# Práctica Imágenes - Detección de objetos

Pablo Mena

Beltrán Rodríguez-Mon

Para la elaboración de esta práctica hemos utilizado YOLOv3 para detectar imágenes de Tablas dentro de documentos.

Para realizar el entrenamiento primero de todo hemos inicializado el modelo de YOLO utilizando unos pesos predefinidos basados en la red **Darknet** (red CNN).

\*Dentro del repositorio de Github se ha omitido la carpeta con los datos de Training y con el modelo, para poder subir el resto.

\*\* El repositorio al completo con los datos están compartidos en esta [Carpeta de Drive](https://drive.google.com/drive/folders/1bEwwj1ibY11t8odqHARaTKmL_bzABeoI?usp=sharing) estando también parcialmente en [Github](https://github.com/beltranmon/noEstructurado)

## Pipeline

Para construir el modelo YOLO, los pasos seguidos han sido los siguientes:

1. Anotación de imágenes
   * Tag de las imágenes según (nombre-label-coordenadas)
   * Estos tags se almacenan en un csv dentro de Data/Source\_Images/Training/vott-csv-export
2. Entrenamiento
   * Descargar los pesos pre-entrenados
   * Entrenar el modelo YOLO custom para la detección de tablas
3. Inferencia/test
   * Detectar tablas en nuevas imágenes

## Repositorio

* 1\_Image\_Annotation: Anotación de imágenes
* 2\_Training: Scripts para el entrenamiento del modelo
* 3\_Inference: Scripts para el test
* Data: Input Data, Output Data y Pesos del modelo
* Utils: Scripts de utilidad

## Getting Started

Entorno Virtual **(Linux/Mac)**:

python3 -m venv env  
source env/bin/activate

Entorno Virtual **(Windows)**:

py -m venv env  
.\env\Scripts\activate

PowerShell

#### Instalación de los paquetes necesarios [Windows, Mac or Linux]

pip install -r requirements.txt

Si falla, puede ser que sea necesario ejecutar: pip install pip --upgrade.

