

Propuesta de Proyecto de Fin de Curso

Título: Luz verde: implementación de un sistema de detección del uso de mascarillas por parte de la ciudadanía ecuatoriana

Integrantes:

- Bryan Manzano Manzano
- Carlos Orellana Fantoni

Nombre del grupo: Los Deep Learners

Problema

Actualmente, con el cambio en los semáforos implementados para la emergencia sanitaria, la cantidad de personas que vuelva a transitar por las ciudades estará aumentando lentamente hasta esperar alcanzar eventualmente los niveles que se tenían regularmente. Sin embargo, el hecho es que a pesar de que los semáforos se encuentren en el nivel “verde”, la emergencia sanitaria continua, y así mismo continúa el peligro de que ocurra un rebrote de la pandemia.

Debido a esto, existe la responsabilidad por parte tanto de los ciudadanos como de las distintas entidades gubernamentales encargadas de amparar por la seguridad del país mediante el cumplimiento de las distintas medidas de salud que se tienen establecidas actualmente, las cuales van cambiando e implementándose a medida que avanza la pandemia [1]. Aun así, existe siempre el riesgo de individuos que, a pesar de la importancia del asunto, no se rijan acorde a estas normas y aumenten la posibilidad de un rebrote. Es por lo tanto necesario que exista una manera de vigilar en lo posible que estas medidas se lleven a cabo, como el uso obligatorio de mascarillas al estar en las calles.

Objetivo

Vigilar mediante imágenes el cumplimiento de las normas de salud como contramedidas para la transmisión del Covid-19 mediante la detección de mascarillas en los ciudadanos ecuatorianos.

