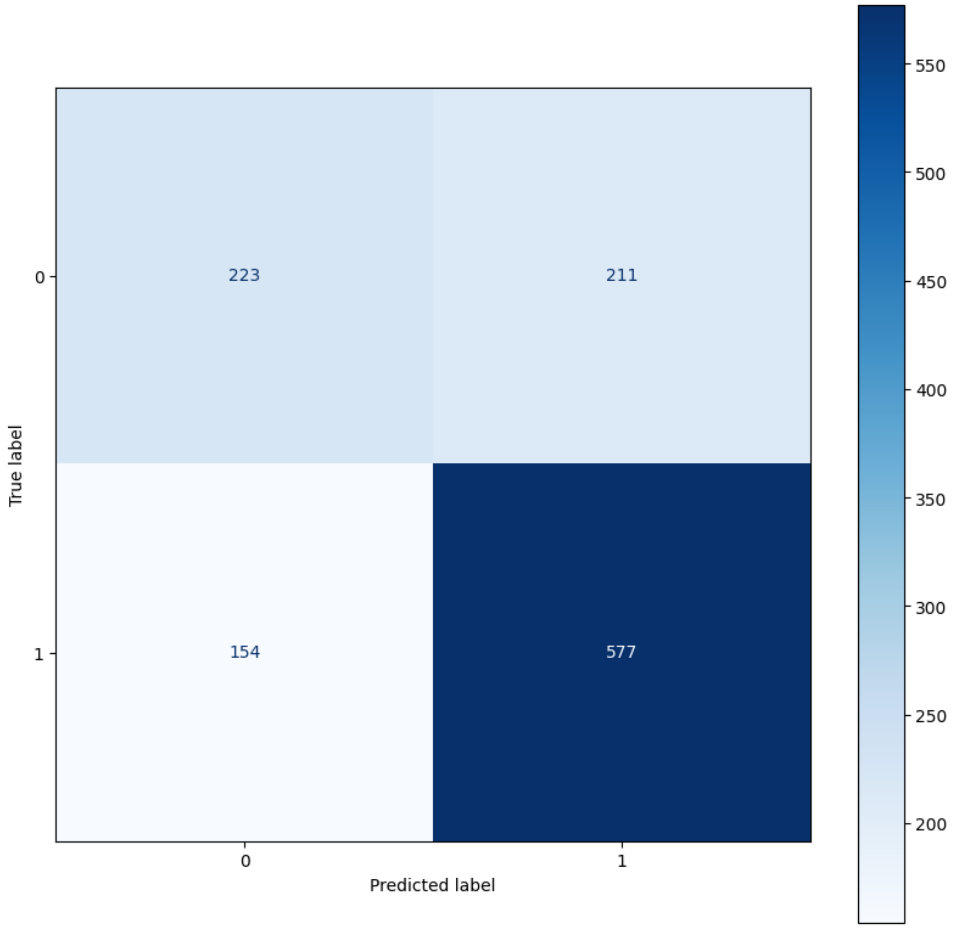


Figura 3.3. Matriz de confusión del modelo desarrollado [Autoría propia].

La matriz de confusión, en la Figura 3.3, muestra que el modelo tiene más aciertos que errores en las predicciones. Dado que es mayor porcentaje la predicción positiva ante un valor positivo esperado, así como la predicción esperada negativa.

3.2 Prototipo Web

3.2.1 Backend

La integración entre el modelo obtenido y el registro de las consultas para el usuario tipo doctor fue realizada como un servicio tipo API REST, que puede ser consumido desde cualquier arquitectura y cumplir con el requisito de escalabilidad necesario según los estándares de ingeniería de software.

La Figura 3.4, a continuación, evidencia el consumo del API REST para creación de una nueva consulta en el sistema desarrollado.