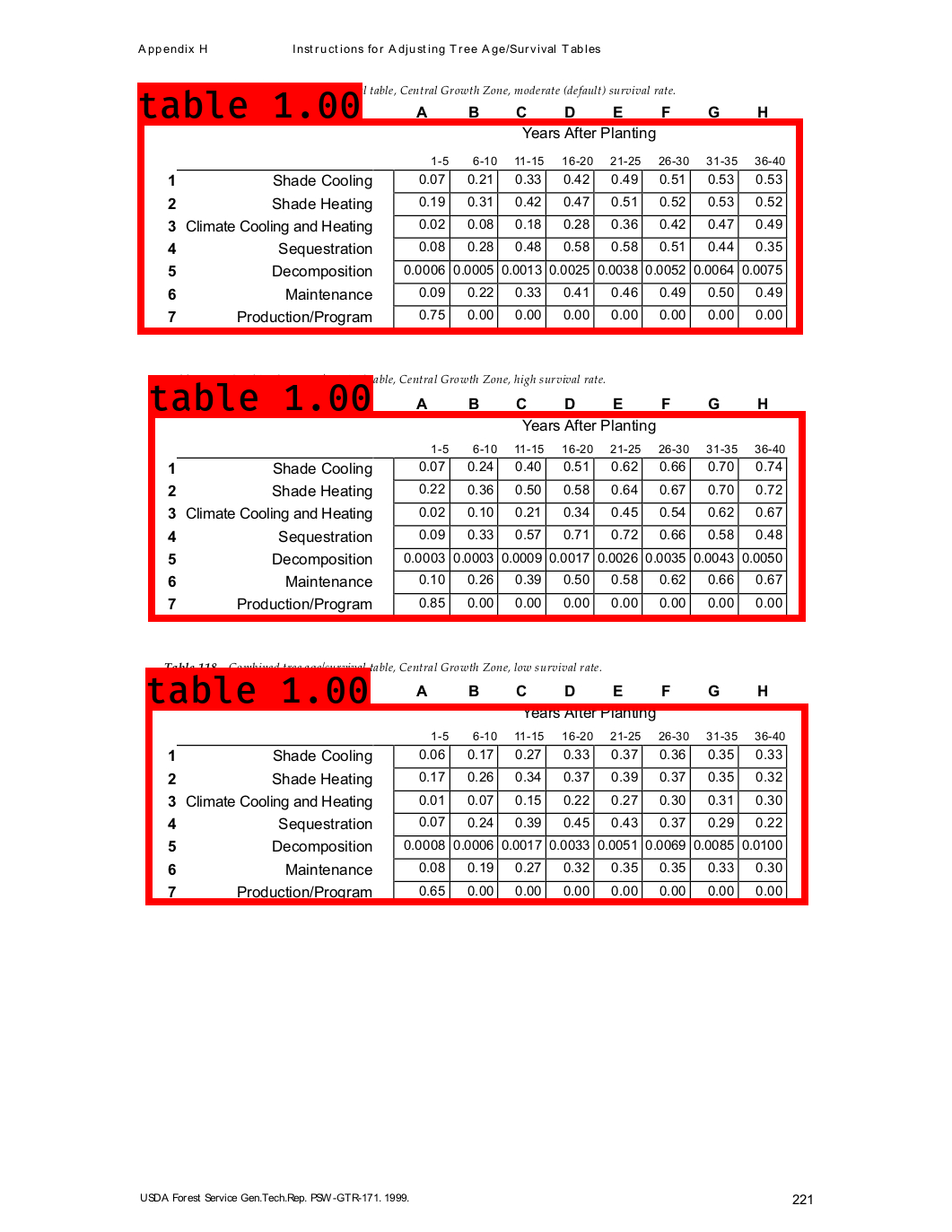
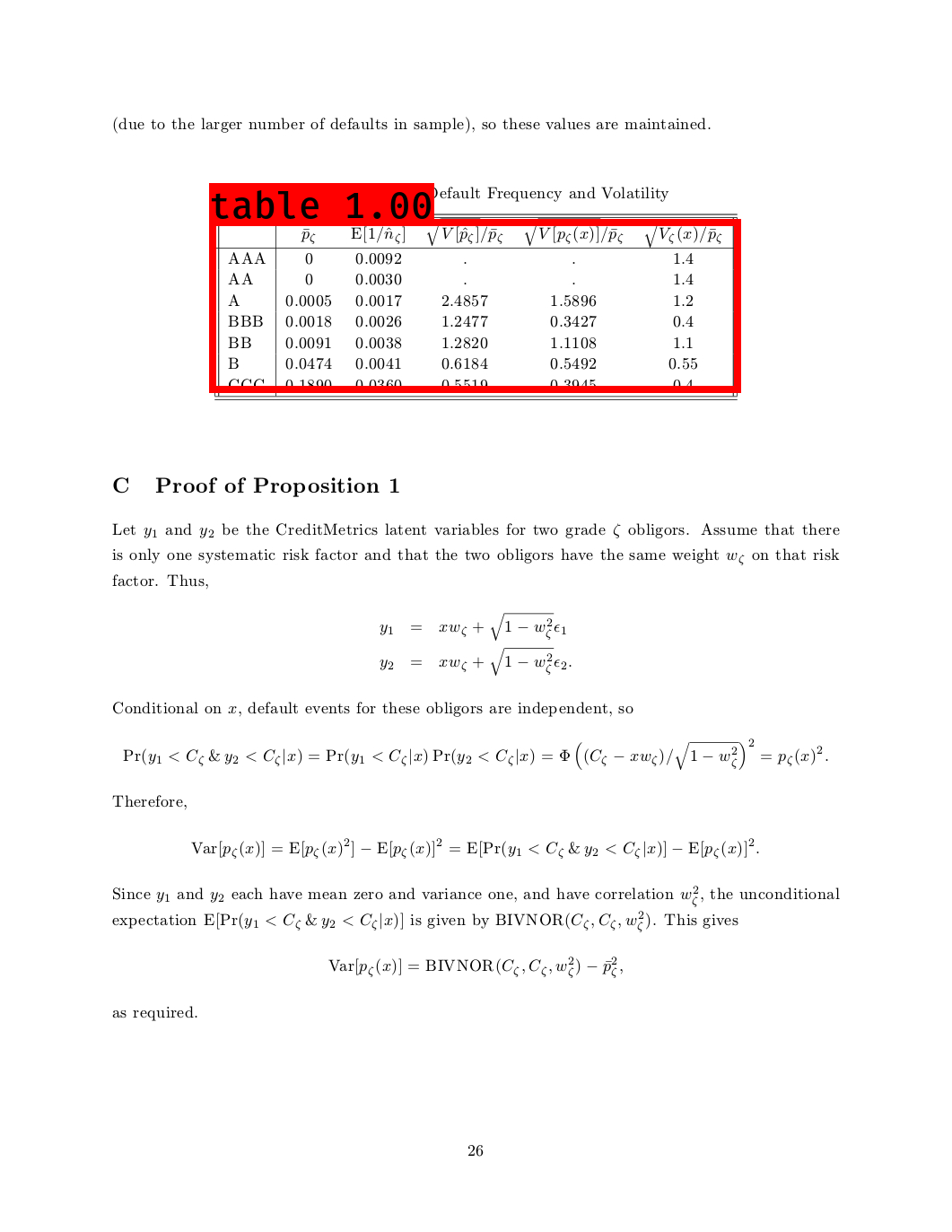
