SV(IA)L

Seguridad Vial basada en Inteligencia Artificial

Alumno:	Roberto Carrero Rodríguez	bo0363
	Juan Manuel Gómez Varas	bp0475
	Alicia González Berrocal	bp0400

22 de Noviembre de 2021

ÍNDICE

OBJETIVO

El objetivo de este sistema inteligente es reducir la cantidad de accidentes ocasionados por distracciones producidas por el error humano. Nuestro sistema inteligente, SVIAL, ayudará a prevenir estas distracciones haciendo más segura la conducción tanto para el propio conductor como para los demás conductores.

INTRODUCCIÓN

El tráfico es un problema de salud pública en todo el mundo. En EE. UU. es la primera causa de muerte hasta los 44 años. Al ser un problema que causa tantos accidentes, debemos tomar medidas para tratar de reducir estos datos. Podemos aplicar medidas para disminuir la cantidad de distracciones al volante y poder así, viajar más seguros.

Según los datos proporcionados por la DGT, el coste anual de los accidentes de carretera se elevaría a 520.000 millones de dólares en todo el mundo.

Aunque los accidentes de tráfico se vienen presentando desde hace varios siglos, es en las últimas décadas cuando ha aumentado este problema. La aparición de nuevas tecnologías, como móviles o el uso del GPS, pueden hacer que apartemos la vista de la carretera.

DEFINICIÓN DEL PROBLEMA

Las distracciones siguen siendo la primera causa de los accidentes mortales de tráfico. Concretamente en 2019, la distracción aparece como factor concurrente en un 28% de los accidentes con víctimas mortales, con 359 fallecidos en ámbito DGT.

Esta cifra, ha llevado a la Dirección General de Tráfico a poner en marcha una nueva campaña de vigilancia y concienciación de los peligros que suponen las distracciones al volante.

Además de llevarse a cabo en muchos países europeos al mismo tiempo, también se está llevando a cabo en zonas urbanas, ya que la Dirección de Transportes ha invitado a muchas policías municipales de los ayuntamientos a trabajar en sus respectivos campos de actuación. La Dirección de Transportes agradece su contribución a la seguridad vial.

DISEÑO DE LA SOLUCIÓN PROPUESTA

Para solucionar el problema descrito anteriormente, se ha llegado a la conclusión de que cuando se produce una distracción los ojos se apartan de la carretera, generando un momento de alto riesgo que puede conllevar accidentes.

El objetivo general del sistema inteligente es detectar la falta de atención del conductor a partir de imágenes tomadas por una cámara.

Esto se lleva a cabo detectando los ojos del conductor y su posición, es decir SV(IA)L detecta cuando el conductor tiene los ojos puestos en la carretera o cuando aparta la mirada de la misma.

HERRAMIENTAS SOFTWARE EMPLEADAS.

En primer lugar, usaremos OpenCV para reconocer la cara y los ojos del conductor. Nuestro objetivo será detectar si está mirando al frente o desvía la mirada de la carretera.

Para que la detección de ojos funcione correctamente, definiremos una región de búsqueda. Veremos como extraer los rectángulos con las regiones del ojo derecho e izquierdo. En el caso de que no se detecten los ojos significará que la persona ha girado la cabeza y que, por lo tanto, está distraída.

Posteriormente, se entrenará al sistema inteligente con imágenes que le servirán para diferenciar si los ojos miran a la carretera o a otro lado.

Para llevar a cabo lo mencionado anteriormente, usaremos la biblioteca de aprendizaje automático TensorFlow, con el apoyo del algoritmo de optimización Adam.

El dataset de imágenes de personas que utilizaremos se obtendrá a partir de la siguiente web: https://generated.photos/faces.
"Las fotos generadas se crean desde cero mediante sistemas de inteligencia artificial" y "todas las imágenes pueden ser utilizadas para cualquier propósito sin preocuparse por copyright, derechos de distribución…". En total descargamos y clasificamos 724 fotos, aquellas en las que la persona estaba mirando, se le puso el nombre "m número" y "n número".

