



Universidad de Montemorelos

**Facultad de Ingeniería y Tecnología
Ingeniería en Sistemas Computacionales**

**Construcción de una aplicación móvil para la detección de vía aérea
difícil usando aprendizaje profundo**

Kevin Aguilar Ruiz

Asesor principal: Dr. Germán Harvey Alférez Salinas

Asesor secundario: Dr. Christian Aguilar Ríos

Montemorelos, Nuevo León, México

26 de abril de 2019

ÍNDICE

I	INTRODUCCIÓN	1
I-A	Antecedentes	1
I-B	Planteamiento del problema	2
I-C	Justificación	2
I-D	Objetivos	2
I-E	Hipótesis	2
II	FUNDAMENTOS TEÓRICOS	2
II-A	Marco teórico	2
II-A1	Aprendizaje automático	2
II-A2	Aprendizaje profundo	3
II-A3	Redes Neuronales Convolucionales	3
II-A4	TensorFlow	3
II-A5	MobileNetV2	3
II-A6	Vía aérea difícil	4
II-B	Estado del arte	4
III	RESULTADOS	4
III-A	Metodología	4
III-A1	Aprobación del proyecto por la Dirección de Posgrado e Investigación (DPI)	5
III-A2	Búsqueda de fotografías de pacientes que fueron evaluados con el método Mallampati	5
III-A3	Entrenamiento del modelo predictivo mediante aprendizaje profundo	5
III-A4	Construcción de la aplicación móvil para la detección de vía aérea difícil	5
III-B	Presentación de los Resultados	6
III-B1	Validación cruzada	8
III-C	Discusión	8
IV	CONCLUSIÓN Y TRABAJO FUTURO	8
V	Agradecimientos	8
References		9
Appendix A: CARTA DE CONSENTIMIENTO		10

Resumen

Siempre que un paciente fuera a entrar al quirófano para ser operado, en caso de que la cirugía requiere anestesia general y con ella, que el paciente sea intubado, un médico especialista tuvo que hacer una revisión previa al paciente, con el motivo de evaluar su vía aérea. Este proceso debe ser hecho a cada paciente para anticipar cualquier problema, como una Vía aérea difícil (VAD), al momento de ser anestesiado. A la detección inadecuada de una VAD se le pueden sumar complicaciones importantes, hasta la muerte. Para solucionar este problema se ha desarrollado una aplicación móvil que utiliza aprendizaje profundo para la detección de una VAD. Como resultado de las pruebas realizadas se obtuvo una exactitud promedio de 82.36%.

Palabras Clave

Vía Aérea Difícil, Aprendizaje Automático, Redes Neuronales Convolucionales.

Construcción de una aplicación móvil para la detección de vía aérea difícil usando aprendizaje profundo

Kevin A. Aguilar, Germán H. Alférez y Christian Aguilar

Facultad de Ingeniería y Tecnología

Universidad de Montemorelos

Email: 1100402@alumno.um.edu.mx, harveyalferez@um.edu.mx, coordmedicina@um.edu.mx

I. INTRODUCCIÓN

A. Antecedentes

Uno de los temores más grandes que se tiene como anestesiólogos es enfrentarse a un paciente con vía aérea difícil ya sea diagnosticada previamente o en el peor de los casos, una vía aérea difícil inesperada. Una vía aérea difícil (VAD) se define como la necesidad de tres o más intentos para la intubación de la tráquea o más de 10 minutos para conseguirla [1]. Aproximadamente ocurre del 1.5%-8% en procedimientos donde se usa anestesia general. La incidencia de la situación “paciente no intubable” o “paciente no ventilable” es de 1/50000 pacientes. Asimismo, el fallo de intubación orotraqueal ocurre en 1/2000 casos programados, aumentando a 1/200 casos en las urgencias [2]. En las gestantes la intubación difícil es de 7.9% y en los casos de intubación muy difícil es de 2%. Si bien la consulta pre-anestésica bien valorada demuestra que la evaluación hecha por el anestesiólogo es muy sólida en la detección de estos tipos de pacientes, el alto índice de vía aérea difícil inesperada y los bajos valores predictivos positivos hace que se reduzca considerablemente su valor como una prueba de diagnóstico confiable [3]. Desde 1993 la Sociedad Americana de Anestesiología (ASA) ha publicado sus guías de manejo en Vía Aérea Difícil. Estas guías dan pautas muy específicas al anestesiólogo para el manejo de estos casos. Específicamente estas guías se enfocan en mantener una buena ventilación y oxigenación ya que una intubación fallida puede originar un aumento de la morbilidad [1], [4].

En base a esta preocupación por la seguridad del paciente se han creado protocolos modificados en los cuáles viendo la realidad del país u hospital se propone agregar dispositivos nuevos en el manejo de vía aérea, listas de chequeo o tecnología que pueden ser útiles en el momento de crisis. Si bien todo este sistema de verificación para la seguridad del paciente está documentado, es una realidad que la falta de implementación o la subvaloración de una intubación difícil ya sea por ego del especialista, por falta de dispositivos de soporte o simplemente por falta de experiencia y/o habilidad del anestesiólogo, ha hecho que este tema siga siendo uno de los más importantes, por no decir, infaltable en cualquier curso o congreso de anestesiología a nivel mundial. Este fue, es y seguirá siendo un tema a tratar ya que la evaluación

subjetiva y objetiva de un anestesiólogo en una vía aérea difícil con el Test de Mallampati por ejemplo tiene una sensibilidad del 80% y especificidad del 50% pero con valor predictivo bajo < 50%. El test más específico y sensible es la distancia esterno-mentoniana que obtuvo una sensibilidad del 82,4% y una especificidad del 88,6% [5], [6]. Estos exámenes de valoración no garantizan en su totalidad que no se tenga alguna vía aérea difícil inesperada. De hecho, “el ASA Closed Claims revela que 34% de las demandas a anestesistas se encuentran en relación a eventos de la vía aérea, y que la dificultad de intubación ha sido la causa más común de daño en las mismas desde los años 90” [2]. Este porcentaje es alto al igual que el impacto que puede generar una intubación fallida ya que el riesgo del paciente de tener una lesión por hipoxia o la muerte siempre está presente. La incapacidad de manejar con éxito una VAD es responsable de 600 muertes anuales y del 25%-30% de las muertes atribuibles a la anestesia [7], [8].

