Reconocimiento de bostezos para la identificación de cansancio utilizando redes convolucionales

JORDI JAROMIL CRUZ MEDRANO[1], JORGE DE LA CALLEJA[1], MA. AUXILIO MEDINA[1], ANTONIO BENITEZ[1], AND HUGO JAIR ESCALANTE[2]

[1] *Universidad Politécnica de Puebla, Departamento de Posgrado, 3er carril del Ejido Serrano S/N,San Mateo Cuanalá, Juan C. BonillaPuebla, México, 72640, Email: jordi.cruz, jorge.delacalleja, maria.auxilio, antonio.benitez @uppuebla.edu.mx*

[2] *Instituto Nacional de Astrofísica, Óptica y Electrónica, Departamento de Ciencias Computacionales, México, 72840*

Compiled December 4, 2017

El artículo presenta un método para la identificación de cansancio en personas a través del reconocimiento de bostezos usando aprendizaje profundo. El método consta de las siguientes etapas: obtención del conjunto de imágenes, pre-procesamiento, extracción de características, y finalmente la identificación. Se usaron diferentes algoritmos de aprendizaje automático con el objetivo de identificar al mejor para aplicarlo en un sistema en tiempo real. Los resultados preliminares indican que el algoritmo de máquinas de soporte vectorial obtiene los mejores resultados considerando un conjunto de imágenes en donde existan más ejemplos positivos que negativos.

OCIS codes: (140.3490) Lasers, distributed feedback; (060.2420) Fibers, polarization-maintaining;(060.3735) Fiber Bragg gratings.

http://dx.doi.org/10.1364/ao.XX.XXXXXX

1. INTRODUCCIÓN

Diversos y numerosos estudios han revelado que conducir fatigado puede llegar a ser más peligroso que conducir en estado de ebriedad, ya que influye en la habilidad para tomar decisiones adecuadas, así como en el desempeño y tiempo para reaccionar a diversos contratiempos que se pueden suscitar en el momento de conducir. Un conductor fatigado es un riesgo para si mismo y para los restantes usuarios de las vías ya que la fatiga produce un importante incremento en el número y amplitud de errores en la conducción, con disminución de la atención y del nivel de atención necesario para manipular un vehículo. Existe un gran porcentaje de accidentes causados por conductores cansado, que en su mayoría ocasionan perdidas monetarias, heridos y hasta muertos. Según el European Transport Safety Council (ETSC) la fatiga al volante causa casi el 20% de los accidentes de los vehículos comerciales de transporte [1].

Una estrategia viable que permita resolver esta problemática consiste en desarrollar sistemas que monitoreen en tiempo real el estado físico de los conductores. Dicha estrategia no es nueva, ya se han desarrollado diversos sistemas alternativos al respecto; algunos de ellos han sido mermados por diferentes inconvenientes que presentan, como el que sea invasivo para el propio conductor, o bien que sea impreciso por factores externos como el estado de la carretera, del vehículo, y hasta de la misma habilidad del conductor [1].

En el año 2007 Yulan Liang, Michelle L. Reyes, y John D. Lee entrenaron un sistema para la detección de personas distraídas, mediante la detección de los ojos utilizando como algoritmo de aprendizaje a las máquinas de soporte vectorial. Mediante el monitoreo en tiempo real de las personas distraídas, el sistema dio resultados óptimos, sin embargo, al implementarlo en un sistema real se obtuvieron más falsos positivos de los que el sistema en simulación detectó. Esto se debió a que el ojo como único factor de detección de distracción resultaba insuficiente, ya que al conducir la mirada difícilmente se encontraba en una posición fija [2] .

En el año 2015 Yan Zhang y Caijian Hua utilizaron un sensor Kinect en conjunto con herramientas de visión por computadora, para desarrollar un sistema eficiente para la detección de la distracción del conductor y el reconocimiento de diversas acciones consideradas distractoras para el conductor. El sistema para realizar la detección consideraba al ojo (la detección de la mirada y parpadeando), la posición del brazo (el brazo derecho hacia arriba, abajo, derecha hacia adelante), orientación de la cabeza, y expresiones faciales. El sistema fue integrado en diferentes módulos fusionados al final, utilizando estrategias diferentes de clasificación: clasificador AdaBoost y modelos ocultos de Markov. La evaluación se realizó con 8 conductores de distinto sexo, edad y nacionalidad, mediante una conducción en un simulador en un periodo de más de 8 horas de grabación. Los resultados mostraron un 85% de exactitud para el tipo de distracción y 90% para la detección de distracción [3].

