

# Student's Attention Monitoring System in Learning Environments based on Artificial Intelligence

Daniel F. Terraza, Mauricio Amaya, Alejandro Piedrahita-Carvajal, Paula A. Rodriguez-Marin, Leonardo Duque-Muñoz, Juan D. Martinez-Vargas

**Abstract**—The students' attention level to the explanation of a given lecture is a factor that might determine the capability of retention and subsequent application of a learned concept. For this reason, students that pay attention are generally more participatory in the learning/teaching process than those who don't, and consequently, they succeed in reaching the competencies proposed in the courses. Hence, it is important to design strategies and tools that help teachers to monitor in a non-invasive way the attention level of the students, allowing them to take actions to modify the dynamics of the lectures when needed. In this work, we introduce a fully automated system to monitor the students' attention based on computer vision algorithms. To this end, we feed a recurrent neural network with one-second sequences generated by facial landmarks. This spatiotemporal analysis of video recordings allows for identifying when a student is attending a given explanation in online educational environments. The system is tested in a database with more than 3000 sequences of students who pay or no attention to online video lectures. Obtained results show that the proposed system is suitable to monitor the students' attention to a particular explanation.

**Index Terms**—Action recognition, LSTM, Monitoring, Students' attention

## I. INTRODUCCIÓN

El término atención, en el área educativa, se refiere a la capacidad de mantener la concentración mientras se ignoran las distracciones, durante un período de tiempo, por ejemplo, mientras se asiste a una conferencia de un tema específico y es considerado como un factor importante para el éxito del aprendizaje [1]. Uno de los principales problemas que enfrenta la educación está relacionado con el bajo desempeño académico que se presenta en todos los niveles, siendo la falta de atención sostenida por parte de los estudiantes, una de las razones principales asociadas con este fenómeno [2]. En los últimos dos años, la transición obligada a la educación virtual causada por la pandemia Covid-19, ha hecho que el desempeño estudiantil baje considerablemente en las instituciones de educación superior, presuntamente por la falta de atención que prestan los estudiantes a las clases online. Esta situación de desatención también desanima a los profesores, generando un círculo vicioso que empeora el proceso de aprendizaje y que no es fácil de romper. Es importante entonces proponer sistemas

de monitoreo que detecten automáticamente si un estudiante esta prestando o no atención a una explicación. Basados en esta información, los docentes podrán tomar acciones para mejorar los contenidos enseñados y así captar la atención de los estudiantes lo que tendrá un impacto significativo en los resultados del aprendizaje [3]–[5].

Una forma de analizar la atención de los estudiantes en el ambiente de aprendizaje (aula de clase, entorno virtual) es a través del monitoreo de su comportamiento. Esta tarea se realiza comúnmente a través de: i) autoinformes, ii) listas de verificación o iii) medidas automáticas. Los autoinformes y las listas de verificación son cuestionarios en los que los estudiantes o terceros reportan indicadores como el nivel de atención, distracción, excitación o aburrimiento percibido en el aula de clase [6]. Sin embargo, la información allí suministrada puede ser subjetiva. Por otro lado, las medidas automáticas se pueden basar en algoritmos de inteligencia artificial que midan, por ejemplo, la atención del estudiante a la clase magistral, o la interacción del estudiante con los demás compañeros [7]. Así, se puede analizar de forma discreta y no invasiva el nivel de atención de los estudiantes analizando su comportamiento [8]–[10].

La tarea de reconocimiento automático de atención o no atención en secuencias de video se puede enmarcar como una tarea de reconocimiento de actividades que es parte fundamental de la visión artificial, con aplicaciones en vigilancia, interacciones hombre-máquina, robótica, entre otras. A diferencia del reconocimiento de objetos en imágenes, para el reconocimiento de actividades se debe modelar de forma conjunta los patrones espaciotemporales contenidos en los videos, lo que dificulta su solución [11]. Utilizando técnicas de aprendizaje automático, esta tarea se ha resuelto de forma general con dos aproximaciones diferentes: i) caracterización manual de los videos, ii) métodos basados en aprendizaje profundo (Deep Learning). Los métodos de caracterización manual se basan en la localización de regiones de interés que codifican de forma local una región de la imagen (cuadro de video). Estas representaciones locales se concatenan a lo largo del tiempo en vectores de características que se utilizan para alimentar un clasificador que resuelve la tarea de reconocimiento [12], [13]. Aunque estos métodos han presentado buenos resultados en tareas específicas, presentan principalmente dos problemas: la localización de las zonas de interés depende del conocimiento previo que se tenga de los videos analizados y requieren segmentos de video con el mismo número de cuadros. Aunque para nuestra tarea el segundo punto no presenta un problema significativo, es poco

Este trabajo fue financiado por el proyecto de investigación con código P20227 de la convocatoria para la formación de banco de elegibles de proyectos de Ciencia, Tecnología, Innovación y Creación para los grupos de Investigación del ITM - 2019.

Laboratorio Máquinas Inteligentes y Reconocimiento de Patrones, Instituto Tecnológico Metropolitano - Medellín, Colombia. Semillero de Investigación en Inteligencia Artificial. e-mail: {danielterrazza212285, mauricioamaya18986, alejandropiedrahita264000}@correo.itm.edu.co, {paularodriguez,leonardoduque, juanmartinez}@itm.edu.co

probable que conozcamos de forma predeterminada en qué parte de cada cuadro de video aparecerá el rostro del estudiante, por lo que la marcación de zonas de interés de forma manual no es una alternativa viable. Por otro lado, los métodos basados en Deep learning por lo general constan de dos etapas: en la primera se obtienen patrones espaciales de los cuadros (frames) del video utilizando redes convolucionales (CNNs) y en la segunda, estos patrones espaciales alimentan redes recurrentes para completar la descripción de las dinámicas espaciotemporales [14]–[16]. Si bien se ha demostrado que esta combinación de arquitecturas (convolucional + recurrente) puede resolver de forma adecuada tareas de reconocimiento de actividades, el entrenamiento de extremo a extremo de este tipo de modelos puede ser bastante costoso computacionalmente y puede requerir una cantidad bastante alta de ejemplos.

