UNIVERSIDAD DEL NORTE

Departamento de ingeniería eléctrica y electrónica

Práctica para IEN 42023: Tópicos Especiales en Deep Learning

Profesor: Winston Percybrooks Bolivar

Estudiantes: Adrián Barranco Carlos, Álvaro Herrada Coronell y Hernán Yepes Fernández

1. INTRODUCCIÓN

El presente documento tiene como objetivo proponer, implementar y evaluar el comportamiento de una red neuronal capaz de detectar un rostro humano en una imagen e identificar si el sujeto detectado porta un tapabocas bien puesto, mal puesto, o si directamente no porta tapabocas, todo esto en tiempo real y con la posibilidad de detectar y clasificar múltiples rostros en simultáneo.

Se tomó como base un modelo **ResNet 18**[1] modificado para identificar debilidades en la red y puntos específicos en los cuales trabajar para implementar una solución satisfactoria a los siguientes requerimientos:

- Proponer, implementar y evaluar estrategias de entrenamiento para hacer más robusta la aplicación final de la red.
- Escalar la aplicación para que trabaje con más de una persona al tiempo.

Para poder proponer soluciones factibles a los problemas identificados, primero se planteó el contexto en el cual se aplicaría la solución propuesta. Para saber cuáles problemas podrían ser solucionados directamente con la aplicación y cuáles in situ con medidas como (no usar gorras, observar siempre a la cámara, correcta iluminación, etc). Se pensó en su implementación en un cubículo de atención al cliente de un banco con máximo 2 sillas para los clientes.

Una vez identificadas las debilidades de la red, y los distintos puntos a mejorar para cumplir con la aplicación final, se evaluaron modificaciones tanto en la arquitectura de la red neuronal y el *dataset* de entrenamiento; como la utilización de distintos modelos de redes neuronales y el posible uso de aplicaciones en paralelo con la red para optimizar su desempeño.

El resto del documento estará conformado de los procedimientos llevados a cabo para proponer la solución, los resultados de las distintas alternativas propuestas y el análisis de cada una. Por último se presentará una breve conclusión del taller, y de las ventajas y retos de la solución obtenida.

2. IDENTIFICACIÓN DEL PROBLEMA Y POSIBLES SOLUCIONES

Se partió de la implementación de un modelo **ResNet 18** modificado a través de un proceso de transfer learning con un dataset de entrenamiento conformado por dos clases: “Mask” y “No

mask”; es decir, sujeto con y sin tapaboca, respectivamente. Se implementó el modelo en distintas condiciones de luminosidad y a distintas distancias para identificar los posibles puntos a mejorar del. Los distintos casos evaluados están consignados en la tabla, donde SI representa que el modelo acertó y NO, que el modelo erró.

Tabla 1. Casos evaluados **ResNet 18**.

ResNet-18 Original	Frente (50 cm)		Cercano (1 m)	
	Luz Baja	Luz Alta	Luz Baja	Luz Alta
Con Tapabocas	SI	SI	NO	NO
Sin Tapabocas	SI	SI	NO	NO
Con la mano	NO	NO	NO	NO
Otro Objeto	NO	NO	NO	NO
Tapaboca mal puesto	NO	NO	NO	NO

n° SI =	4
n° NO =	16
Accuracy =	13,33%

SI = Acierto
NO = Desacierto

Tomando el modelo **ResNet 18** modificada como referencia se encontraron los siguientes puntos de mejora:

- Red engañada:
    - Red predice “mask” cuando la mano u otro objeto cubre la boca.
    - Red predice “mask” cuando el tapaboca está mal puesto.
- Técnicos:
    - La red no funciona con bajos niveles de iluminación.
    - Los resultados oscilan significativamente aún cuando el rostro permanece inmovil.
    - La red no funciona cuando los rostros se encuentran a gran distancia.
    - La red presenta dificultades identificando tapabocas de color oscuro (Negro, azul turquí).
    - La red se confunde cuando el tapabocas es de color similar al tono de piel del portador.
    - La red se confunde cuando con imágenes de cuerpo entero
    - La red se confunde cuando el individuo presenta accesorios. (Gorras, gafas e incluso barba).
- Limitaciones de diseño:
    - La red no fue diseñada para identificar múltiples rostros en una sola imagen.