En este trabajo de investigación se presentan resultados preliminares para la identificación de cansancio por medio de imágenes que muestren bostezo de personas. Para ello se creó un conjunto de imágenes obtenidas de diferentes buscadores de internet y bancos de datos. Esto con el fin de generar un compendio de imágenes para extraer características usando aprendizaje profundo y clasificarlas aplicando diversos algoritmos de aprendizaje automático y así determinar que combinación es la más idónea, al momento de entrenar un sistema que funcione

en prototipo.

2. APRENDIZAJE PRODUNDO

El aprendizaje profundo es un enfoque relativamente nuevo que representa el acercamiento más cercano en el funcionamiento del sistema nervioso humano, por lo que sus algoritmos están relacionados con las redes neuronales artificiales como lo muestra la figura 1 para su implementación. Su objetivo principal es el poder jerarquizar las características de la información obteniendo niveles de abstracción superiores formadas por diversos niveles de profundidad, los cuales dependerán del nivel de abstracción que el problema requiera. Las redes convolucionales han brindado los mejores resultados para dar solución a diversos problemas, particularmente en el área de aprendizaje automático [4] [5] [6].

En el aprendizaje profundo existen niveles de profundidad y la profundidad del algoritmo dependerá del nivel de abstracción que el problema requiera. También dependiendo de la profundidad dependerá las herramientas necesarias. El rendimiento de las representaciones poco profundas se puede mejorar de manera significativa mediante la adopción en el aumento de datos mejorando con ello como los resultados finales como una mejor abstracción de la información[7].

Como en la mayoría de los enfoques utilizados para el área del aprendizaje automático estos constan de diversos algoritmos que permiten la implementación de estos enfoques para el entrenamiento de los sistemas [7].

A. Redes Convolucionales

Actualmente los modelos jerárquicos de aprendizaje de automático, tales como las redes neuronales de convolución (CNN) están siendo usados para resolver diversos problemas. Las CNN además de ser un método que permite el entrenamiento de redes altamente profundas siendo sustancialmente más sofisticadas, permiten un nivel nuevo de abstracción. Su arquitectura particular permite entrenar redes profundas multicapas, siendo además eficaces en la clasificación de imágenes [5].

Para lograr entender su funcionamiento es preciso conocer sus 3 tareas básicas las cuales son: campos receptivos locales, los pesos compartidos, y agrupamiento de capas[8].

a) Campos receptivos locales(Pool):

Para una red de convolución se considerarán los datos de entrada como una matriz de nxn como lo muestra la figura 1, siendo esta matriz la capa de entrada de la red. En una red neuronal normal la capa de entrada se interconecta con la primera capa oculta. Ya que los tamaños de los datos de entrada pueden llegar a ser muy grandes, no se conectará cada pixel a la siguiente capa, en su lugar las conexiones serán realizadas por pequeñas regiones de la matriz original como se muestra en la Figura 1; estas regiones son conocidas como campos receptivos de los cuales se extraen las características más relevantes obtenidas de esa región para la adquisición de un nodo de la capa oculta consecuente.

Como lo muestra la Figura 2 el campo receptivo se moverá nodo a nodo, también conocido como la longitud del paso en este caso al ser una longitud de uno, se obtiene una matriz de ,

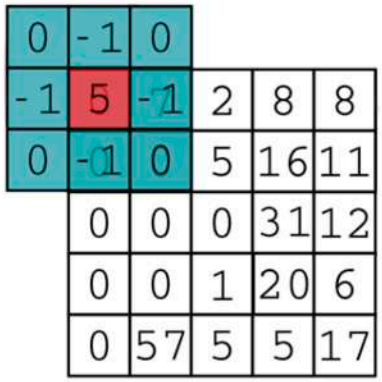

Fig. 1. Representación de los datos de entrada en una Red de Convolución [8].

la longitud de paso en nodos que da el campo receptivo puede variar y de eso dependerá el tamaño de la matriz resultante [5] [7] [8].