## II. PLANTEAMIENTO DEL PROBLEMA

Dado el contexto mencionado anteriormente, con este trabajo queremos saber si es posible desarrollar un sistema automático basado en algoritmos de inteligencia artificial y visión por computador que permita analizar secuencias de video para monitorear la atención de los estudiantes en ambientes educativos. El sistema deberá tener en cuenta la estructura espaciotemporal de las secuencias de video y deberá incluir conocimiento previo de la tarea a desarrollar, por ejemplo, que el estado de atención se puede calcular a partir de diferentes señales visuales como la mirada y el movimiento de la cabeza y las posturas corporales [17]–[19]. En la mirada sobresalen el uso de marcadores que permiten conocer hacia donde el estudiante dirige su atención, el movimiento de cabeza se refiere a la dirección en que se mueve la cabeza, por ejemplo en [2], clasifican al estudiante en activo, transcribiendo, inútil, distraído y en transición, siguiendo su movimiento de cabeza.

Por la situación relacionada con el Covid-19, la metodología estará enfocada en sistemas de educación virtual y al seguimiento del rostro, sin embargo, se diseñará con la premisa de ser fácilmente escalable a ambientes presenciales.

## III. SOLUCIÓN PROPUESTA

Para dar solución al problema planteado, proponemos utilizar una metodología que combina los mejores atributos de la caracterización manual y de los métodos basados en Deep Learning. Por un lado, basados en conocimiento previo, definimos un conjunto de marcadores del rostro de las personas que pueden ser calculados utilizando arquitecturas pre-entrenadas de redes convolucionales como los es Multi-task Cascaded Convolutional Networks (MTCNN). Este proceso reduce la cantidad de parámetros a estimar en el modelo dado que la parte de extracción de patrones espaciales no requiere entrenamiento. Después, estos patrones se entregan a una red recurrente para completar el modelado espaciotemporal de los videos y así resolver por completo la tarea de reconocimiento de atención.

## IV. METODOLOGÍA

En este trabajo, el monitoreo de atención de estudiantes en un ambiente de educación se realiza de la siguiente forma: Primero, en cada uno de los cuadros de video en una ventana de análisis de un segundo (30 cuadros), se detecta el rostro del estudiante y se toman los puntos de referencia (*landmarks*) y el cuadro delimitador (*bounding box*). Con estos datos, se genera una secuencia que se utiliza para alimentar una red recurrente *Long-Short-Term-Memory* LSTM que retorna la clase a la que pertenece cada secuencia de puntos, i.e., atención o no atención. Dada la situación actual por la pandemia Covid-19, el estudio se realizó en ambientes de educación virtual, pero dado su diseño, se puede extender fácilmente a sistemas de educación presencial (aulas de clase). El esquema general del sistema se puede ver en la Fig. 1.

### A. Recopilación de Datos

En el estudio participaron 15 estudiantes de Ingeniería Electrónica e Ingeniería de Sistemas del Instituto Tecnológico Metropolitano (Medellín, Colombia), todos miembros del Semillero en Inteligencia Artificial. El estudio fue aprobado por el Comité de Ética del Instituto Tecnológico Metropolitano - ITM de Medellín - Colombia. Cada estudiante escogió libremente dos videoconferencias online cortas (máximo 5 minutos), una con un tema de interés y otra con un tema cualquiera. Los estudiantes se instruyeron para recortar los videos en segmentos continuos en los que prestaron atención y aquellos en los que no prestaron atención a las videoconferencias. Como resultado, cada estudiante compartió dos carpetas con videos de no más de 3 minutos cada uno, una con la etiqueta atención y una con la etiqueta no atención. Se hizo una inspección visual de los videos recolectados para corroborar que correspondieran a estudiantes atendiendo videoconferencias. Cabe aclarar que no se especificaron poses o conductas predeterminadas que fueran identificadas como atención o no atención y que sesgaran el modelo propuesto. Todos los videos se registraron con las cámaras de computadores portátiles (laptops) a una tasa de adquisición de 30 fps. Como parte del preproceso, se tomaron segmentos de un segundo de cada video para generar la base de datos. Como resultado, se obtuvieron 2340 segmentos de un segundo de la clase *atención* y 1142 pertenecientes a la clase *no atención*. Cabe anotar que los videos tienen diferentes niveles de iluminación, de ángulos de adquisición de la cámara, de distancia entre la cámara y el sujeto, complicando el diseño del modelo de aprendizaje. La Fig. 2 muestra ejemplos de los datos adquiridos.

