Implementasi Mask RCNN

Menggunakan Python3, Keras, dan Tensorflow

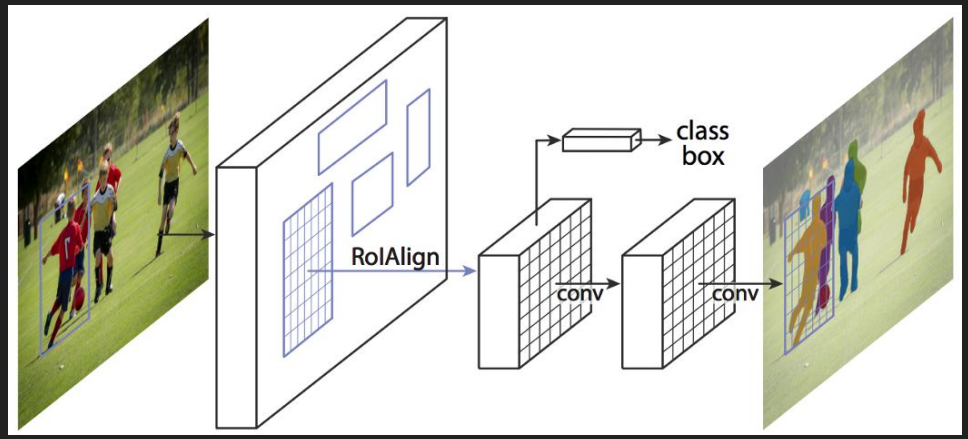
https://github.com/tomytw/Mask-RCNN-TF2.0

C14180006 - Tomy Widjaja
C14180029 - Andrew Firman
C14180046 - Gregorius Nicholas Goenawan

Apa itu Mask RCNN

Mask RCNN adalah framework Deep Neural Network yang terdiri dari dua stage:

1. Scan Image dan Generate Proposal
2. Klasifikasi Proposal dan Generate Bounding Box & Mask



Modul di Mask RCNN

Di Mask RCNN terdiri dari 3 modul utama:

1. Backbone = backbone yang digunakan Resnet101 yang berguna sebagai feature extractor. Layers bagian awal ini diharapkan dapat mengekstraksi low level features (seperti edges dan corners)
   Terdapat FPN (Feature Pyramid Network) di atas Backbone yang berfungsi untuk merepresentasikan objek di banyak skala.
2. Region Proposal Network: Light weight neural network yang scan image secara “sliding window” atau bergeser.
3. ROI Classifier & Bounding Box Regressor -> Mengeluarkan output Class (FG/BG) & Bounding Box Refinement.

Langkah - langkah Instalasi (di laptop/pc pribadi)

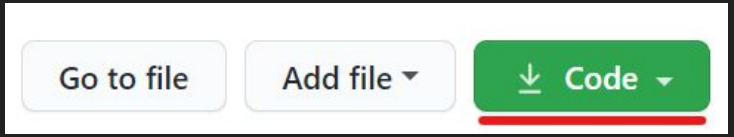
1. Install modul-modul berikut (disarankan di environment baru):

```
numpy
scipy
Pillow
cython
matplotlib
scikit-image
tensorflow==2.0.0
keras==2.3.1
opencv-python
h5py
imgaug
IPython[all]
```

Requirements di atas bisa di save ke file ‘requirements.txt’
Jalankan: pip3 install -r requirements.txt

Langkah - langkah Instalasi

2. Download repository ‘https://github.com/matterport/Mask_RCNN’



3. Download pre-trained COCO weights (mask_rcnn_coco.h5) = https://github.com/matterport/Mask_RCNN/releases/download/v1.0/mask_rcnn_coco.h5 (letakkan file di root folder repository)

4. Jalankan `pip install git+https://github.com/philferriere/cocoapi.git#subdirectory=PythonAPI`

5. Install Visual C++ 2015 Build Tools: https://go.microsoft.com/fwlink/?LinkId=691126

Langkah - langkah Instalasi

6. Ubah beberapa file di bawah untuk menjalankan demo (penyesuaian dengan tensorflow versi > 2 karena di repository menggunakan tensorflow 1.x)

- Perubahan di file mrcnn/model.py:

- tf.log() → tf.math.log()
- tf.sets.set_intersection() → tf.sets.intersection()
- tf.sparse_tensor_to_dense() → tf.sparse.to_dense()
- tf.to_float() → tf.cast([value], tf.float32)
- indices = tf.stack([tf.range(probs.shape[0]), class_ids], axis=1) →
  indices = tf.stack([tf.range(tf.shape(probs)[0]), class_ids], axis = 1)

Langkah - langkah Instalasi

- Perubahan di file mrcnn/model.py:

- mrcnn_bbox = KL.Reshape((s[1], num_classes, 4), name="mrcnn_bbox")(x) →
  if s[1]==None:
  mrcnn_bbox = KL.Reshape((-1, num_classes, 4), name="mrcnn_bbox")(x)
  else:
  mrcnn_bbox = KL.Reshape((s[1], num_classes, 4), name="mrcnn_bbox")(x)
- from keras.engine import saving → from tensorflow.python.keras.saving import hdf5_format
- saving.load_weights_from_hdf5_group_by_name(f, layers) → hdf5_format.load_weights_from_hdf5_group_by_name(f, layers)
- saving.load_weights_from_hdf5_group(f, layers) → hdf5_format.load_weights_from_hdf5_group(f, layers)

Langkah - langkah Instalasi

- Perubahan di file mrcnn/model.py (untuk menjalankan training):

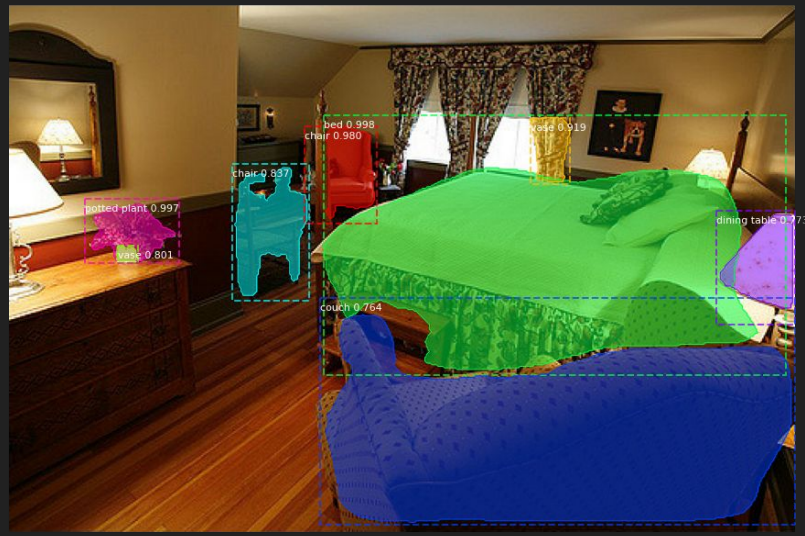
  - tf.random_shuffle → tf.random.shuffle
  - Di comment:
    # if layer.output in self.keras_model.losses:
    #continue
  - Tambah di bagian awal method compile():
    self.keras_model.metrics_tensors = []

- Perubahan di file mrcnn/util.py (untuk menjalankan training):

  - tf.log → tf.math.log

Demo Program (Melihat Hasil Deteksi)

- Demo Mask RCNN dilakukan dengan menggunakan pretrained model mask_rcnn_coco.h5 (Model yang sudah di train menggunakan MS COCO Dataset) yang dapat mendeteksi 80 kelas / objek



Langkah- Langkah Train Model dengan Custom Dataset(Ringkas)