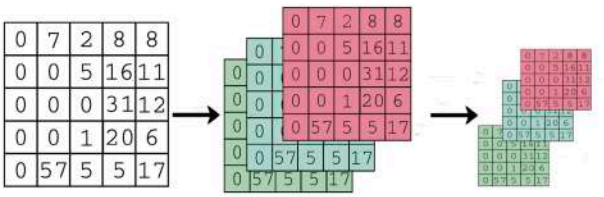

Fig. 2. Se muestra la obtencion de una nueva matriz de información obtenida de la caracterización de una imagen previa [8].

b) Pesos Compartidos:

Cada neurona de la capa oculta cuenta con un mismo sesgo y peso al ser conectados, por lo que la capa oculta detecta exactamente la misma característica previamente detectada. Por esta razón se le llega a llamar al mapa de la capa oculta como mapa de características. La estructura de la red descrita hasta el momento solo ha mostrado un mapa de características, y para una convolución completa es necesario de diferentes mapas de características que responde a los pixeles de entrada como lo muestra la Figura 3 del cual extrajeron 3 mapas cada una del mismo tamaño pero diferentes sub-regiones [6], [5], [8].

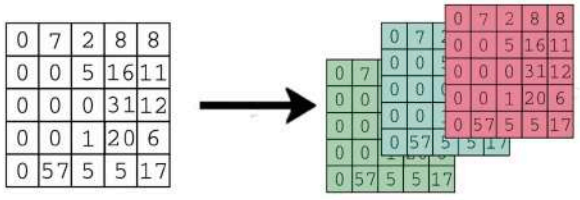

Fig. 3. Se muestra las 3 capas de características generadas [8].

c) Agrupamiento de capas:

La agrupación de capas se utilizan por lo general inmediatamente después de las capas convolucionales. Lo que las capas de la puesta en común hacen es simplificar la información en la salida de la capa convolucional como se muestra en la Figura 4. Esta tarea es similar que los campos receptivos locales pero

en este caso se simplifica la información obtenida en la capa oculta para poder al final obtener un vector de información el cual se convertiría en la capa de entrada para la red en su entrenamiento[6], [5], [8].

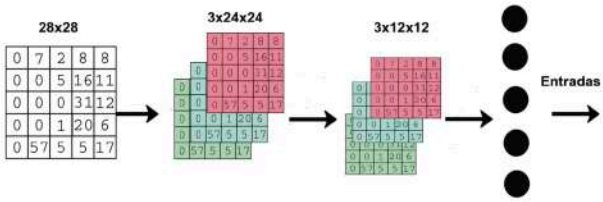

Fig. 4. Se muestra la imagen de datos obtenidas de una capa oculta de diversos pesos y sesgos [8].

3. METODOLOGÌA

A. CREACIÓN DEL COMPENDIO DE IMÁGENES

Se creó una base de datos con 100 imágenes de personas bostezando y 100 imágenes de personas realizando diferentes gestos. Estas imágenes fueron obtenidas de diversas páginas de internet como son Pinterest, Photobucket, We Heart It, y Google; usando palabras clave únicamente en la selección de las imágenes de personas bostezando, mientras que las demás fueron seleccionadas considerando las primeras 30 imágenes encontradas en el buscador. Cabe mencionar que por ser imágenes extraídas de internet, éstas varían tanto en la posición de la persona, edad, iluminación, distancia, nitidez y otros factores como se muestra en la figura .