Modelos de análisis y definición del problema

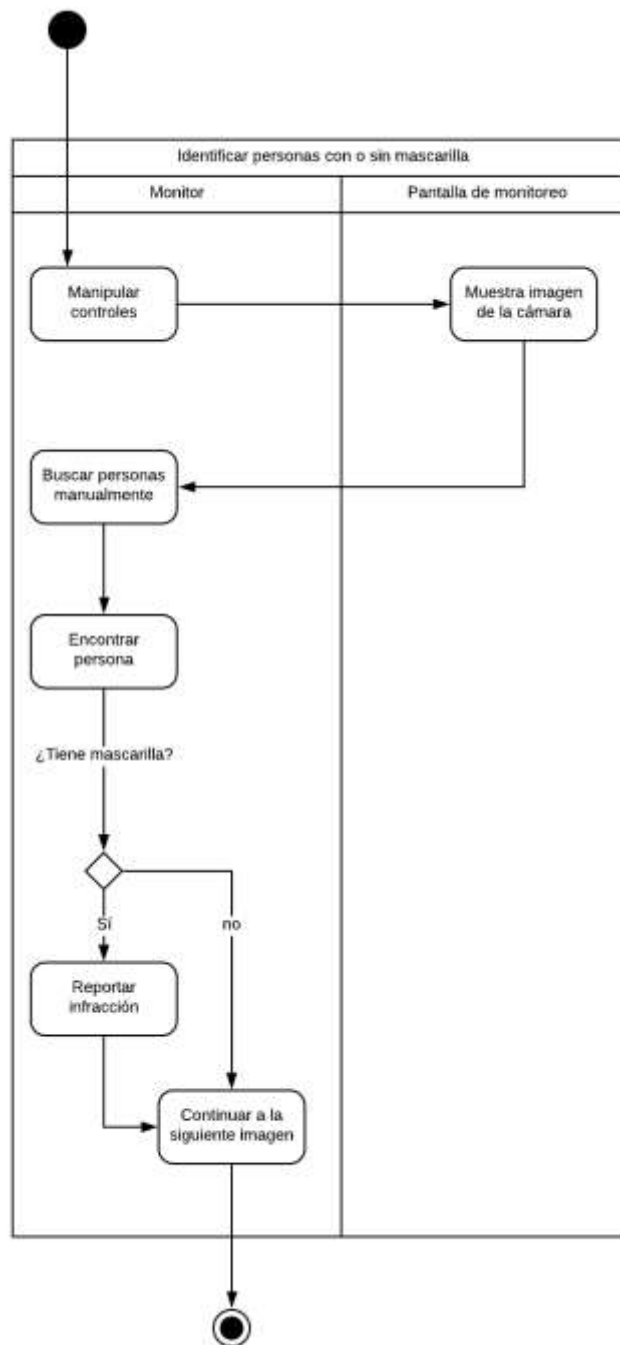


Figura 1 Modelo de definición del problema

Modelos completos del diseño de la solución

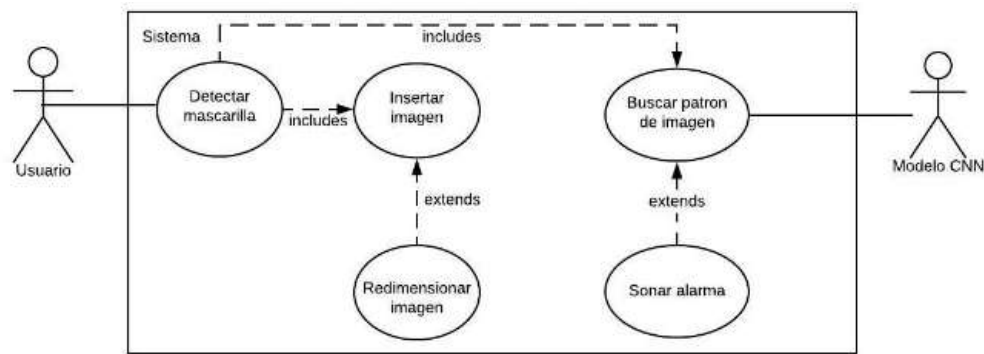


Figura 2 Casos de uso del sistema propuesto

Escenarios de Casos de Uso

El esquema utilizado para la descripción de los escenarios de casos de uso fue tomado a partir de un documento realizado por Gutiérrez et. al [2] sobre la generación automática de pruebas del sistema.

Tabla 1 Escenario de Caso de Uso "Detectar mascarilla"

Nombre de caso de uso	Detectar mascarilla
Precondiciones	No relevante
Actor (es)	Usuario
Descripción	Detectar si una persona en una imagen lleva o no mascarilla
Secuencia Principal	<ol style="list-style-type: none">1. El usuario ingresa una imagen al sistema2. El sistema envía la imagen al modelo CNN3. El modelo busca un patrón en la imagen4. El sistema envía una respuesta al usuario
Errores/Secuencia Alternativa	<ol style="list-style-type: none">1.1. Si la imagen ingresada no es del tamaño adecuado para el modelo CNN, usar caso de uso "Redimensionar imagen"4.1. Si se detectó una persona sin mascarilla, llamar al caso de uso "Sonar alarma"4.2. Si no hay persona en la imagen, el sistema
Resultados	<ol style="list-style-type: none">4. El sistema le indica al usuario si la persona en la imagen tiene o no una mascarilla puesta
Postcondiciones	No relevante

Tabla 2 Escenario de Caso de Uso "Redimensionar imagen"

Nombre de caso de uso	Redimensionar imagen
Precondiciones	Usuario inserta una imagen con dimensiones diferentes a las de la entrada del modelo CNN
Actor (es)	Sistema
Descripción	Cambiar las dimensiones de una imagen a las aceptadas por el modelo CNN
Secuencia Principal	<ol style="list-style-type: none">1. El sistema verifica la dimensión que el modelo CNN necesita
Errores/Secuencia Alternativa	
Resultados	<ol style="list-style-type: none">1. El sistema cambia las dimensiones de la imagen
Postcondiciones	No relevante

Tabla 3 Escenario de Caso de Uso "Sonar Alarma"

Nombre de caso de uso	Sonar alarma
Precondiciones	Modelo CNN detectó una persona sin mascarilla en la entrada
Actor (es)	Sistema
Descripción	Se inicia una alarma indicando que se encontró una persona sin mascarilla en la imagen ingresada
Secuencia Principal	1. El sistema lanza una alarma
Errores/Secuencia Alternativa	
Resultados	1. El usuario recibe una alarma en el sistema indicando que la persona en la imagen no lleva mascarilla
Postcondiciones	No relevante

Descripción de arquitectura del modelo CNN propuesto

La solución propuesta busca implementar DL mediante el uso de redes neuronales convolucionales (CNN), las cuales mostrarán en las imágenes de entrada si es que:

- Hay personas en la imagen
- Las personas tienen una mascarilla puesta

Se optó por implementación de modelo con arquitectura CNN debido a que estas son las mejores en cuanto a reconocimiento de patrones en imágenes, las cuales pueden encontrarse ubicadas en diferentes posiciones del espacio tridimensional y además representadas en múltiples formas en cuanto a mascarillas se refiere. Como ejemplo tenemos la arquitectura YOLO propuesto por Redmon et al. [3]. La Figura 3 muestra la arquitectura del modelo en detalle. La red consta de 24 capas convolucionales, seguidas de 2 capas densas.

