

Sistema de visión para detectar y alarmar el estado de cansancio de un conductor.

Vision system to detect and alarm the state of fatigue of a car driver.

Néstor Antonio Morales Navarro.
Instituto Tecnológico de Tuxtla Gutiérrez, Universidad Valle de México.
nmorales@ittg.edu.mx.

Gilberto Méndez Sántiz (2). Estudiante, I.T. de Tuxtla Gutiérrez. gilberto.mendez.santiz@gmail.com.

Eliver Castillejos Lara (3). Estudiante, I.T. de Tuxtla Gutiérrez. eliverlara@gmail.com.

Oswaldo Brindis Velázquez (4). I. T. de Tuxtla Gutiérrez, Universidad Valle de México. obrindis@ittg.edu.mx.

Salomón Velasco Bermúdez (5). I. T. de Tuxtla Gutiérrez. svelasco@ittg.edu.mx.

Artículo recibido en octubre 03, 2017; aceptado en diciembre 19, 2017.

Resumen.

En este trabajo se presenta el desarrollo de un dispositivo, el cual brinda una alternativa de ayuda para evitar accidentes automovilísticos que son un problema de hoy en día, ya sea a causa de distracciones o por otros factores que afectan a los conductores. Utilizando tecnologías como visión artificial, la cual permitirá a partir del uso de diversas técnicas, el procesamiento y análisis de los datos recabados a través de un dispositivo conformado por una tarjeta de control Raspberry pi. El resultado de este análisis permite al dispositivo tomar la decisión de activar o no una alerta auditiva en caso de que el conductor del automóvil este presentando cuadros de cansancio o sueño, las pruebas realizadas arrojan un alto porcentaje de efectividad, sin embargo, existen factores ambientales como la iluminación que en ocasiones afecta el correcto funcionamiento del algoritmo.

Palabras clave: Accidentes automovilísticos, visión artificial, Raspberry.

Abstract.

In this paper, the development of a device is presented, which provides an alternative of help to avoid automobile accidents that are a problem of today, whether due to distractions or other factors that affect drivers. Using technologies like artificial vision which will allow us, from the use of diverse techniques, the processing and analysis of the data collected through a device conformed by a control card Raspberry Pi, the result of this analysis allows the device to take the decision to activate or not an audible warning in case the driver of the car is presenting pictures of fatigue or sleep, the tests carried out show a high percentage of effectiveness, however there are environmental factors such as illumination, which occasionally affect the correct functioning of the algorithm.

Keywords: Car accidents, artificial vision, Raspberry.

1. Introducción.

La necesidad de transportarse de un lugar a otro es una actividad que se ha convertido en algo indispensable para llevar a cabo diversas actividades cotidianas, es algo rutinario y que absolutamente todos realizan, existen diferentes medios para poder realizar esta actividad, pero sin duda alguna la manera más utilizada es a través de medios terrestres.

La mayoría de las personas a lo largo del día utilizan ya sea un automóvil personal o el transporte público para llegar a sus respectivos destinos, pero al hacer uso de estos transportes no garantiza llegar sanos y salvos a nuestros destinos.

Los accidentes están a la orden del día, donde las personas necesitan llegar a sus lugares de trabajo de manera inmediata y puntual, por lo tanto, en muchas ocasiones eso provoca un estrés de quien conduce un vehículo, lo que genera que por alguna distracción del conductor se causen accidentes que provocan afectaciones no sólo en lo personal sino también a terceras personas, además de ello se tienen pérdidas monetarias hacia el automóvil personal o al transporte público, y no sólo es eso, en algunas ocasiones hay afectaciones a la salud de las personas porque se llegan a fracturar y en algunas ocasiones se tiene la pérdida de extremidades o genera discapacidades motrices.

En el año del 2014 por datos arrojados en el INEGI en Chiapas sucedieron 2106 accidentes automovilísticos de los cuales 2090 sucedieron por culpa del conductor, en ese mismo año de los accidentes automovilísticos mencionados, en 2090 hubo pérdidas humanas, INEGI (2014). De acuerdo a los registros el horario con mayor número de víctimas mortales se presenta entre las 18:00 y las 20:59 horas, con el 18.7%; en segundo lugar se obtuvo que el rango es de 21:00 a 23:59 horas con 16.2%. Para el año 2015 el número de accidentes disminuyó ya que fueron 1,960 accidentes automovilísticos, de los cuales 1,956 fueron por culpa del conductor del vehículo, de estos accidentes en 93 se tuvieron pérdidas humanas, INEGI (2015).