Existen muchas pruebas para detectar una vía aérea difícil, algunos autores inclusive combinaron estas pruebas para hacerlas más sólidas en su diagnóstico con buenos resultados. Otros autores mencionan que realmente no existe una vía aérea difícil, sino que es una interacción compleja entre el paciente, el anestesiista, el equipo disponible y otras circunstancias. Casi nunca se toma en cuenta que hay situaciones complejas que no tienen que ver con el paciente mismo sino con el entorno que conlleva a ese paciente a poder tener una vía aérea difícil o no. Por ejemplo, una vía aérea normal diagnosticada previamente se puede convertir en una vía aérea difícil si es que el anestesiólogo no tiene experiencia en intubación o hubo error en dosis de medicamentos que ayuden a que el paciente esté en un buen plano anestésico para realizar dicho procedimiento. Otro ejemplo tiene que ver con la “fijación” en la vía aérea. En este caso, por querer intubar al paciente no se ve el entorno en cuanto a signos vitales generales. Todas estas situaciones generan un estrés en el anestesiólogo tanto pre, intra y postoperatorio sobre todo si no se cuenta con el apoyo necesario de la institución en que se trabaja en cuanto a tecnología, protocolos establecidos o simplemente por no tener lo básico necesario para una situación de vía aérea difícil prevista como imprevista.

En estos últimos años las aplicaciones (*apps*) móviles están siendo utilizadas en la medicina. Esto se puede evidenciar con *apps* que ayudan al diagnóstico o que son de apoyo al

médico en la toma de decisiones. Hasta donde conocemos, solamente existen dos *apps* para apoyar a los anestesiólogos en su labor. La primera es una *app* que provee al paciente un expediente con la información necesaria para que, si en un futuro necesita una intervención quirúrgica que requiere intubación, los médicos sepan que esa persona ya fue evaluada anteriormente con vía aérea difícil [9]. La segunda *app* se centra en las experiencias de sus usuarios. De esta manera, ya sea un médico u otro personal que realizó una intervención y tuvo problemas con un paciente debido a una vía aérea difícil puedan compartir esa experiencia para que otras personas tengan retroalimentación viendo los casos de otros expertos de la salud y cómo ellos manejaron la situación [10]. A pesar de que existen múltiples factores predictivos de VAD, ninguno de ellos es totalmente fiable debido a la baja sensibilidad, especificidad y valor predictivo positivo de todos ellos [3], [11].

B. Planteamiento del problema

Siempre que un paciente fuera a entrar al quirófano para ser operado, en caso de que la cirugía requiere anestesia general y con ella, que el paciente sea intubado, un médico especialista tuvo que hacer una revisión previa al paciente, con el motivo de evaluar su vía aérea. Este proceso debe ser hecho a cada paciente para anticipar cualquier problema, como una Vía aérea difícil (VAD), al momento de ser anestesiado. A la detección inadecuada de una VAD, se le pueden sumar complicaciones importantes, hasta la muerte.

C. Justificación

La incapacidad de manejar con éxito una VAD es responsable de 600 muertes anuales y del 25%-30% de las muertes atribuibles a la anestesia [7], [8]. Mucha de la incapacidad del manejo, es por no detección de una VAD, pues depende de la percepción de los criterios de cada anestesiólogo. Si bien, la experiencia de aplicar los criterios mejora la detección oportuna de una VAD, el problema de la subjetividad (como en todas las detecciones sensibles), está presente. La objetividad que pudiera aportar una *app*, podría reducir el riesgo de enfrentarse a una VAD no esperada. La incidencia de la situación “paciente no intubable” o “paciente no ventilable” es de 1/50000 pacientes; el fallo de Intubación orotraqueal ocurre en 1/2000 casos programados, aumentando a 1/200 casos en las urgencias [2].

D. Objetivos

Nuestro objetivo es experimentar con redes neuronales convolucionales aplicando principios de aprendizaje profundo para lograr la detección de una VAD por medio de una aplicación móvil. Para este fin se cumplieron los siguientes objetivos específicos:

- Tomar fotografías de vías aéreas. Clasificar fotografías de acuerdo al Mallampati con 4 clases: I, II, III y IV.
- Entrenar un modelo predictivo con las fotografías de las 4 clases de Mallampatti: I, II, III y IV.

- Construir una aplicación móvil que utilice el modelo entrenado mediante aprendizaje profundo para la clasificación de fotografías de intubación.
- Realizar la evaluación del modelo de detección de vía aérea difícil para la intubación mediante validación cruzada.

E. Hipótesis

Ho: No hay concordancia entre la clasificación obtenida por un anestesiólogo experto y las hechas por la aplicación móvil.

Ha: Hay concordancia entre la clasificación obtenida por un anestesiólogo experto y las hechas por la aplicación móvil.

II. FUNDAMENTOS TEÓRICOS

A. Marco teórico

En esta sección se presenta el marco teórico. En la Figura 1 se muestra la relación de los conceptos.

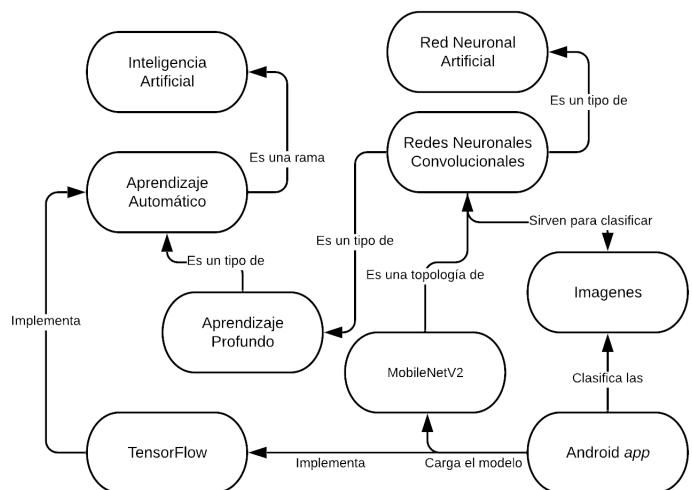


Fig. 1. Mapa conceptual.