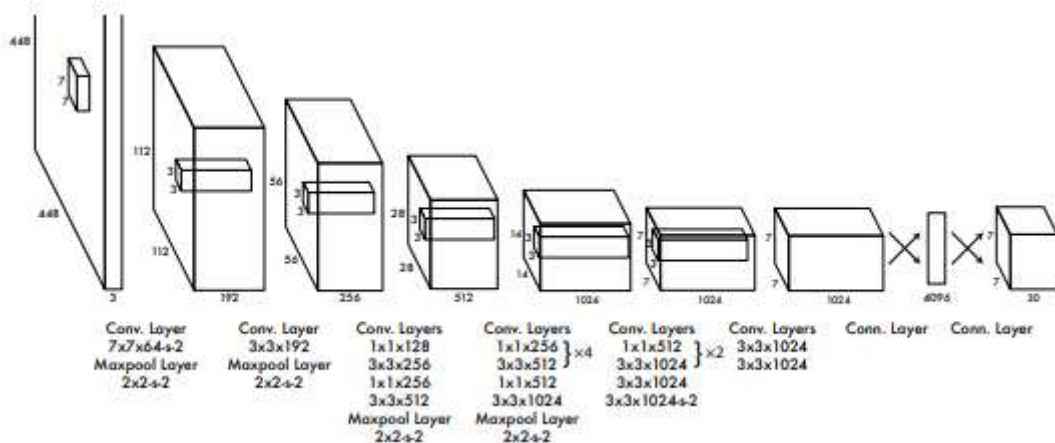


Figura 3 Arquitectura del modelo YOLO

La solución consta de 3 capas convolucionales y 3 capas de pooling. Para las capas convolucionales, la cantidad de filtros aplicados se elevó en potencia de 2 empezando desde 64 filtros en la primera capa; para el tamaño de los filtros, en cambio, se redujo en 2 a medida que se avanzaba en las capas. En el caso de las capas de pooling, se utilizaron filtros de 2x2 al ser esta cantidad el estándar a utilizar, y el tipo de pooling a utilizar fue el average pooling.

Originalmente se prefirió utilizar average pooling pues, según lo investigado [4], este resulta óptimo que utilizar max o min pooling, los cuales se utilizan para fotos con pixeles más brillantes

y oscuros respectivamente; esto es porque las imágenes que se utilizarán como dataset train/test serán tomadas a diferentes momentos del día, por lo que la luz resulta un factor variante (es decir, no solo son fotos brillantes u oscuras). Sin embargo, después de varias pruebas comparativas se optó que, para el dataset que se tiene actualmente, el uso de max pooling daba mejores resultados.

Como referencia para el aumento de número de filtros se utilizó información respecto al sistema Yolo, el cual iba aumentando en potencia de 2. Es decir, en nuestro caso pasamos de 64 a 128.

Finalmente, para el model base se tomó la arquitectura de prueba que se tiene en la documentación de TensorFlow [5], y se la modificó aplicando las bases y patrones encontrados en el modelo Yolo durante nuestra investigación.

Arquitectura de la solución propuesta

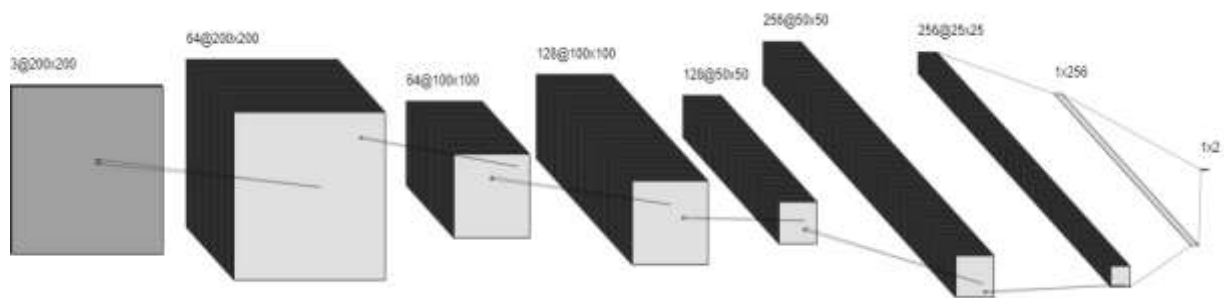


Figura 4 Arquitectura del modelo CNN propuesto

Descripción de la implementación

Para llevar a cabo la arquitectura se decidió por usar el lenguaje de programación Python, el cual ofrece una gran cantidad de librerías orientadas a la inteligencia artificial y el respectivo entrenamiento de machine learning. De estas últimas, hemos optado por usar Keras (en TensorFlow), ya que ofrece una gran variedad de funcionalidades y en especial la implementación de CNN. Para instalar las librerías utilizadas se usaron los siguientes comandos:

- `pip install tensorflow-cpu`
- `pip install matplotlib`

Para usar el modelo generado, se presentará al usuario final una interfaz gráfica, como se puede ver en la Figura 5. El programa carga el modelo entrenado y pide al usuario cargar una imagen. Al cargarla, se envía esta imagen al modelo, y el resultado obtenido (tanto la clase clasificada como la precisión de la clasificación) se muestra en pantalla.

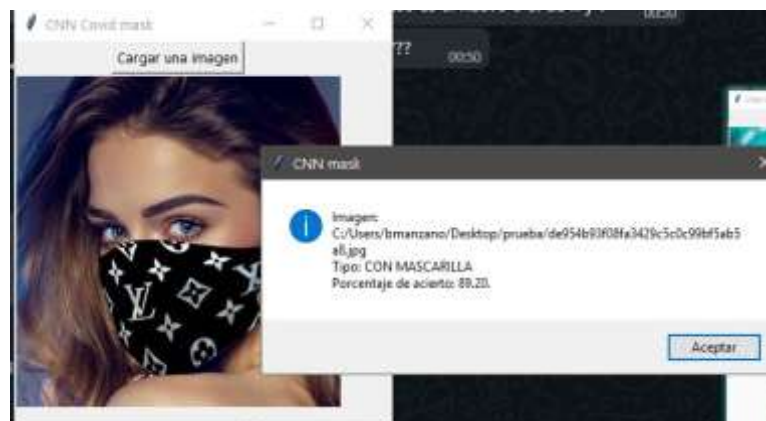


Figura 5 Interfaz gráfica

Hiper-parámetros utilizados

Tabla 4 Hiper-parametros utilizados

Hiper-parámetro	Valor/modo
Número de épocas	20
Tamaño de entrada	200x200
Train/test set	80% - 20%
Validation set	20%
Semilla	123
Dropout	0.3
Pooling	Max-pooling
Núm. filtros	64, 128, 256
Tamaño filtros	5, 3, 3
Data Augmentation	Valor
Rotación	0.2 ($[-20\% * 2\pi, 20\% * 2\pi]$)
Zoom	0.2 (80% - 120%)

El número de épocas se lo utilizó a partir del método del codo en el gráfico de pérdida, el cual nos mostró que la curva dejaba de ser pronunciada a partir de las 20 épocas.

El tamaño de entrada fue escogido de manera arbitraria luego de analizar el rango de tamaños de las diferentes imágenes que fueron utilizadas para el entrenamiento del modelo. Puesto que estas imágenes eran enfocadas en el rostro de una persona y sus alrededores cercanos, el tamaño de las imágenes era considerablemente menor al de una imagen panorámica, de aquí que se usara 200x200.

Para la división de los datos de entrenamiento, testeo y evaluación, se dividió así el dataset en base tanto a la implementación de la arquitectura YOLO como de la experiencia propia entrenando modelos.

Se utilizó dropout pues en iteraciones iniciales la precisión de entrenamiento era demasiado alta en comparación a la validación, por lo que se tenía overfitting. Se probó utilizando dropout también de 0.5, pero los resultados ya eran notables con 0.3.

Como se explicó anteriormente, los valores de los filtros (el tamaño y la cantidad) fueron tomados a base de la arquitectura YOLO, incrementando en potencia de 2 para la cantidad, y disminuyendo en 2 para el tamaño.

Resultados

De las diferentes iteraciones que se tuvo en el transcurso del proyecto con respecto al entrenamiento del modelo CNN, el último modelo generado (Figura 6) logró alcanzar un accuracy mayor a 0.95 con la data de entrenamiento, mientras que para la data de validación se alcanzaron accuracies cercanos a 0.95. Con esto se logró superar el accuracy del modelo base en aproximadamente 2%.

Otra mejoría significativa fue superar el underfitting que se veía en los resultados originales (Figura 7). El primer modelo presentaba picos de bajada en su gráfico de accuracy del set de validación, lo que significaba que el modelo generalizaba demasiado. Aumentando el número de capas se logró mejorar el modelo.