Principales factores que causan accidentes carreteros.

Factor Humano: Los factores humanos son la causa del mayor porcentaje de accidentes de tránsito, debido a las principales causas siguientes:

- Conducir bajo los efectos del alcohol, medicinas y estupefacientes.
- Realizar maniobras imprudentes y de omisión por parte del conductor, por ejemplo; no respetar los señalamientos viales.
- Conducir a exceso de velocidad (produciendo vuelcos, salida del automóvil de la carretera, derrapes).
- Salud física del conductor (ceguera, daltonismo, sordera)
- Conducir con fatiga, cansancio o con sueño.

De los accidentes automovilísticos registrados el 35% de ellos fue porque el conductor presentaba con fatiga, cansancio o con sueño, INEGI (2014).

2. Métodos.

Para este proyecto se desarrolló un sistema capaz de alertar al conductor de un automóvil cuando se detecte distracción o estado de cansancio mediante un dispositivo aplicando visión artificial en tiempo real. En la figura 1 se puede observar el funcionamiento general del sistema.

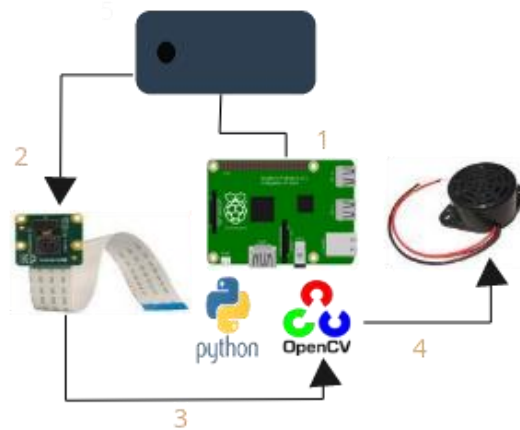


Figura 1. Componentes del dispositivo..

1. Raspberry es una tarjeta de control y el principal componente del dispositivo, utiliza un sistema basado en linux y puede realizar diversas tareas como una computadora normal, se utilizará para poder realizar la actividad de tomar el video con un módulo de cámara especial para esta tarjeta.
2. Se activa el módulo de cámara de la tarjeta raspberry para llevar a cabo el monitoreo en tiempo real del rostro del conductor.
3. Mediante la biblioteca OpenCV manejada a través del lenguaje de programación python se realiza el procesamiento del video captado por la cámara para determinar el estado del conductor.
4. Si se detecta un comportamiento anormal en el pestañeo del conductor se hace activar una alarma sonora.

Todos estos componentes deberán estar ensamblados dentro de una carcasa la cual se pueden apreciar sus dimensiones en la figura 2.

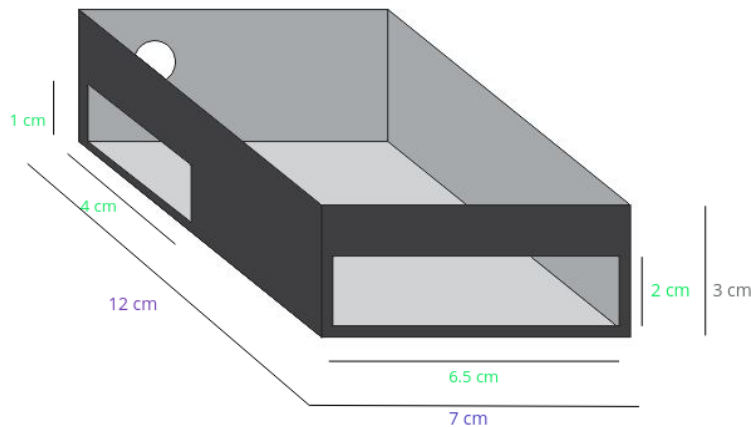


Figura 2. Dimensiones de la carcasa del dispositivo.

Tecnologías involucradas.

Visión artificial: Se puede definir la “Visión Artificial” como un campo de la “Inteligencia Artificial” que, mediante la utilización de las técnicas adecuadas, permite la obtención, procesamiento y análisis de cualquier tipo de información especial obtenida a través de imágenes digitales. La visión artificial la componen un conjunto de procesos destinados a realizar el análisis de imágenes. Estos procesos son: captación de imágenes, memorización de la información, procesado e interpretación de los resultados, Cabeza (2010).

Python: Es un lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis que favorezca un código legible. Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional.

