

# Transferencia de Aprendizaje Mediante Redes Neuronales Convolucionales para el Reconocimiento de Conductores Distraídos

## *Transfer learning using convolutional neural networks for driver distraction recognition*

**Cristopher Bazan <sup>1\*</sup>, Zaid Sanchez <sup>2</sup>, Ricardo Rodriguez <sup>3</sup>**

*Facultad de Ingeniería Mecánica, Universidad Nacional de Ingeniería. Av. Túpac Amaru, Rímac. Lima. Perú*

Recibido (Received): 13 /10/2018

Aceptado (Accepted): 15/12/2018

---

### RESUMEN

En el presente trabajo se plantea identificar si una persona se encuentra distraída o no, cuando está conduciendo un vehículo. Esto se logra mediante la clasificación de imágenes de conductores para determinar si están atentos o distraídos usando redes neuronales convolucionales (CNN) y dos técnicas para mejorar el algoritmo, las cuales son Transferencia de Aprendizaje e Ingeniería de Características. Para realizar el entrenamiento se usó imágenes de la competición de Kaggle, en las cuales se hicieron operaciones de rotación y aumento del tamaño para poder tener más datos y obtener mejores resultados. Posteriormente se usó la red extractora de características VGG16, que es un modelo pre-entrenado, de la cual se reemplazaron sus últimas capas para reducir el sobre-entrenamiento y adaptarla a nuestro algoritmo. Los resultados obtenidos en el clasificador nos dan una eficiencia de entrenamiento y validación del 99.30% y 99.46% respectivamente.

*Palabras Clave: Transferencia de Aprendizaje, Ingeniería de Características, Conductores Distraídos, Redes Neuronales Convolucionales*

### ABSTRACT

In this work, we propose to identify if a person is distracted or not, when it comes to driving a vehicle. We experimented with the classification of the images of the controllers to determine if they are attentive or distracted using convolutional neural networks (CNN) and two techniques to improve the algorithm, such as learning transfer and feature engineering. To carry out the training, images of the Kaggle competition were used, in which rotations and expansion operations were performed to obtain more data and obtain better results. Subsequently, the feature extraction network VGG16 was used as a pre-trained model, from which its last layers were replaced to reduce overtraining and adapt it to our algorithm. The results obtained are impressive with a training efficiency and validation of 99.30% and 99.46% respectively.

*Keywords: Transfer learning, Feature Engineering, distracted driver, convolutional neural networks*

---

## 1. INTRODUCCION

Con la gran cantidad de automóviles que circulan a diario a nuestro alrededor, se ha incrementado la tasa de accidentes de tránsito alrededor del mundo. En el año 2017, en el Perú, se registraron 88 168 accidentes de tránsito en los cuales murieron 2 826 personas [1]. El principal causante de esto es la irresponsabilidad del conductor al realizar otras tareas al momento de conducir [2], lo que provoca que no preste atención a la conducción y de un momento a otro pueda ocasionar un accidente. Se han identificado 9 acciones que si se realizan mientras se está conduciendo un vehículo generan gran falta de atención en el camino.

El creciente avance de la tecnología en el área de inteligencia artificial, nos ha permitido proponer una metodología más eficiente que sea capaz de identificar si un conductor está distraído o está conduciendo normalmente, a través del análisis de imágenes del conductor, captadas en el vehículo, y para ello se considera la detección de 10 distintos tipos de acciones del conductor, de los cuales solo una de ellas es conducción segura. Con esta metodología se podrá identificar cuando la persona no está prestando atención y así poder alertarla para prevenir un posible accidente.

---

\* Corresponding author.:  
E-mail: cristopher.bazan@uni.pe

## 2. ESTUDIOS RELACIONADOS

En esta sección se resumen algunos de los artículos más relevantes para la detección de conductores distraídos.

Baheti et al. [3] expone el uso de la arquitectura VGG16 para detectar conductores distraídos usando la base de datos creada por Abouelnaga [4]. El modelo consiste en aplicar varias técnicas de regularización (como “dropout” y “L2 regularization”) para evitar el sobre entrenamiento de la red neuronal. Los resultados muestran que el modelo logra una eficiencia de 96.31% en el conjunto de prueba que se compone de 10 categorías.