Para obtener todas las imágenes a la vez y no descargarlas de foto en foto, se ha desarrollado un script en Python que permite bajar las fotos necesarias de una vez.

Todo esto ha sido realizado en un entorno virtual creado en Anaconda con la versión 3.9.7 de Python.

DESARROLLO DEL SOFTWARE REALIZADO

Funciones programadas:

Hemos creado 3 funciones que nos ayudarán a realizar lo contado anteriormente:

-Set_label:

Recibe como parámetro de entrada el nombre de la imagen a clasificar.
Devuelve un array dependiendo de si está "mirando" o "no mirando".
Hemos cambiado los nombres de las imágenes identificándolos con "m número" y "n número".

-Load data:

No recibe ningún parámetro de entrada.
Devuelve un array con los datos de entrenamiento.
Esta función se encarga de crear un array con los datos de cada imagen que se va a usar para entrenar nuestra red.
Recorta las imágenes (usando OpenCV con haarcascade) de tal forma que sólo nos quedará la parte que nos interesa analizar, los ojos. Usa el directorio de imágenes de entrenamiento (TRAIN_DIR) y además guarda este array en train_data.npy.

-Create Test:

No recibe ningún parámetro de entrada.
Devuelve un array con los datos de testeo.
Esta función se encarga de crear un array con los datos de cada imagen que se va a usar para testear nuestra red neuronal. Usa el directorio de imágenes de entrenamiento (TEST_DIR) y guarda, además, este array en test_data.npy.

PRUEBAS Y TEST

Para realizar la prueba se introdujo 14 fotografías en la carpeta test, 10 no mirando y 4 mirando.

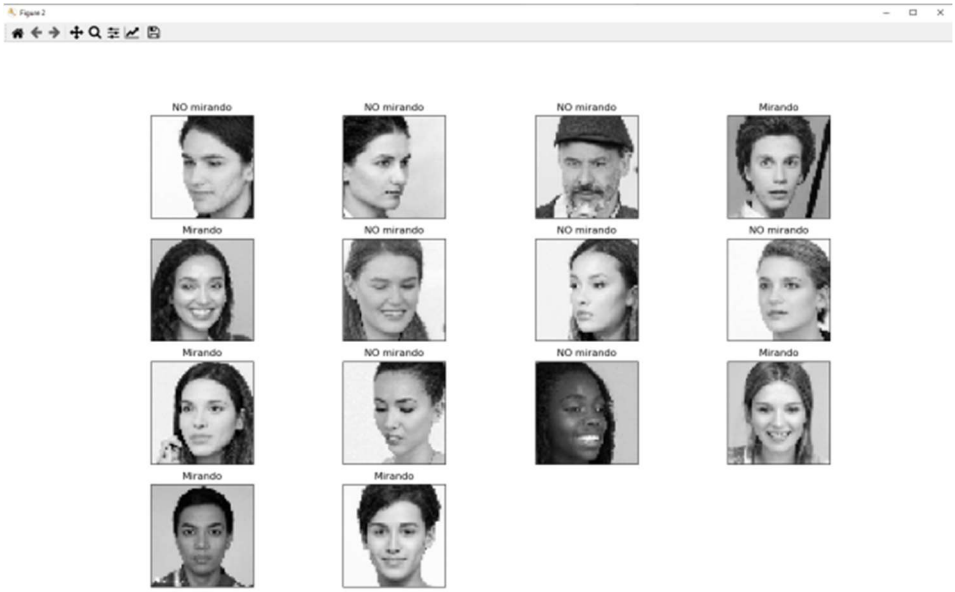

Ilustración 1: Resultado de la ejecución del sistema inteligente.

Como se puede ver en la *Ilustración 2* no identifica el mismo número de personas mirando (6) y no mirando (8), pero algunas de las que nosotros consideramos de forma subjetiva que no miraban, también se puede considerar que sí.

Resultados obtenidos durante las experimentaciones:

Durante los primeros entrenamientos, entrenamos la red neuronal con 10 epochs, pero el porcentaje de acierto oscilaba entre 65-70%. Decidimos aumentar el número de epochs a 30 y nuestro % de acierto también aumentó. Hemos usado Adam https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/ como optimizador con una tasa de aprendizaje de 0.001.

