

Universidad Latina de Panamá

Estudiantes:

Lisandro García 128315327

Humberto Tapia 8-905-1643

Carlos Jiménez Rao 8-874-411

Proyecto:

Sistema de alerta inteligente mediante cámaras para la prevención de robos.

Materia:

Control Digital

Profesor:

Pablo González

**Introducción**

Los principales dispositivos a nivel mundial, en la actualidad funcionan gracias a que están escrito en un lenguaje de programación.

Estos lenguajes sirven para que el humano pueda comunicarse con los diversos equipos informáticos y electrónicos que pueden ser encontrados en el mercado, dicho esto los lenguajes de programación tienen una gran usabilidad en el área de la electrónica.

Gracias a esto hemos decidido crear un dispositivo de seguridad accesible de bajos recursos, utilizando como base el lenguaje de programación Python y diversas librerías que pueden ser encontradas en su gran red de repositorios.

Este proyecto funciona con librerías de Open-CV para la detección de objetos en la fuente de video, MIME para él envió de correos electrónicos, imutils para el procesamiento de imágenes y flask para lograr montar la visualización de la cámara en un entorno web.

**Problema de investigación**

Planteamiento del problema Aun cuando han surgido alternativas en materia de seguridad, aún no han sido suficientes teniendo en cuenta que los tiempos de robo son cada vez más cortos. Por ello, el presente proyecto está enfocado en dar un complemento de seguridad a los locales comerciales mediante el uso de tecnologías como la integración de la Raspberry Pi, un sensor de movimiento y una cámara que permitan reducir el riesgo de robo y tener toma de decisiones por parte de los dueños de los locales comerciales en caso de robo, generando así una nueva opción de seguridad.

**Antecedentes**

Rodarte, Gutiérrez y Pérez (2011) propusieron un sistema de seguridad de bajo costo con una red de sensores de movimiento PIR, enlazados por RF a un sistema de hardware libre Arduino, sirviendo este como interfaz. Aunque dicho proyecto versa sobre el robo en hogares, no se menciona en el mismo una forma de que el usuario conozca la vulnerabilidad de manera remota. En contraste, el proyecto que se explicita en este artículo utiliza un dispositivo más potente —la placa Raspberry Pi—, con la posibilidad de integrarle una cámara y procesar la captura de una imagen para ser enviada a través de correo electrónico; a nuestro juicio, esto resulta más práctico que la idea propuesta por los autores mencionados.

Zeki, Eldaw, Ibrahim, Haruna y Abdulkareem (2013) consideraron un sistema automático de control de seguridad interactivo, similar al propuesto en tanto captura una imagen del intruso; sin embargo, se enfoca en lugares de la vivienda, gestionado mediante un portal web. Frente a esta idea, el proyecto que se presenta aquí tiene como valor agregado el uso de un tiempo definido para dar como sospechosa a una persona que se encuentre dentro de la zona de detección, ya que está enfocado hacia los locales comerciales.

Sarthak, Vaibhav y Goyal (2014) exploraron en su artículo el uso de Raspberry Pi y envíos de mensajes de texto para domótica, generando control sobre la casa, en tanto al accionar un switch, se envía la alerta de uso de este a través de correo electrónico; pero no implementa algún sensor para darle cierta seguridad a este sistema. Adicionalmente, esta iniciativa pasa por alto el campo de los locales comerciales o negocios: utiliza un algoritmo realizado en el lenguaje de Python, aunque sí aborda el control de seguridad en las casas. El presente proyecto, a su turno, utiliza un sensor PIR que cubre un ángulo de 100 grados, de forma tal que no es necesario que el usuario esté en la ubicación para que funcione, motivo que lo hace apto para locales comerciales.

**Justificación**

Hoy en día los sistemas de intrusión son un elemento más en las tiendas y empresas, siendo cada vez más habituales también en los hogares particulares. Protegen sus bienes y activos de posibles robos proporcionando una alta seguridad mediante su alta tecnología. Paralelamente actúan como elemento disuasorio y permiten su automatización con sistemas de encendido de luces, puertas u otros dispositivos. El principal funcionamiento de los Sistemas de Seguridad está basado en la detección de robo, intrusión e incendio mediante sensores de varias tecnologías conectados a Centrales de alarmas. Las centrales disponen de elementos de salida con los que avisar del estado del sistema. Cuando uno de los elementos sensores detectan una situación de riesgo estos transmiten inmediatamente el aviso a la central, esta procesa la información recibida y ordena en repuesta la emisión de señales sonoras o luminosas alertando de la situación. Todos estos elementos poseen un control contra sabotaje de manera que si en algún elemento se corta la alimentación o se produce la rotura de algunos de sus componentes se enviará una señal a la central de alarma para que ésta accione los elementos de señalización y aviso correspondientes.

**Pregunta de investigación**

Como se puede mejorar la seguridad y vigilancia de residencias, comercios e industrias aprovechando las tecnologías actuales y aplicando el internet de las cosas.

**Objetivo General**

Implementar un sistema de detección automática de movimientos corporales o facial, utilizando técnicas de procesamiento digital de imágenes.

**Objetivos Específicos**

* Desarrollar un software utilizando lenguaje en Python, que utilice las librerías de OPEN-CV que cumplan la función de detectar los movimientos corporales o faciales mediante una cámara.
* Utilizar un Raspberry que permita procesar las señales digitales obtenidas del software de detección y envié una copia del video a un servidor en la nube.
* Incentivar el desarrollo de la investigación en el área de procesamiento de imágenes.

**Hipótesis**

Si es posible la implementación de un sistema automático de detección de movimientos corporales o faciales utilizando técnicas de procesamiento digital de imágenes que contribuya a la seguridad y prevención de robos.

**Reducción de pérdidas, robos y vandalismo:** Las cámaras de vigilancia no solo capturan y almacenan más videos que los sistemas analógicos, sino que la calidad de la transmisión de video también es muy superior. Los dispositivos IP ofrecen imágenes en alta definición y ángulos de visión más amplios. Esta calidad proporciona a las empresas la capacidad de ver claramente lo que está sucediendo, dentro y alrededor de sus instalaciones.

**Monitoreo remoto:** esta podría ser una de las ventajas más valoradas de un sistema de vigilancia IP. Con un sistema de video altamente integrado, los dueños de los negocios pueden monitorear la actividad en sus transmisiones de vigilancia desde cualquier parte del mundo, siempre que tengan una conexión a Internet.

El video inteligente se trata del cambio de las imágenes hacia datos, desde la simple recopilación de información para investigación forense, hasta el análisis y comprensión del contexto de los datos capturados. Utilizar inteligencia artificial y algoritmos derivados de big data puede proporcionar información inmediata y predicciones prospectivas.

La necesidad de proporcionar capacidades inteligentes al video, junto con el desarrollo de sistemas de vigilancia basados en la nube, ha llevado a la evolución de cámaras inteligentes conectadas en red. Estas cámaras poseen un poderoso software de análisis y un dispositivo de almacenamiento capaz de implementar capturas y análisis locales (donde los datos se generan y se procesan), proporcionando información valiosa en vivo, sin los efectos de la disponibilidad o la latencia de la red.

**Código Python**

Ejecución

source ~/.profile

workon cv3

python main.py

MAIN.PY

#En este archive va la programación principal donde el programa toma las capturas de imágenes cuando detecta algo frente a ella y se configura el entorno web con flask.

import cv2

import sys

#import smtplib

from mail import sendEmail

from flask import Flask, render\_template, Response

from camera import VideoCamera

from flask\_basicauth import BasicAuth

import time

import threading

email\_update\_interval = 10 # cantidad de tiempo que se toma el intervalo de cada fotografia al detectar a la persona

video\_camera = VideoCamera(flip=True) # creates a camera object, flip vertically

object\_classifier = cv2.CascadeClassifier("models/facial\_recognition\_model.xml") # an opencv classifier

# Usuario y contraseña de donde se ve el video en vivo

app = Flask(\_\_name\_\_)

app.config['BASIC\_AUTH\_USERNAME'] = 'control'

app.config['BASIC\_AUTH\_PASSWORD'] = 'control'

app.config['BASIC\_AUTH\_FORCE'] = True

basic\_auth = BasicAuth(app)

last\_epoch = 0

def check\_for\_objects():

global last\_epoch

while True:

try:

frame, found\_obj = video\_camera.get\_object(object\_classifier)

if found\_obj and (time.time() - last\_epoch) > email\_update\_interval:

last\_epoch = time.time()

print ("Mandando Correo...")

sendEmail(frame)

print ("Listo!")

except:

print((("Error al mandar el correo: "), sys.exc\_info()[0]))

@app.route('/')

@basic\_auth.required

def index():

return render\_template('index.html')

def gen(camera):

while True:

frame = camera.get\_frame()

yield (b'--frame\r\n'

b'Content-Type: image/jpeg\r\n\r\n' + frame + b'\r\n\r\n')

@app.route('/video\_feed')

def video\_feed():

return Response(gen(video\_camera),

mimetype='multipart/x-mixed-replace; boundary=frame')

if \_\_name\_\_ == '\_\_main\_\_':

t = threading.Thread(target=check\_for\_objects, args=())

t.daemon = True

t.start()

app.run(host='0.0.0.0', debug=False)

MAIL.PY

#En este archivo se parametriza, configura y se programa el envio del correo electrónico.

import smtplib

from email.mime.multipart import MIMEMultipart

from email.mime.text import MIMEText

from email.mime.image import MIMEImage

# Correo electronico donde se envia la imagen

fromEmail = ('pfinal1122@gmail.com')

fromEmailPassword = ('PF12345678')

# Correo electronico donde se recibe la imagen

toEmail = ('carlos.rao25@gmail.com')

def sendEmail(image):

msgRoot = MIMEMultipart('related')

msgRoot['Subject'] = 'Camara de Seguridad'

msgRoot['From'] = fromEmail

msgRoot['To'] = toEmail

msgRoot.preamble = 'Raspberry pi security camera update'

msgAlternative = MIMEMultipart('alternative')

msgRoot.attach(msgAlternative)

msgText = MIMEText('Smart security cam found object')

msgAlternative.attach(msgText)

msgText = MIMEText('<img src="cid:image1">', 'html')

msgAlternative.attach(msgText)

msgImage = MIMEImage(image)

msgImage.add\_header('Content-ID', '<image1>')

msgRoot.attach(msgImage)

smtp = smtplib.SMTP('smtp.gmail.com', 587)

smtp.starttls()

smtp.login(fromEmail, fromEmailPassword)

smtp.sendmail(fromEmail, toEmail, msgRoot.as\_string())

smtp.quit()

CAMERA.PY

#En este archive se hace la respective congifuracion de la camara.

import cv2

from imutils.video.pivideostream import PiVideoStream

import imutils

import time

import numpy as np

class VideoCamera(object):

def \_\_init\_\_(self, flip = False):

self.vs = PiVideoStream().start()

self.flip = flip

time.sleep(2.0)

def \_\_del\_\_(self):

self.vs.stop()

def flip\_if\_needed(self, frame):

if self.flip:

return np.flip(frame, 0)

return frame

def get\_frame(self):

frame = self.flip\_if\_needed(self.vs.read())

ret, jpeg = cv2.imencode('.jpg', frame)

return jpeg.tobytes()

def get\_object(self, classifier):

found\_objects = False

frame = self.flip\_if\_needed(self.vs.read()).copy()

gray = cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY)

objects = classifier.detectMultiScale(

gray,

scaleFactor=1.1,

minNeighbors=5,

minSize=(30, 30),

flags=cv2.CASCADE\_SCALE\_IMAGE

)

if len(objects) > 0:

found\_objects = True

# Para dibujarle los Cuadros a las fotos

for (x, y, w, h) in objects:

cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2)

ret, jpeg = cv2.imencode('.jpg', frame)

return (jpeg.tobytes(), found\_objects)