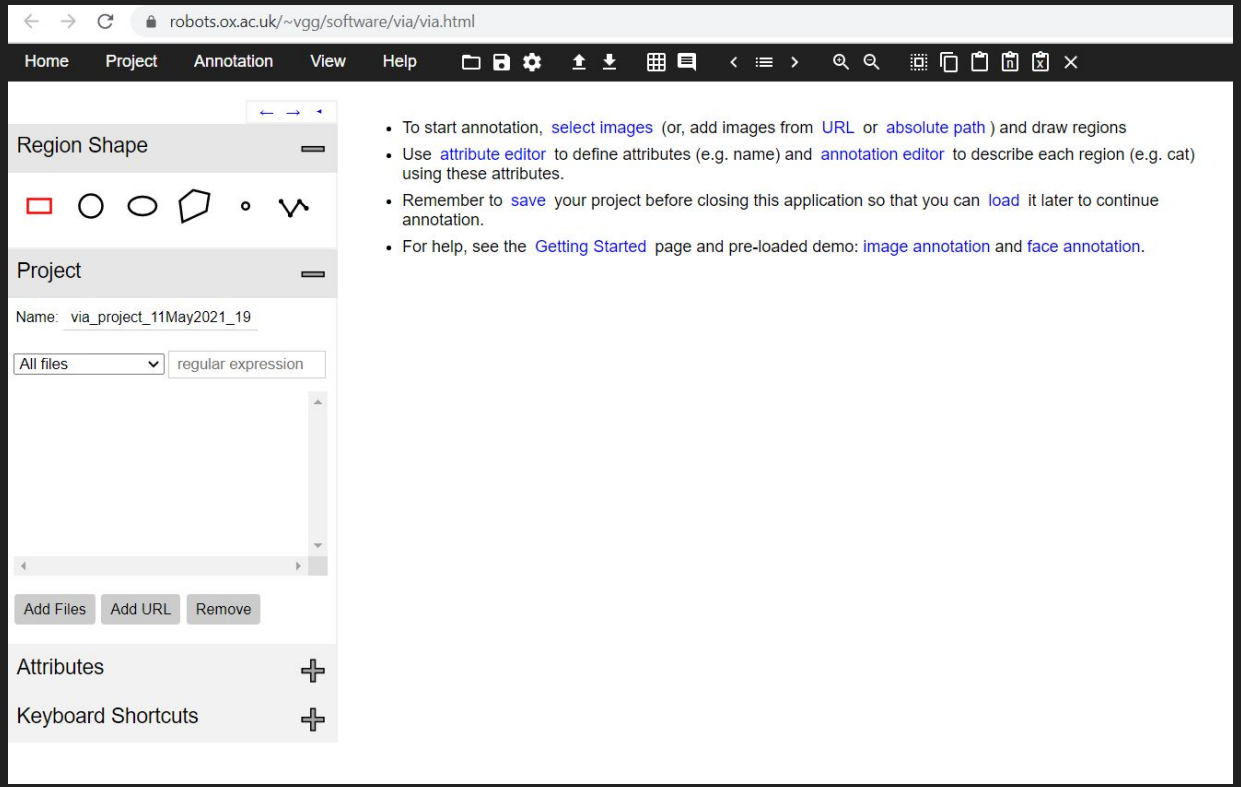
1. Membutuhkan GPU Power yang kuat karena backbone Resnet101 (disarankan training di Colab)
2. Menentukan Config
3. Jika menggunakan dataset custom, maka perlu melakukan labeling dan masking menggunakan VIA Annotator
4. Siapkan Dataset-> Training Dataset and Validation dataset
5. Create Model -> pilih pre trained weights (coco atau imagenet)
6. Train

Langkah- Langkah Implementasi ‘Splash’ pada Model Balloon

1. Mengubah config menjadi mode ‘inference’
2. Load pre trained weight dari dataset balloon -> mask_rcnn_baloon.h5
3. Tentukan path gambar yang mau di splash ke image_path dan tentukan juga output path sebagai lokasi output hasil inference
4. Jalankan test-balloon.ipynb
5. Hasil ‘Splash’ akan di save di directory output path yang kita tentukan

