28-10-2019



Bermello Burgos Brenda

Braganza Zorilla Samuel

EVALUACIÓN DE TÉCNICAS DE INTERPOLACIÓN CONVENCIONAL PARA AUMENTO DE RESOLUCIÓN EN VIDEOS

**Contenido**

[**RESUMEN** 2](#_Toc24402596)

[**OBJETIVOS** 2](#_Toc24402597)

[**JUSTIFICACIÓN** 3](#_Toc24402598)

[**METODOLOGÍA** 4](#_Toc24402599)

[ Recursos 4](#_Toc24402600)

[ Instalación OpenCV 4](#_Toc24402601)

[ Procedimiento 5](#_Toc24402602)

[**BIBLIOGRAFÍA** 7](#_Toc24402603)

# **RESUMEN**

La problemática principal que se plantea en el presente proyecto es la generación u obtención de vídeos con resolución aumentada (SR) a partir de la disminución de la resolución a la cuarta parte del tamaño de estos (LR) aplicando diferentes técnicas convencionales.

A partir de este punto pasamos de procesamiento digital de imágenes a visión computacional ya que se abordará la detección de rostros en cada uno de los estados en que se encuentren los videos: video original, video en alta resolución SR y video en baja resolución LR.

En un área de interés la superresolución obtiene más pixeles por lo tanto este proceso estima información de imagen de baja resolución por medio de interpolación y la coloca en imagen o video con mayor resolución.

# **OBJETIVOS**

1. Identificar técnicas de interpolación convencionales de imágenes y videos.
2. Aplicar las técnicas de interpolación de videos para inicialmente bajar la resolución de un video de prueba y luego subirla para su posterior comparativa.
3. Extraer la cantidad de rostros que se detectan tanto en el video de baja resolución como en los de alta resolución.
4. Generar la comparativa de las técnicas de interpolación implementadas e identificar la más óptima.

# **JUSTIFICACIÓN**

Existen diversos usos muy útiles y eficaces de la interpolación. En el campo de la construcción existe un software llamado ArcGIS empleado para crear mapas, realizar análisis espaciales, administrar información geográfica, etc. Dentro de una de sus funcionalidades es capaz de crear una superficie de elevación a partir de un conjunto de mediciones de muestra gracias a la interpolación.

En el campo de la informática forense, es posible identificar la marca y modelo de la cámara que ha generado una imagen o video en dispositivos que poseen una mediante algoritmos de interpolación cromática aplicados en el proceso de formación de la imagen.

# **METODOLOGÍA**

Debido a las limitaciones que posee la visión del ser humano el procesamiento de imágenes se ha transformado en un factor importante en la búsqueda de información, en la mejora de calidad y aspecto de las imágenes.

En muchas ocasiones nos hemos presentado ante la situación de querer arreglar una imagen que no está muy clara, los objetos se visualizan borrosos, no está nítida la imagen y no nos permite la detección correcta de los objetos que se encuentran. El objetivo de usar técnicas convencionales es aumentar la resolución y así poder evaluar cual tiene mejor calidad de conversión. Para poder lograr este objetivo recurrimos a la utilización de algoritmo de detección de rostros.

Localizar áreas dentro de una imagen o video que contienen un rostro se denomina detección de rostros.

## Recursos

Durante del desarrollo del proyecto se empleará el uso de las siguientes herramientas:

-Python

Python es un lenguaje de programación interpretado desarrollado como proyecto Open Source por Guido Van Rossum, multiplataforma, intuitivo y de fácil entendimiento.

-OpenCV

OpenCV (Open Source Computer Vision) es la librería de visión de computador más importante y usada. Es multiplataforma tanto en sistemas operativos como en lenguajes de programación.

-Numpy

Numpy es una librería (Numerical Python) facilita poderosas estructuras de datos, implementando matrices y matrices multidimensionales. Estas estructuras de datos garantizan cálculos eficientes con matrices.

## Instalación OpenCV

Para la instalación de OpenCV en Windows es necesario seguir los siguientes pasos:

1. Tener instalado Python, para este proyecto se usó Python 3.6.0.
2. Instalar el paquete Numpy desde el símbolo del sistema (cmd): pip install numpy
3. Verificar si Numpy está instalado correctamente, escribiendo en el IDLE de Python ‘import numpy as np’
4. Descargar OpenCV en la versión deseada ingresando a la página de SourceForge, doble click y extraer la información.
5. Ir a la carpeta opencv/build/Python/3.6
6. Copiar el archivo cv2.pyd en C:\Program Files\Python36\Lib\site-packages
7. Desde IDLE de python verificar: import cv2

## Procedimiento

El presente proyecto se dividirá en cuatro etapas:

* + - 1. Generación de video en baja resolución.

Se obtendrá y cargará un video cuya particularidad será la visualización de rostros. Para generar el video con menor resolución utilizaremos el método Pirámide Gaussiana que al ser usado una vez la imagen resultante será exactamente un cuarto de la superficie. Cabe recalcar que cuando se reduce el tamaño de una imagen o video se pierde información.

* + - 1. Generación de videos con resolución aumentada.

Para aumentar la resolución del video generado (baja resolución) se procederá a utilizar técnicas de interpolación:

Bilineal: Implementa un filtro de media para los ejes horizontal y vertical.

Relación pixel-área: Utiliza la relación entre el área y los pixeles, similar al vecino más cercano.

Vecino más cercano: Permite cambiar el valor de los pixeles con el valor del vecino próximo a interpolar.

Bicúbica: Simliar a la interpolación bilineal pero preserva mejor los detalles.

Lanzcos: Realiza interpolación de 8x8 pixeles.

Por cada técnica de interpolación se generará un nuevo video.

* + - 1. Detección de rostro.

Para el proceso de detección de rostro utilizaremos el algoritmo Haar Cascades.

El siguiente diagrama muestra de manera general el procedimiento de detección de rostros.

Imagen de entrada

Imagen integral

Extracción de características

Clasificación

Imagen con rostro detectado

En este proceso existen 3 etapas:

Imagen integral: Permite extraer característica de manera rápida. Cada punto de la imagen integral contiene el resultado de la suma de los valores de todos los puntos ubicados a la izquierda y encima de la imagen original.

Extracción de características: Las características son formas geométricas, compuestas por rectángulos grises y blancos denominados filtros Haar.

Clasificación: Combinación de los grupos clasificadores complejos para descartar los rangos que no contienen rostros.

* + - 1. Comparativa.

En esta etapa se procederá a realizar la comparativa con los videos en alta resolución y el de baja resolución en base a los resultados obtenidos con el video original. Para esto se obtendrá el porcentaje de error de cada uno de los videos por medio del número de rostros detectados.

# **BIBLIOGRAFÍA**

calameo. (2008). *calameo.* Recuperado el 25 de 10 de 2019, de https://es.calameo.com/read/0052422633cdeefb3aa8e

ECURED. (02 de 03 de 2010). *Python.* Recuperado el 25 de 10 de 2019, de https://www.ecured.cu/Python

Openinventionnetwork. (18 de 08 de 2015). *Oficina de Software Libre Universidad de la Laguna.* Recuperado el 25 de 10 de 2019, de Oficina de Software Libre Universidad de la Laguna: https://osl.ull.es/software-libre/opencv-libreria-vision-computador/

theailearner. (15 de 11 de 2018). *theailearner.* Recuperado el 26 de 10 de 2019, de theailearner: https://theailearner.com/tag/image-interpolation-opencv-python/

UNIVERSIDAD COMPLUTENSE DE MADRID. (2016). *UNIVERSIDAD COMPLUTENSE DE MADRID.* Recuperado el 27 de 10 de 2019, de UNIVERSIDAD COMPLUTENSE DE MADRID: https://eprints.ucm.es/49879/1/Memoria%20TFG.pdf