**Steps and Links of Table-Transformer Model.**

In PubTables-1M: Towards complete table extraction from unstructured sources, authors Brandon Smock, Rohith Pesala, and Robin Abraham put forth the Table Transformer concept. To measure advancements in table extraction from unstructured documents, table structure identification, and functional analysis, the authors provide PubTables-1M, a brand-new dataset. The authors develop two DETR models known as Table Transformers, one for table identification and one for table structure recognition.

**Model github repo is in** : [microsoft/table-transformer: Model training and evaluation code for our dataset PubTables-1M, developed to support the task of table extraction from unstructured documents. (github.com)](https://github.com/microsoft/table-transformer)

* It contains two folder detr and src where we have to use src folder to train for both Table Detection and table Structure Recognition. Where the table detection code will detect the whole table with confidence label and with the coordinates of the bounding box around the table. And each rows and columns for the Table structure with the confidence label and bounding box for each row and column, altogether it will show a comprehensive structure.
* And the detr folder contains the detectron model which will help us to detect the tables, which acts as a wrapper for this object detection model.

It has 575,305 pages of annotated documents that have tables on them for table detection. 947,642 completely annotated tables with comprehensive location (bounding box) data and text content for identifying table structure and performing functional analysis. All table rows, columns, and cells (including empty cells), as well as other marked structures like column headers and projected row headers, have complete bounding boxes in image and PDF coordinates.

All tables and pages were rendered as pictures. For every word that appears in each table and page graphic, there are bounding boxes and text. Additional cell attributes weren't employed in the model's training at this time. To ensure the annotations are as noise-free as feasible, they had also canonicalized cells in the headers and used a number of quality check procedures. For further details you can check in there given paper : [2110.00061.pdf (arxiv.org)](https://arxiv.org/pdf/2110.00061.pdf)

* Also they had provided the weights of the pretrained models for both Table detection and Table structure detection which can be used to train on our new data.
* They trained the models for 20 epochs only.

You can change the number of epochs in the structure.config and dectron.config file at the src folder. And also can reduce the size of the data at the text files of the table structure and pdf annotation dataset folder for training purpose.

For the evaluation metrices used in the table transformer researcher had focused on the GriTs method, [2203.12555.pdf (arxiv.org)](https://arxiv.org/pdf/2203.12555.pdf)

To download the dataset please follow the following link : [Microsoft Research Open Data (msropendata.com)](https://msropendata.com/datasets/505fcbe3-1383-42b1-913a-f651b8b712d3)

**Steps to Train and Test the Model.**

First we need to create the environment if it’s not there in the conda environment list. The environment name will be tables-detr.

**Step 1 :** Check whether the env is there or not by using cmd in linux terminal :

‘**conda env list** ‘ where env name will be : **tables-detr**

If the environment is not installed then we have to create the environment by going to this particular path which is : **cd SageMaker/ table-transformer/**

Then run this command to create the environment : **conda env create -f environment.yml**

**Step 2 :** Then activate the env using : **source activate tables-detr**

**Step 3 :** After activation go to the folder where main.py is present :

**cd SageMaker/table-transformer/src/**

**Step 4 :** Then run this command to train the table structure model :

(tables-detr2) C:\Users\malvi\ML\_project\table-transformer-main\src>python main.py --data\_type structure --config\_file structure\_config.json --data\_root\_dir C://Users//malvi//Desktop//Main//PhD//course\_work//ML//Project//data//pubTable//PubTables-1M-Image\_Table\_Structure\_PASCAL\_VOC//PubTables1M-Structure-PASCAL-VOC

And

Then run this command to train the table detection model :

(tables-detr2) C:\Users\malvi\ML\_project\table-transformer-main\src>python main.py --data\_type detection --config\_file detection\_config.json --data\_root\_dir C://Users//malvi//Desktop//Main//PhD//course\_work//ML//Project//data//pubTable//PubTables-1M-Image\_Page\_Detection\_PASCAL\_VOC//PubTables1M-Detection-PASCAL-VOC