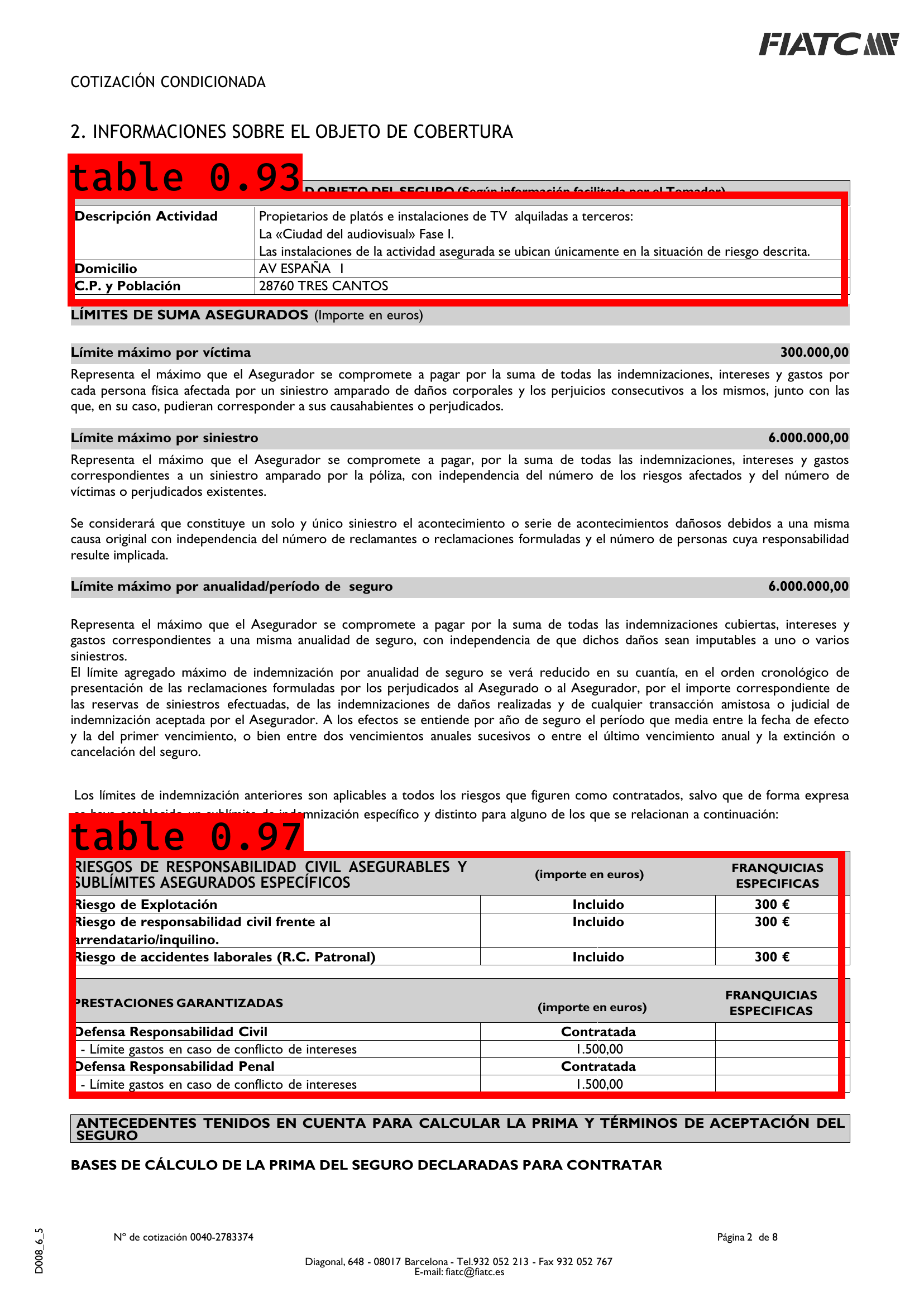
## Entrenamiento del modelo