Python es un lenguaje de programación multiparadigma. Esto significa que más que forzar a los programadores a adoptar un estilo particular de programación, permite varios estilos: programación orientada a objetos, programación imperativa y programación funcional, otros paradigmas están soportados mediante el uso de extensiones.

Una característica importante de Python es la resolución dinámica de nombres; es decir, lo que enlaza un método y un nombre de variable durante la ejecución del programa (también llamado enlace dinámico de métodos).

Otro objetivo del diseño del lenguaje es la facilidad de extensión. Se pueden escribir nuevos módulos fácilmente en C o C++. Python puede incluirse en aplicaciones que necesitan una interfaz programable.

Aunque la programación en Python podría considerarse en algunas situaciones hostil a la programación funcional tradicional del Lisp, existen bastantes analogías entre Python y los lenguajes minimalistas de la familia Lisp como puede ser Scheme, Martelli (2006).

OpenCV: es una biblioteca libre de visión artificial originalmente desarrollada por Intel. Desde que apareció su primera versión alfa en el mes de enero de 1999, se ha utilizado en infinidad de aplicaciones. Desde sistemas de seguridad con detección de movimiento, hasta aplicaciones de control de procesos donde se requiere reconocimiento de objetos. Esto se debe a que su publicación se da bajo licencia BSD, que permite que sea usada libremente para propósitos comerciales y de investigación con las condiciones en ella expresadas.

Open CV es multiplataforma, existiendo versiones para GNU/Linux, Mac OS X y Windows. Contiene más de 500 funciones que abarcan una gran gama de áreas en el proceso de visión, como reconocimiento de objetos (reconocimiento facial), calibración de cámaras, visión estéreo y visión robótica.

El proyecto pretende proporcionar un entorno de desarrollo fácil de utilizar y altamente eficiente. Esto se ha logrado realizando su programación en código C, C++ y python optimizados, aprovechando además las capacidades que proveen los procesadores multinúcleo. OpenCV puede además utilizar el sistema de primitivas de rendimiento integradas de Intel, un conjunto de rutinas de bajo nivel específicas para procesadores Intel.

Clasificador de Cascada: Mejor conocido como algoritmo o clasificador Haar, fue el primer framework de detección de objetos propuesto por Paul Viola y Michael Jones en 2001 que permitía el análisis de imágenes en tiempo real, haciendo uso de una función matemática (Wavelet Haar) propuesta por Alfred Haar en 1909.

Los clasificadores haar, definen regiones rectangulares sobre una imagen en escala de grises (imagen integral) y al estar formada por un número finito de rectángulos, se puede obtener un valor escalar que consiste en sumar los píxeles de cada rectángulo, en base a una serie de clasificadores en cascada. Cada clasificador determina si la subregión se trata del objeto buscado o no. A diferencia de otros algoritmos, este sólo invierte capacidad de procesamiento a las subregiones que posiblemente representen un rostro, Bradski, G. & Kaebler, A (2012).

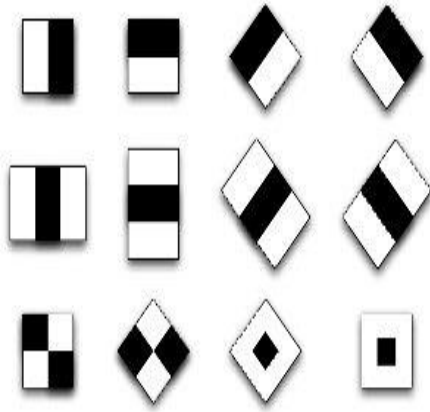


Figura 3. Clasificadores Haar.

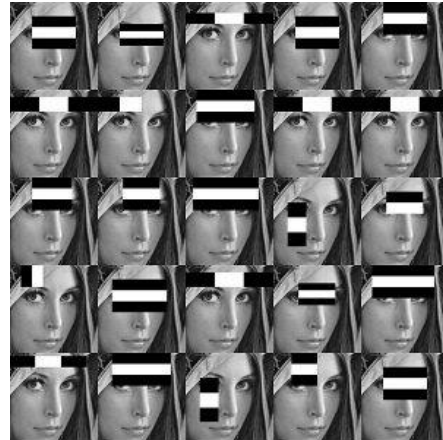


Figura 4. Cálculo de Haar sobre imagen.

3. Desarrollo.

El desarrollo de este proyecto está enfocado a dos tareas principales: El ensamblado del dispositivo y el funcionamiento del algoritmo sobre el cual trabaja el dispositivo.