Torres et al. [5] desarrolla una arquitectura convolucional estructurada en dos partes, una convolucional y la otra totalmente conectada para una clasificación binaria, es decir para conductores distraídos que usan el celular y para conductores no distraídos. Los resultados de esta investigación muestran una eficiencia del 99%, usando una cámara de video estándar.

Ngan et al. [6] usa una red convolucional particular denominada MS-FRCNN para detectar varios objetos simultáneamente dentro de una misma imagen o video. El modelo propuesto consiste en detectar la actividad del conductor, analizando la posición de las manos, del celular y del volante. La obtención de las imágenes que sirvieron de entrada al algoritmo, proviene de distintos puntos de vista dentro automóvil. Los resultados obtenidos son 94.2% de eficiencia para la detección del uso del celular, y de 93% para la detección de las manos en el volante, además de obtener una buena velocidad de procesamiento 0,06 - 0,09 FPS.

Koeswiady et al. [7] propone una red convolucional para la detección de conductores distraídos usando una red pre-entrenada VGG19 para la extracción de características. Este modelo detecta 4 tipos de clases, y sus resultados son comparados con XGboost, método que ha presentado soluciones exitosas en competiciones Kaggle. Los resultados muestran un 95% de eficiencia en la etapa de test.

Nuestra metodología consta de usar las primeras 13 capas convolucionales de la arquitectura VGG16, para luego implementar una capa convolucional y una completamente conectada, consiguiendo así mejores resultados a nivel de eficiencia.

## 3. METODOLOGÍA

Para el desarrollo de este artículo, se utilizaron principalmente dos técnicas ampliamente conocidas

doi: <https://doi.org/10.21754/tecnia.v28i2.549>

como Transferencia de Aprendizaje (en inglés “Transfer learning”) e Ingeniería de Características (en inglés “Feature Engineering”)

Como lo señala Yosinqui et al. [8] La Transferencia de aprendizaje es una técnica de aprendizaje automático, que consiste en entrenar una arquitectura neuronal para una determinada tarea y que esta arquitectura puede ser reutilizada para otra tarea similar. Comúnmente se utiliza transferencia de aprendizaje de dos formas:

\* Modelo Pre-entrenado: Consiste en utilizar un modelo pre-entrenado, y reemplazar sus últimas capas por otras, para que de esa manera se extraigan las características del nuevo conjunto de datos.

\* Sintonización de la red convolucional: Esta estrategia consiste en sintonizar los pesos de las capas superiores usando propagación hacia atrás (en inglés “backpropagation”)

La Ingeniería de Características es un método que permite aumentar la base de datos original, con la finalidad de que el algoritmo sea menos propenso a los cambios que puedan suceder en los datos de entrenamiento, y de esa manera generalizar aún más los resultados. Este aumento de la base de datos se realiza rotando, trasladando y agrandando la imagen.

Es relevante precisar que para la implementación del algoritmo se utilizaran redes VGG, propuesta por Simoyan and Zisserman [9] debido a la buena relación costo/beneficio. Las redes VGG utilizan múltiples capas convolucionales 3x3 en secuencia, de manera que se puedan extraer características más complejas y también la combinación de tales características.

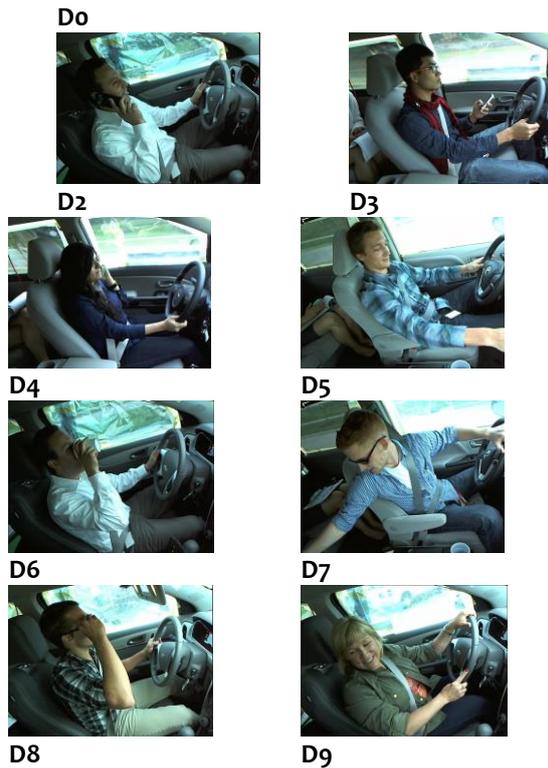
En la Tabla 1 se muestra las distintas categorías a clasificar. Asimismo, en las imágenes siguientes se muestran algunos ejemplos de estas categorías.