Una vez identificadas las debilidades del modelo se plantearon sus posibles causas para poder atacar directamente el problema, poder mejorar su robustez y permitir su implementación con múltiples individuos. Se pensó en las siguientes como posibles causas y soluciones:

PROBLEMA	SOLUCIÓN
La arquitectura de de la red está mal configurada para la aplicación	- Modificar. - Modificar los parámetros a entrena de la red.
La arquitectura del modelo no es lo suficientemente potente.	- Utilizar otro modelo de red
La red es engañada o confundida	- Incluir datos de entrenamiento que contengan accesorios (gorras, gafas, bigote, barba) - Incluir datos de entrenamiento con tapabocas de distintos colores
No identifica imágenes de cuerpo entero o sólo detecta un único sujeto	- Preprocesar la imagen para identificar y recortar el/los rostro(s). Solo el/los recorte(s) serían evaluados por la red neuronal, de forma independiente.

Tabla 2. Problemas y posibles soluciones.

Los problemas asociados con baja iluminación y mala calidad de la imagen capturada por la cámara no son abordados, ya que se asume que para la implementación del modelo las condiciones lumínicas son ideales, todos los bancos cumplen con condiciones de buena iluminación. Además, las limitaciones acarreadas por el Hardware deben ser asumidas por el cliente que vaya a implementar la solución, cámaras de una buena calidad deben ser implementadas.

3. PROCEDIMIENTO

Habiendo identificado los puntos a mejorar y posibles soluciones se planteó la siguiente estrategia de mejora. En primer lugar, se tuvieron en cuenta diferentes arquitecturas de redes neuronales residuales (ResNet) incluyendo la **ResNet 18** inicial para mejorar el modelo, ya que este tipo de arquitectura destaca en el campo de la visión por computadora debido a su rendimiento. De igual manera, se optó por cambiar el *dataset* de entrenamiento por uno más amplio, con el fin de reducir el riesgo de *overfitting* y mejorar la capacidad de respuesta de la red [2]. Luego, se decidió hacer uso de una red de reconocimiento de rostros de tipo **Haar Cascade** [3] con el fin de evaluar las tres clases de la red de reconocimiento de tapabocas únicamente en la parte de la imagen general correspondiente al rostro detectado, eludiendo así el problema de la distancia a la que se debe tomar la fotografía. Posterior a esto, planteó la posibilidad de generalizar el código para que la detección y evaluación del rostro se realicen en tiempo real haciendo uso de una *webcam*. Por último, se propuso mejorar este proceso para que fuera capaz de detectar y evaluar múltiples rostros humanos en simultáneo.

Una vez definida la estrategia de solución, se procedió a su implementación en código, iniciando con la etapa de reconocimiento de rostros humanos, que, como se explicó anteriormente, se consiguió haciendo uso de una red neuronal pre entrenada **Haar Cascade**, que es capaz de detectar las coordenadas de uno o varios rostros humanos dentro de una imagen dada. Haciendo uso de esta red, se programó el reconocimiento facial en tiempo real a través de la *webcam* y se añadió un recuadro sobre cada rostro identificado en la imagen general que captura la webcam, de tal forma que el algoritmo de reconocimiento de tapabocas itere sobre cada rostro, tal y como se observa en la siguiente imagen.

```
import cv2
from PIL import Image
import numpy as np


face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')


#Open webcam via OpenCV
font = cv2.FONT_HERSHEY_COMPLEX
webcam = cv2.VideoCapture(0)
key = ord('0')

while key != ord('q'):
    check,frame = webcam.read()
    height,width = frame.shape[:2]
    key = cv2.waitKey(60)
    img = Image.fromarray(frame)
    img = np.asarray(img)
    gray = cv2.cvtColor(img, cv2.COLOR_RGB2GRAY)
    # Detect faces
    faces = face_cascade.detectMultiScale(gray, 1.1, 3)
    # Draw rectangle around the faces and crop the faces
    for (x, y, w, h) in faces:
        #Verde-mask (0,255,0)
        #Azul-Incorrect (255,0,0)
        #Rojo-No mask (0,0,255)
        bgr = (0, 255, 0)
        cv2.rectangle(frame, (x, y), (x + w, y + h), bgr, 2)

    cv2.imshow('frame', frame)
webcam.release()
print("Camera off.")
cv2.destroyAllWindows()
```