Ensamblado de componentes.

1. Cámara: El módulo de la cámara se debe de conectar en el bus de datos específico para la cámara como se muestra en la siguiente figura.



Figura 5. Módulo de cámara insertado en raspberry.

2. Buzzer: Para conectar el buzzer se debe de contar con dos cables y para el módulo del GPS se necesitan 4 más. De acuerdo a los pines de la tarjeta de control Raspberry Pi para poder activar el buzzer debemos de conectar el cable negro al pin 6(GND) y el cable rojo al pin 15.

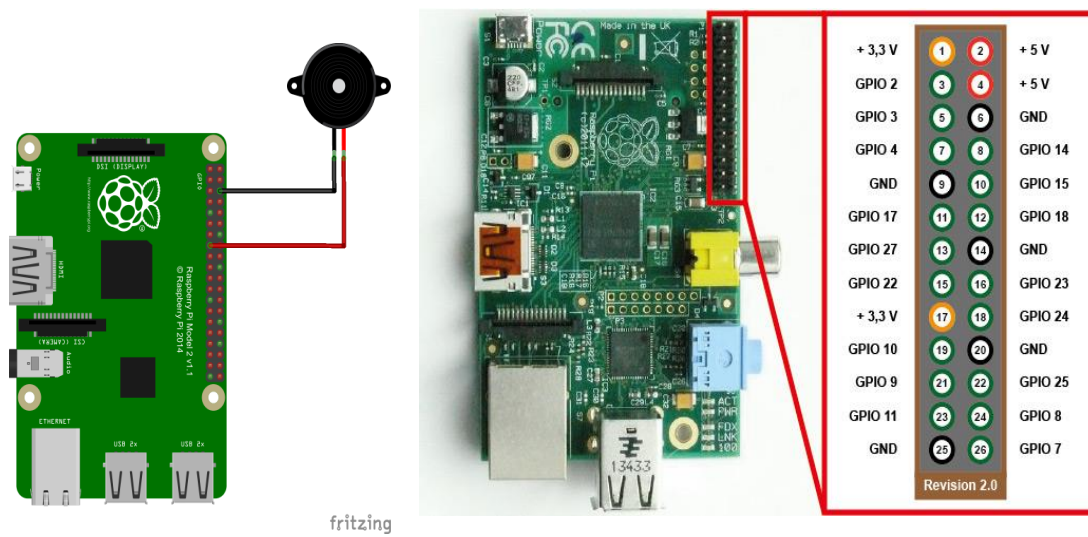


Figura 6. Conexión de buzzer en Raspberry Pi.

3. Carcasa: Después de tener todo correctamente conectado se introduce en la carcasa de modo que no se desconecte ni un cable para el correcto funcionamiento del dispositivo.



Figura 7. Componentes dentro de la carcasa.



Figura 8. Vista frontal del dispositivo ensamblado.



Figura 9. Vista trasera del dispositivo ensamblado.

Funcionamiento del algoritmo.

El dispositivo a través del módulo cámara que tiene integrado realiza la toma de video en tiempo real y mediante la tarjeta de control Raspberry pi mediante diversos filtros se busca detectar anomalías en la persona que está conduciendo, es decir, que la persona este cerrando los ojos por mucho tiempo, se está quedando dormida al frente del volante o se distraiga.

Durante el tiempo que trabaja el dispositivo está procesando el video y detectando el rostro de la persona, el procesamiento de los frames del video se realiza haciendo uso de la biblioteca OpenCV, el primer paso en el análisis de los frames procesados es, mediante el uso de los clasificadores Haar realizar la detección del rostro de la persona, una vez que se detecta el rostro se buscan los ojos para poder determinar ciertos comportamientos como parpadeos o posibles estados de sueño de la persona, este proceso se muestra en la figura 10.

El funcionamiento de la alarma es determinado de acuerdo a los ojos de la persona, si la persona cierra los ojos al frente de volante por más de 400 milisegundos que es la velocidad media del ojo de un parpadeo, en respuesta a ello se debe de activar la alarma para tratar de despertar al conductor, en caso de que solo sea un parpadeo por parte del conductor no sería motivo para activar la alarma.