### B. Reconocimiento y Detección de Rostros

La primera parte del proceso consiste en reconocer y detectar rostros en cada uno de los cuadros de video. Para esto se utilizó la arquitectura conocida como Multi-Task Cascade Convolutional Neural Network (MTCNN). En resumen, MTCNN desarrolla dos tareas principales, la detección de rostros (*bounding boxes*) y la localización de *landmarks* en los rostros detectados. En la primera parte del proceso, las imágenes se pasan por una primera red convolucional que

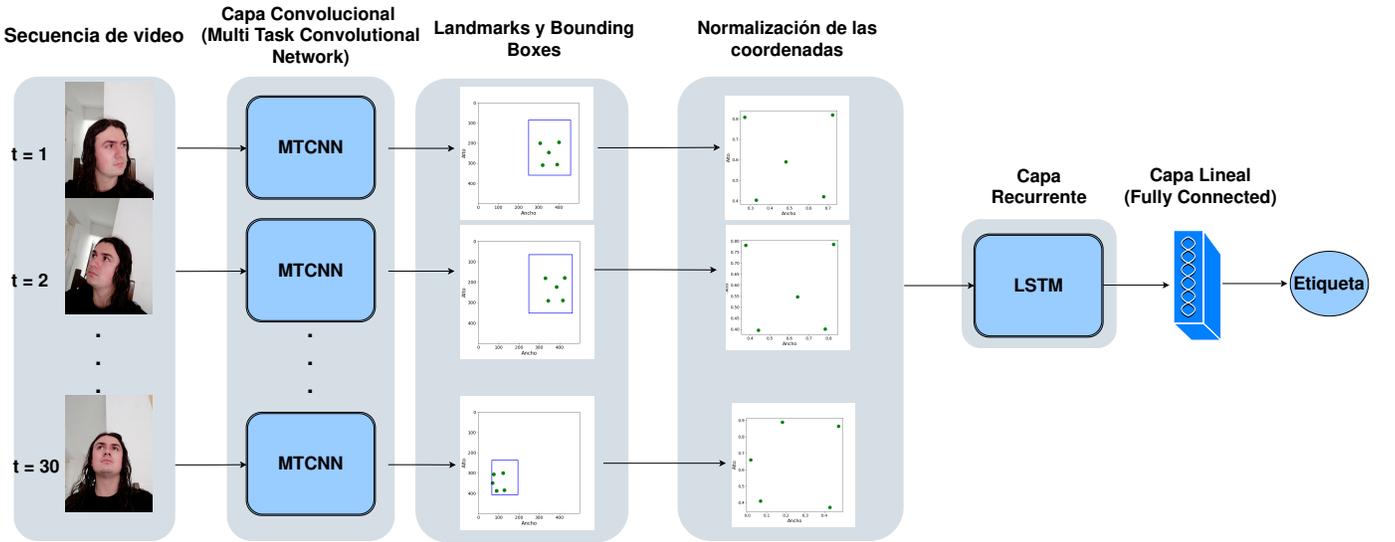


Fig. 1. Esquema general de la metodología propuesta. Cada cuadro de la secuencia de un segundo pasa por el modelo MTCNN para extraer los *bounding boxes* y *landmarks*. Después, esta información se normaliza y se ingresa a una LSTM cuya salida se utiliza para calcular la etiqueta de clase de la secuencia analizada.



Fig. 2. Ejemplos de los datos adquiridos para el monitoreo de atención

entrega las diferentes regiones que son candidatas para tener un rostro. Escogimos este modelo dado que en la base de datos Face Detection Data Set Benchmark - Fddb [20] que tiene ejemplos de rostros similares a los obtenidos en nuestro proceso de adquisición, obtiene una precisión cercana al 95% con aproximadamente 500 falsos positivos y una tasa de acierto del 95% en la tarea de detección de rostros [21].

Después, las regiones obtenidas son procesadas por otra red convolucional que refina las regiones candidatas. En la última etapa, las regiones que aún persisten como posibles rostros son procesadas por una tercera red para finalmente obtener las regiones en las que se detectó un rostro y los *landmarks* de los mismos. Como resultado, además de la posición superior izquierda  $(x_0, y_0)$  e inferior derecha  $(x_1, y_1)$  del *bounding box*, se obtienen las coordenadas  $x, y$  de los siguientes *landmarks*: centro del ojo izquierdo, centro del

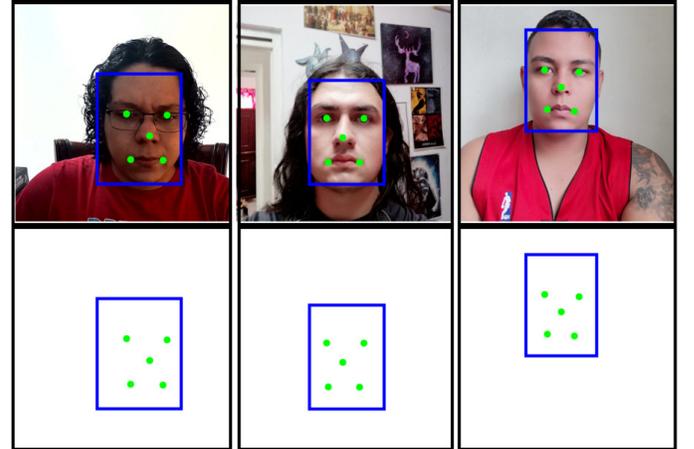


Fig. 3. Ejemplos del resultado del proceso con MTCNN para la detección de rostros. En la parte superior se muestran tres cuadros de video mientras que en el centro y en la parte inferior se muestran los *bounding boxes* y *landmarks* que entrega el modelo MTCNN.

ojo derecho, nariz, comisura derecha de la boca, comisura izquierda de la boca. Las coordenadas en el eje  $x$  (horizontal) de los *landmarks* se almacenan en un vector  $\mathbf{x}_x \in \mathbb{R}^{5 \times 1}$  mientras que las coordenadas del eje  $y$  (vertical) se almacenan en un vector  $\mathbf{x}_y \in \mathbb{R}^{5 \times 1}$ .