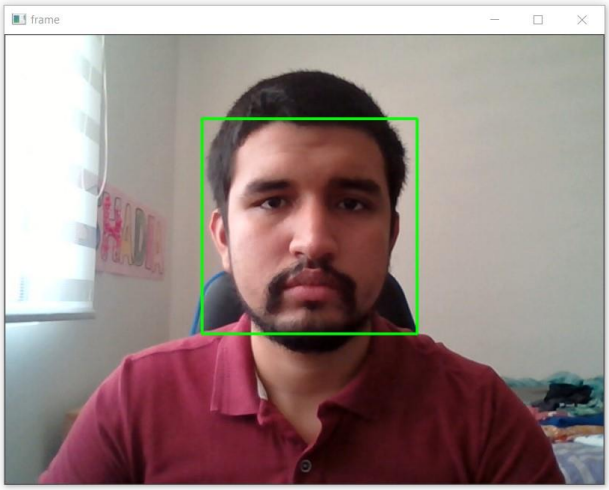

Figura 1. Detector de rostro **Haar Cascade**.

Simultáneamente, se cambió el dataset de entrenamiento de la red neuronal de detección de máscara, por uno que cuenta con aproximadamente tres mil imágenes [2] distribuidas de forma homogénea en cada una de las tres clases a evaluar, ya que una mayor cantidad de datos produce un mejor rendimiento de la red y reduce la probabilidad de que ocurra overfitting.

Hecho esto, se entrenaron y evaluaron las redes neuronales **Resnet 18**, **34** y **152** con este nuevo dataset, realizando transfer learning en cada una de ellas con un total de 5 épocas de entrenamiento por red, esto con el fin de escoger la que mejor realizara el proceso de clasificación no sólo con con un banco de imágenes de evaluación sino también con el uso en tiempo real una vez acoplado el reconocimiento de rostros. Tras finalizar este proceso de evaluación, se escogió como red de reconocimiento de máscaras la **Resnet 152**, pues el rendimiento conseguido con esta demostró ser superior a las otras redes evaluadas. Se considera que este rendimiento se debe a la cantidad de parámetros entrenables y por consiguiente, la mejora en la capacidad de extracción de características, lo que le permite realizar una mejor inferencia sobre la imagen de entrada. Es necesario aclarar que cada una de las redes ResNet fue modificada para que su salida tuviera únicamente las tres clases que requerimos para esta aplicación en particular.

Posteriormente, surgió el problema de acople entre la red de reconocimiento de rostros y la red de clasificación de máscaras, ya que el código inicial permitía utilizar la red **Resnet 18** únicamente con una imagen en formato .jpg como entrada. Sin embargo, la red de reconocimiento facial, tiene como salida las coordenadas del rostro dentro de la imagen capturada, con lo cual, tras una serie de transformaciones, fue posible convertir la información dentro de las coordenadas dadas en un tensor utilizable por la red de reconocimiento de máscaras, tras lo cual, se optó por aplicar un código de colores sencillo para cada caso a evaluar: Un recuadro color verde alrededor del rostro con el texto “Wearing mask” si se detecta que el rostro porta máscara de forma adecuada; Color amarillo con el texto “Incorrect mask”, si se detecta que la máscara está mal puesta (es decir, se tiene la nariz por fuera del tapabocas); por

último, color rojo con el texto “No mask”, si el rostro detectado no porta máscara. Esto puede verse en las siguientes imágenes:

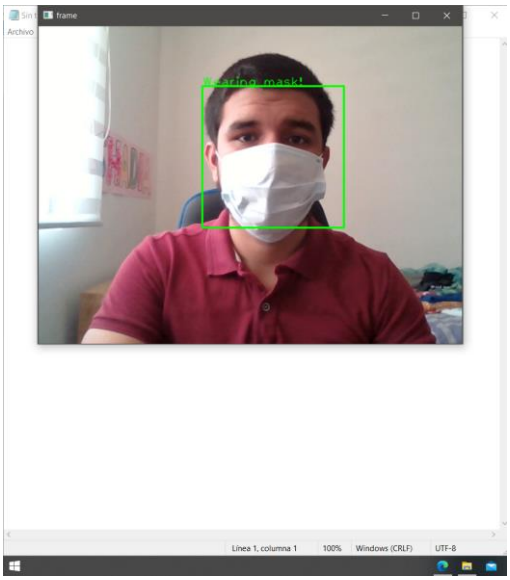

Figura 2a. Detección de tapabocas en caso de “uso correcto”.

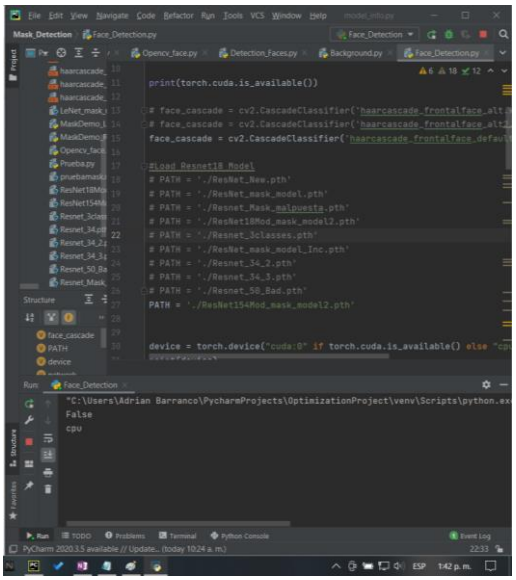

Figura 2b. Código **ResNet 152** + **Haar Cascade**:

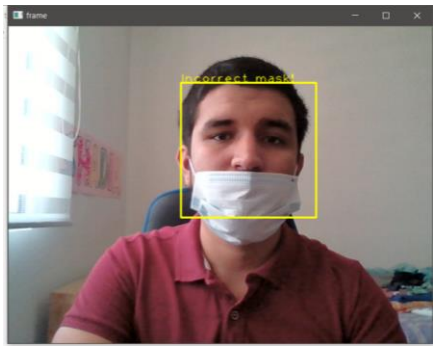

Figura 3a. Detección de tapabocas en caso de “uso incorrecto”.

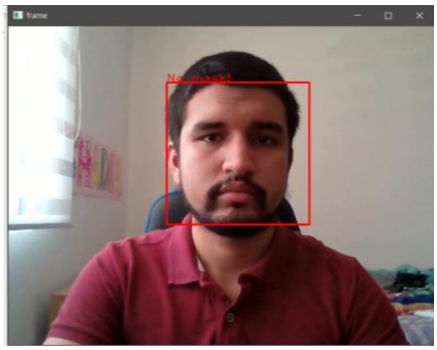

Figura 3b. Detección de tapabocas en caso de uso “sin tapabocas”.

Esta aplicación funciona de forma satisfactoria, sin embargo, fue necesario realizar pequeñas modificaciones en el código para que este pudiera detectar y evaluar múltiples rostros, ya que, al introducir más de uno, la dimensión de la variable img, que contiene el tensor de la imagen original capturada por la *webcam*, se volvía incompatible con la entrada de la red de clasificación de máscaras, generando un error cuando se intentaba probar con múltiples sujetos. Esto ya que la red de reconocimiento facial tiene la capacidad de reconocer varios rostros en simultáneo, con lo que se hizo necesario tener esta característica en cuenta a la hora de acoplar el reconocimiento de máscaras haciendo uso de una variable auxiliar llamada imgP para evitar modificar la variable img original. Tras esto, el programa realiza de forma automática la

detección y evaluación de diferentes rostros en simultáneo y en tiempo real, dando cumplimiento al segundo objetivo planteado en la actividad, tal y como se observa en las siguientes imágenes.

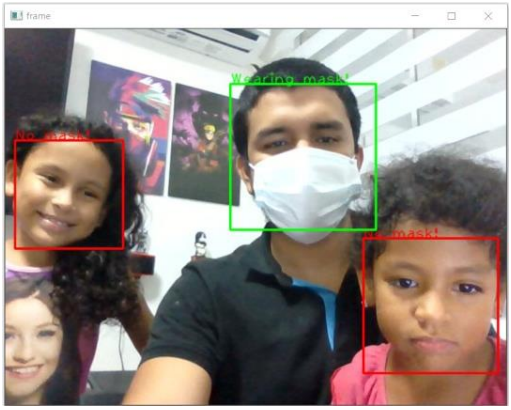

Figura 4a. Detección de rostros y máscaras para múltiples sujetos (uno correcto, dos sin tapabocas).

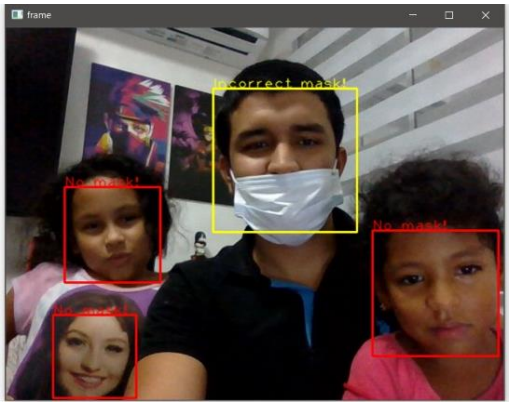

Figura 4b. Detección de rostros y máscaras para múltiples sujetos (uno incorrecto, dos sin tapabocas).

En la figura 4.b) se puede observar que la detección de rostros identifica el rostro estampado en la camiseta de la niña y este luego es procesado por la red neuronal ResNet. Esto no es contado como una falla o un engaño a la red, ya que esta fue entrenada utilizando imágenes de personas, por lo que técnicamente está cumpliendo su función. Identificar la imagen de la mujer en la camiseta y detectar que no tiene tapabocas.

Por último, es necesario aclarar que esta red, pese a ser sustancialmente mejor que la red inicial, sigue presentando dificultades asociadas tanto a la etapa de reconocimiento facial como a la etapa de clasificación de máscaras. Algunas debilidades detectadas en la red ocurren en los siguientes casos:

- Bajas condiciones de luz en las que se captura la imágen.
- Sujetos a evaluar con objetos o accesorios objetos sobre su rostro (como lentes o gorro además del tapabocas).
- Cámara con baja resolución.
- Tapabocas de color oscuro (negro, azul turquí)
- Sujeto observando lejos de la cámara (rostro de perfil).

Esto afecta directamente la capacidad de la red de reconocer la existencia de un rostro humano en la imagen que captura. También se observó que es posible que la red se confunda y detecte objetos o sombras varias como rostros humanos, sin embargo, esto no repercute de forma directa en el rendimiento general de la aplicación, ya que a lo largo de nuestras pruebas notamos que rara vez falla en la detección de rostros humanos reales. Por otra parte, el clasificador funciona de forma satisfactoria en las categorías “Wearing mask” y “No mask” en distancias de aproximadamente 2.5 metros para diversos rostros, sin embargo, la clase que más se le dificulta a esa distancia es la de “incorrect mask”, que logra funcionar de forma correcta en

distancias cortas (entre 0.5 y 1.5 metros de la cámara), pero tiende a fallar en distancias más largas. Consideramos que se debe a que, por limitaciones relacionadas con la calidad de imagen, al aumentar la distancia, se disminuye el tamaño en pixeles del rostro detectado, dificultando la detección de un tapabocas puesto incorrectamente, debido a la falta de resolución en la imagen. Cosa que no ocurre en los casos correctos e incorrectos ya que es mucho más notoria la diferencia.

4. CONCLUSIONES Y RECOMENDACIONES

Tras la realización de la actividad planteada es posible concluir que la aplicación presenta un desempeño satisfactorio en distancias cortas, con una tasa de acierto elevada comprobada a lo largo de nuestras pruebas con uno o más sujetos, lo que nos permite inferir que, con una mejora en el hardware de la cámara, los resultados en distancias mayores también mejorarán proporcionalmente, sin embargo, en caso de no contarse con los recursos necesarios para implementar una mejora de este tipo, se sugiere la utilización de una red neuronal de aumento de resolución de la imagen de cada rostro detectado siempre que esta tenga un tamaño inferior a una determinada resolución, con el fin de que los datos de entrada de la red de clasificación de máscaras tenga la mayor definición posible reduciendo así el error en la salida.

Por otra parte se presentan fallos en la de detección de rostros cuando el sujeto porta accesorios adicionales al tapabocas en su rostro, concluimos que esto es producto de que la red neuronal de detección de rostros no cuenta con un dataset lo suficientemente variado, con lo cual, se sugiere que al momento de realizar la detección, el sujeto porte únicamente el tapabocas para evitar este tipo de fallos. Sin embargo, como estrategia de mejora, se sugiere reentrenar la red neuronal con un dataset más variado que le permita reconocer sin dificultades rostros humanos con múltiples accesorios adicionales al tapabocas. Una solución alternativa que no requeriría el reentrenamiento de la red sería requerir que el sujeto retirase cualquier accesorio al entrar al banco, como es el caso comúnmente (retirar cascos, gorras, sombreros, gafas oscuras, etc.)

Adicionalmente, se observaron fallos relacionados con el tapabocas que utiliza el sujeto o sujetos a evaluar, encontrando que la red sólo es capaz de detectar de forma correcta los tapabocas de colores similares a los típicos azul celeste y blanco, esta limitación surge directamente del dataset utilizado para el entrenamiento de este modelo, ya que sólo cuenta con imágenes de este tipo de máscaras, por ello, se sugiere reentrenar el modelo con un dataset con máscaras de colores más variados, con el fin de que sea posible reconocer múltiples tipos de máscara con menos dificultades.

También se sugiere que el sujeto esté mirando de frente hacia la cámara mientras se requiera su monitoreo, para sobrellevar la limitación de no identificación de rostros de perfil. Si se desea atacar esta limitación directamente desde la red neuronal se podría reentrenar utilizando un dataset de tapabocas bien puestos, mal puestos y no utilizados de perfil; así como reentrenar el detector facial **haar cascade** para que identifique y recorte satisfactoriamente rostros de perfil y/o a distintas inclinaciones.

5. ANEXOS

- Script principal de detección de máscaras en tiempo real.
- Red neuronal de detección de máscaras
- Red neuronal de detección facial
- Los archivos COPYRIGHT y LICENSE correspondientes al uso de la red neuronal de detección facial
- Enlaces a los *datasets* utilizados

6. REFERENCIAS

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 10-Dec-2015. [Online]. Available: https://arxiv.org/abs/1512.03385v1. [Accessed: 20-Apr-2021].

[2] A. Cabani, K. Hammoudi, H. Benhabiles, and M. Melkemi, "MaskedFace-Net – A dataset of correctly/incorrectly masked face images in the context of COVID-19," *Smart Health*, 28-Nov-2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352648320300362?via%3Dihub. [Accessed: 20-Apr-2021].

[3] S. Soo -Object detection using Haar-cascade Classifier. Institute of Computer Science, *University of Tartu,* 28-Nov-2020. [Online]. Available: https://www.academia.edu/download/58974392/Object_detection_using_Haar-final20190420-25685-1k5lqm7.pdf. [Accessed: 20-Apr-2021]