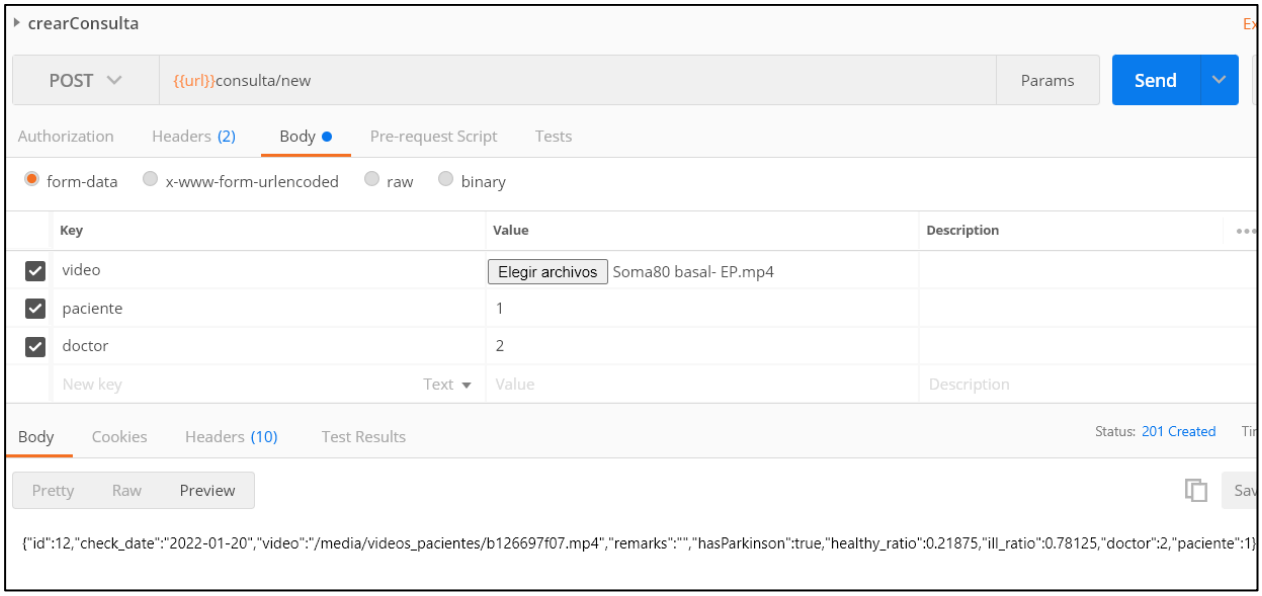


Figura 3.4. Conexión con el backend desde Postman [Autoría propia].

3.2.2 Frontend

La interfaz de usuario para el prototipo web, cumplió los requerimientos del cliente, respecto a la previsualización del video de pacientes durante el registro de las consultas, así como el cálculo automático del diagnóstico temprano de Parkinson que se convierte en una herramienta muy valiosa para los médicos durante la evaluación de sus pacientes.

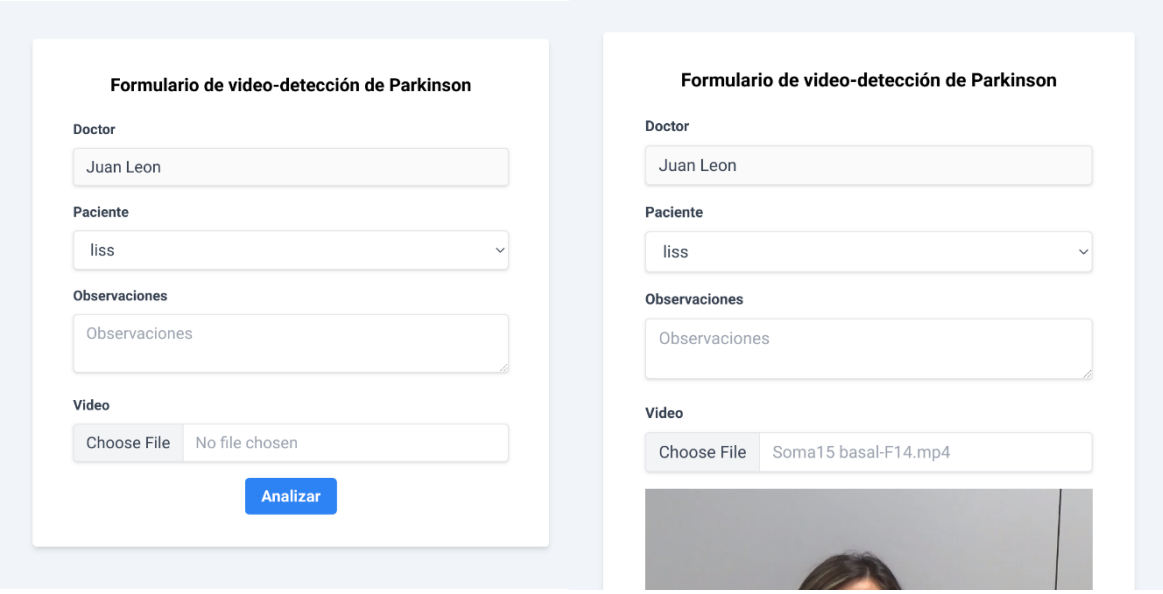


Figura 3.5. Formulario implementado para registro de consultas [Autoría propia].