1) *Aprendizaje automático*: El aprendizaje automático (*machine learning* en inglés) es la ciencia (y el arte) de la programación de computadoras para que estas puedan aprender de los datos. También puede describirse como el campo de estudio que le da a las computadoras la capacidad de aprender sin ser programadas explícitamente [12]. El aprendizaje es un fenómeno multifacético. Los procesos de aprendizaje incluyen la adquisición de nuevos conocimientos declarativos, el desarrollo de habilidades motoras y cognitivas a través de la instrucción o la práctica, la organización de nuevos conocimientos en general, representaciones efectivas y el descubrimiento de nuevos hechos y teorías a través de la observación y la experimentación. Desde el inicio de la era de la informática, los investigadores se han esforzado por implantar tales capacidades en las computadoras. Resolver este problema ha sido, y sigue siendo, un objetivo de largo alcance cada vez más desafiante y fascinante en el campo de la inteligencia artificial (IA) [13].

2) *Aprendizaje profundo*: El aprendizaje profundo (*deep learning* en inglés) es una forma de aprendizaje automático que permite a las computadoras aprender de la experiencia y comprender el mundo en términos de una jerarquía de conceptos. Debido a que la computadora reúne los conocimientos de la experiencia no es necesario que un operador humano de computadora especifique formalmente todo el conocimiento que necesita la computadora. La jerarquía de conceptos permite que la computadora aprenda conceptos complicados al construirlos a partir de los más simples [14]. Las redes neuronales artificiales (ANNs, por sus siglas en inglés) son el núcleo del aprendizaje profundo. Las ANNs son versátiles, potentes y escalables, lo que las hace ideales para abordar tareas de aprendizaje automático grandes y complejas, tales como clasificar miles de millones de imágenes (por ejemplo, Google Images). Asimismo, estas sirven para potenciar los servicios de reconocimiento de voz (por ejemplo, Siri de Apple), recomendar videos a cientos de millones de usuarios todos los días (por ejemplo, YouTube), o aprender a vencer al campeón del mundo en el juego de Go al examinar millones de juegos pasados y luego jugar contra sí mismo (AlphaGo de DeepMind) [12].

3) *Redes Neuronales Convolucionales* : Una red neuronal convolucional (CNN, por sus siglas en inglés) es un algoritmo de aprendizaje profundo que puede tomar una imagen de entrada, asignar importancia (pesos y sesgos aprendibles) a varios aspectos/objetos en la imagen y poder diferenciar uno de otro. La arquitectura de una CNN es análoga a la del patrón de conectividad de las neuronas en el cerebro humano y se inspiró en la organización de la corteza visual. Las neuronas individuales responden a los estímulos solo en una región restringida del campo visual conocida como campo receptivo. Una colección de dichos campos se superponen para cubrir toda el área visual [15]. Las CNNs se utilizan comúnmente para el análisis de imágenes visuales.

Una red neuronal convolucional tiene la siguiente topología [16]:

- *Capa de convolución*: en la capa de convolución se calcula la salida del producto punto entre un área de la(s) imagen o imágenes de entrada y una matriz de peso llamada filtro. El filtro se desliza por toda la imagen repitiendo la misma operación del producto punto. Por ejemplo, en la Figura 2 se considera una imagen de 3x3, al igual que otra matriz de 2x2 que es el filtro. La última es la nueva matriz con los resultados de la convolución.

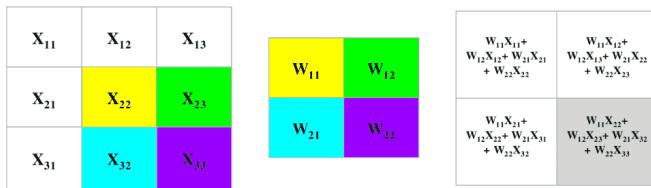


Fig. 2. Operación de convolución [16].

- *Capa de agrupación*: la capa de agrupación se utiliza para reducir las dimensiones espaciales, pero no la profundidad, en una red neuronal convolucional. En esta capa se

puede utilizar *max pooling* mediante la cuál se toma el número más alto (el área más sensible de la imagen) del área de entrada (una matriz $n \times m$). En la Figura 3 se muestra un ejemplo con la operación *max pooling*.

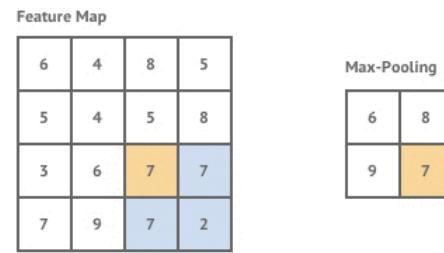


Fig. 3. Max Pooling [16].

- *Capa de no linealidad*: en la capa de no linealidad se usa la función de activación ReLU. La función de activación ReLU devuelve 0 por cada valor negativo en la imagen de entrada, mientras que devuelve el mismo valor por cada valor positivo. En la Figura 4 se muestra un ejemplo con la operación ReLU.

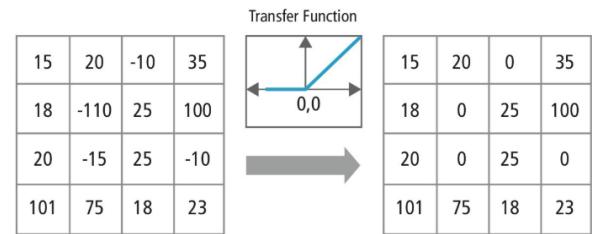


Fig. 4. Función de activación ReLU [16].

- *Capa totalmente conectada*: en esta capa se aplana la salida de la última capa de convolución y se conecta cada nodo de la capa actual con el otro nodo de la siguiente capa.

4) *TensorFlow*: TensorFlow¹ es una biblioteca de software de código abierto para el cálculo numérico distribuido mediante grafos de flujo de datos. TensorFlow fue creado por Google y es compatible con muchas de sus aplicaciones de aprendizaje automático [12]. TensorFlow, como su nombre lo indica, es un *framework* para definir y ejecutar cálculos que involucran tensores. Un tensor es una generalización de vectores y matrices a dimensiones potencialmente más altas. Internamente, TensorFlow representa los tensores como matrices n-dimensionales de tipos de datos base. Cada elemento en el tensor tiene el mismo tipo de datos, y el tipo de datos siempre se conoce. La forma (es decir, el número de dimensiones que tiene y el tamaño de cada dimensión) puede ser conocida solo parcialmente. El rango de un tensor es su número de dimensiones. Los sinónimos para rango incluyen orden o grado o n-dimensión [17].