Como pre-proceso, se normalizaron los *landmarks* de cada rostro con respecto a su *bounding box* para lograr que la información relevante sea dada por la posición de los *landmarks* con respecto al rostro y no por la ubicación del rostro en la imagen. La normalización se realiza de la siguiente forma:

$$\hat{x}_x^j = \frac{x_x^j - x_0}{x_1 - x_0} \quad (1)$$

$$\hat{x}_y^j = \frac{y_1 - y_0}{y_1 - y_0}, \quad (2)$$

donde  $(x_x^j, x_y^j) \in \mathbb{Z}^+$  son las coordenadas  $x$  y  $y$  del  $j$ -ésimo

*landmark*,  $j = 1, \dots, 5$ , y  $(\hat{x}_x^j, \hat{x}_y^j) \in [0, 1]$  son las coordenadas normalizadas. Concatenando todas las coordenadas normalizadas, se crea un vector  $\hat{\mathbf{x}} \in (0, 1)^{10 \times 1}$  de características para cada cuadro. Cabe anotar que dado que los datos se tomaron con las cámaras de computadores portátiles, se restringió el número de rostros en cada cuadro a 1. Sin embargo, si se utilizaran diferentes métodos de adquisición, se podrían procesar tantos rostros como aparezcan en cada cuadro. Este proceso se realiza para cada uno de los 30 cuadros de cada segmento, por lo que los *landmarks* normalizados podrían verse como una secuencia de vectores  $\hat{\mathbf{x}}_t, t = 1, \dots, 30$ .

### C. Redes LSTM y BiLSTM

Las redes *Long Short Term Memory* - LSTM son un caso particular de redes recurrentes (RNNs) que tienen la habilidad de aprender dependencias a largo plazo presentes en los datos, almacenando la información útil en celdas de memoria [22]. Cada unidad LSTM se compone de una celda de memoria y tres compuertas: entrada, salida y olvido. Con esta estructura, la LSTM controla cuál información debe olvidar y cuál debe recordar a lo largo del tiempo. De forma detallada, la compuerta de entrada (input)  $\mathbf{i}_t \in [0, 1]^{p \times 1}$  junto con una compuerta intermedia  $\mathbf{c}_t^* \in [-1, 1]^{p \times 1}$  controlan la información que se almacena en la memoria  $\mathbf{c}_t \in [-1, 1]^{p \times 1}$  en el instante de tiempo  $t$ . La compuerta de olvido  $\mathbf{f}_t \in [0, 1]^{p \times 1}$  controla la información que se debe eliminar o se debe mantener en la memoria proveniente del instante de tiempo  $t-1$ . Por último, la compuerta de salida  $\mathbf{o}_t \in [0, 1]^{p \times 1}$  controla qué información de la celda de memoria se debe utilizar para la salida de la LSTM en el instante de tiempo  $t$ . Resumiendo, la Eq. 3 describe las operaciones realizadas por la LSTM:

$$\mathbf{i}_t = \sigma(\mathbf{U}_i \hat{\mathbf{x}}_t + \mathbf{W}_i \mathbf{h}_{t-1} + \mathbf{b}_i) \quad (3a)$$

$$\mathbf{f}_t = \sigma(\mathbf{U}_f \hat{\mathbf{x}}_t + \mathbf{W}_f \mathbf{h}_{t-1} + \mathbf{b}_f) \quad (3b)$$

$$\mathbf{c}_t^* = \tanh(\mathbf{U}_c \hat{\mathbf{x}}_t + \mathbf{W}_c \mathbf{h}_{t-1} + \mathbf{b}_c) \quad (3c)$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \mathbf{c}_t^* \quad (3d)$$

$$\mathbf{o}_t = \sigma(\mathbf{U}_o \hat{\mathbf{x}}_t + \mathbf{W}_o \mathbf{h}_{t-1} + \mathbf{b}_o), \quad (3e)$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t). \quad (3f)$$

donde  $\hat{\mathbf{x}}_t$  son los *landmarks* normalizados de un segmento en el instante de tiempo  $t$ ,  $\mathbf{h}_t \in [-1, 1]^{p \times 1}$  es el estado oculto de la LSTM,  $\mathbf{W}_* \in \mathbb{R}^{p \times p}$ ,  $\mathbf{U}_* \in \mathbb{R}^{p \times 10}$  y  $\mathbf{b}_* \in \mathbb{R}^{p \times 1}$  son los parámetros de la red,  $\sigma$  es la función sigmoide, el operador  $\odot$  denota la multiplicación punto a punto, y  $p \in \mathbb{Z}^+$  es la dimensión del estado oculto y la memoria de la LSTM. La Fig.4 describe el proceso de la red LSTM. Una vez toda la secuencia de *landmarks* pasa por la LSTM, la probabilidad de que cada segmento pertenezca a cada una de las clases (*atención, no-atención*) se calcula de la siguiente forma:

$$\mathbf{y} = \text{softmax}(\mathbf{W}_y \mathbf{h}_{30} + \mathbf{b}_y), \quad (4)$$

donde  $\mathbf{y} \in [0, 1]^{2 \times 1}$  son las probabilidades de que la muestra analizada pertenezca a cada una de las 2 clases ( $\sum_{k=1}^2 y_i^k = 1$ ),  $\mathbf{W}_y \in \mathbb{R}^{2 \times p}$  y  $\mathbf{b}_y \in \mathbb{R}^{2 \times 1}$  son parámetros de la red y  $\mathbf{h}_{30} \in [-1, 1]^{p \times 1}$  es el último estado oculto de la LSTM.