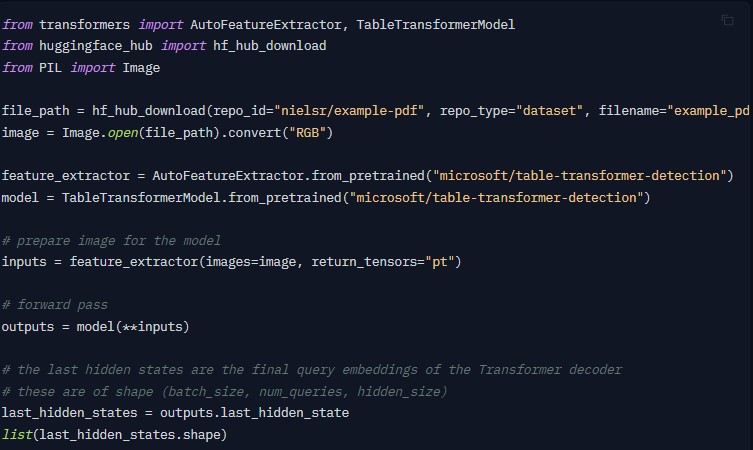
Then the training will start and the model weights will get store in the output folder inside the PubTables1M-Structure-PASCAL-VOC folder. We can use this model in the inference stage to test on top of our new data.

**Huggingface Table-Transformer.**

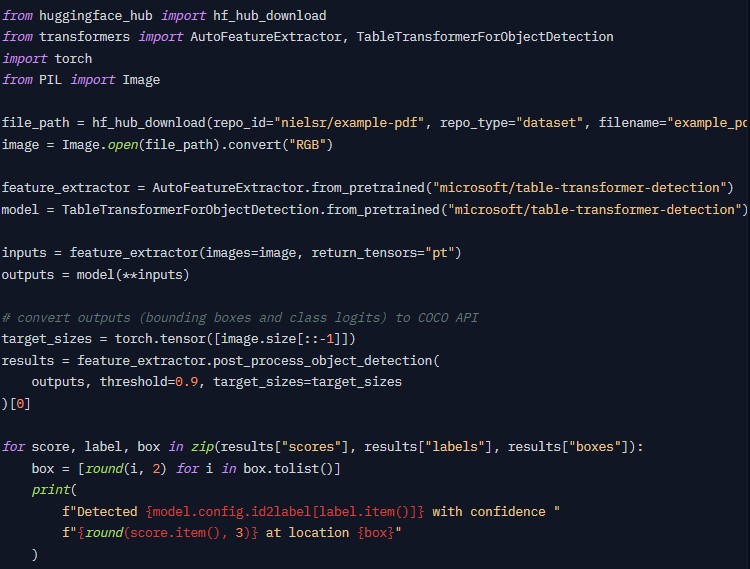
Hugging face has inferenced the Table Transformer model by using the pre-trained models for both Table Detection and Table Structure recognition.

* Two models were published by the authors: one for document table detection and the other for table structure recognition (the task of recognising the individual rows, columns etc. in a table).
* In order to prepare photos and optional targets for the model, one can utilize the AutoFeatureExtractor API ([Auto Classes (huggingface.co)](https://huggingface.co/docs/transformers/main/en/model_doc/auto#transformers.AutoFeatureExtractor)). A DetrFeatureExtractor ([DETR (huggingface.co)](https://huggingface.co/docs/transformers/main/en/model_doc/detr#transformers.DetrFeatureExtractor)) will be loaded in the background as a result.

**TableTransformerModel**



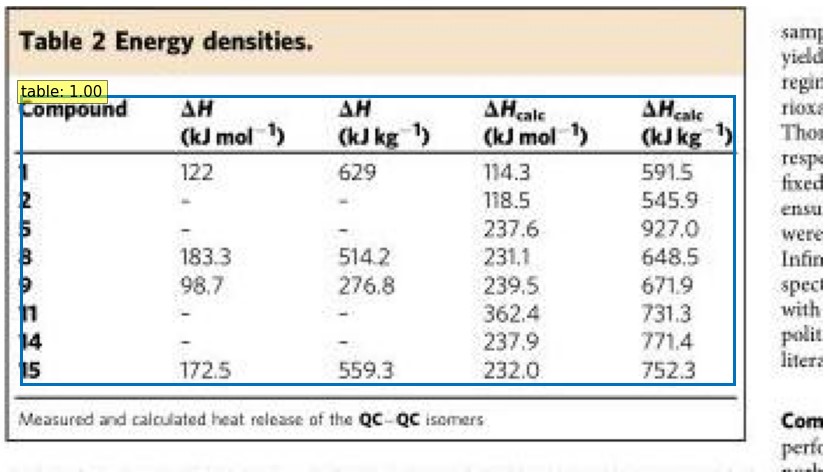
**TableTransformerForObjectDetection**

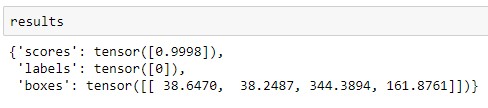
****

\*Before running the codes we need to install the dependencies : [Dependencies\_Installation (sagemaker.aws)](https://beazley-gpu.notebook.eu-west-2.sagemaker.aws/notebooks/Devmallya/Dependencies_Installation.ipynb)