5) *MobileNetV2*: MobileNet es una familia de modelos de visión por computadora con tecnología móvil para TensorFlow. MobileNet fue diseñada para maximizar la precisión de los

¹<https://www.tensorflow.org>

modelos de manera efectiva tomando en cuenta los recursos restringidos para una aplicación integrada o en el dispositivo. Estos modelos son pequeños, de baja latencia, baja potencia, y parametrizados para cumplir con las restricciones de recursos de una variedad de casos de uso. Se pueden construir para la clasificación, detección, incrustaciones y segmentación de manera similar a como se utilizan otros modelos populares a gran escala, como Inception [18].

Las MobileNets se basan en una arquitectura optimizada que utiliza convoluciones separables en profundidad para construir redes neuronales ligeras y profundas. Estas MobileNets fueron presentadas en el año 2017 e introdujeron dos hiper-parámetros globales simples, un multiplicador de ancho y un multiplicador de resolución, que intercambian eficientemente la latencia y la precisión. Estos hiper-parámetros permiten que el creador de modelos elija el modelo del tamaño correcto para su aplicación en función de las restricciones del problema [19].

La MobileNetV2 mejora el rendimiento de vanguardia de los modelos móviles en múltiples tareas y puntos de referencia, así como en un espectro de diferentes tamaños de modelos. La arquitectura de MobileNetV2 se basa en una estructura residual invertida en la que la entrada y la salida del bloque residual son capas delgadas de cuello de botella opuestas a los modelos residuales tradicionales que utilizan representaciones expandidas en la entrada. Además, MobileNetV2 usa convoluciones ligeras en profundidad para filtrar características en la capa de expansión intermedia [20]. La estructura básica se muestra en la Figura 5.

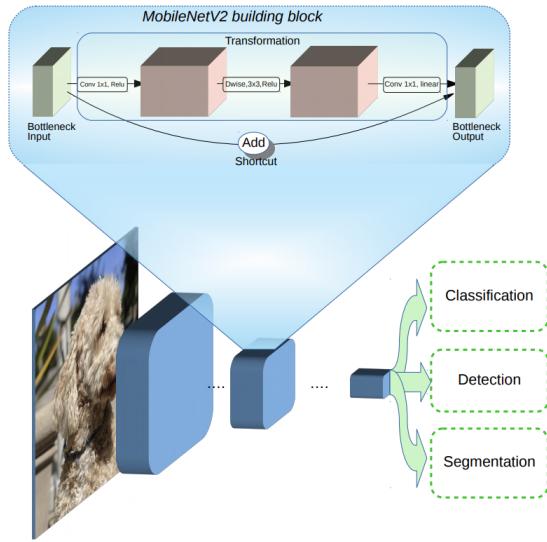


Fig. 5. Descripción general de la arquitectura MobileNetV2 [21].

MobileNetV2 se basa en las ideas de MobileNetV1 [19], utilizando la convolución separable en profundidad como bloques de construcción eficientes. Sin embargo, MobileNetV2 introduce dos nuevas características a la arquitectura: 1) cuellos de botella lineales entre las capas y 2) conexiones de acceso directo entre los cuellos de botella. Los cuellos de botella codifican las entradas y salidas intermedias del modelo, mientras que la capa interna encapsula la capacidad del modelo para transformarse de conceptos de nivel inferior,

como píxeles, a descriptores de nivel superior, como categorías de imágenes. Finalmente, al igual que con las conexiones residuales tradicionales, los accesos directos permiten un entrenamiento más rápido y una mayor precisión [21].

En la Tabla 1 se puede observar cómo se organizan los bloques de cuellos de botella en MobileNetV2. t representa la tasa de expansión de los canales, c representa el número de canales de entrada y n la frecuencia con la que se repite el bloque. Finalmente, s indica si la primera repetición de un bloque usó un paso de 2 para el proceso de reducción de muestreo [22].

TABLE I
LA ARQUITECTURA DE MOBILENETV2

Input	Operator	t	c	n	s
$224^2 \times 3$	conv2d	-	32	1	2
$122^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1x1	-	1280	1	1
$7^2 \times 1280$	avgpool 7x7	-	-	1	-
$1 \times 1 \times 1280$	conv2d 1x1	-	k	-	-

6) *Vía aérea difícil:* Una vía aérea difícil (VAD) se define como la necesidad de tres o más intentos para la intubación de la tráquea o más de 10 minutos para conseguirla [1]. Aproximadamente ocurre del 1.5%-8% en procedimientos donde se usa anestesia general. También se define como la situación clínica en la que un anestesiólogo experimentado con capacitación convencional tiene dificultad para ventilar la vía aérea superior con Mascarilla Facial (MF), Intubación Endotraqueal (IET), o ambas.

B. Estado del arte

No se encontraron en la literatura científica *apps* validadas por los colegios y/o consejos de anestesiología. Estas ayudarían a identificar a pacientes con vía aérea difícil y así poder contar con una ayuda extra y sobre todo que de un soporte al anestesiólogo en su evaluación diaria de pacientes. Por otra parte, en los últimos años ha habido en el mundo un despertar científico e industrial enfocado en la inteligencia artificial, no solo en temas como las finanzas, el transporte o la robótica, también en la medicina. De hecho, en la Universidad de Montemorelos se han realizado proyectos de investigación que han demostrado que es posible detectar ciertas enfermedades tales como nevo sospechoso de melanoma [23] y glaucoma [24] mediante inteligencia artificial. Además, se ha utilizado la inteligencia artificial para estudiar la relación entre caries dental y diabetes [25].

III. RESULTADOS

A. Metodología

A continuación se describen las actividades realizadas en cada una de las etapas de esta investigación.

1) *Aprobación del proyecto por la Dirección de Posgrado e Investigación (DPI)*: Con el fin de formalizar el proyecto ante la Universidad de Montemorelos, los autores de esta investigación sometieron la propuesta de la presente ante la DPI de dicha universidad. Esta propuesta fue aprobada tanto por el Comité de Ética en Investigación como por el Comité de Investigación Institucional de la Universidad de Montemorelos. El código registrado de este proyecto es el 2018-17 con el nombre de “Construcción de una aplicación móvil para la detección de vía aérea difícil usando aprendizaje profundo”.