Contoh hasil ‘Splash’ dengan weight balloon

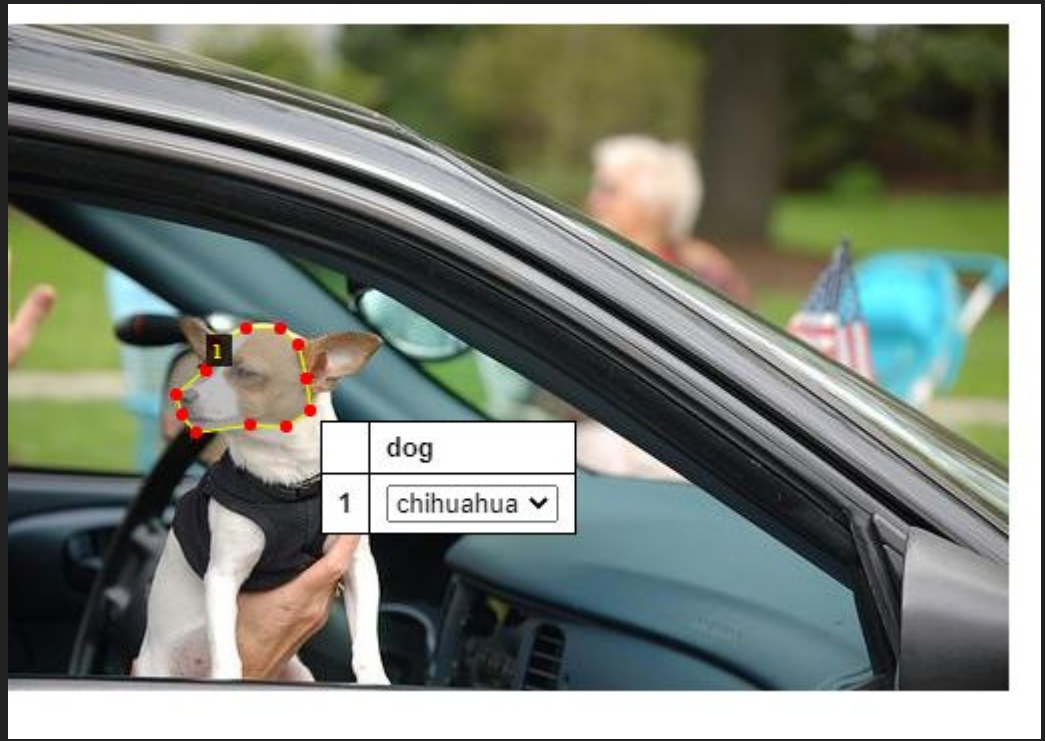
VIA Usage VGG Image Annotation



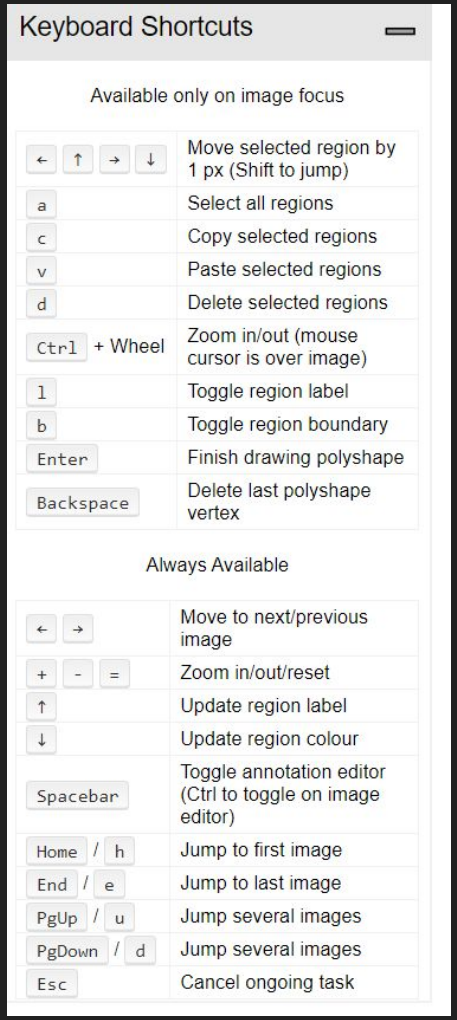
1. Add Files
2. Pilih Region shape, pilih yang paling kanan.
3. Lalu Start Drawing Point-point
4. ESC> cancel, backspace> rm last,

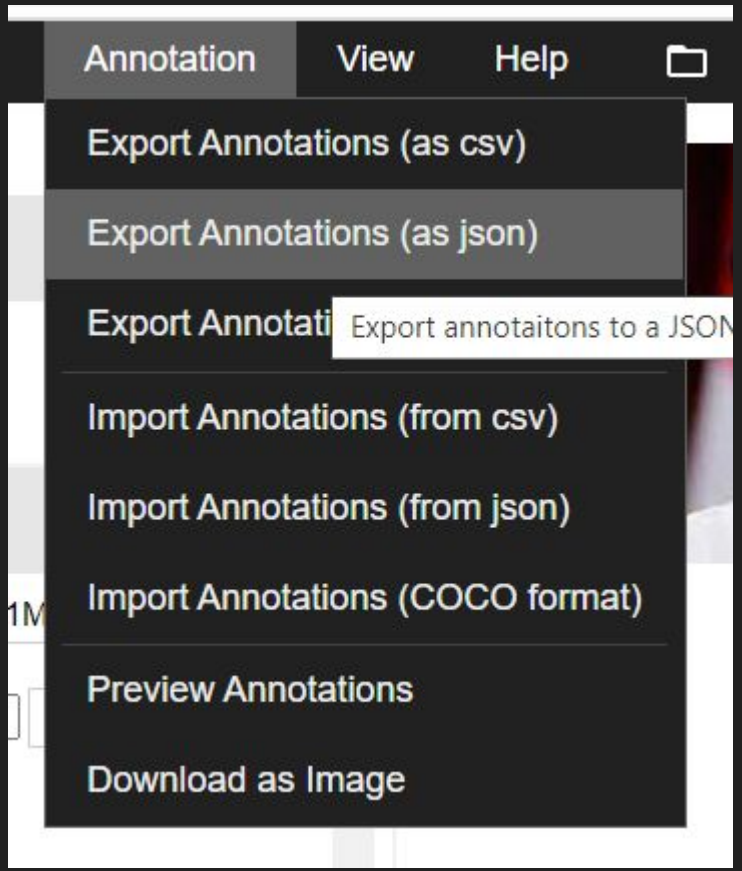
   Enter> Finish

Contoh: Chihuahua



Keyboard Shortcut(ext)





Saat sudah selesai, Klik Menu ‘Annotation’ pada Menu bar, lalu pilih Export Annotations (as JSON)

Thank you