TABLA 1. Categorías posibles a identificar

	Descripción	Designación
D0	Conducción normal	Seguro
D1	Usar el celular – Derecha	No seguro
D2	Hablar por teléfono – Derecha	No seguro
D3	Usar el celular – Izquierda	No seguro
D4	Hablar por teléfono – Izquierda	No seguro
D5	Operar la radio	No seguro
D6	Beber	No seguro
D7	Voltearse	No seguro
D8	Maquillarse o arreglarse el cabello	No seguro
D9	Hablar con el pasajero	No seguro



D1



Considerando estas técnicas, se diseñó la metodología mostrada en la Figura 1.

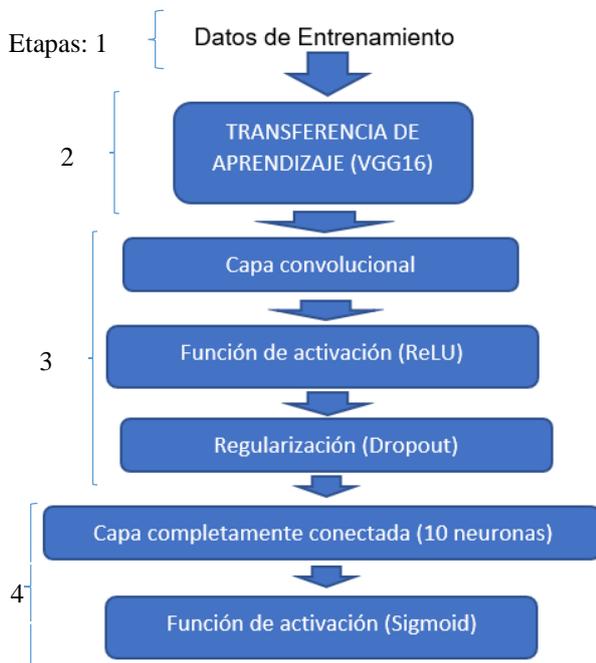


Figura 1. Configuración de la arquitectura utilizada en el algoritmo de clasificación.

Etapa 1: Se aumenta la base de datos realizando rotaciones de 10 y 15 grados y agrandando la imagen, y de esa manera la cantidad de datos de entrenamiento se incrementa, lo que nos permitirá hacer más robusto

al algoritmo entrenado y así poder generalizar mejor los resultados.

Etapa 2: Estas imágenes son redimensionadas, de 640x480 a 224x224 píxeles, de tal manera que estas tengan los requerimientos en tamaño para alimentar la red extractora de características VGG16. Esta arquitectura extraerá e identificará patrones de las imágenes de entrada.

Etapa 3: Posteriormente, se agregará una capa convolucional que permita extraer patrones aún más complejos. Esta capa es acompañada de la función de activación y la capa de regularización, cuya función es reducir el sobre-entrenamiento.

Etapa 4: Finalmente se agrega una capa neuronal completamente conectada de 10 neuronas, acompañada de la Función de Activación (Sigmoid), para clasificar las 10 categorías de la base de datos.

#### 4. EXPERIMENTOS

En la tabla 2, se muestra la distribución de los datos de entrenamiento, validación y prueba de cada clase. Del total de datos obtenidos (22424 datos) de [9], el 70% (15697 datos) de estos son de entrenamiento, el 23.31% (5227 datos) para la validación y el 6.69% (1500) para probar el algoritmo.

TABLA 2. Distribución de los datos obtenidos para Entrenamiento, Validación y Prueba.