2) *Búsqueda de fotografías de pacientes que fueron evaluados con el método Mallampati*: Se tomaron 179 fotografías de la escala de Mallampati a alumnos universitarios de la Escuela de Medicina de la Universidad de Montemorelos. Estas fotografías fueron tomadas en un lapso de 1 mes con un iPhone 6s por un estudiante de medicina. Para tomar las fotografías se realizó lo siguiente:

- 1) A cada estudiante de primer ingreso de medicina se le propuso participar en esta investigación.
- 2) El estudiante que deseaba participar firmó un consentimiento en el cual se le indicó el procedimiento y el uso de sus fotografías. El Apéndice A muestra la carta de consentimiento.
- 3) Para la toma de fotografías de la escala de Mallampati se le pidió al estudiante que abriera lo más que pudiera su boca y que sacara la lengua lo más afuera posible.
- 4) Al cumplir con lo anterior, se le tomó de la fotografía al estudiante. Una fotografía por estudiante.

3) *Entrenamiento del modelo predictivo mediante aprendizaje profundo*: En este paso se re-entrenó la red neuronal convolucional MobileNetV2 con el fin de crear un nuevo modelo de clasificación a partir de las fotografías de Mallampati. Se eligió esta topología ya que en comparación con InceptionV3 [26], se obtiene un peso menor a nivel de megabytes del modelo. Este es un factor importante en esta investigación con el fin de que especialistas en la salud puedan utilizar esta app aún en dispositivos de baja gama.

En la Tabla 2 se muestra la comparación entre los dos modelos, en la cual se puede observar que el tamaño del modelo generado mediante MobileNetV2 es casi 10 veces menor que el modelo generado mediante InceptionV3.

TABLE II
COMPARATIVA DE TAMAÑO ENTRE LOS MODELOS

Modelo	Tamaño Original	Tamaño Generado
Inception_V3	95.3 MB	83.4 MB
Mobilenet_V2_1.0_224	14.0 MB	8.76 MB

Con el fin de entrenar el modelo, fue necesario instalar las siguientes herramientas:

- 1) Se instaló Anaconda versión 5.2.0 a través del instalador para Windows de su página oficial².
- 2) Se creó un ambiente de desarrollo con Python versión 3.5.5.

²<https://www.anaconda.com/distribution/#download-section>

- 3) En el ambiente creado de Python se instaló TensorFlow versión 1.10.0.
- 4) Se descargó el código retrain.py del proyecto de GitHub de TensorFlow. Este código carga el modelo pre-entrenado con MobileNetV2 con imágenes del conjunto de datos de ImageNet y reentrena el modelo de clasificación con las fotos de Mallampati³.
- 5) Se realizaron varias pruebas preliminares a nivel de medición de exactitud del modelo con las siguientes clases: 40 en la clase Mallampati 1, 36 en la clase Mallampati 2, 43 en la clase Mallampati 3 y 40 en la clase Mallampati 4. 5 imágenes de cada clase, aparte de las que fueron utilizadas en el entrenamiento, fueron separadas para la realización de las pruebas. Debido a la similitud de las fotografías en las clases Mallampati 1 y 2 se obtuvo un bajo nivel de exactitud (*accuracy* en inglés) en la evaluación. Es por esto que se decidió juntar las clases en pares, quedando así, solo dos clases, Mallampati 1-2 y Mallampati 3-4. Las fotografías en estos dos grupos se almacenaron en dos carpetas diferentes. El comando con el cual se realizó el re-entrenamiento del modelo fue el siguiente:

```
python -m scripts.retrain
--output_graph=retrained_graph.pb
--output_labels=retrained_labels.txt
--image_dir=photos --tfhub_module
https://tfhub.dev/google/imagenet/
mobilenet_v2_100_224/feature_vector/2
```

En esta instrucción se indicó lo siguiente:

- El script con el que se ejecutará el re-entrenamiento: retrain.py
- El modelo a crear: retrained_graph.pb
- La lista con las etiquetas para la clasificación: retrained_labels.txt
- La carpeta donde se encuentran las fotografías. En este caso el folder photos contiene las dos carpetas con las clases Mallampati 1-2 y Mallampati 3-4.
- En base a cuál modelo se realizará el re-entrenamiento. En este caso, el modelo es Mobilenet_V2_1.0_224.

El re-entrenamiento tardó 6 minutos en una computadora con las siguientes características: procesador Intel Core i7-7700HQ 2.8 GHz, 16 GB DDR4 1300 MHz de memoria RAM, tarjeta gráfica NVIDIA GeForce GTX 1060 Max-Q 6 GB GDDR5 8008 MHz y sistema operativo Windows 10 versión 1809.

4) *Construcción de la aplicación móvil para la detección de vía aérea difícil*: En este paso se creó una aplicación móvil que utiliza el modelo entrenado para la detección de vía aérea difícil. Específicamente se realizó lo siguiente:

- 1) Se instaló Android Studio⁴.
- 2) Se creó una nueva app.
- 3) Se agregaron las clases Classifier y TensorflowImageClassifier de TFMobile al proyecto. La clase Classifier es una interfaz genérica para interactuar con diferentes motores de reconocimiento, mientras que la clase TensorflowImageClassifier la implementa y esta es la que crea

³<http://bit.ly/tensorflow-retrain>

⁴<https://developer.android.com/studio#downloads>

el clasificador para etiquetar las fotografías utilizando TensorFlow. Estas clases fueron tomadas del sitio web oficial de TensorFlow en GitHub⁵ para dispositivos Android.

- 4) Se agregó el modelo ya entrenado en la carpeta de *assets* de la *app*.
- 5) Se crearon 3 pantallas. La primera es una pantalla inicial que muestra un botón para comenzar. En la segunda pantalla se muestra una imagen la cual ejemplifica como es que se debe de tomar la fotografía al mismo tiempo que un botón para abrir la cámara. En la ultima pantalla se muestran los resultados que da el modelo a nivel de exactitud. Ver la Figura 6.

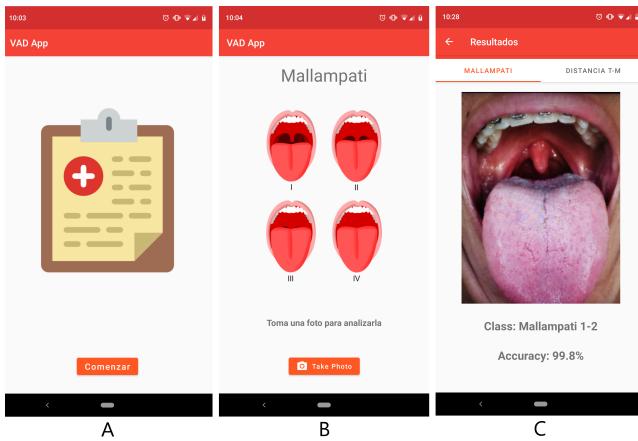


Fig. 6. Pantallas de la *app*: A) Pantalla principal, B) Pantalla donde se toma la fotografía, C) Pantalla donde se muestran los resultados.