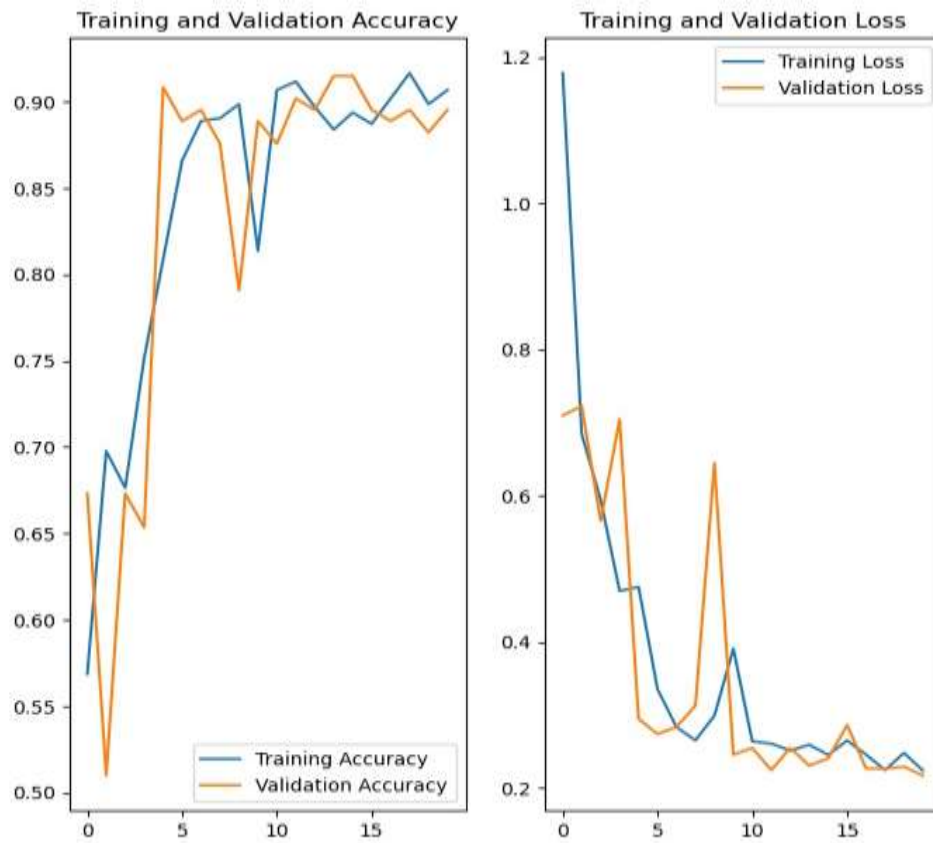


Figura 6 Resultados del modelo implementado

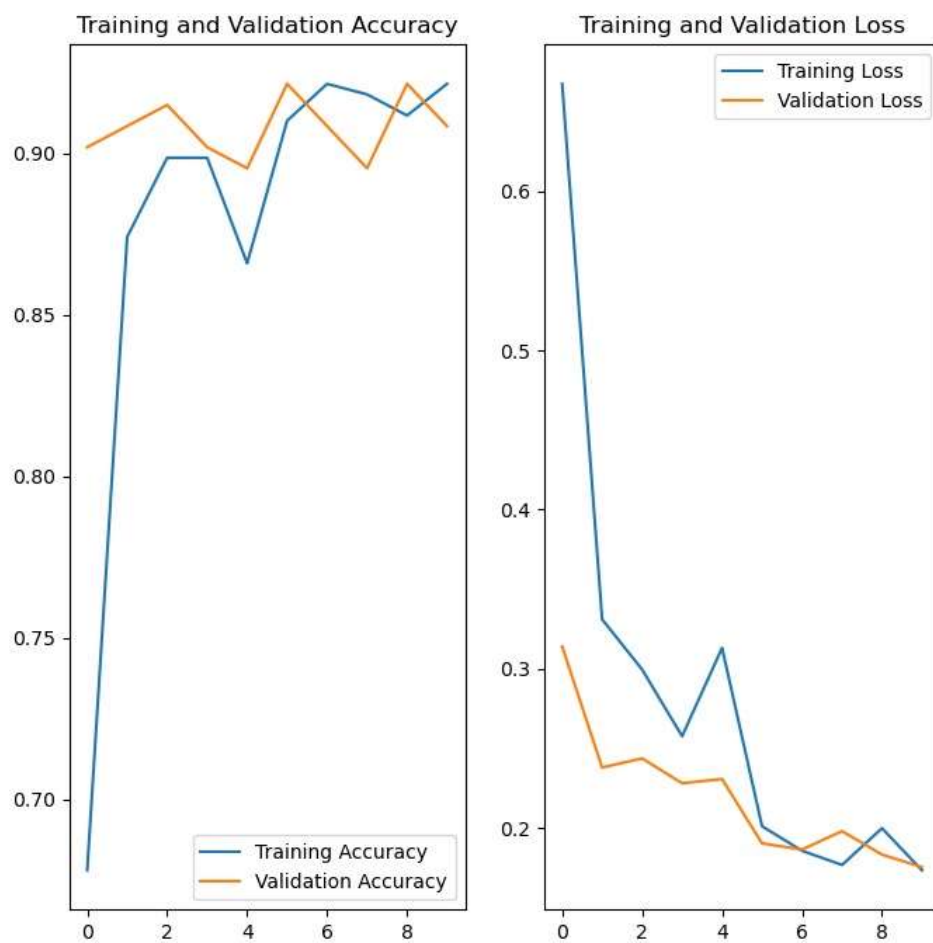


Figura 7 Resultados del modelo original

Adicional a esto, al momento de probar el modelo con imágenes de personas que no tenían mascarilla pero que de alguna manera estaban con otro objeto en su rostro (barbas o manos), el modelo continúa catalogando erróneamente la imagen. Anteriormente se probaron estas mismas imágenes con los primeros modelos, y aunque no se logró que el modelo distinguiera que eran objetos diferentes a mascarillas en una cantidad significativa, el modelo final alcanzó el menor valor de precisión errónea entre los modelos diseñados, además de poder catalogar correctamente a unos casos de personas con barba como sin mascarilla.

Contribuciones

Se diseñaron modelos utilizando Average pooling debido a la forma en la cual se conseguirían las imágenes en la aplicación.

La precisión en comparación al modelo original (TensorFlow) aumentó para los casos correctos y disminuyó para los incorrectos, y se logró superar el underfitting.

Conclusiones

- La arquitectura CNN propuesta supera en precisión a la original en $\sim 2\%$.
- Hace falta idear una estrategia para los casos de personas con barba (bajas accuracies).
- Utilizar imágenes con mayor contexto (no solo el rostro de la persona) mejoró en gran manera al modelo.
- El modelo puede ser mejorado aún más si se aplican mejores técnicas para escoger los hiper-parámetros.

Como posibles trabajos futuros, se puede implementar el modelo diseñado con un dataset más diverso, que incluya los casos de personas con otros objetos cubriendo sus rostros, a modo de que reconozca la diferencia entre estos y las mascarillas.

Adicional a esto, también se pueden incluir imágenes de personas en diversos momentos del día, donde la falta de luz afecte los casos catalogados. El objetivo de esto sería principalmente el poder probar de manera efectiva el uso del average pooling, pues era una propuesta que como tal no pudo probarse tanto como se esperaba originalmente, y esto permitiría descartarlo o implementarlo de manera definitiva.

