

# OP7. Data Analysis with Art images from The Louvre

**Juan David Hernández\***

*Universidad Nacional de Colombia*

**Andres Pinzon\*\***

*Universidad Antonio Nariño*

**Ariel Rivera Castro\*\*\***

*Universidad Nacional de Ingenieria*

*Asesor: Arturo Sanchez*

18 de Mayo 2021

## Índice

<b>1. Introducción</b>	<b>2</b>
<b>2. Metodología: Extracción y análisis de pinturas del museo de Louvre.</b>	<b>3</b>
2.1. Programa de extracción de imágenes de una URL dada . . . . .	3
2.1.1. Descripción. . . . .	3
2.1.2. Librerías necesarias para ejecutar el programa. . . . .	3
2.2. Identificar las personas que se encuentran en cada imagen . . . . .	4
2.2.1. Descripción. . . . .	4
<b>3. El experimento y los resultados</b>	<b>7</b>
<b>4. Conclusiones y Recomendaciones</b>	<b>10</b>
4.1. Programa de extracción de imágenes de una URL dada . . . . .	10
4.2. Identificar las personas que se encuentran en cada imagen . . . . .	10
<b>5. Referencias</b>	<b>11</b>

---

\* e-mail: judhernandezra@unal.edu.co

\*\* e-mail: cpinzon31@uan.edu.co

\*\*\* e-mail: ariverac@uni.pe

## Resumen

Usamos el framework para python ImageAIque [?] para observar sus capacidades y alcances con la detección de objetos en imágenes y extraer esos objeto de la imagen y se uso como Dataset de analisis, pinturas del Museo del Louvre [?]. Se creo un metodo automatico para desde el notebook en Jupyterhud de la redclara poder extraer imagenes de cuadros desde la web del Museo del Louvre, se pudo en algunas pinturas seleccionadas contar correctamente el numero de persosnas presentes, unas 20 personas, pudimos obtener estadisticas automatizadas del numero de personas presentes en pinturas por año o por siglo. Las capacidades de este tipo de herrameintas podrian ser utiles en el analisis de imagenes astronomicas de forma automatizada o en espectroscopia, etc.

- <https://github.com/OlafenwaMoses/ImageAI>,
- <https://collections.louvre.fr/en/page/apropos>

## 1. Introducción

ImageAI es un framework que proporciona métodos muy convenientes y potentes para realizar la detección de objetos en imágenes y extraer esos objeto de la imagen. La clase de detección de objetos es compatible con RetinaNet, Yolov3 y TinyYOLOv3.

Construido pensando en la simplicidad, ImageAI admite una lista de algoritmos de aprendizaje automático de última generación para la predicción de imágenes, la predicción de imágenes personalizadas, la detección de objetos, la detección de videos, el seguimiento de objetos de video y los entrenamientos de predicciones de imágenes. ImageAI actualmente admite la predicción de imágenes y el entrenamiento utilizando 4 algoritmos de aprendizaje automático diferentes entrenados en el conjunto de datos ImageNet-1000. ImageAI también admite la detección de objetos, la detección de video y el seguimiento de objetos utilizando COCO dataset. Por último, ImageAI le permite entrenar modelos personalizados para realizar la detección y el reconocimiento de nuevos objetos.

Eventualmente, ImageAI proporcionará soporte para aspectos más amplios y especializados de la Visión por Computadora, incluyendo y no limitado al reconocimiento de imágenes en entornos especiales y campos especiales.

## 2. Metodología: Extracción y análisis de pinturas del museo de Louvre.

### 2.1. Programa de extracción de imágenes de una URL dada

#### 2.1.1. Descripción.

Crear un programa capaz de extraer imágenes de la página del museo de Louvre, Francia. Luego con un software de análisis de imagen se identificar las personas que había en cada cuadro y finalmente analizar cómo cambia el número de personas promedio por cuadro según la época en la que se realizó la pintura.

#### 2.1.2. Librerías necesarias para ejecutar el programa.

Antes de ejecutar el notebook es necesario instalar las siguientes librerías y ejecutando los siguientes comandos.

```
!pip install --user bs4
!pip install --user urllib.request
!pip install --user PIL
!pip install --user io

from bs4 import BeautifulSoup
from urllib.request import urlopen
from PIL import Image
import requests
import pandas as pd
from io import BytesIO
```

Para este código vamos a utilizar la librería BeautifulSoup, la cual identifica los diferentes elementos de una URL dada. Luego creamos una función (get\_images), que crea una lista con todos los objetos tipo imagen que detectó la librería BeautifulSoup.

Acá se encuentra el código original escrito para python2 y que utiliza la librería urllib2, la cual no es compatible con python3 (a diferencia de urllib.request que fue la utilicé en este código). El código corresponde a la tercera respuesta del foro.

Para extraer imágenes se crea dos funciones utilizando la librería BeautifulSoup, de la siguiente forma. La cual recibe como variable una URL.

```
def make_soup(url):
    html = urlopen(url).read()
    return BeautifulSoup(html)

def get_images(url):
    soup = make_soup(url)
    images = [img for img in soup.findAll('img')]
    image_links = [each.get('src') for each in images]
    return image_links
```

Finalmente, con la librería PIL, usando el comando `Image.open`, extraemos la imagen de la URL dada.

```
im1 = Image.open(requests.get(get_images('https://collections.louvre.fr/en/ark:/53355/cl010208581')[2], stream=True).raw)
im1
```

## 2.2. Identificar las personas que se encuentran en cada imagen

### 2.2.1. Descripción.

La identificación de objetos se realiza a través del módulo [imageai](#) el cual tiene la capacidad de comparar elementos dentro de una imagen con un modelo para indicar que clase de objeto es, en este caso son personas.

El identificador fue implementado con base al código indicado [aquí](#) y modificado para que realizara la detección automáticamente.

Para realizar la identificación de personas se instalan las siguientes librerías

```
!pip install --user tensorflow==2.4.0
!pip install --user imageai --upgrade
```

Se actualizan las siguiente:

```
!pip install --user keras==2.4.3 numpy==1.19.3 pillow==7.0.0 scipy==1.4.1 h5py==2.10.0
```

Ahora se importa el módulo `imageai` y `OS`. También se ejecuta una línea que va a indicar donde está el path del directorio donde están los modelos y las pinturas.



Figura 1: Extraemos una imagen de la URL dada.

```
from imageai.Detection import ObjectDetection
import os
```

Los modelos se pueden descargar directamente del repositorio indicado en la parte superior donde se encuentra el código original.

Ahora con las líneas siguientes se llama al modelo, en este caso es resnet50 y en la cuarta línea se carga el modelo. Finalmente en la línea 5 se indica el camino donde está guardada la imagen (en caso de que esté guardada en un lugar diferente al lugar donde está el archivo de python se le indica el path) y en la segunda parte del código se le indica donde guardar una copia de la imagen analizada donde se muestra en un recuadro la persona que identificó y el porcentaje de coincidencia con el modelo.

```
execution_path = os.getcwd()

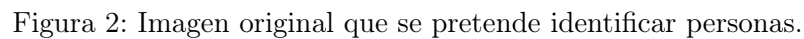
detector = ObjectDetection()
detector.setModelTypeAsYOLOv3()
detector.setModelPath( os.path.join(execution_path , "yolo.h5"))
detector.loadModel()
detections = detector.detectObjectsFromImage(input_image=os.path.join(execution_path , "image.jpg"))

import matplotlib.pyplot as plt
import matplotlib.image as img
image = img.imread('image.jpg')
plt.imshow(image)
plt.show()

import matplotlib.pyplot as plt
import matplotlib.image as img
image = img.imread('image.jpg')
plt.imshow(image)
plt.show()
```

El código anterior identifica 80 objetos en cada imagen. Luego de recorrer la imagen se obtendrá lo siguiente

```
image = img.imread('imagenew.jpg')
plt.imshow(image)
plt.show()
```



Para realizar las gráficas, se utiliza la librería seaborn que se instala mediante el siguiente comando

Usando los resultados encontrados por el programa de identificación de personas en imágenes, el número promedio de personas por año que se encontró fue el siguiente:

Y por siglo

7

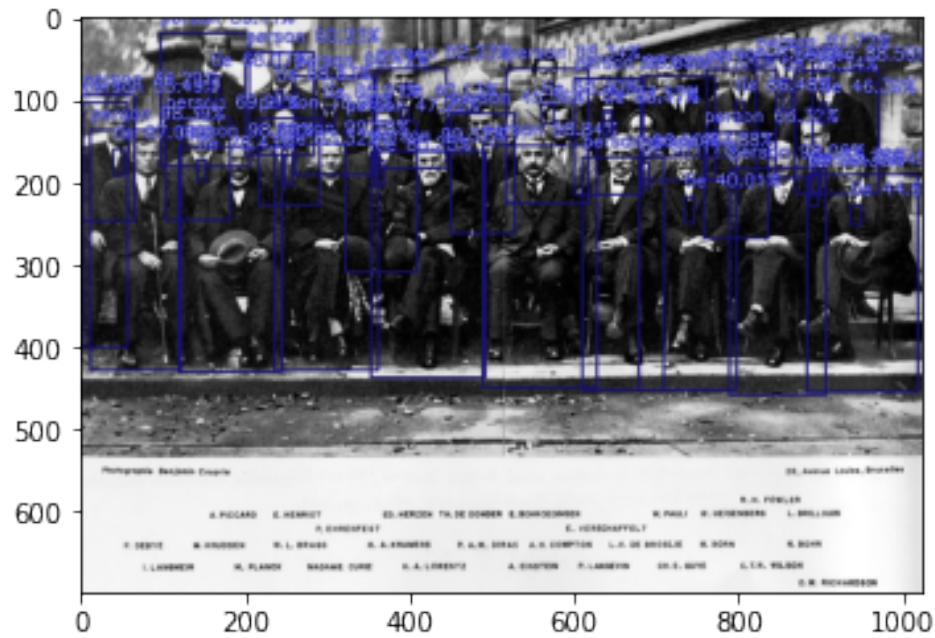


Figura 3: Imagen en la que se identifico personas

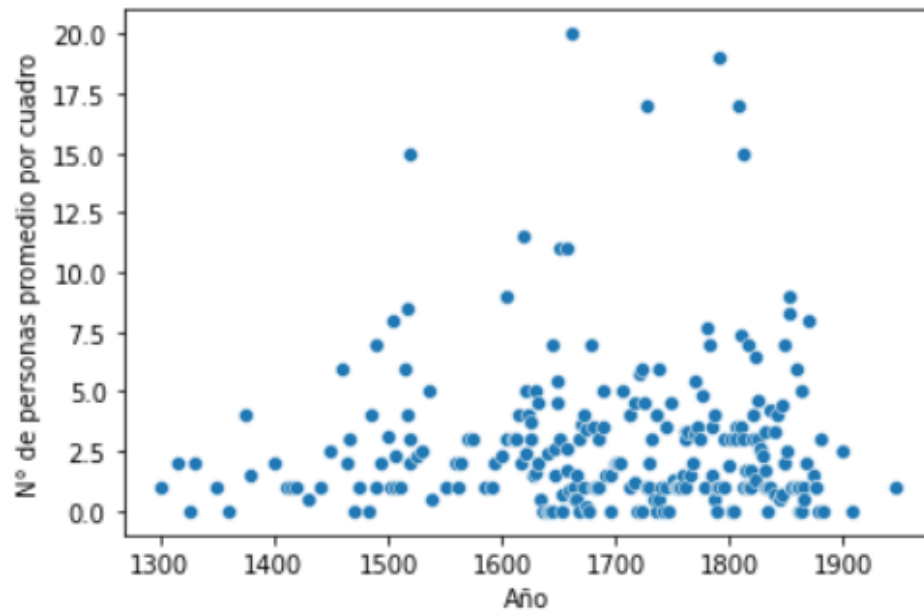


Figura 4: número promedio de personas por año que se encontró



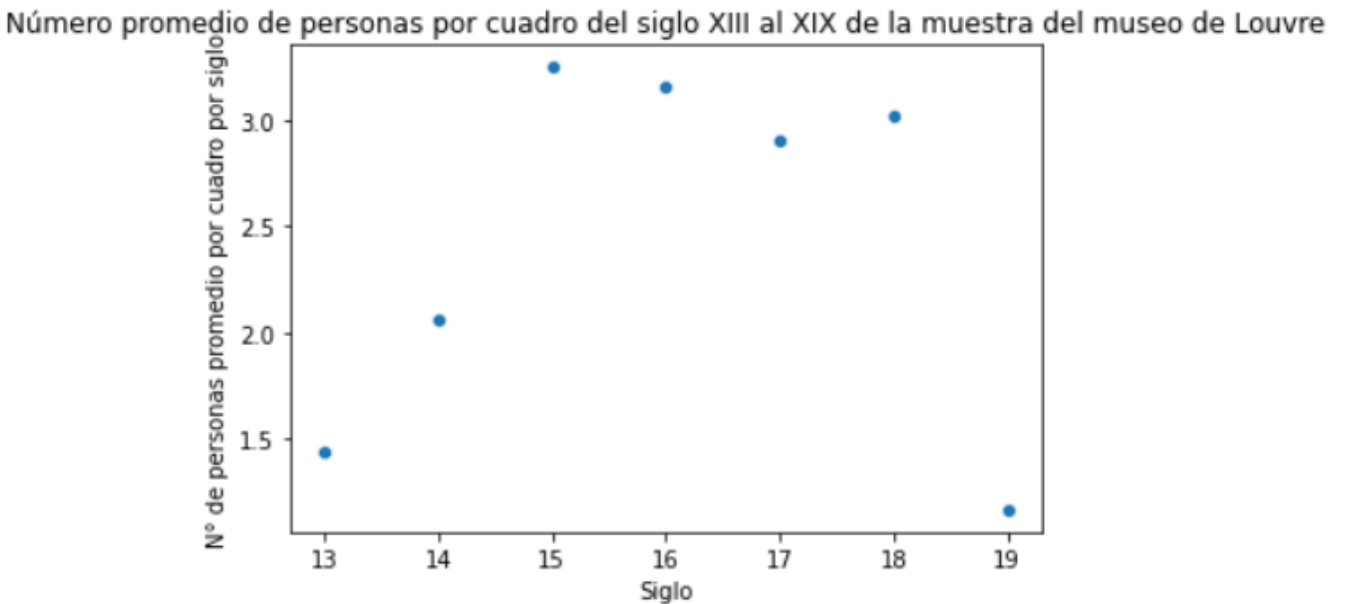


Figura 5: número promedio de personas por siglo que se encontró

```
plt.ylabel("numero de personas promedio por cuadro por siglo")
plt.xlabel("Siglo")
plt.title("numero promedio de personas por cuadro del siglo XIII
          al XIX de la muestra del museo de Louvre")
```

La gráfica nos dice que en el siglo 13 y 14 había una tendendencia a pintar retratos, puesto que el número promedio de personas por cuadro era menor a 2.1. En el siglo 15 se encuentra que el número de personas por cuadro es mayor comparado con los otros siglos, aunque, la diferencia es de 1.6 comparado con el siglo 13, lo que nos dice que en la muestra son pocos los cuadros en los que se pintaba multitudes.

Podemos extraer una imagen de cada siglo para observar la diferencia entre las pinturas. Para esto recurrimos al dataframe original. Por ejemplo en el siglo 18.

Unnamed: 0	ARK	Object name/Title	Author	Date	Inventory number	Collection	fecha	personas	
1	1	cl010208742	Portrait de l'artiste et de son frère Achille	Devéria, Eugène ; France	1836	RF 2593; MV 6017	Département des Peintures	1836.0	2
2	2	cl010214786	Wa-ta-we-huch-a-nah (Le général commandant). I...	Catlin, George ; Amérique	1846	LP 6684	Département des Peintures	1846.0	1
3	3	cl010214788	Ah-wun-ne-wa-he (L'oiseau de tonnerre)	Catlin, George ; Amérique	1846	LP 6686	Département des Peintures	1846.0	1
5	5	cl010241037	Entrevue de Napoléon Ier et de la reine Louise...	Berthon, René Théodore ; France	1810	INV 2534; MV 8455; MR 1188	Département des Peintures	1810.0	15
8	8	cl010241164	Sujet tiré de l'histoire de Cromwell	Hippolyte Bruyères ; France	1832	INV 3087; LP 124; MV 6930	Département des Peintures	1832.0	11
...	...	...	...	...	...	...	...	...	
977	977	cl010065885	Le Soir, dit aussi Les Illusions perdues	Gleyre, Charles ; Suisse	1825 / 1850 (2e quart du XIXe siècle)	INV 10039	Département des Peintures	1825.0	0
980	980	cl010066329	Liseuse couronnée de fleurs ou La muse de Virg...	Corot, Jean-Baptiste Camille ; France	1845	RF 2599	Département des Peintures	1845.0	2
982	982	cl010066340	Paysage inspiré de la vue de Frascati	Michallon, Achille Etna ; France	1822	INV 6633; L 3853	Département des Peintures	1822.0	0
989	989	cl010066377	Intérieur au Mas-Bilier, près de Limoges.	Corot, Jean-Baptiste Camille ; France	1850 / 1875 (3e quart du XIXe siècle)	RF 1611	Département des Peintures	1850.0	11
994	994	cl010066422	Le singe peintre, dit aussi Intérieur d'atelier.	Decamps, Alexandre-Gabriel ; France	1849	RF 1375	Département des Peintures	1849.0	7

Figura 6: Estadísticas y numero de personas encontrados en pinturas en el siglo 18.

## 4. Conclusiones y Recomendaciones

### 4.1. Programa de extracción de imágenes de una URL dada

- A partir del siglo 15 comienzan a retratarse cuadros con más personas en promedio. Antes de este año no hay cuadros con 10 personas o más en la muestra del museo de Louvre.
- El software tiene dificultades para identificar personas que se encuentran en un segundo plano o que se encuentren en posiciones poco naturales, es por esto que en promedio el programa identifica menos personas de las que se pueden observar.
- La extracción de imágenes es un proceso que demora 10s por imagen lo que se traduce como un proceso que toma varias horas cuando es una muestra superior a las 500 imágenes.

### 4.2. Identificar las personas que se encuentran en cada imagen

- Los modelos que se usaron determinaron una aceptable identificación de personas, esto debido a que fueron entrenados con relativamente pocas imágenes, lo cual generaba que el patrón de coincidencias se tuviera que configurar a partir de 30 %.

- La lectura de las imágenes por parte del identificador sólo se podía generar exclusivamente sobre formatos definidos como jpg,png, etc y no sobre arrays que representaran la pintura.

## 5. Referencias

- ImageAI (v2.1.6) <https://github.com/OlafenwaMoses/ImageAI>,
- Museo del Louvre <https://collections.louvre.fr/en/page/apropos>
- Olafenwa,Moses.ImageAI : Object Detection. <https://github.com/OlafenwaMoses/ImageAI/blob/master/imageai/Detection/README.md#firstdetection>
- Beautiful Soup. <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>