El código más relevante en términos de la aplicación de inteligencia artificial es el que se encuentra detrás de la segunda pantalla en donde se toma la fotografía. Específicamente, en esta segunda pantalla se carga el modelo de clasificación, previamente entrenado, que se va a utilizar para clasificar la imagen. Un fragmento del código para cargar el modelo se presenta en el Listado de código 1.

En las líneas 3-6 se declaran las variables que se necesitan para cargar el modelo. Estas variables se describen a continuación:

- INPUT_SIZE: es el tamaño de entrada del modelo. En nuestro caso, la topología de MobileNetV2 que se utilizó tiene un tamaño de entrada de 224x224 píxeles.
- IMAGE_MEAN y IMAGE_STD: estas variables indican el rango de entrada esperado de la red neuronal que se está utilizando. Algunas redes neuronales utilizan la entrada en el rango [0,255], otras [0,1], otras [-1,1], etc.
- INPUT_NAME y OUTPUT_NAME: son los nombres de entrada y salida del modelo que provienen del script de entrenamiento retrain.py.
- MODEL_FILE y LABEL_FILE: son los nombres de los archivos del modelo y sus etiquetas que se encuentran dentro de la carpeta *assets* de la *app*.

En la línea 11 se crea la interfaz de clasificación. En las líneas 21-40 se encuentra la función que carga el modelo clasificador

el cual se ejecuta en el método *onCreate* en la línea 18, esta función toma las variables previamente definidas para crear el clasificador. Esta función utiliza la *TensorFlowInferenceInterface*, que es la que se encarga de cargar el modelo usando el *assetManager*. Esto es similar a un *tf.Session* con TensorFlow en Python. El código fuente completo para cargar el modelo de clasificación se encuentra en la cuenta personal de GitHub⁶.

```

1  public class MallampatiActivity extends
2      AppCompatActivity {
3
4      public static final int INPUT_SIZE = 224;
5      public static final int IMAGE_MEAN = 0;
6      public static final float IMAGE_STD = 255.0f;
7      public static final String INPUT_NAME = "Placeholder"
8          ;
9      public static final String OUTPUT_NAME =
10         final_result";
11     private static final String MODEL_FILE = "file:///-
12         android_asset/graphM.pb";
13     private static final String LABEL_FILE = "file:///-
14         android_asset/labelsM.txt";
15
16     public Classifier classifier;
17
18     @Override
19     protected void onCreate(Bundle savedInstanceState) {
20         super.onCreate(savedInstanceState);
21         setContentView(R.layout.activity_mallampati);
22
23         initTensorFlowAndLoadModel();
24
25     private void initTensorFlowAndLoadModel() {
26         executor.execute(new Runnable() {
27             @Override
28             public void run() {
29                 try {
30                     classifier = TensorFlowImageClassifier.create
31                         (
32                             getAssets(),
33                             MODEL_FILE,
34                             LABEL_FILE,
35                             INPUT_SIZE,
36                             IMAGE_MEAN,
37                             IMAGE_STD,
38                             INPUT_NAME,
39                             OUTPUT_NAME);
40                 } catch (final Exception e) {
41                     throw new RuntimeException("Error
42                         initializing TensorFlow!", e);
43                 }
44             }
45         });
46     }
47 }
```

Listado de código 1. noneFragmento de código para cargar el modelo de clasificación.

B. Presentación de los Resultados

En las evaluaciones realizadas con 20 fotografías de prueba (5 por cada clase) la exactitud promedio del modelo generado fue de 82.36%. En las Tablas 3-6 se muestran los resultados de las evaluaciones obtenidas con 10 fotografías de las clases Mallampati 1 y 2 (5 fotografías de clase Mallampati 1 y 5 fotografías de la clase Mallampati 2) y Mallampati 3 y 4 (5 fotografías de clase Mallampati 3 y 5 fotografías de la clase Mallampati 4). Cada fotografía muestra el porcentaje de exactitud del modelo al detectar Mallampati 1-2 o Mallampati 3-4.

⁵<http://bit.ly/tensorflow-android-demo>

⁶http://bit.ly/Mallampati_Activity

En la Tabla 5, se puede observar que dentro de las evaluaciones de Mallampati 3, 4 de cada 5 imágenes son clasificadas correctamente. En las evaluaciones de Mallampati 4 sucede lo mismo.

TABLE III
EVALUACIÓN DE IMÁGENES CON MALLAMPATI 1

Foto	Mallampati 1-2	Mallampati 3-4
	99.99%	0.01%
	99.97%	0.03%
	66.44%	33.56%
	88.04%	11.96%
	96.53%	3.47%
Promedio	90.19%	9.81%

TABLE IV
EVALUACIÓN DE IMÁGENES CON MALLAMPATI 2

Foto	Mallampati 1-2	Mallampati 3-4
	89.2%	10.8%
	64.56%	35.44%
	99.81%	0.19%
	89.82%	10.18%
	76.93%	23.07%
Promedio	84.06%	15.94%

TABLE V
EVALUACIÓN DE IMÁGENES CON MALLAMPATI 3

Foto	Mallampati 1-2	Mallampati 3-4
	6.39%	93.61%
	0.33%	99.67%
	0.06%	99.94%
	0.13%	99.87%
Promedio	89.98% 19.38%	10.02% 80.62%

TABLE VI
EVALUACIÓN DE IMÁGENES CON MALLAMPATI 4

Foto	Mallampati 1-2	Mallampati 3-4
	0.08%	99.92%
	31.33%	68.67%
	1.46%	98.54%
	92.56%	7.44%
Promedio	1.68% 25.42%	98.32% 74.58%

1) *Validación cruzada:* La Tabla 7 muestra los resultados de sensibilidad (*sensitivity* en inglés) y especificidad (*specificity* en inglés) que se obtuvieron en la validación cruzada del modelo generado. Los términos antes mencionados se definen a continuación [27]:

- Sensibilidad: Qué tan bueno es el modelo en detectar pacientes con vía aérea difícil. La fórmula para obtener la sensibilidad es la siguiente:

$$\text{Sensitivity} = \frac{TP}{(TP + FN)}$$

- Especificidad: Qué tan bueno es el modelo en detectar pacientes sin vía aérea difícil. La fórmula para obtener la especificidad es la siguiente:

$$\text{Specificity} = \frac{TN}{(TN + FP)}$$

A continuación se describen los conceptos de Verdadero Positivo (TP, por sus siglas en inglés), Verdadero Negativo (TN, por sus siglas en inglés), Falso Positivo (FP, por sus siglas en inglés), Falso Negativo (FN, por sus siglas en inglés) en el contexto del presente estudio:

- TP: una persona a la que se le dijo que tiene una vía aérea difícil y que realmente la tiene.
- TN: una persona a la que se le dijo que no tiene una vía aérea difícil y que realmente no la tiene.
- FP: una persona a la que se le dijo que tiene una vía aérea difícil, aunque no la tiene.
- FN: una persona a la que se le dijo que no tiene una vía aérea difícil a pesar de que realmente si la tiene.