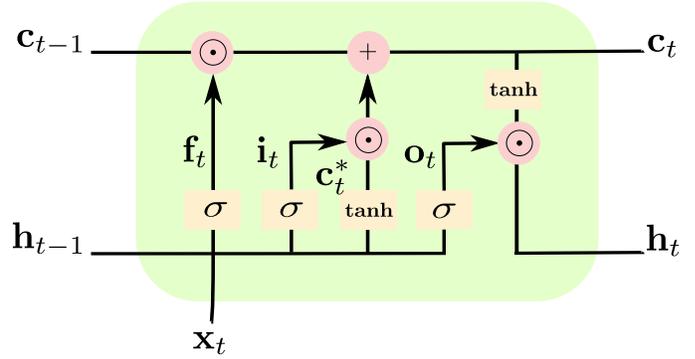


Fig. 4. Proceso de una unidad de red LSTM.

Las redes LSTM pueden ver afectado su rendimiento dado que consideran únicamente el flujo de información en un sentido. Este problema se puede solucionar parcialmente si se considera que la información de la secuencia en ambas direcciones puede contener información importante. Consecuentemente, en [23] se propusieron las redes LSTM bi-direccionales (Bi-LSTM) que ingresa la secuencia desde el inicio hasta el final ( $t = 1, \dots, 30$  en nuestro caso) y desde el final hasta el principio ( $t = 30, \dots, 1$ ). De esta forma se analiza información del pasado y del futuro de un instante de tiempo en particular. Para esto, la secuencia  $\mathbf{x}_f = \{\mathbf{x}_1, \dots, \mathbf{x}_{30}\}$  se pasa por una LSTM como la descrita en la Fig. 4, y la secuencia  $\mathbf{x}_b = \{\mathbf{x}_{30}, \dots, \mathbf{x}_1\}$  se pasa por otra LSTM. Al final, los estados ocultos de cada una de las redes se concatenan para calcular la salida descrita en la Eq. 4.

### D. Diseño del Experimento y Sintonización de Hiperparámetros

Los datos se dividieron en conjuntos de entrenamiento (60%), validación (20%) y prueba (20%). Dado que todos los *landmarks* normalizados se encuentran en el rango entre 0 y 1, no se realizó ningún proceso adicional de estandarización.

En nuestra implementación, utilizamos una LSTM (o Bi-LSTM) que toma como entrada las  $K = 10$  características correspondientes a las coordenadas normalizadas de cada uno de los *landmarks* que retorna el modelo MTCNN en cada instante de tiempo (cada cuadro del video). El tamaño de los estados ocultos toma los valores  $p = [20, 40, 60, 80, 100]$ . La salida de la red LSTM tiene una activación ReLU. Después, en la capa de salida, el modelo tiene dos neuronas ocultas con activación softmax para estimar la probabilidad de que cada secuencia de video pertenezca a cada una de las clases. Para evitar el sobreajuste, se utiliza dropout = 0.2 en las conexiones entre la última iteración de los estados ocultos de la LSTM ( $\mathbf{h}_{30}$ ) y las dos neuronas de la capa de salida. En el conjunto de prueba se utiliza únicamente el valor de  $p$  que maximiza la tasa de acierto en el conjunto de validación. Para comparar, la metodología se implementa con una red recurrente (vanilla-RNN) en sus arquitecturas normal y bi-direccional (BiRNN), entrenada con los mismos parámetros de la LSTM.

Se utilizó Adam como algoritmo de optimización con una tasa de aprendizaje de 0.01 [24]. El tamaño del batch se ajustó en 16, se utilizó *cross-entropy* como función de costo,

TABLE I

RESULTADOS DEL ENTRENAMIENTO PARA DIFERENTES VALORES DE  $p$  DE LA RED NEURONAL RECURRENTE EN SUS DOS CONFIGURACIONES RNN Y BiRNN

$p$	RNN		BiRNN	
	Acc Ent (%)	Acc Val (%)	Acc Ent (%)	Acc Val (%)
20	76.43	74.46	92.62	93.40
40	76.78	77.34	90.28	93.04
60	81.23	79.50	90.28	92.90
80	81.45	77.70	92.91	92.81
100	82.44	78.60	92.55	88.49

TABLE II

RESULTADOS DEL ENTRENAMIENTO PARA DIFERENTES VALORES DE  $p$  DE LA RED LSTM EN SUS DOS CONFIGURACIONES LSTM Y BiLSTM

$p$	LSTM		BiLSTM	
	Acc Ent (%)	Acc Val (%)	Acc Ent (%)	Acc Val (%)
20	97.04	90.65	95.15	90.65
40	95.87	94.87	95.82	92.45
60	96.77	95.14	94.34	91.91
80	93.70	94.78	95.87	91.91
100	96.09	92.45	93.76	93.71

y se entrenó durante 100 épocas. Todos los modelos se implementaron en Google Colaboratory utilizando la librería Pytorch [25].

## V. RESULTADOS

En esta sección evaluamos el rendimiento de todos los modelos utilizados: RNN, BiRNN, LSTM, BiLSTM. Para escoger el valor óptimo de  $p$  (tamaño de los estados ocultos de las redes recurrentes), se utilizó la tasa de acierto de clasificación (*Accuracy*) en el conjunto de validación, como se muestra en las Tablas I y II. Se puede observar que en general, las dos configuraciones de *LSTM* obtienen los mejores resultados. Adicionalmente, para la RNN, la diferencia entre la arquitectura normal y la arquitectura bi-direccional es aproximadamente de 20 puntos mientras que para las dos configuraciones de LSTM la diferencia no es significativa.