Bibliografía

- [1] El Comercio, «La mayoría de cantones decide abrir sus playas paulatinamente,» 5 Agosto 2020. [En línea]. Available: <https://www.elcomercio.com/actualidad/cantones-playas-ecuador-apertura-covid19.html>. [Último acceso: 18 Agosto 2020].
- [2] J. Gutiérrez, M. Escalona y M. Mejías, «Departamento de Lenguajes y Sistemas Informáticos - Universidad de Sevilla,» 2006. [En línea]. Available: <http://www.lsi.us.es/~javierj/publications/Hito02.pdf>. [Último acceso: 18 Agosto 2020].
- [3] J. Redmon, S. Divvala, R. Girshick y A. Farhadi, «You Only Look Once: Unified, Real-Time Object Detection,» de The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, 2016.
- [4] M. Basavarajaiah, «Maxpooling vs minpooling vs average pooling,» 8 Febrero 2019. [En línea]. Available: <https://medium.com/@bdhuma/which-pooling-method-is-better->

maxpooling-vs-minpooling-vs-average-pooling-95fb03f45a9. [Último acceso: 27 Julio 2020].

- [5] Tensorflow, «Convolutional Neural Network (CNN),» 9 Febrero 2020. [En línea]. Available: <https://www.tensorflow.org/tutorials/images/cnn>. [Último acceso: 2020 Agosto 27].
- [6] Na8, «¿Cómo funcionan las Convolutional Neural Networks? Visión por Ordenador,» 29 Noviembre 2018. [En línea]. Available: <https://www.aprendemachinelearning.com/como-funcionan-las-convolutional-neural-networks-vision-por-ordenador/>. [Último acceso: 27 Julio 2020].
- [7] AMP Tech, «Cómo hacer un clasificador de imagenes desde cero con Tensorflow,» 2018. [En línea]. Available: https://www.youtube.com/watch?v=EAqb20_4Rdg&t=849s. [Último acceso: 27 Julio 2020].
- [8] Universidad Nacional Abierta y a Distancia, «Diagramas de Actividades,» [En línea]. Available: http://stadium.unad.edu.co/ovas/10596_9839/diagramas_de_actividades.html. [Último acceso: 27 Julio 2020].