TABLE VII
RESULTADOS DE LAS PRUEBAS

	Sensibilidad	Especificidad
Mallampati 1-2	1	0.8
Mallampati 3-4	0.8	1
Promedio	0.9	0.9

C. Discusión

De acuerdo con la Tabla 7 en donde se exponen los resultados de la validación cruzada del modelo generado, se puede observar que tanto la sensibilidad como la especificidad en promedio fueron de 90%. Estos valores son 10% y 40% mayores que los resultados de sensibilidad y especificidad médica respectivamente presentada por Rodríguez et al (2014) en el caso del Test de Mallampati. Se valida la hipótesis: Hay concordancia entre la clasificación obtenida por un anestesiólogo experto y las hechas por la aplicación móvil.

IV. CONCLUSIÓN Y TRABAJO FUTURO

Este trabajo de investigación se realizó con el fin de proveer una herramienta adicional en la evaluación de pacientes prequirúrgicos para la detección de una VAD mediante el uso de redes neuronales convolucionales. A través de TensorFlow se extendió el modelo MobileNetV2 con 159 fotografías tomadas en la Escuela de Medicina de la Universidad de Montemorelos.

En las evaluaciones realizadas con 20 fotografías de prueba, se obtuvo un resultado promedio del 82.36% de exactitud en las evaluaciones entre Mallampati 1-2 y Mallampati 3-4. Además, los resultados de la validación cruzada en términos de sensibilidad y especificidad fueron de 90%.

Como trabajo futuro, ya que este proyecto fue desarrollado con TFMobile, una tecnología que fue desplazada por TFLite, se actualizará la app a esta nueva tecnología. De igual manera se extenderá la muestra de fotografías de pacientes con diferentes clases dentro de la escala de Mallampati con el fin de mejorar aún más la clasificación. Asimismo, se introducirá un nuevo modelo para la clasificación de la distancia tiro-mentoniana.

V. AGRADECIMIENTOS

Este trabajo de investigación no hubiese sido posible sin la supervisión del Dr. Germán H. Alférez a quien le agradezco su apoyo incondicional para la realización de este proyecto. Asimismo, expreso mi agradecimiento al Dr. Christian Aguilar y al estudiante de medicina Jedidja Arden Joseph de la Escuela de Medicina de la Universidad de Montemorelos, por su apoyo y disponibilidad en la realización de las diferentes actividades requeridas en esta investigación.