El entrenamiento del modelo se ha realizado haciendo *Fine-Tuning* del modelo original. Los resultados han sido

45/45 [==============================] - 622s 14s/step - loss: 2072.1919 - val\_loss: 270.7197  
Epoch 2/51  
45/45 [==============================] - 517s 11s/step - loss: 189.7256 - val\_loss: 134.2318  
Epoch 3/51  
45/45 [==============================] - 516s 11s/step - loss: 110.2705 - val\_loss: 95.2559  
Epoch 4/51  
45/45 [==============================] - 535s 12s/step - loss: 78.6412 - val\_loss: 68.6279  
Epoch 5/51  
45/45 [==============================] - 620s 14s/step - loss: 61.8053 - val\_loss: 55.1787  
Epoch 6/51  
45/45 [==============================] - 649s 14s/step - loss: 50.5018 - val\_loss: 45.4948  
Epoch 7/51  
45/45 [==============================] - 676s 15s/step - loss: 43.4575 - val\_loss: 39.7827  
Epoch 8/51  
45/45 [==============================] - 674s 15s/step - loss: 38.1779 - val\_loss: 36.2439  
Epoch 9/51  
45/45 [==============================] - 647s 14s/step - loss: 34.1646 - val\_loss: 32.1615  
Epoch 10/51  
45/45 [==============================] - 681s 15s/step - loss: 31.5358 - val\_loss: 30.2518  
Epoch 11/51  
45/45 [==============================] - 625s 14s/step - loss: 28.8158 - val\_loss: 27.2826  
Epoch 12/51  
45/45 [==============================] - 629s 14s/step - loss: 27.4354 - val\_loss: 26.2810  
Epoch 13/51  
45/45 [==============================] - 553s 12s/step - loss: 25.5320 - val\_loss: 24.6966  
Epoch 14/51  
45/45 [==============================] - 550s 12s/step - loss: 24.4519 - val\_loss: 23.3203  
Epoch 15/51  
45/45 [==============================] - 615s 14s/step - loss: 23.3815 - val\_loss: 22.2603  
Epoch 16/51  
45/45 [==============================] - 625s 14s/step - loss: 22.4803 - val\_loss: 21.5432  
Epoch 17/51  
45/45 [==============================] - 632s 14s/step - loss: 21.6809 - val\_loss: 20.7737  
Epoch 18/51  
45/45 [==============================] - 622s 14s/step - loss: 20.9248 - val\_loss: 21.1336  
Epoch 19/51  
45/45 [==============================] - 640s 14s/step - loss: 20.3810 - val\_loss: 19.3579  
Epoch 20/51  
45/45 [==============================] - 646s 14s/step - loss: 19.8482 - val\_loss: 19.5318  
Epoch 21/51  
45/45 [==============================] - 635s 14s/step - loss: 19.3444 - val\_loss: 18.9209  
Epoch 22/51  
45/45 [==============================] - 591s 13s/step - loss: 18.8952 - val\_loss: 18.6198  
Epoch 23/51  
45/45 [==============================] - 545s 12s/step - loss: 18.7238 - val\_loss: 18.3770  
Epoch 24/51  
45/45 [==============================] - 525s 12s/step - loss: 18.2791 - val\_loss: 17.8375  
Epoch 25/51  
45/45 [==============================] - 509s 11s/step - loss: 18.0045 - val\_loss: 17.1076  
Epoch 26/51  
45/45 [==============================] - 506s 11s/step - loss: 17.6227 - val\_loss: 17.2048  
Epoch 27/51  
45/45 [==============================] - 510s 11s/step - loss: 17.3180 - val\_loss: 16.5084  
Epoch 28/51  
45/45 [==============================] - 513s 11s/step - loss: 17.2350 - val\_loss: 16.1463  
Epoch 29/51  