Para ejecutar el programa basta con tener una versión de Python (en nuestro caso, Python 3.9.7) y las dependencias necesarias.
Una vez instalado todo lo necesario, se lanza el programa con el comando "python svIAl.py"

```
(tf) D:\Users\Roberto\Desktop\svIAlProyect>python svIAl.py
```

```
--
Training Step: 147  | total loss: 0.32106 | time: 4.525s
| Adam | epoch: 021 | loss: 0.32106 - acc: 0.8969 | val_loss: 0.44126 - val_acc: 0.8481 -- iter: 434/434
--
Training Step: 154  | total loss: 0.38187 | time: 4.542s
| Adam | epoch: 022 | loss: 0.38187 - acc: 0.8856 | val_loss: 0.44182 - val_acc: 0.8425 -- iter: 434/434
--
Training Step: 161  | total loss: 0.39064 | time: 4.557s
| Adam | epoch: 023 | loss: 0.39064 - acc: 0.8753 | val_loss: 0.42946 - val_acc: 0.8287 -- iter: 434/434
--
Training Step: 168  | total loss: 0.29815 | time: 4.541s
| Adam | epoch: 024 | loss: 0.29815 - acc: 0.8880 | val_loss: 0.44318 - val_acc: 0.8204 -- iter: 434/434
--
Training Step: 175  | total loss: 0.21472 | time: 4.755s
| Adam | epoch: 025 | loss: 0.21472 - acc: 0.9245 | val_loss: 0.57371 - val_acc: 0.8370 -- iter: 434/434
--
Training Step: 182  | total loss: 0.30870 | time: 4.610s
| Adam | epoch: 026 | loss: 0.30870 - acc: 0.9152 | val_loss: 0.43386 - val_acc: 0.8508 -- iter: 434/434
--
Training Step: 189  | total loss: 0.27353 | time: 4.496s
| Adam | epoch: 027 | loss: 0.27353 - acc: 0.9196 | val_loss: 0.40924 - val_acc: 0.8591 -- iter: 434/434
--
Training Step: 196  | total loss: 0.18183 | time: 4.511s
| Adam | epoch: 028 | loss: 0.18183 - acc: 0.9511 | val_loss: 0.66765 - val_acc: 0.8481 -- iter: 434/434
--
Training Step: 203  | total loss: 0.39056 | time: 4.503s
| Adam | epoch: 029 | loss: 0.39056 - acc: 0.9358 | val_loss: 0.59467 - val_acc: 0.8370 -- iter: 434/434
--
Training Step: 210  | total loss: 0.38951 | time: 4.452s
| Adam | epoch: 030 | loss: 0.38951 - acc: 0.9122 | val_loss: 0.42538 - val_acc: 0.8508 -- iter: 434/434
--
```

Ilustración 3: Entrenamiento del sistema inteligente

CONCLUSIONES

El SV(IA)L trata de contribuir a uno de los problemas que más afectan a la sociedad actualmente como son los accidentes de tráfico. En la actualidad los accidentes de tráfico por distracciones son los que más están aumentando.

Este proyecto ha sido desarrollado con éxito cumpliendo con la labor para la cual fue pensado, no obstante, se han encontrado varios problemas a lo largo de su desarrollo como el análisis del dataset, tampoco se ha logrado un 100% de aciertos a la hora de determinar si la cara mira o no.

LINEAS FUTURAS

Se trataría de implementar el sistema en un vehículo de motor, incluyendo modificaciones en el software de detección de imágenes ya que no se basaría en una foto sino en una imagen capturada por una cámara en tiempo real y poder comunicárselo a el conductor en forma de un actuador, ya sea un pitido.

También se trataría de mejorar el software para que SV(IA)L sea capaz de diferenciar entre una distracción y momentos de la conducción en los que se aparta la mirada, pero no por una distracción, como, por ejemplo, mirar a los retrovisores o mirar al freno de mano.