REFERENCES

- [1] J. L. Apfelbaum, C. A. Hagberg, R. A. Caplan, C. D. Blitt, R. T. Connis, D. G. Nickinovich, C. A. Hagberg, R. A. Caplan, J. L. Benumof, F. A. Berry, C. D. Blitt, R. H. Bode, F. W. Cheney, R. T. Connis, O. F. Guidry, D. G. Nickinovich, and A. Ovassapian, "Practice guidelines for management of the difficult airway," *Anesthesiology*, vol. 118, no. 2, pp. 251–270, feb 2013.
- [2] B. García, "Valoración preoperatoria de la vía aérea difícil ¿hay algo nuevo?", <https://anestesiayreanimacion.org/2015/valoracion-preoperatoria-de-la-via-aerea-dificil-hay-algo-nuevo/>, 2015, accessed: 11- May- 2018.
- [3] A. K. Nørskov, C. V. Rosenstock, J. Wetterslev, G. Astrup, A. Afshari, and L. H. Lundstrøm, "Diagnostic accuracy of anaesthesiologists' prediction of difficult airway management in daily clinical practice: a cohort study of 188 064 patients registered in the danish anaesthesia database," *Anaesthesia*, vol. 70, no. 3, pp. 272–281, dec 2014.
- [4] J. Campos, "Guías, algoritmos y recomendaciones durante el manejo de la vía aérea difícil en el paciente sometido a cirugía torácica: ¿están respaldados por la evidencia científica?" *Revista Española de Anestesiología y Reanimación*, vol. 65, no. 1, pp. 1–4, jan 2018.
- [5] A. M. Rodríguez, J. N. Pascual, L. P. Ferrer, J. F. Domínguez, J. B. Chaves, and E. M. González, "Validez de los predictores de vía aérea difícil en medicina extrahospitalaria," *Anales del Sistema Sanitario de Navarra*, vol. 37, no. 1, pp. 91–98, apr 2014.
- [6] E. R. García and J. R. Cedeño, "Valor predictivo de las evaluaciones de la vía aérea difícil," *Trauma*, vol. 8, no. 3, pp. 63–70, 2005.
- [7] M. Gómez-Ríos, L. Gaitini, I. Matter, and M. Somri, "Guías y algoritmos para el manejo de la vía aérea difícil," *Revista Española de Anestesiología y Reanimación*, vol. 65, no. 1, pp. 41–48, jan 2018.
- [8] T. Cook, N. Woodall, and C. Frerk, "Major complications of airway management in the UK: results of the fourth national audit project of the royal college of anaesthetists and the difficult airway society. part 1: Anaesthesia †," *British Journal of Anaesthesia*, vol. 106, no. 5, pp. 617–631, may 2011.
- [9] E. Shanahan, J. H.-C. Huang, A. Chen, A. Narasimhan, and R. Tang, "Difficultintubationapp.com – a difficult airway electronic record," *Canadian Journal of Anesthesia/Journal canadien d'anesthésie*, vol. 63, no. 11, pp. 1299–1300, aug 2016.
- [10] L. V. Duggan, S. L. Lockhart, T. M. Cook, E. P. O'Sullivan, T. Dare, and P. A. Baker, "The airway app: exploring the role of smartphone technology to capture emergency front-of-neck airway experiences internationally," *Anaesthesia*, vol. 73, no. 6, pp. 703–710, mar 2018.
- [11] T. Shiga, Z. Wajima, T. Inoue, and A. Sakamoto, "Predicting difficult intubation in apparently normal patients meta-analysis of bedside screening test performance," *Anesthesiology*, vol. 103, no. 2, pp. 429–437, 2005. [Online]. Available: <http://dx.doi.org/>
- [12] A. Géron, *Hands-On Machine Learning with Scikit-Learn and TensorFlow*. O'Reilly UK Ltd., 2017. [Online]. Available: https://www.ebook.de/de/product/26464006/aurelien_geron_hands_on_machine_learning_with_scikit_learn_and_tensorflow.html
- [13] R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, Eds., *Machine Learning*. Springer Berlin Heidelberg, 1983.
- [14] K. G. Kim, "Book review: Deep learning," *Healthcare Informatics Research*, vol. 22, no. 4, p. 351, 2016.
- [15] S. Saha, "A comprehensive guide to convolutional neural networks - the eli5 way," <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>, Dec. 2018, accessed: 14- Mar- 2019.
- [16] A. Zerium, "Demystifying convolutional neural networks," <https://medium.com/@eternalzer0dayx/demystifying-convolutional-neural-networks-ca17bdc75559>, Sep. 2018, accessed: 16- Sep- 2018.
- [17] "Tensors," <https://www.tensorflow.org/guide/tensors>, accessed: 14- Mar- 2019.
- [18] A. G. Howard and M. Zhu, "Mobilenets: Open-source models for efficient on-device vision," <https://ai.googleblog.com/2017/06/mobilenets-open-source-models-for.html>, Jun. 2017, accessed: 16- Sep- 2018.
- [19] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications."
- [20] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilnetv2: Inverted residuals and linear bottlenecks," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.
- [21] M. Sandler and A. Howard, "Mobilnetv2: The next generation of on-device computer vision networks," <https://ai.googleblog.com/2018/04/mobilenetv2-next-generation-of-on.html>, Apr. 2018, accessed: 14- Mar- 2019.
- [22] P.-L. Prâve, "Mobilnetv2: Inverted residuals and linear bottlenecks," <https://towardsdatascience.com/mobilenetv2-inverted-residuals-and-linear-bottlenecks-8a4362f4ffd5>, Apr. 2018, accessed: 14- Mar- 2019.
- [23] C. Marín, G. H. Alférez, J. Córdova, and V. González, "Detection of melanoma through image recognition and artificial neural networks," in *IFMBE Proceedings*. Springer International Publishing, 2015, pp. 832–835.
- [24] M. Espinoza, G. Alférez, and J. Castillo, "Prediction of glaucoma through convolutional neural networks," in *Proceedings of the 2018 International Conference on Health Informatics & Medical Systems*, 2018, pp. 90–95.
- [25] G. Alférez, J. Jiménez, H. H. Navarro, M. González, R. Domínguez, A. Briones, and H. H. Villalvazo, "Aplicación de ciencia de datos en las historias clínicas de una clínica ubicada en el noreste de méxico para corroborar la relación entre caries dental y diabetes," in *Proceedings of the 2o Congreso de Investigación Universitaria de la División Interamericana*, 2016, pp. 236–241.
- [26] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
- [27] Y. Greenfield, "Precision, recall, sensitivity and specificity," <http://yuvalg.com/blog/2012/01/01/precision-recall-sensitivity-and-specificity/>, Jan. 2012, accessed: 21- Mar- 2019.

APPENDIX A
CARTA DE CONSENTIMIENTO



Dirección de Investigación e Innovación.
CARTA DE CONSENTIMIENTO FOTOS

Nombre de la Investigación:

Diferencia en las mediciones de la Vía Aérea entre el método convencional y una aplicación. 2018

Estimado (a) Señor/ Señora:

Introducción / objetivo: La Escuela de Medicina y la Facultad de Ingeniería y Tecnología está realizado un proyecto de investigación en el cual el objetivo del estudio es descubrir mediante una aplicación móvil si el/la paciente tiene un pronóstico alto de Vía aérea difícil y por ende una intubación difícil. El estudio se realizará en diferentes hospitales de la República Mexicana y Hospitales Internacionales.

Procedimientos: Se necesita para alimentar al software un par de fotos, una de frente con la boca abierta, lengua afuera y otra foto de perfil de la misma persona para poder tener una base de datos bastante amplia y evitar lo menos posible el error. En el equipo de cómputo en el que se realizará el experimento, los investigadores responsables almacenarán las fotos ofrecida por la red neuronal profunda. Ni su nombre ni las fotografías de su rostro aparecerán en el documento final de proyecto de fin de carrera ni en el artículo científico, solo serán de referencia para nuestra base de datos.

Beneficios: Usted no recibirá un beneficio directo por las fotografías que se le tomarán. Sin embargo, si usted acepta participar estará colaborando con el avance científico.

Confidencialidad: Su nombre siempre será confidencial. Su nombre ni fotografía no aparecerá en los documentos relacionados al proyecto ni en las exposiciones o publicaciones de la misma. Las fotografías originales las conservará el investigador responsable en un lugar seguro y no serán publicadas.

Compensación: Usted no recibirá una compensación financiera por sus fotos ni por el experimento realizado.

Participación voluntaria /retiro: La participación en este estudio es absolutamente voluntaria. Usted está en plena libertad de negarse a participar o de retirar su participación del mismo en cualquier momento. Su decisión de participar o de no participar no afectará de ninguna manera la forma de cómo le traten en la Universidad de Montemorelos.

Números a contactar: Si usted tiene algún pregunta, comentario o preocupación con respecto al proyecto, por favor comuníquese con el investigador responsable del proyecto: Dr. Christian Aguilar Rios y a la siguiente dirección de correo electrónico: coordmedicina@um.edu.mx.

Si usted acepta participar en el estudio y si Ud. Lo desea, le entregaremos una copia de este documento que le pedimos sea tan amable de firmar.

Consentimiento para su participación en el estudio

Su firma indica su aceptación para participar voluntariamente en el presente estudio.

Nombre del participante:

fecha:

Firma: _____

Nombre de la persona que obtiene el consentimiento

fecha:
