



Formato de Registro de Proyectos Modulares

Thu Feb 20 2020 10:35:26 GMT-0500 (EST)

Título del Proyecto Modular :

Cooler class

Área de Participación: Educación

Proyecto Modular: Módulo I - "Arquitectura y Programación de Sistemas", Módulo II - "Sistemas Inteligentes", Módulo III - "Sistemas Distribuidos"

Integrantes del Proyecto:

Nombre del Alumno	Código	Carrera	Correo electrónico
Eduardo Atonatiuh Camarena Santamaría	211666167	INCO	lalo.a.camarena@gmail.com
Diego Salvador Gonzalez Camacho	212662262	INCO	diegogcp10@gmail.com

Asesor(es) del Proyecto:

Nombre Profesor	Código	Departamento	Correo electrónico
<<Nombre Profesor (1er Asesor):>>	<<Código (1er Asesor)>>	<<Departamento (1er Asesor)>>	<<E-mail (1er Asesor)>>
<<Nombre Profesor (2do Asesor):>>	<<Código (2do Asesor)>>	<<Departamento (2do Asesor)>>	<<E-mail (2do Asesor)>>
2do Asesor externo:	<<Nombre de la Entidad donde Labora (2do Asesor)>>		

Firma
Eduardo Atonatiuh Camarena Santamaría

Vo. Bo.
<<Nombre Profesor (1er Asesor):>>

Guadalajara, Jalisco.



RESUMEN DEL PROYECTO

Thu Feb 20 2020 10:35:26 GMT-0500 (EST)

RESUMEN

Cuando se está televisando una clase o conferencia, la cámara enfoca al pizarrón o a la pantalla. En eso el profesor se atraviesa y con su cuerpo bloquea parte de la imagen. Y no se ve a dónde apunta o qué está señalando.

La idea es con la computadora alterar la imagen transmitida de manera que el cuerpo del profesor se borre o se haga transparente, y vuelva a aparecer el texto que se encuentra oculto por él.

El proyecto será una aplicación de escritorio en la cual los maestros podrán dar clases en vivo a sus alumnos, y para mejorar la experiencia de los alumnos a la hora de escribir nota, los profesores desaparecerán cuando estorben el pizarrón.

La aplicación será desarrollada en su completud en python ya que ambos integrantes conocemos bien el lenguaje y este tiene todas las herramientas que necesitamos para hacer la aplicación.

JUSTIFICACIÓN

Módulo I	<p>El sistema será un paquete de python y estará compuesto de 4 sub-paquetes para realizar los siguientes trabajos:</p> <p>Model: este sub-paquete contendrá todo lo relacionado a el modelo de visión artificial, se usará mask-RCNN, desde preprocesado de archivos y el entrenamiento del modelo.</p> <p>database: este sub-paquete contendrá todo lo relacionado a la base de datos, se usará sqlalchemy como ORM, de esta manera podremos programar cada tabla de la base de datos como una clase de python y de la misma manera podremos hacer diferentes funciones para llamar a la base de datos desde python en ves de escribir todo un query cada ves que se necesite acceder a la base de datos.</p> <p>Transmition: este paquete contendrá todo lo relacionado a la transmisión de datos, desde las hacer las llamadas a la base de datos (para hacer el login de los usuarios, mostrar su información, etc.) hasta conectar a los usuarios a las clases y transmitirles el livestream de la clase, se usará flask para hacer este paquete.</p> <p>Window: Será todo el front-end de la aplicación, se usará pyqt5 para desarrollar el front-end</p>
Módulo II	<p>Visión artificial.</p> <p>El proyecto usará visión artificial para localizar al maestro para que cada que el maestro estorbe el pizarrón este se transparente.</p> <p>El algoritmo que se utilizará para esto es una Mask-RCNN (Region-based Convolutional Neural Network) este tipo de redes neuronales son muy usadas en visión computacional ya que esta red nos regresa el un outline casi perfecto del objeto (depende del entrenamiento) en ves de una caja que encierre al objeto.</p> <p>La diferencia de esta red a otras es que en vez de analizar la imagen por sectores la analizan toda rápidamente en busca de patrones y una ves encontrado un patrón, lo extrapola de la imagen y usa una SVM (support vector machine) para tomar una decisión de si el objeto es de la(s) clase(s) que está entrenado identificar.</p>



UNIVERSIDAD DE GUADALAJARA

CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E INGENIERIAS
DIVISIÓN DE ELECTRONICA Y COMPUTACIÓN
COORDINACIÓN DE INGENIERÍA EN COMPUTACIÓN

Módulo III

Comunicación en tiempo real.

Al ser clases en vivo el proyecto transmitirá video en tiempo real se necesita que una computadora capture el video y lo transmita a otras computadoras, la computadora que esté transmitiendo el video aceptará conexiones de nuevos equipos en cualquier momento mientras la transmisión esté activa.

Para lograr esto usaremos Flask y openCV, openCV será usado para capturar el video ya que tiene un módulo para capturar video desde la webcam del equipo y usaremos flask para aceptar las conexiones a las clases, esto funcionará haciendo una llamada a la base de datos para comprobar la sesión así como que el estudiante esté registrado a la clase y si nos acepta el request nos conectará a la transmisión. De la misma manera flask será usado para transmitir el video ya que este nos permite mandar respuestas con funciones especiales que pueden ser interrumpidas y reanudadas, de esta manera se puede transmitir video sin preocuparnos de que se pierda la conexión momentáneamente o que se pierdan algunos paquetes en la red.

ACTIVIDADES

Eduardo - Programación de parte backend (servidor de la aplicación, el modelo, transmisión y la base de datos).

Diego - Programación de el frontend y parte del backend (transmisión y base de datos)