Como puede observarse en la Figura 3.5, el formulario para el registro de consultas corresponde al rol de médico; en el cual selecciona un paciente existente o registra uno nuevo, ingresa las observaciones de la consulta y el video de monitoreo para su respectivo análisis.

Figura 3.6. Resultados del diagnóstico durante las consultas [Autoría propia].

La Figura 3.6 muestra los resultados del análisis del video, con los porcentajes correspondientes a predicción positiva y negativa para Párkinson; información útil que complementará el diagnóstico del médico evaluador.

Los respectivos manuales para uso del prototipo web, así como de instalación de sus componentes se encuentran en los apéndices A y B.

CAPÍTULO 4

4. CONCLUSIONES Y RECOMENDACIONES

Conclusiones

- El modelo desarrollado alcanzó un porcentaje de precisión de aproximadamente el 70%, lo que permite sea considerado como un apoyo al diagnóstico temprano de pacientes con la enfermedad de Párkinson.
- La falta de diversidad de expresiones en múltiples imágenes de una misma persona limita el rendimiento de las redes de aprendizaje profundo y aumenta el *overfitting* del modelo.
- La tasa de aprendizaje y los momentos del optimizador fueron los hiper-parámetros que más influyeron en la estabilización del comportamiento del error del modelo.

Recomendaciones

- Solicitar a los pacientes realizar gesticulaciones o imitar expresiones durante el monitoreo, para diversificar las imágenes que se utilizan durante el entrenamiento del modelo.
- Solicitar a los pacientes no cubrir su cara con ningún objeto (ni sus manos) durante la grabación.

BIBLIOGRAFÍA

[1] L. Alonso-Recio, J. M. Serrano-Rodríguez, F. Carvajal-Molina, A. Loeches-Alonso y P. Martín-Plasencia, «Reconocimiento de expresiones faciales de emociones en la enfermedad de Parkinson: una revisión teórica,» *neurologia.com,* 2012.

[2] S. R. Livingstone, E. Vezer, L. M. McGarry, A. E. Lang y F. A. Russo, «Deficits in the Mimicry of Facial Expressions in Parkinson's Disease,» *Frontiers in Psychology,* 2016.

[3] M. Arroyo Menéndez y L. Finkel Morgenstern, «Dependencia e impacto social de la enfermedad de Parkinson,» *REVISTA ESPAÑOLA DE DISCAPACIDAD,* vol. 1, nº 2, 2013.

[4] A. Esteva, K. Chou, S. Yeung, N. Naik, A. Madani, A. Mottaghi, Y. Liu, E. Topol, J. Dean y R. Socher, «Deep learning-enabled medical computer vision,» *Digital medicine,* pp. 4,5, 2021.

[5] A. Storstein y O.-B. Tysnes, «Epidemiology of Parkinson's disease,» *Journal of neural transmission,* 2006.

[6] W. Rivas Asanza y B. Mazón Olivo, Redes neuronales artificiales aplicadas al reconocimiento de patrones, Machala: Editorial UTMACH, 2018.

[7] S. Xu, Z. Wang, J. Sun, Z. Zhang, Z. Wu, T. Yang, G. Xue y C. Cheng, «Using a deep recurrent neural network with EEG signal to detect Parkinson’s disease,» de *Annals of translational medicine*, 2020.

[8] S. Lee, R. Hussein y M. J. McKeown, «A Deep Convolutional-Recurrent Neural Network Architecture for Parkinson’s Disease EEG Classification,» *2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP),* 2020.

[9] U. Anusri, G. Dhatchayani, Y. Princely Angelinal y S. Kamalraj, «An Early Prediction of Parkinson’s Disease Using Facial Emotional Recognition,» *Journal of Physics: Conference Series,* 2021.

[10] B. Jin, Y. Qu, L. Zhang y Z. Gao, «Diagnosing Parkinson Disease Through Facial Expression Recognition: Video Analysis,» *JOURNAL OF MEDICAL INTERNET RESEARCH,* 2020.

[11] M. R. Ali, T. Myers, E. Wagner, H. Ratnu, E. R. Dorsey y E. Hoque, «Facial expressions can detect Parkinson's disease: preliminary evidence from videos collected online,» *Digital Medicine,* 2021.

[12] [En línea]. Available: http://grupo.us.es/gtocoma/pid/pid10/RedesNeuronales.htm. [Último acceso: 26 11 2021].

[13] E. García Sánchez, *Introducción a las redes neuronales de convolución.,* Zaragoza, 2019.

[14] A. Hidaka y T. Kurita, «Data Visualization for Deep Neural Networks Based on Interlayer Canonical Correlation Analysis,» *Transactions of the Institute of Systems Control and Information Engineers,* 2018.

[15] M. Lorenzo, F. Larussi, V. Cifuentes y G. Rodriguez, «Clasificador multiclase con redes neuronales convolucionales,» de *XLIX Jornadas Argentinas de Informática e Investigación Operativa*, 2020.

[16] A. Martín-Plasencia, «Diseño de redes neuronales antagónicas para estegoanálisis,» 2020.

[17] K. Simonyan y A. Zisserman, «Very Deep Convolutional Networks for Large-Scale Image Recognition,» de *International Conference on Learning Representations 2015*, San Diego, 2014.

[18] S. Charan Arishanapally, «Sai Charan Arishanapally,» 16 4 2019. [En línea]. Available: https://saicharanars.medium.com/building-vgg19-with-keras-f516101c24cf. [Último acceso: 23 1 2022].

[19] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke y A. Rabinovich, «Going Deeper with Convolutions,» de *2015 IEEE Conference on Computer Vision and Pattern Recognition*, Boston, 2014.

[20] G. Huang, Z. Liu, L. van der Maaten y K. Q. Weinberger, «Densely Connected Convolutional Networks,» de *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, 2017.

[21] B. Zoph, V. Vasudevan, J. Shlens y Q. V. Le, «Learning Transferable Architectures for Scalable Image Recognition,» de *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, 2017.

[22] I. Challenger-Pérez, Y. Díaz Ricardo y R. A. Becerra-García, «El lenguaje de programación Python,» 2014.

[23] OpenCV, «Open Source Computer Vision,» [En línea]. Available: https://docs.opencv.org. [Último acceso: 30 11 2021].

[24] Google, «TensorFlow,» Google, [En línea]. Available: https://www.tensorflow.org. [Último acceso: 25 11 2021].

[25] Keras, «Keras,» [En línea]. Available: https://keras.io. [Último acceso: 30 11 2021].

[26] S. Dauzon, A. Bendoraitis y A. Ravindran, Django: Web Development with Python, Birmingham, 2016.

[27] T. M. D. team, «matplotlib,» [En línea]. Available: https://matplotlib.org. [Último acceso: 1 12 2021].

[28] R. Borja-Robalino, A. Monleon-Getino y J. Rodellar Benedé, «Estandarización de Métricas de Rendimiento para Clasificadores Machine y Deep Learning,» de *VI Congreso Internacional de Ciencia, Tecnología e Innovación para la Sociedad, CITIS*, Guayaquil, 2020.

[29] K. He, X. Zhang, S. Ren y J. Sun, «Deep Residual Learning for Image Recognition,» de *2016 IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, 2016.

[30] J. Martinez Heras, «Precision, Recall, F1, Accuracy en clasificación,» 9 10 2020. [En línea]. Available: https://www.iartificial.net/precision-recall-f1-accuracy-en-clasificacion/. [Último acceso: 23 1 2022].

APÉNDICES

APÉNDICE A

Manual de instalación del sistema de video detección de Parkinson

El código fuente tiene dos directorios principales, uno llamado *backend* y otro *frontend*, en donde se encuentra la implementación de dichos componentes del sistema.

El *backend* fue desarrollado usando el *framework* **Django**. El *frontend* fue desarrollado usando el *framework* **React**.

A continuación, se describe la instalación y ejecución del sistema.

Instalación de Backend

El backend del proyecto ha sido desarrollado y probado con las siguientes dependencias:

- Python 3.8

Una vez cumplida dicha dependencia, realizar los siguientes pasos:

1. Ingresar al directorio ***backend***.
2. Instalar las librerías de Python de las cuales depende el proyecto, ejecutando el comando **pip install -r requirements.txt**. La figura A.1 muestra la salida del comando, conforme se instalan cada uno de los requerimientos especificados en el archivo *requirements.txt.*

```
(.venv) xavierfigueroav@friday:~/Documents/integradorapp$ cd backend/
(.venv) xavierfigueroav@friday:~/Documents/integradorapp/backend$ pip install -r requirements.txt
Collecting absl-py==1.0.0
  Using cached absl_py-1.0.0-py3-none-any.whl (126 kB)
Collecting asgiref==3.2.10
  Using cached asgiref-3.2.10-py3-none-any.whl (19 kB)
Collecting astunparse==1.6.3
  Using cached astunparse-1.6.3-py2.py3-none-any.whl (12 kB)
Collecting black==21.12b0
  Using cached black-21.12b0-py3-none-any.whl (156 kB)
Collecting cachetools==4.2.4
  Using cached cachetools-4.2.4-py3-none-any.whl (10 kB)
Collecting certifi==2020.6.20
  Using cached certifi-2020.6.20-py2.py3-none-any.whl (156 kB)
Collecting chardet==3.0.4
  Using cached chardet-3.0.4-py2.py3-none-any.whl (133 kB)
Collecting click==8.0.3
  Using cached click-8.0.3-py3-none-any.whl (97 kB)
```

Figura A.1. Salida del comando *pip install* en la línea de comandos [Autoría propia].

3. Una vez instaladas las dependencias, se procede a configurar la base de datos en el proyecto. Para esto, es necesario editar el archivo ***settings.py*** que se encuentra en el directorio ***backend/backend***. La configuración depende del motor de base de datos que se requiera usar. El proyecto fue probado con Sqlite. Para más información sobre cómo configurar otros motores, leer https://docs.djangoproject.com/en/4.0/ref/settings/. La figura A.2 muestra las líneas de código necesarias para configurar un proyecto con Sqlite.

Figura A.2. Sección del archivo settings.py en donde se configura el motor de bases de datos a utilizar [Autoría propia].

4. Una vez configurada la base de datos, ejecutar el comando **python manage.py migrate**. La figura A.3 muestra la salida del comando mencionado conforme se crean las entidades iniciales de la base de datos a utilizarse.

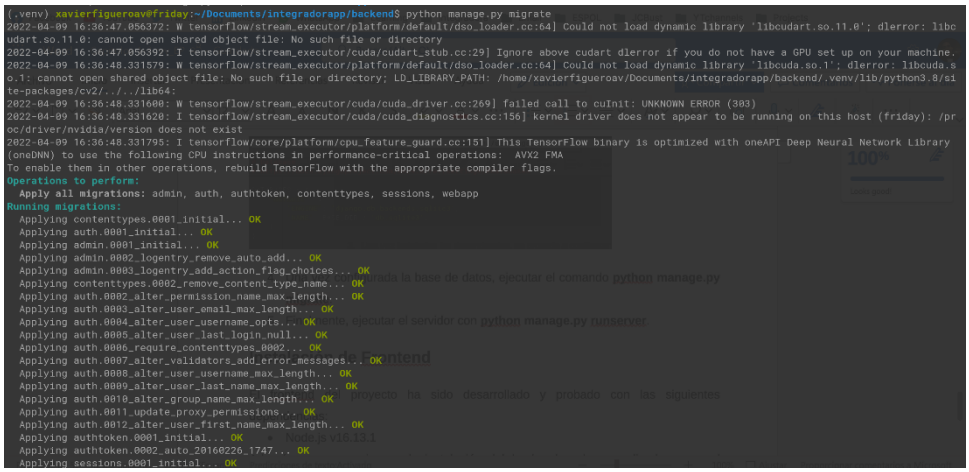


Figura A.3. Salida del comando *python manage.py migrate* en la línea de comandos [Autoría propia].

5. Finalmente, ejecutar el servidor con **python manage.py runserver**. La figura A.4 muestra la salida esperada una vez se ha puesto en servicio del servidor.

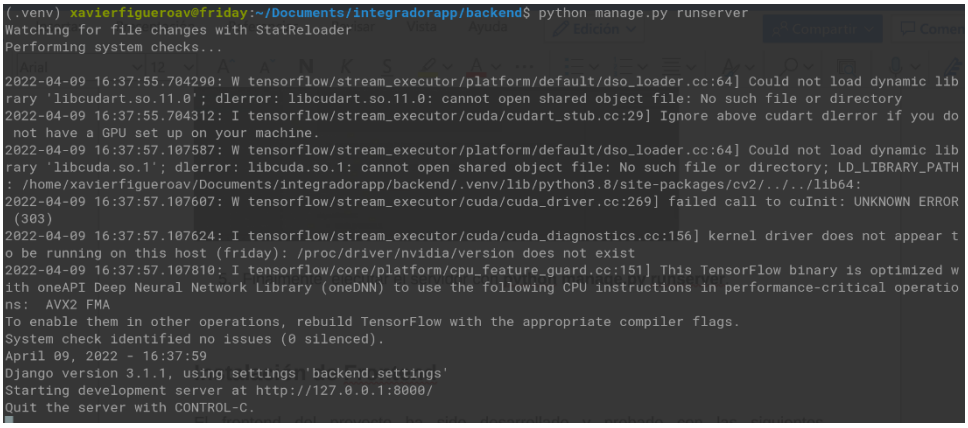


Figura A.4. Salida del comando *python manage.py runserver* en la línea de comandos [Autoría propia].

Instalación de Frontend

El frontend del proyecto ha sido desarrollado y probado con las siguientes dependencias:

- Node.js v16.13.1

Además, es necesario que la instalación del *backend* se haya realizado y que el servidor esté en ejecución.

Una vez cumplida dicha dependencia, realizar los siguientes pasos:

1. Ingresar al directorio ***frontend***.
2. Ejecutar **npm install** para instalar las dependencias del proyecto. La figura A.5 muestra la salida del comando mencionado, conforme se instalan las dependencias necesarias.

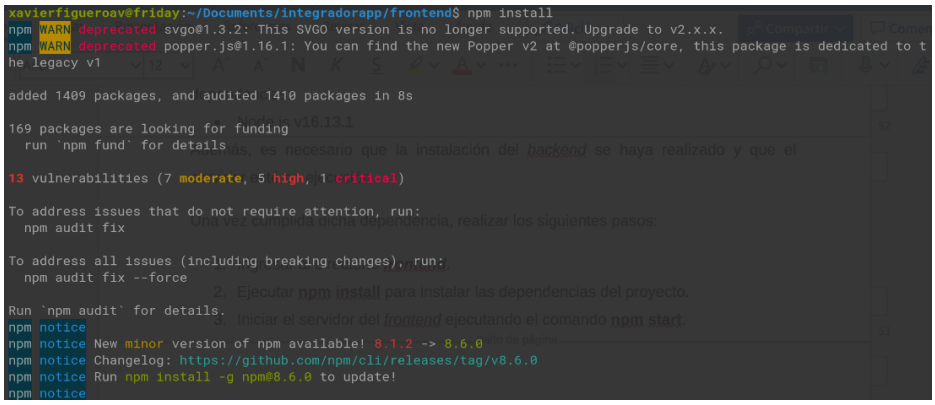


Figura A.5. Salida del comando *npm install* en la línea de comandos [Autoría propia].

3. Iniciar el servidor del *frontend* ejecutando el comando **npm start**. Automáticamente se abrirá una ventana del navegador con la página de inicio del sistema. La figura A.6 muestra la pantalla de inicio en el navegador una vez que se encuentre en funcionamiento el servidor frontend.

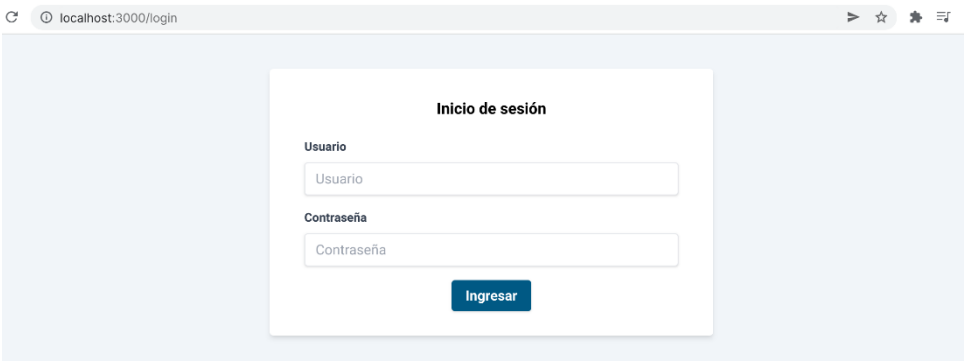


Figura A.6. Página principal del sistema abierta automáticamente luego de ejecutar el comando *npm start* [Autoría propia].