	Entrenamiento	Validación	Prueba	Total
D0	1742	597	150	
D1	1587	530	150	
D2	1622	545	150	
D3	1642	554	150	
D4	1629	547	150	
D5	1619	543	150	
D6	1628	547	150	
D7	1402	450	150	
D8	1337	424	150	
D9	1489	490	150	
<b>Total</b>	<b>15697</b>	<b>5227</b>	<b>1500</b>	<b>22424</b>

Después de separar los datos, se procede a desarrollar la metodología descrita sobre los datos de entrenamiento, para lo cual se incrementó de 15 697 a 62 788 imágenes.

#### 5. RESULTADOS

El número de épocas de entrenamiento serán 10, cada época de entrenamiento tiene una duración de 98

segundos aproximadamente, lo que da un tiempo estimado de 980 segundos de entrenamiento, y la eficiencia de entrenamiento y validación fueron de 99.30% y 99.46% respectivamente. En la Figura 2 y en la Figura 3, se muestran dos curvas de aprendizaje con 10 épocas de entrenamiento, las cuales muestran las métricas de eficiencia y pérdida. Además, en la Figura 4 se observa la matriz de confusión de los resultados de validación. En esta matriz, cada fila representa la cantidad de ejemplos de una determinada categoría, mientras que en las columnas se predice a que categoría pertenecen dichas imágenes. De ese modo, la diagonal de la matriz de confusión muestra los ejemplos correctamente clasificados.

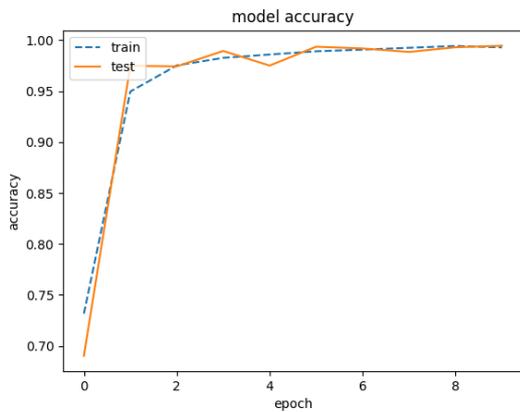


Figura 2. Evolución de la eficiencia en 10 épocas de entrenamiento.

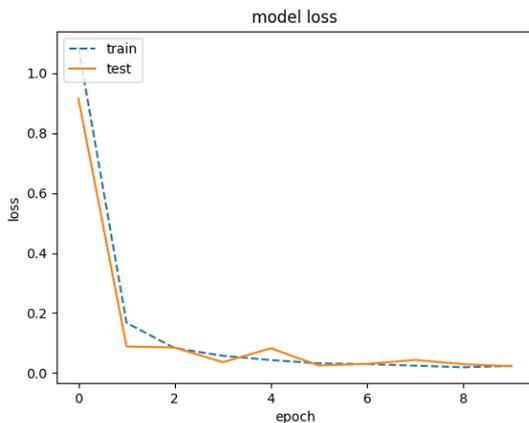


Figura 3. Evolución de la pérdida en 10 épocas de entrenamiento.

Con la finalidad de demostrar la generalización del algoritmo, este algoritmo fue evaluado en otro conjunto de prueba de 1500 imágenes, cuya eficiencia obtenida fue de 0.9946, tal como se detalla en la matriz de confusión de la Figura 5. Esta eficiencia indica que el modelo se puede generalizar a la clasificación de conductores distraídos, considerando que la cámara se encuentra en la parte superior del auto cerca del copiloto.

D0	593	0	0	1	0	1	0	0	0	2
D1	0	530	0	0	0	0	0	0	0	0
D2	0	1	543	0	0	0	1	0	0	0
D3	0	0	0	554	0	0	0	0	0	0
D4	0	0	0	1	546	0	0	0	0	0
D5	3	0	1	0	0	538	1	0	0	0
D6	0	1	1	0	0	0	545	0	0	0
D7	0	0	0	0	0	0	1	449	0	0
D8	1	0	1	0	0	0	0	0	421	1
D9	9	0	0	0	1	0	0	0	0	480
	D0	D1	D2	D3	D4	D5	D6	D7	D8	D9