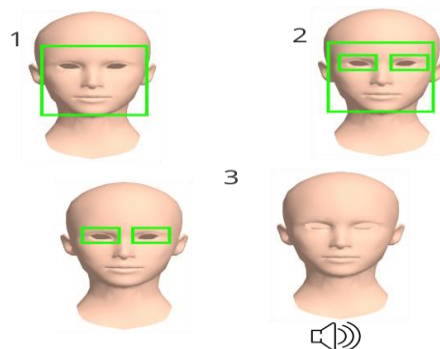


Figura 10. Proceso de detección de rostro y ojos del algoritmo.

Pruebas y resultados.

La evaluación del funcionamiento y efectividad del dispositivo se llevó a cabo con dos escenarios posibles:

1. Persona con condiciones de vista normal (sin anteojos)
2. Persona con algún problema visual (con anteojos)

Pruebas en persona 1.

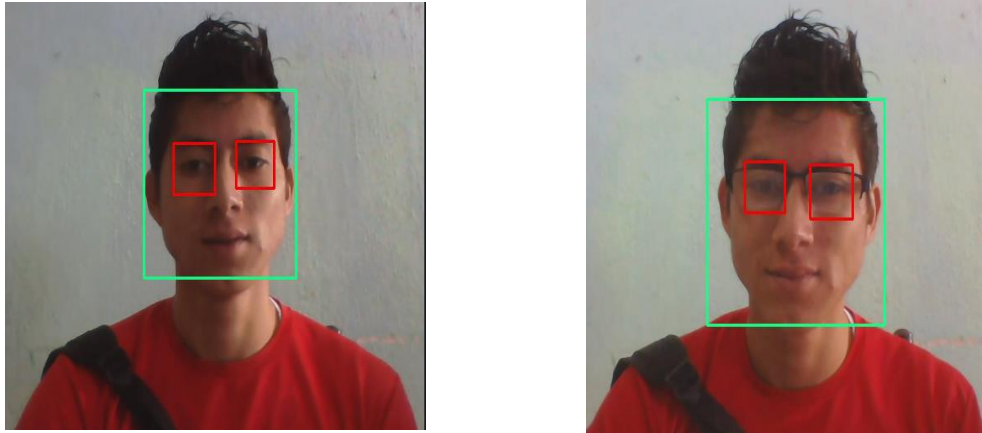


Figura 11. Detección de ojos y rostro en persona 1 sin y con anteojos.

Pruebas en persona 2.

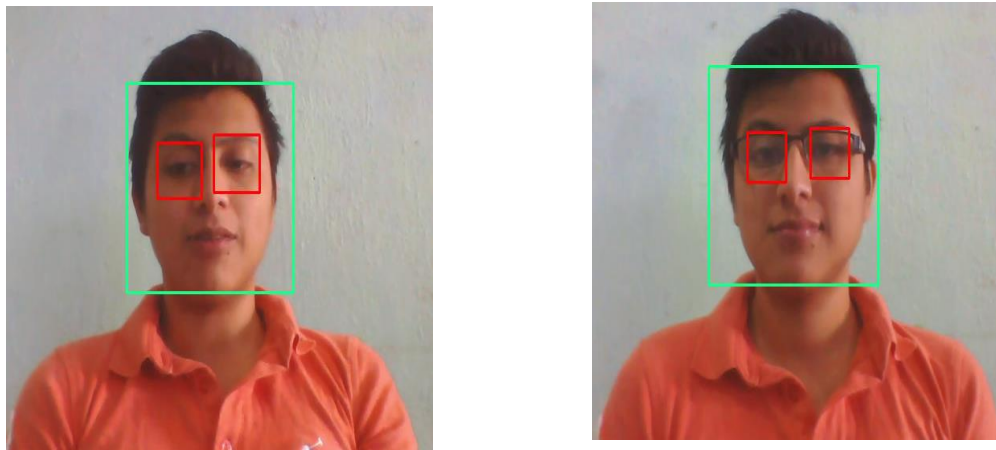


Figura 12. Detección de ojos y rostro en persona 2 sin y con anteojos.

Durante las pruebas se pudo observar que el factor iluminación en el ambiente donde actúa el dispositivo tiene repercusiones en el correcto funcionamiento de este, principalmente ocasiona lo que se denomina como “Falsos positivos” como se puede ver en la figura 13, este problema se puede resolver haciendo que el dispositivo realice una calibración automática en la iluminación percibida por la cámara, lo que se espera sea realizado en mejoras futuras al dispositivo.