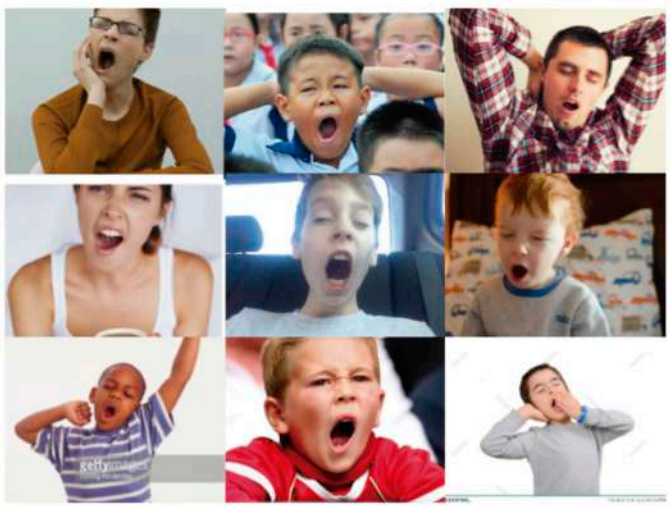

Fig. 5. Ejemplos de las imágenes de personas boztesando obtenidas en la búsqueda.

A.1. CARACTERIZACIÓN Y PRE-PROCESAMIENTO

Se utilizo una red pre-entrenada con el fin de poder detectar rostros de personas famosas, en este caso la misma solo se utilizo para la fase de caracterización de las imágenes, utilizando el Toolbox de Matlab llamado Matconvnet especializado en funciones de aprendizaje profundo. Las imágenes fueron transferidas al código por medio de un vector las imágenes eran procesadas

Fig. 6. Ejemplos de las imágenes de personas boztesando obtenidas en la búsqueda usada como ejemplos negativos.

por medio del método de Convolución de imágenes descritas anteriormente, como resultado se obtuvieron una matriz de los vectores de la caracterización de la imagen, teniendo como resultado una matriz por cada compendio de imágenes de 100 x 4096.

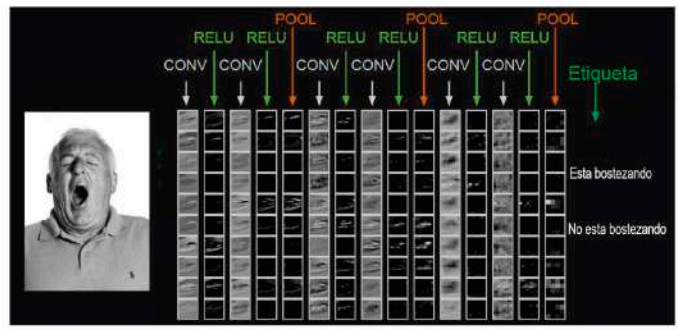

Fig. 7. Estructura de la red de MatConvNet.

A.2. CREACIÓN DE LOS ARCHIVOS .CSV

Para poder hacer la clasificación utilizando la herramienta weka los documentos a utilizar deben de cumplir con ciertas características, como ser de un formato determinado. Dado que la integración de los datos obtenidos por Matlab con Excel son más eficientes se opto por la utilización de documentos con extensión .csv (comma-separated values) uno de los formatos permitidos en weka. En la creación de los documentos en la primera fila se deben de poner los atributos al no ser atributos conocidos se le ingresaron los caracteres "Var" seguido del valor correspondiente a esa celda la cual va desde el 1 hasta el 4096.

Por ultimo en la ultima columna se le agrego a la clase perteneciente ya sea si es un vector adquirido de una imagen de una persona bostezando dándole el valor de "bostezo " en caso contrarió se le agrego el valor de "random" como se muestra en la tabla 6.

4. RESULTADOS

Se utilizó la herramienta weka para la clasificación de los datos utilizando diversos algoritmos que el mismo programa ofrece.

Table 1. Encabezado del archivo .csv como la asignación de la clasificación.

Var1	Var2	Var3	Clase
0	34.957623	9.91843423	Random
9.90911	0	4.957623	Random
5.2481308139	0	6.23435454	Random
9.91843423	0	0	Bostezo
4.783455	4.957623	34.957623	Bostezo
6.23435454	9.90911	0	Bostezo

Los experimentos se realizaron combinando el compendio de imágenes en una proporción de 50/50 para la primera prueba en donde se cuenta con la misma cantidad de imágenes positivas como de imágenes negativas; la segunda se realizó con una proporción de 25/75 en donde se utilizaron más imágenes negativas que positivas; el tercero con una proporción de 75/25 con más imágenes positivas.