Considerando el tamaño de los estados ocultos, se puede observar que no hay una diferencia considerable entre los valores utilizados, y tampoco existe un patrón que indique que entre más grande  $p$  mejor codificada la información de las secuencias de video en los estados ocultos de la red. Por estas razones, se escogió un  $p$  diferente para cada modelo: RNN  $p = 100$ , BiRNN  $p = 80$ , LSTM  $p = 60$  y BiLSTM  $p = 100$ .

Con estos valores de  $p$  procedimos a calcular tanto las matrices de confusión (Fig. 5) como la tasa de acierto de clasificación (Tabla III) en el conjunto de prueba para cada uno de los modelos. Con respecto a la arquitectura RNN, se puede observar que los mejores resultados se obtienen con la configuración bidireccional. Como este tipo de redes no poseen la habilidad de mantener información relevante a largo plazo, es muy probable que tener información en uno de los estados ocultos tanto del pasado como del futuro ayude a dar un contexto general del video, aumentando la capacidad de la red de clasificar correctamente las secuencias. El aumento en este caso de casi 12 puntos en la tasa de acierto es considerable.

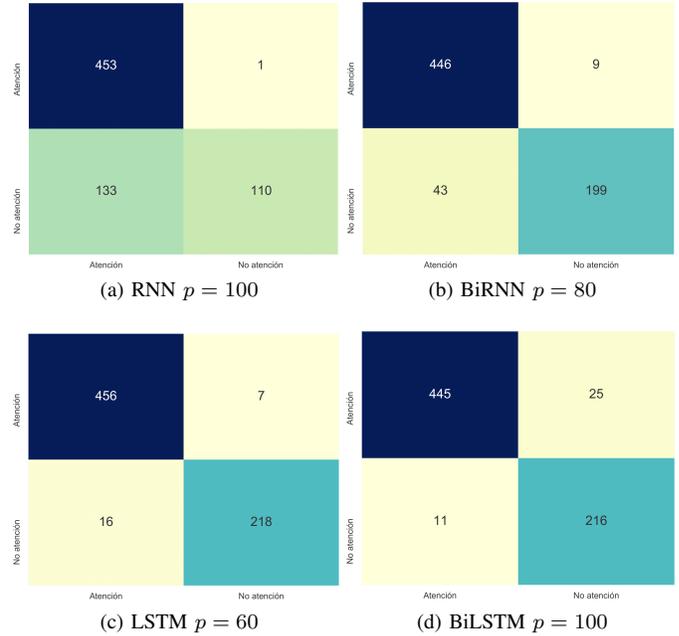


Fig. 5. Matrices de confusión calculadas en el conjunto de prueba de cada uno de los modelos con su valor óptimo de  $p$ .

TABLE III

RESULTADOS EN EL CONJUNTO DE PRUEBA CON EL VALOR ÓPTIMO DE  $p$  PARA CADA UNO DE LOS MODELOS

Modelo	$p$	Acc prueba
RNN	100	80.77%
BiRNN	80	92.73%
LSTM	60	96.70%
BiLSTM	100	94.84%

Adicionalmente, se puede observar que en general, las dos configuraciones de RNN tienden a confundir secuencias de no atención con secuencias de atención, como se puede observar en la posición (2,1) de las Figs.5a y 5b.

Con respecto a la arquitectura LSTM, se puede observar que la diferencia entre su versión normal y su versión bidireccional no es tan amplia como en el caso de la RNN. En efecto, para este tipo de redes, los mejores resultados se obtienen con la versión normal. Dado que este tipo de redes tienen la capacidad de conservar la información relevante de largo plazo, y dado que las secuencias contienen únicamente 30 cuadros, es muy probable que la información del futuro que se incluye en la BiLSTM, ya esté contenida en los estados ocultos de la red normal. Se puede observar que la LSTM continúa etiquetando mal segmentos de no atención (ver posición (2,1) de Fig. 5c), mientras que la BiLSTM comete más errores etiquetando secuencias de atención como secuencias de no atención (posición (1,2) de Fig. 5d).

De forma general, de acuerdo con la Tabla III, el mejor modelo se obtiene utilizando la arquitectura LSTM. Esto confirma nuestra hipótesis de que un correcto monitoreo de atención en un ambiente educativo requiere un análisis que sea capaz de tener en cuenta la información contenida en una secuencia de video.

Finalmente, con el modelo basado en LSTM, hicimos

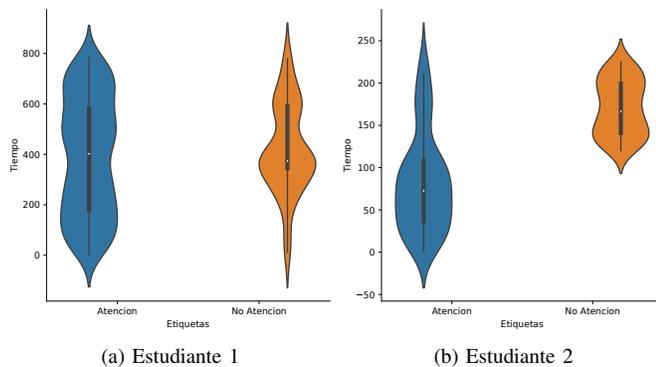


Fig. 6. Informe del monitoreo de atención de un estudiante. En azul se ve el tiempo en el que el estudiante atendió la conferencia mientras que en naranja se muestra el tiempo en el que se distrajo.

un último experimento que consiste en analizar el comportamiento de dos estudiantes ajenos a la creación de la base de datos, para generar un reporte del nivel de atención que prestaron los estudiantes. Para esto, los estudiantes, sin conocimiento de que iba a ser monitoreados, atendieron una conferencia corta de alrededor de 5 minutos. Los resultados obtenidos se pueden ver en la Fig. 6. Se puede observar que el primer estudiante estuvo atento la mayor parte del tiempo. Sin embargo, el segundo estudiante fue perdiendo interés a medida que transcurría la conferencia. Este tipo de monitoreos pueden ser útiles para que tanto el docente como la institución tomen decisiones pertinentes o se acerquen a los estudiantes para encontrar el motivo de la pérdida de atención.