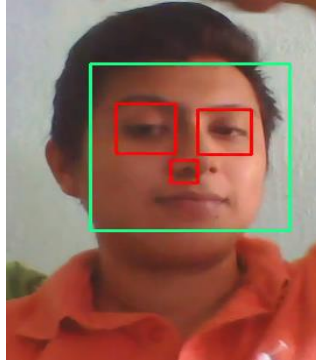


Figura 13. Detección de “Falso positivo”.

Conclusiones.

La visión artificial posee un área de conocimiento muy extensa e interesante para el desarrollo de proyectos que apliquen alguna de las técnicas del procesamiento de imágenes, durante el desarrollo de este proyecto solamente se exploró un pequeño sector de esta área, pero eso ha sido suficiente para lograr un producto funcional aunque cabe mencionar que faltan por realizar algunas cosas en los diferentes módulos, en la parte física, que es el dispositivo, se considera que el aspecto físico aún puede ser mejorado, incluso, el algoritmo para la detección de distracciones y estados de sueño aún puede ser pulido para poder resolver problemáticas que puedan surgir, por ejemplo cuando ocurren cambios en la iluminación (como se ilustró en las pruebas realizadas), es decir, aún puede mejorar la efectividad del algoritmo para que pueda trabajar de manera más eficiente en cualquier ambiente.

Referencias Bibliográficas.

- Bradski, G.& Kaebler, A.** (2012). Learning OpenCV. O'Reilly Media, Inc, 1005 Gravenstein Hwy N, Sebastopol, CA 95472, EE. UU.
- Cabeza, H. R.** (2010). Visión artificial aplicada a la industria. In INFAIMON, editor, The title of the book, volume 10 of 14, page 28, Edificio TR2 C/. Colon, 1 08222 Terrassa. Universidad Politécnica de Cataluña, JCEE'10.
- INEGI.** (2014). Accidentes de tránsito terrestre en zonas urbanas y suburbanas. Obtenido http://www.inegi.org.mx/est/lista_cubos/consulta.aspx?p=adm&c=8.
- INEGI.** (2015). Accidentes de tránsito terrestre en zonas urbanas y suburbanas. Obtenido de http://www.inegi.org.mx/est/lista_cubos/consulta.aspx?p=adm&c=1.
- Martelli, A.** (2006). Python in a Nutshell: A Desktop Quick Reference. In a Nutshell (O'Reilly). O'Reilly Media, 1005 Gravenstein Hwy N, Sebastopol, CA 95472, EE. UU.

Información de los autores.



Néstor Antonio Morales Navarro, Maestro en Ciencias en Ingeniería Mecatrónica por el Instituto Tecnológico de Tuxtla Gutiérrez en 2010. Es profesor en el área de Ingeniería en Sistemas Computacionales del Instituto Tecnológico de Tuxtla Gutiérrez desde 2012 y en el área de Licenciatura en Ingeniería Mecatrónica de la Universidad Valle de México Campus Tuxtla desde 2011. Se especializa en el área de Visión e Inteligencia Artificial.



Eliver Castillejos Lara, estudiante de ingeniería en sistemas computacionales en el Instituto Tecnológico de Tuxtla Gutiérrez, sus áreas de interés se relacionan con el desarrollo de software, microcontroladores y software libre.



Gilberto Méndez Sántiz, estudiante de ingeniería en sistemas computacionales en el Instituto Tecnológico de Tuxtla Gutiérrez, sus áreas de interés se relacionan con el desarrollo de software y microcontroladores.



Osvaldo Brindis Velázquez, Maestro en Ciencias en Ingeniería Mecatrónica, Jefe del Departamento de Ingeniería Eléctrica y Electrónica del Instituto Tecnológico de Tuxtla Gutiérrez (ITTG), las áreas de interés en las que investiga y desarrolla proyectos son el control automático, control robusto, control inteligente, automatización industrial, electrónica de potencia. Es profesor en el área de Ingeniería en Sistemas Computacionales del Instituto Tecnológico de Tuxtla Gutiérrez desde 2012 y en el área de Licenciatura en Ingeniería Mecatrónica de la Universidad Valle de México Campus Tuxtla desde 2011.



Salomón Velasco Bermúdez, Ingeniero en Electrónica, Jefe del Departamento de Servicios Escolares y docente de la carrera Ingeniería Electrónica en el Instituto Tecnológico de Tuxtla Gutiérrez (ITTG). Su área de interés es el diseño de sistemas electrónicos y sus aplicaciones. Colaboró con el Departamento de Ciencias Básicas en donde impartió las asignaturas de Probabilidad y Estadística, Algebra Lineal, Métodos Numéricos y Fundamentos de Física.