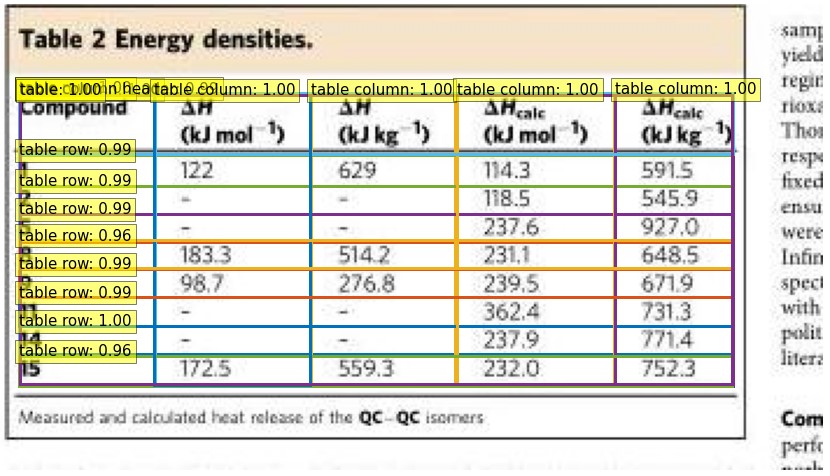
Our Table Transformer inference model is present at : [Table\_Transformer\_Beazley\_data (sagemaker.aws)](https://beazley-gpu.notebook.eu-west-2.sagemaker.aws/notebooks/Devmallya/Table_Transformer_Beazley_data.ipynb)

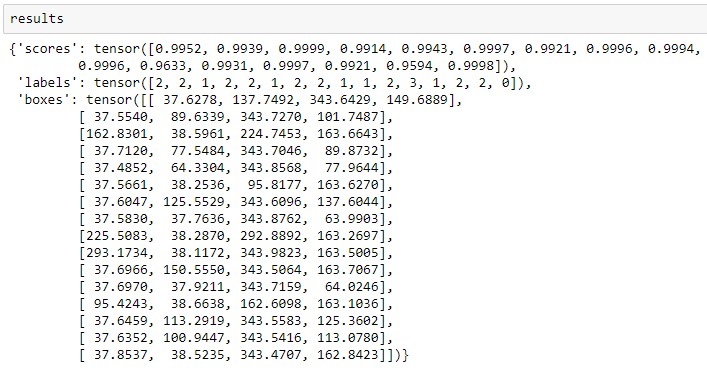
**Table Detection Output**

****

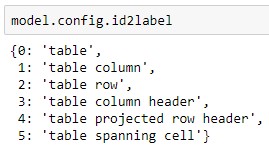
****

**Table Structure Recognition Output**

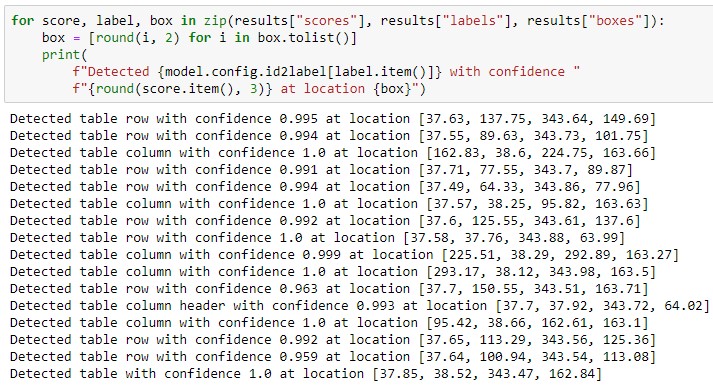
****

****

**Number of Tables and Columns**

****

**Confidence score and Bounding Boxes Coordinates**

****