APÉNDICE B

Manual de uso del sistema de video detección de Parkinson

Una vez instalado el sistema según lo descrito en el Manual de instalación, el sistema será accesible a través del navegador web, usando la URL http://localhost:3000, si es que el administrador no configuró un dominio o URL diferente.

Para hacer uso del sistema, sigue los siguientes pasos:

1. Ingresar a la URL proporcionada por el administrador.
2. Si es la primera vez que ingresas, se te redirigirá a la pantalla de inicio de sesión.
3. En esta pantalla, ingresa las credenciales proporcionadas por el administrador del sistema. La figura B.1 muestra los campos requeridos para inicio de sesión en la pantalla inicial del sistema.

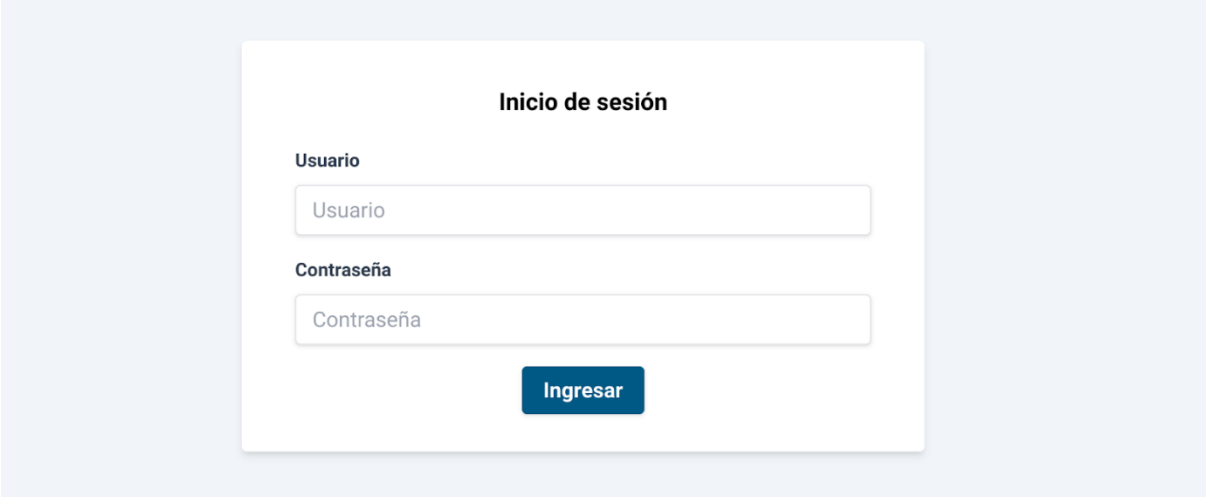


Figura B.1. Formulario de inicio de sesión [Autoría propia].

4. Una vez iniciada la sesión, se te redirigirá a la pantalla principal del sistema, la cual es el formulario para ingresar el video del paciente. La figura B.2 muestra los campos que el médico debe llenar, para posteriormente adquirir la predicción del sistema.

Figura B.2. Formulario de video detección de Parkinson [Autoría propia].

5. En esta pantalla tendrás que llenar los siguientes campos.
    a. Paciente. Puedes elegir entre la lista de pacientes registrados o registrar uno nuevo en ese momento. La figura B.3 muestra el formulario que se desplegará si se elige crear un ***Nuevo paciente***.

Figura B.3. Formulario de creación de paciente embebido en el formulario de video detección de Parkinson [Autoría propia].

    b. Observaciones. El llenado de este campo es opcional. Sirve para agregar información que pueda ser relevante en el futuro cuando se revisen los resultados obtenidos para este paciente.

c. Video. Este campo es obligatorio y debes subir un video en formato .mp4. La figura B.4 permite observar que luego de seleccionar el video, se puede utilizar la barra de desplazamiento inferior para elegir la porción del video que se desea sea analizada.