## VI. CONCLUSIONES

En este trabajo, propusimos un sistema de monitoreo de atención de estudiantes basado en algoritmos de visión por computador e inteligencia artificial. El sistema tiene como objetivo generar alertas que indiquen si un estudiante está perdiendo interés en sus estudios. De esta forma el docente o la institución pueden tomar medidas pertinentes orientadas a mejorar su metodología de enseñanza y así recuperar la atención del estudiante. El sistema se basa en la adquisición de video de los estudiantes durante una conferencia. Dicho video es segmentado cada segundo, cada secuencia es caracterizada y procesada, y al final se le asigna una etiqueta de atención o no atención. Para la caracterización de cada secuencia se utilizó el modelo MTCNN. En la etapa de clasificación, dada la estructura temporal de las secuencias, utilizamos redes recurrentes con arquitectura LSTM, las cuales tienen la capacidad de preservar información relevante a lo largo de la secuencia.

Los resultados de clasificación en la etapa de prueba muestran que con una configuración simple de LSTM se pueden obtener tasas de clasificación de entre 90.65 a 95.14%, incluso mejorando a su configuración BiLSTM.

Por último, con el mejor modelo se realizó un seguimiento de dos estudiantes mientras estos atendían a una conferencia corta. Se pudo evidenciar que el sistema identificó aquellos instantes en los cuales los estudiantes perdieron la atención. Este resultado se podría utilizar para retroalimentar a los

docentes y a la institución educativa para que se tomen medidas que incrementen la motivación de los estudiantes.

Como trabajo futuro se plantea extender el modelo para analizar videos de clases presenciales donde aparezca más de una persona, lo que plantea como retos adicionales principalmente: i) hacer seguimiento de cada persona en el video de forma automática, ii) determinar, además de los marcadores faciales, qué otro tipo de comportamiento puede estar relacionado con la atención, como por ejemplo, la pose, y iii) ubicación de la cámara dentro del salón de clase. Adicionalmente, se plantea la posibilidad de estimar a qué tipo de materiales o recursos educativos se les presta más atención, para proponerle a los docentes incluirlos dentro del desarrollo de sus metodologías.

Dado que el modelo de detección de rostros basado en MTCNN no puede reconocer la posición de ojos y boca (abiertos o cerrados), se propone explorar otro tipo de modelos como el propuesto en [26] que genera más *landmarks* que pueden ayudar a reconocer otras actitudes de los estudiantes cuando atienden conferencias (presenciales o virtuales).

## AGRADECIMIENTOS

Los autores quieren agradecer al programa de Jóvenes Investigadores e Innovadores 2020 del Instituto Tecnológico Metropolitano - ITM y al Semillero de Investigación en Inteligencia Artificial por la ayuda para recolectar la base de datos.

## REFERENCES

- [1] E. F. Risko, N. Anderson, A. Sarwal, M. Engelhardt, and A. Kingstone, "Everyday attention: Variation in mind wandering and memory in a lecture," *Applied Cognitive Psychology*, vol. 26, no. 2, pp. 234–242, 2012.
- [2] D. Dinesh, A. N. S., and K. Bijlani, "Student analytics for productive teaching/learning," in *2016 International Conference on Information Science (ICIS)*, 2016, pp. 97–102.
- [3] J. B. Heppen and S. B. Therriault, "Developing early warning systems to identify potential high school dropouts. issue brief," *National High School Center*, 2008.
- [4] C. Márquez-Vera, A. Cano, C. Romero, A. Y. M. Noaman, H. Mousa Fardoun, and S. Ventura, "Early dropout prediction using data mining: a case study with high school students," *Expert Systems*, vol. 33, no. 1, pp. 107–124, 2016.
- [5] C.-M. Chen, J.-Y. Wang, and C.-M. Yu, "Assessing the attention levels of students by using a novel attention aware system based on brainwave signals," *British Journal of Educational Technology*, vol. 48, no. 2, pp. 348–369, 2017.
- [6] S. K. D'Mello, S. D. Craig, J. Sullins, and A. C. Graesser, "Predicting affective states expressed through an emote-aloud procedure from autotutor's mixed-initiative dialogue," *International Journal of Artificial Intelligence in Education*, vol. 16, no. 1, pp. 3–28, 2006.
- [7] J. Whitehill, Z. Serpell, Y.-C. Lin, A. Foster, and J. R. Movellan, "The faces of engagement: Automatic recognition of student engagement from facial expressions," *IEEE Transactions on Affective Computing*, vol. 5, no. 1, pp. 86–98, 2014.
- [8] S. K. D'Mello, S. D. Craig, and A. C. Graesser, "Multimethod assessment of affective experience and expression during deep learning," *International Journal of Learning Technology*, vol. 4, no. 3-4, pp. 165–187, 2009.
- [9] S. K. D'Mello and A. Graesser, "Multimodal semi-automated affect detection from conversational cues, gross body language, and facial features," *User Modeling and User-Adapted Interaction*, vol. 20, no. 2, pp. 147–187, 2010.
- [10] D. Canedo, A. Trifan, and A. J. Neves, "Monitoring students' attention in a classroom through computer vision," in *International Conference on Practical Applications of Agents and Multi-Agent Systems*. Springer, 2018, pp. 371–378.