En primer lugar se realizó la experimentación con diversos algoritmos, seleccionando aquellos con mayor porcentaje de instancias clasificadas correctamente para una experimentación completa. Así, los mejores algoritmos fueron: Naive Bayes Multidimensional (NaiveBayes), máquinas de soporte vectorial (SMO), árboles aleatorios (RandomForest) y LIBlinear (LibLinear). En la tabla 2 se muestran los resultados obtenidos con un balance de 50-50 del cual se cuenta con el mismo número de imágenes positivas como negativas en donde el algoritmo más destacado fue RandomForest, con una exactitud promedio de 81%.

Table 2. Resultados obtenidos considerando un conjunto balanceado de imágenes 50-50.

50-50 Bostezo con Aleatorio				
Semilla	NaiveBayes	SMO	RandomForest	LibLinear
1	80.102	81.6327	81.6327	80.102
5	79.0816	77.551	81.6327	78.0612
10	79.5918	78.5714	80.6122	78.5714
15	82.1429	79.0816	79.0816	78.0612
20	81.6327	80.6122	82.6531	80.102
PROMEDIO	80.5102	79.48978	81.12246	78.97956

En la tabla 3 se muestran los resultados obtenidos con un balance de 75-25 del cual se cuenta con más imágenes positivas que negativas en donde el algoritmo más destacado fue SMO que supera a los resultados de la tabla 2 en todos sus promedios, con una exactitud de 88%.

En la tabla 4 se muestran los resultados obtenidos con un balance de 25-75 del cual se cuenta con más imágenes negativas que positivas del cual el algoritmo de LibLinear se destacó, pero SMO nuevamente resultados cercanos a los mostrados en la Tabla 2.

5. CONCLUSIONES

En el presente artículo se presentaron los resultados preliminares para identificar bostezos con el objetivo de reconocer a personas cansadas. De acuerdo a los resultados se puede comentar que es más conveniente usar en el entrenamiento un porcentaje de 75% de imágenes positivas, usando el algoritmo de Maquinas de Soporte Vectorial, debido a que obtiene una exactitud de 88.41%

Table 3. Resultados obtenidos considerando un conjunto balanceado de imágenes 75-25.

25-75 Bostezo con Aleatorio				
Semilla	NaiveBayes	SMO	RandomForest	LibLinear
1	77.2358	82.9268	79.6748	82.1138
5	82.1138	82.9268	79.6748	84.5528
10	82.1138	83.7398	79.6748	84.5528
15	80.4878	82.1138	79.6748	81.3008
20	79.6748	82.1138	80.4878	82.9268
PROMEDIO	80.3252	82.7642	79.8374	83.0894

Table 4. Resultados obtenidos considerando un conjunto balanceado de imágenes 25-75.

75-25 Bostezo con Aleatorio				
Algoritmo	NaiveBayes	SMO	RandomForest	LibLinear
1	85.7143	88.0952	79.6748	86.5079
5	88.0952	88.8889	79.6748	86.5079
10	88.0952	87.3016	79.6748	86.5079
15	86.5079	88.8889	79.6748	87.3016
20	85.7143	88.8889	80.4878	84.9268
PROMEDIO	86.82538	88.4127	79.8374	86.35042

REFERENCES

1. F. CEA (2014).
2. Y. Liang, M. L. Reyes, and J. D. Lee, IEEE transactions on intelligent transportation systems **8**, 340 (2007).
3. Y. Zhang and C. Hua, Optik-International Journal for Light and Electron Optics **126**, 4501 (2015).
4. R. Arrabales, "Xataka," urlhttp://neuralnetworksanddeeplearning.com/ (2016).
5. Y. Bengio *et al.*, Foundations and trends® in Machine Learning **2**, 1 (2009).
6. Y. LeCun, Y. Bengio, and G. Hinton, Nature **521**, 436 (2015).
7. M. Nielsen, "Deep learning," urlhttp://neuralnetworksanddeeplearning.com/ (2016).
8. K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, arXiv preprint arXiv:1405.3531 (2014).