Figura 4. Matriz de confusión para los datos de Validación

D0	149	0	0	1	0	0	0	0	0	0
D1	0	150	0	0	0	0	0	0	0	0
D2	0	0	150	0	0	0	0	0	0	0
D3	0	0	0	149	1	0	0	0	0	0
D4	0	0	0	0	150	0	0	0	0	0
D5	0	0	0	0	0	150	0	0	0	0
D6	0	0	0	0	0	0	150	0	0	0
D7	0	0	0	0	0	0	0	150	0	0
D8	1	1	1	0	0	0	0	0	146	1
D9	0	1	0	0	0	0	0	0	1	148
	D0	D1	D2	D3	D4	D5	D6	D7	D8	D9

Figura 5. Matriz de confusión para los datos de Validación

## 6. CONCLUSIONES

En el presente artículo se muestra un sistema robusto y eficiente para detectar conductores distraídos que además indica la causa de la distracción. En el experimento realizado, nuestra red presenta una eficiencia de más del 99% usando transferencia de aprendizaje e ingeniería de características, y ello combinado al echo que nuestra solución requiere una cámara estándar.

Otra conclusión importante, es que nuestro algoritmo detecto algunos datos, en el conjunto de entrenamiento, incorrectamente etiquetados; es decir, en el conjunto de entrenamiento existen imágenes que fueron etiquetadas según el criterio del ser humano, sin embargo, nuestro algoritmo detecto imágenes que fácilmente podrían ser identificadas como de otra categoría, lo que indicaría la eficiencia obtenida podría ser aún mayor. Aquí se muestran algunas de ellas.

Por lo tanto, se concluye que la solución presentada es prometedora y que pronto podrá ser evaluada en un sistema embebido ubicado en un auto, de tal manera que pueda ser una alternativa a considerar en la seguridad del conductor y los pasajeros.



Figura 6: Ejemplo de Validación, etiquetado como Do, cuyo nombre en la data de entrenamiento es “img\_12470”



Figura 7: Ejemplo de Validación, etiquetado como Do, cuyo nombre en la data de entrenamiento es “img\_16927”

- en International Conference on Artificial Neural Networks Springer, Cham, 72-79
- [6] Hoang Ngan Le T, Zheng Y, Zhu C, Luu K y Savvides M 2016 Multiple Scale Faster-RCNN Approach to Driver's Cell-Phone Usage and Hands on Steering Wheel Detection en Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 46-53
- [7] Koesdwiady A, Bedawi S M, Ou C y Karray F 2017 End-to-end deep learning for driver distraction recognition en International Conference Image Analysis and Recognition Springer, Cham, 11-18
- [8] Yosinski J, Clune J, Bengio Y y Lipson H 2014. How transferable are features in deep neural networks? en Advances in neural information processing systems 3320-3328
- [9] Simonyan K y Zisserman A 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.
- [10] StateFarm: State Farm Distracted Driver Detection. [Consultado 5 Jul 2018]. Disponible en: <https://www.kaggle.com/c/state-farm-distracted-driver-detection>.



Los artículos publicados por TECNIA pueden ser compartidos a través de la licencia Creative Commons: CC BY 4.0 Perú. Permisos lejos de este alcance pueden ser consultados a través del correo [revistas@uni.edu.pe](mailto:revistas@uni.edu.pe)

## REFERENCIAS

- [1] Oficina De Tecnología De Información Ministerio De Transportes Y Comunicaciones. [Consultado 25 Jul 2018]. Disponible en: <https://www.mtc.gob.pe/cnsv/estadistica.html>
- [2] Ministerio de Salud – MINSA, “Situación de los accidentes de tránsito en el Peru”. [Consultado 25 Jul 2018]. Disponible en: <https://elcomercio.pe/peru/accidentes-transito-750-personas-murieron-carreteras-peru-2018-noticia-574739>
- [3] Baheti B, Gajre S y Talbar S 2018 Detection of distracted driver using convolutional neural network en Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops 1032-1038.
- [4] Abouelnaga Y, Eraqi H M y Moustafa M N 2017 Real-time distracted driver posture classification. arXiv preprint arXiv:1706.09498.
- [5] Torres R, Ohashi O, Carvalho E y Pessin G 2017 A Deep Learning Approach to Detect Distracted Drivers Using a Mobile Phone