45/45 [==============================] - 514s 11s/step - loss: 16.9569 - val\_loss: 16.8566  
Epoch 30/51  
45/45 [==============================] - 516s 11s/step - loss: 16.8506 - val\_loss: 15.9374  
Epoch 31/51  
45/45 [==============================] - 525s 12s/step - loss: 16.6143 - val\_loss: 16.0619  
Epoch 32/51  
45/45 [==============================] - 523s 12s/step - loss: 16.3675 - val\_loss: 15.2876  
Epoch 33/51  
45/45 [==============================] - 514s 11s/step - loss: 16.2447 - val\_loss: 15.9445  
Epoch 34/51  
45/45 [==============================] - 581s 13s/step - loss: 16.1473 - val\_loss: 14.9769  
Epoch 35/51  
45/45 [==============================] - 478s 11s/step - loss: 15.9140 - val\_loss: 15.4020  
Epoch 36/51  
45/45 [==============================] - 471s 10s/step - loss: 15.8976 - val\_loss: 15.4180  
Epoch 37/51  
45/45 [==============================] - 472s 10s/step - loss: 15.7109 - val\_loss: 15.4537  
Epoch 38/51  
45/45 [==============================] - 472s 10s/step - loss: 15.7216 - val\_loss: 15.3927  
Epoch 39/51  
45/45 [==============================] - 471s 10s/step - loss: 15.5720 - val\_loss: 15.3672  
Epoch 40/51  
45/45 [==============================] - 473s 11s/step - loss: 15.4707 - val\_loss: 14.6734  
Epoch 41/51  
45/45 [==============================] - 472s 10s/step - loss: 15.3267 - val\_loss: 14.9112  
Epoch 42/51  
45/45 [==============================] - 471s 10s/step - loss: 15.2894 - val\_loss: 14.7180  
Epoch 43/51  
45/45 [==============================] - 471s 10s/step - loss: 15.2245 - val\_loss: 14.7386  
Epoch 44/51  
45/45 [==============================] - 474s 11s/step - loss: 15.1195 - val\_loss: 14.4612  
Epoch 45/51  
45/45 [==============================] - 473s 11s/step - loss: 15.0545 - val\_loss: 14.4236  
Epoch 46/51  
45/45 [==============================] - 472s 10s/step - loss: 15.0266 - val\_loss: 14.6731  
Epoch 47/51  
45/45 [==============================] - 472s 10s/step - loss: 14.9318 - val\_loss: 14.3230  
Epoch 48/51  
45/45 [==============================] - 472s 10s/step - loss: 14.8995 - val\_loss: 14.3872  
Epoch 49/51  
45/45 [==============================] - 472s 10s/step - loss: 14.8085 - val\_loss: 14.6400  
Epoch 50/51  
45/45 [==============================] - 471s 10s/step - loss: 14.7438 - val\_loss: 14.6279  
Epoch 51/51  
45/45 [==============================] - 474s 11s/step - loss: 14.7772 - val\_loss: 14.2174

## Resultados

Las salidas del modelo están almacenadas en Data/Source\_Images/Test\_Image\_Detection\_Results. Dentro de este directorio hay varias imágenes para mostrar la efectividad del modelo.



#### Testing del Detector

Para detectar nuevas tablas es necesario ejecutar el Script de 3\_Inference:

python Detector.py

La salida se almacena en Data/Source\_Images/Test\_Image\_Detection\_Results

## Conclusión

El modelo funciona bastante bien, especialmente considerando que el entrenamiento del modelo original (Darknet) no se ha realizado con un dataset mínimamente semejante al objetivo de nuestro enfoque (identificar tablas en documentos).