The relevant details from an invoice in commercial applications certainly involves a lot of high spec algorithms. They **identify relevant parts first and extract the details afterwards.**

However, my first starting point would be to get all the text from an invoice (e.g. via tesseract). If you have a decent photo, tesseract will be able to OCR the content. The next step would be to identify relevant content, such as payment amount, names, and bank account numbers. This may be possible by hardcoded rules to some extent. Alternatively, one could use NLP-like models to detect certain sequences. With some effort, this should work out well since invoices are relatively structured documents.

Mask RCNN

<https://towardsdatascience.com/instance-segmentation-using-mask-r-cnn-7f77bdd46abd>

<https://github.com/matterport/Mask_RCNN>

<https://github.com/matterport/Mask_RCNN/blob/master/samples/shapes/train_shapes.ipynb>

<https://medium.com/@nandumpilicode/how-anyone-having-a-pc-with-no-gpu-can-train-custom-object-detection-dataset-within-1-hour-74a7b7b7fd4e>

<https://www.robots.ox.ac.uk/~vgg/software/via/via_demo.html>

<https://towardsdatascience.com/building-a-custom-mask-rcnn-model-with-tensorflow-object-detection-952f5b0c7ab4>

Article

<https://www.pyimagesearch.com/2018/11/19/mask-r-cnn-with-opencv/>

 Through my experimentation, I observed that training a Mask RCNN model requires fewer images than training a Faster RCNN model to get to the same accuracy.

# **Custom objects**

<https://towardsdatascience.com/building-a-custom-mask-rcnn-model-with-tensorflow-object-detection-952f5b0c7ab4>

<https://github.com/priya-dwivedi/Deep-Learning/tree/master/Custom_Mask_RCNN>

\*\*\*\*\*\*

<https://github.com/RomRoc/maskrcnn_train_tensorflow_colab>

<https://github.com/RomRoc/maskrcnn_train_tensorflow_colab/blob/master/maskrcnn_custom_tf_colab.ipynb>

Mask - RCNN

You don't necessarily have to resize it before hand.

you can use this option in the model config file to set the size limit for your training.

image\_resizer {

keep\_aspect\_ratio\_resizer {

min\_dimension: 600

max\_dimension: 1024

}

}

Please make sure all the bounding boxes are in range with the image dimensions. i.e. the within the range of width and height of the image. Then the boxes and the images will be auto resized according to the parameter set here

Tutorial

<https://www.linkedin.com/pulse/mask-rcnn-custom-data-set-idly-vada-dosa-abhilash-reddy-yerasi/>

# **GOOGLE COLLAB**

<https://github.com/vijendra1125/Custom-Mask-RCNN-using-Tensorfow-Object-detection-API>

<https://medium.com/@vijendra1125/custom-mask-rcnn-using-tensorflow-object-detection-api-101149ce0765>

Image resize py

<https://www.dlology.com/blog/how-to-train-an-object-detection-model-with-mmdetection/>

1. Pixel annotation tool via – for labelling

SSD – Mobile net for custom object

<https://pythonprogramming.net/training-custom-objects-tensorflow-object-detection-api-tutorial/>

To crop

<https://github.com/sugi-chan/clothes_segmentation/blob/master/inspect_fashion%20eval_examples.ipynb>

rescale ROIs

<https://www.pyimagesearch.com/2018/11/19/mask-r-cnn-with-opencv/>

<https://stackoverflow.com/questions/51220865/resize-bounding-box-according-to-image>

<https://stackoverflow.com/questions/38601285/finding-new-position-x-y-after-resizing-image>

Class IDs in mask RCNN

<https://github.com/matterport/Mask_RCNN/issues/372>

\*\*

<https://github.com/SUYEgit/Surgery-Robot-Detection-Segmentation/blob/master/surgery.py>

<https://github.com/matterport/Mask_RCNN/issues/372>

Docker GPU here

<https://medium.com/innovation-machine/deploying-object-detection-model-with-tensorflow-serving-part-2-55e6c638822e?>

Convert .h5 to tensorflow

<https://towardsdatascience.com/deploying-keras-models-using-tensorflow-serving-and-flask-508ba00f1037>

<https://hackernoon.com/how-we-improved-tensorflow-serving-performance-by-over-70-f21b5dad2d98>

<https://towardsdatascience.com/deploying-keras-models-using-tensorflow-serving-and-flask-508ba00f1037>

<https://www.dlology.com/blog/how-to-convert-trained-keras-model-to-tensorflow-and-make-prediction/>

<https://ngc.nvidia.com/catalog/containers/nvidia:tensorflow/tags>

<https://github.com/bendangnuksung/mrcnn_serving_ready>

# **Data Augmentation and hyper parameter tuning**

<https://www.kaggle.com/c/data-science-bowl-2018/discussion/53495>

import imgaug # https://github.com/aleju/imgaug (pip3 install imageaug)

augmentation = imgaug.augmenters.Fliplr(0.5)

model.train(dataset\_train, dataset\_val,

learning\_rate=config.LEARNING\_RATE,

epochs=40,

layers='heads',

augmentation=augmentation)

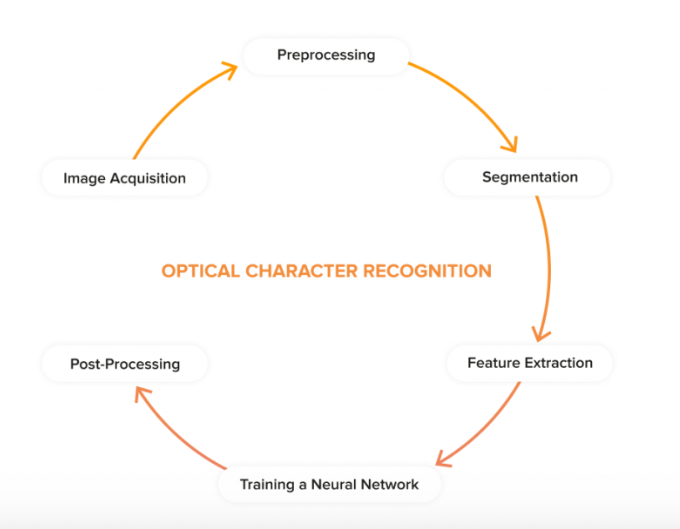
<https://github.com/matterport/Mask_RCNN/issues/281>

## **Scores**

<https://github.com/fastlater/DSB_2018_Parameter_Selection_For_Matterport_Mask_RCNN/blob/master/comparison_of_Matterport_Mask_RCNN_parameters.md>

Optimize training with hyper parameters

\*\* <https://github.com/matterport/Mask_RCNN/wiki>



CNN - depth knowledge

https://brohrer.github.io/how\_convolutional\_neural\_networks\_work.html