Luz verde: implementación de un sistema de detección del uso de mascarillas por parte de la ciudadanía ecuatoriana

PROBLEMA

Actualmente, con el cambio en los semáforos implementados para la emergencia sanitaria, la cantidad de personas que vuelva a transitar por las ciudades estará aumentando lentamente hasta esperar alcanzar eventualmente los niveles que se tenían regularmente. Sin embargo, a pesar de que las entidades hallan dispuestos de normativas a seguir en estas etapas, el riesgo de individuos que estas no se rijan es vigente lo que aumentaría la posibilidad de un rebrote.

OBJETIVO GENERAL

- Vigilar mediante imágenes el cumplimiento de las normas de salud como contramedidas para la transmisión del Covid-19 mediante la detección de mascarillas en los ciudadanos ecuatorianos.

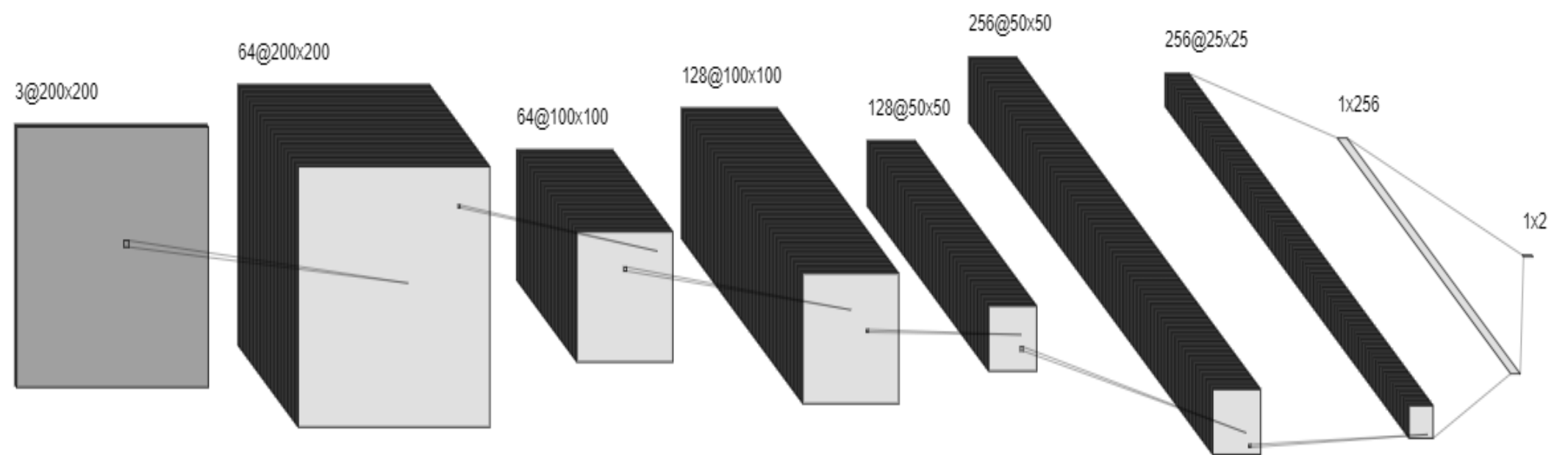


SOLUCIÓN

- La solución propuesta busca implementar DL mediante el uso de redes neuronales convolucionales (CNN), las cuales mostrarán en las imágenes de entrada si es que:

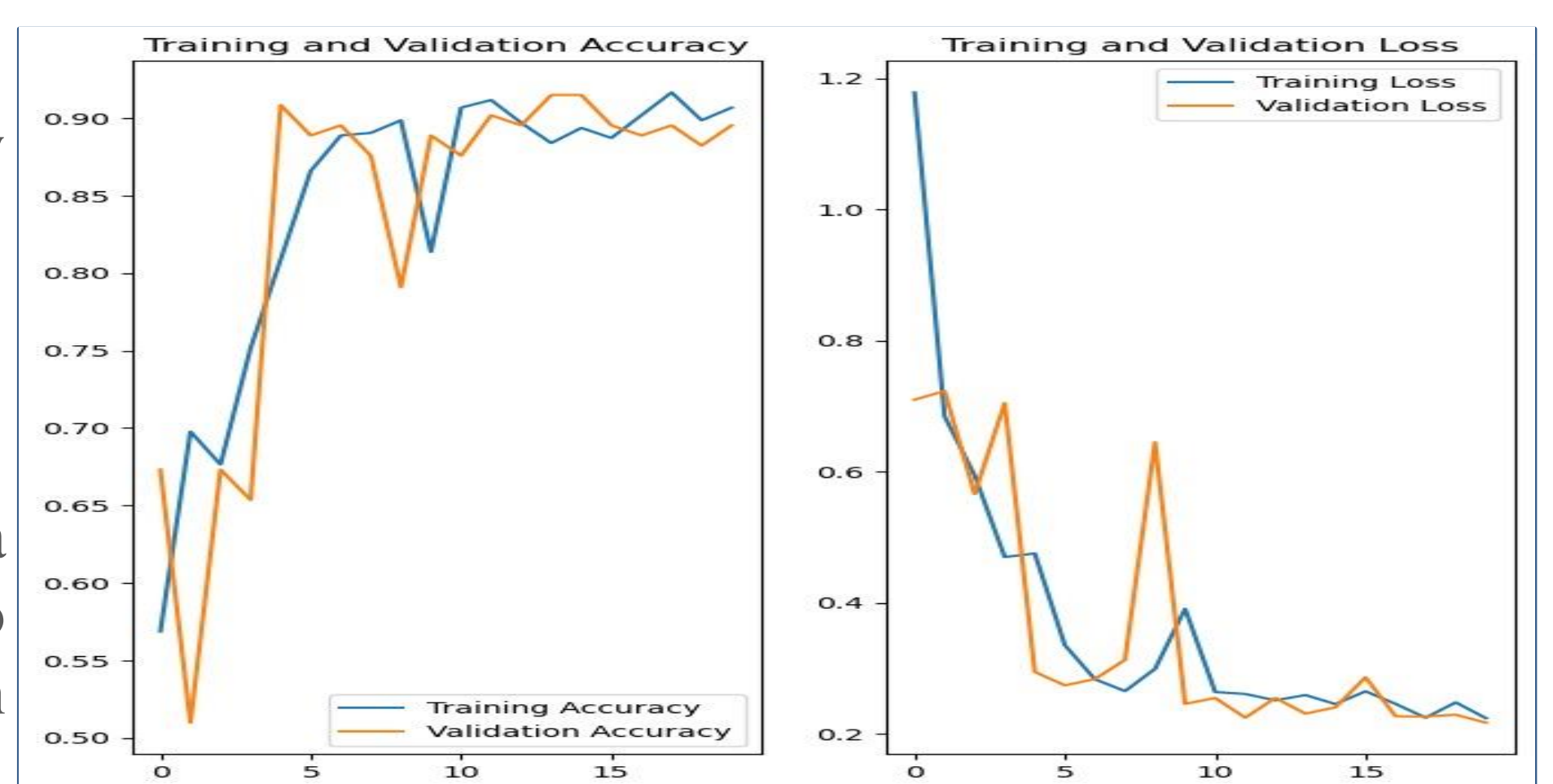
- Hay personas en la imagen
- Las personas tienen una mascarilla puesta

- 3 capas convolucionales
- Numero de filtros aumentando en potencia de 2 empezando desde 64 filtros en la primera capa
- Tamaño de filtros disminuyendo en 2 empezando con tamaño 5 (3 en el resto de las capas)



RESULTADOS

- El modelo final alcanza en el entrenamiento un accuracy >0.95, con un accuracy de validación de aprox. 0.95
- Se superó el underfitting que se tenía en el modelo original
- Para casos donde hay un objeto diferente a una mascarilla en la cara (barbas o manos), el modelo continua catalogando erróneamente la imagen, pero con una precisión significativamente menor.



CONCLUSIONES

- La arquitectura CNN propuesta supera en precisión a la original en ~ 2%
- Hace falta idear una estrategia para los casos de personas con barba (bajos accuracies)
- Utilizar imágenes con mayor contexto (no solo el rostro de la persona) mejoró en gran manera al modelo
- El modelo puede ser mejorado aún más si se aplican mejores técnicas para escoger los hiper-parámetros

NUESTRA CONTRIBUCIÓN

- Comparación con Average pooling debido a la forma en la cual se conseguirían las imágenes en la aplicación
- La precisión en comparación al modelo original (TensorFlow) aumentó para los casos correctos y disminuyó para los incorrectos