- [11] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2014, pp. 1725–1732.
- [12] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, "Action recognition by dense trajectories," in *CVPR 2011*, 2011, pp. 3169–3176.
- [13] X. Wang, D. Chen, T. Yang, B. Hu, and J. Zhang, "Action recognition based on object tracking and dense trajectories," in *2016 IEEE International Conference on Automatica (ICA-ACCA)*. IEEE, 2016, pp. 1–5.
- [14] Z. Zhang, Z. Lv, C. Gan, and Q. Zhu, "Human action recognition using convolutional lstm and fully-connected lstm with different attentions," *Neurocomputing*, vol. 410, pp. 304–316, 2020.
- [15] J. Su, W. Byeon, J. Kossaihi, F. Huang, J. Kautz, and A. Anandkumar, "Convolutional tensor-train lstm for spatio-temporal learning," *arXiv preprint arXiv:2002.09131*, 2020.
- [16] L. Wei, S. Zhao, O. F. Bourahla, X. Li, F. Wu, Y. Zhuang, J. Han, and M. Xu, "End-to-end video saliency detection via a deep contextual spatiotemporal network," *IEEE Transactions on Neural Networks and Learning Systems*, 2020.
- [17] J. Zaletelj and A. Košir, "Predicting students' attention in the classroom from kinect facial and body features," *EURASIP journal on image and video processing*, vol. 2017, no. 1, pp. 1–12, 2017.
- [18] M. S. Young, S. Robinson, and P. Alberts, "Students pay attention! combating the vigilance decrement to improve learning during lectures," *Active Learning in Higher Education*, vol. 10, no. 1, pp. 41–55, 2009.
- [19] A. S. Won, J. N. Bailenson, and J. H. Janssen, "Automatic detection of nonverbal behavior predicts learning in dyadic interactions," *IEEE Transactions on Affective Computing*, vol. 5, no. 2, pp. 112–125, 2014.
- [20] V. Jain and E. Learned-Miller, "Fddb: A benchmark for face detection in unconstrained settings," UMass Amherst technical report, Tech. Rep., 2010.
- [21] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499–1503, 2016.
- [22] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [23] Z. Huang, W. Xu, and K. Yu, "Bidirectional lstm-crf models for sequence tagging," *arXiv preprint arXiv:1508.01991*, 2015.
- [24] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
- [25] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 8024–8035.
- [26] A. Bulat and G. Tzimiropoulos, "How far are we from solving the 2d & 3d face alignment problem?(and a dataset of 230,000 3d facial landmarks)," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1021–1030.



**Daniel F. Terraza-Arciniegas** Tecnólogo en Electrónica del Instituto Tecnológico Metropolitano (2017), Técnico en sistemas del SENA (2013), estudiante de Ingeniería Electrónica del Instituto Tecnológico Metropolitano. Entre sus áreas de interés se encuentran principalmente los sistemas de visión e inteligencia artificial enfocados al análisis de datos.



**Mauricio Amaya** Tecnólogo en Electrónica (2017) y estudiante de Ingeniería Electrónica del Instituto Tecnológico Metropolitano. Entre sus áreas de interés se encuentran principalmente los sistemas de visión e inteligencia artificial enfocados al análisis de emociones.



**Alejandro Piedrahita-Carvajal** Estudiante de Tecnología en Sistemas de Información del Instituto Tecnológico Metropolitano. Entre sus áreas de interés se encuentra principalmente el desarrollo de aplicaciones web y la inteligencia artificial, especialmente en los sistemas de reconocimiento facial para entornos educativos y de aprendizaje.



**Paula A. Rodríguez-Marin** Administradora de Sistemas Informáticos de la Universidad Nacional de Colombia Sede Manizales (2008), Magíster en Ingeniería de Sistemas de la Universidad Nacional de Colombia Sede Medellín (2013), Doctora en Ingeniería – Ingeniería de Sistemas de la Universidad Nacional de Colombia Sede Medellín (2018). Docente ocasional del ITM desde enero del 2018 del departamento de Sistemas de Información. Entre sus áreas de interés se encuentra principalmente la informática en la educación aplicando técnicas de inteligencia artificial, especialmente los sistemas de recomendación para el apoyo a los procesos de enseñanza – aprendizaje.



**Leonardo Duque-Muñoz** Ingeniero Electrónico de la Universidad Nacional de Colombia sede Manizales (2009), Magíster en Ingeniería de la misma universidad (2012, tesis meritosa) y Doctor en Ingeniería Electrónica (2019, magna cum laude) de la Universidad de Antioquia. Docente Asistente del Programa de Ingeniería Mecatrónica del Instituto Tecnológico Metropolitano, vinculado desde enero del año 2018. Sus principales intereses de investigación son el procesamiento de señales cerebrales, (electroencefalografía y magnetoencefalografía) para procesamiento emocional y reconocimiento de patologías. Implementación de algoritmos y metodologías para procesamiento de grandes volúmenes de datos basado en técnicas de aprendizaje de máquina e inteligencia artificial.



**Juan D. Martínez-Vargas** Recibió los títulos de Ingeniero Electrónico (2009), Magíster en Ingeniería (2011, summa cum laude) y Doctor en Ingeniería (2017, summa cum laude) de la Universidad Nacional de Colombia, sede Manizales. Sus intereses de investigación incluyen la integración de áreas como estadística Bayesiana, aprendizaje de máquina e inteligencia artificial para apoyar el procesamiento de señales e imágenes. Actualmente se desempeña como docente de tiempo completo e investigador del Instituto Tecnológico Metropolitano - ITM, y es el líder del Grupo de Investigación Máquinas Inteligentes y Reconocimiento de Patrones (MIRP).