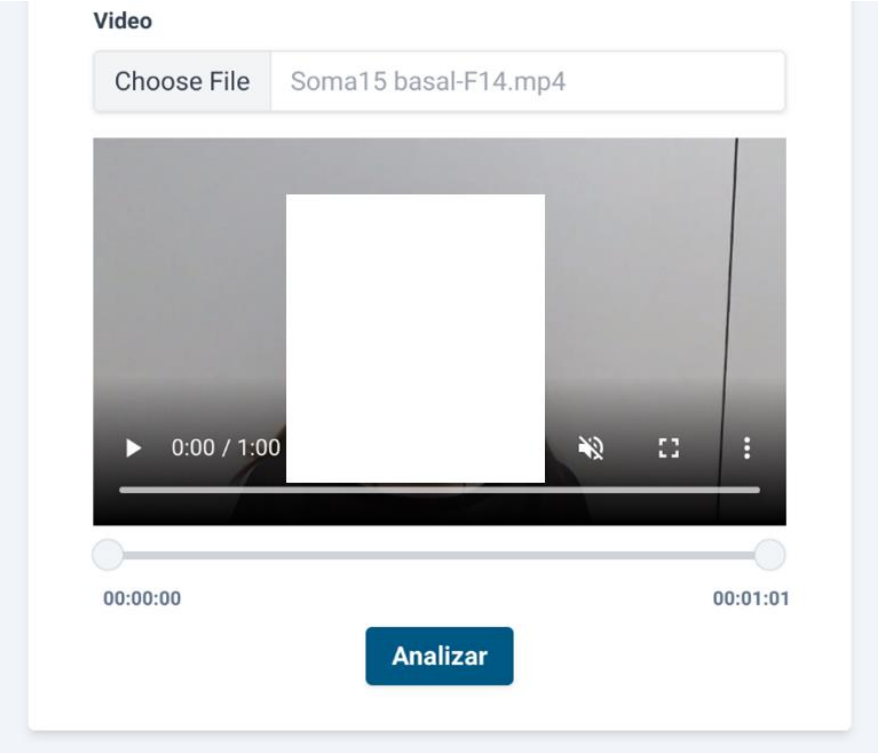


Figura B.4. Reproductor de video en el formulario de video detección de Parkinson (la identidad del paciente se ha mantenido oculta para este reporte) [Autoría propia].

6. Una vez llenados todos los campos, puedes dar clic en Analizar. En caso de que exista algún error, este será señalado para que lo corrijas y podrás volver a intentarlo.
7. Como se observa en la figura B.5, una vez que los resultados estén listos, se mostrarán en pantalla.

Figura B.5. Resultados de la video detección de Parkinson [Autoría propia].

8. En la parte inferior de los resultados, podrá retroalimentar el modelo para su mejora. La figura B.6 muestra los campos que debe colocar el médico en caso de tener observaciones o desacuerdos con la predicción otorgada.

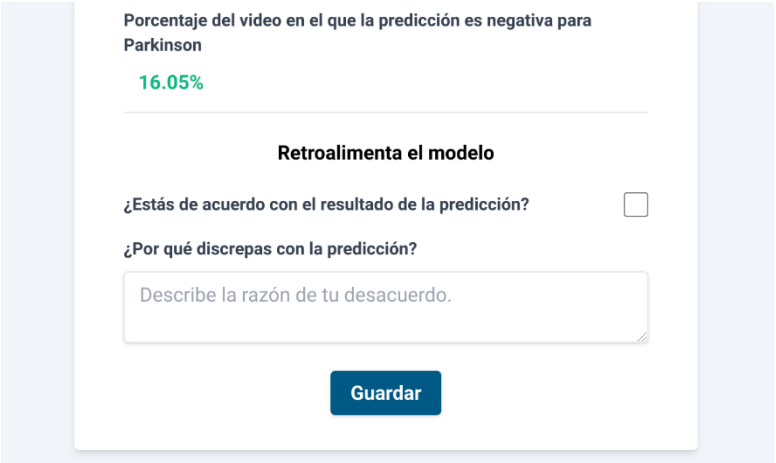


Figura B.6. Formulario de retroalimentación del modelo [Autoría propia].

9. Si estás de acuerdo con los resultados del modelo, debes marcar la casilla correspondiente y presionar Guardar. Si no estás de acuerdo, debes dejar la casilla sin marcar y opcionalmente agregar una explicación de tu desacuerdo; y luego presionar Guardar.
10. Luego de presionar Guardar, se te pedirá confirmación, ya que se te redirigirá al formulario principal y no podrás volver.

11. Luego de terminado todo este ciclo análisis-evaluación-retroalimentación, puedes analizar a un nuevo paciente o cerrar sesión si así lo deseas, usando el botón que se encuentra en la esquina superior derecha.