THE DEVELOPER'S CONFERENCE

Identificando árvores e rede elétrica com o Tensorflow

Tamara Mendes

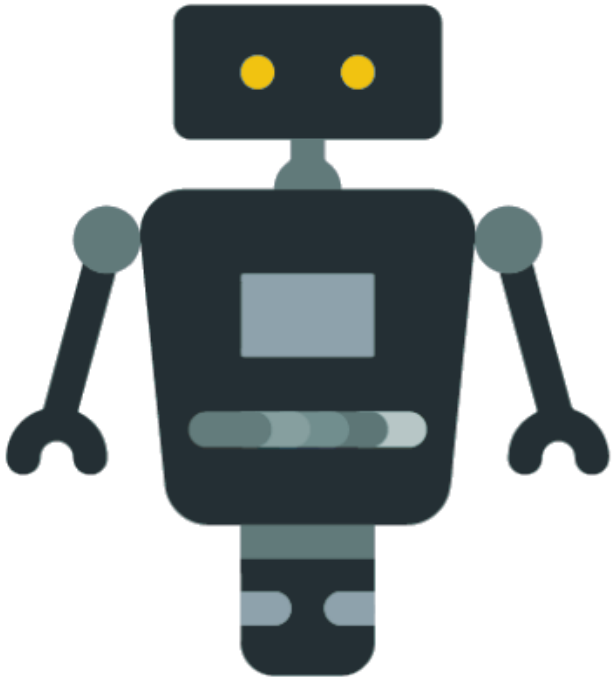
CONCERT TECHNOLOGIES

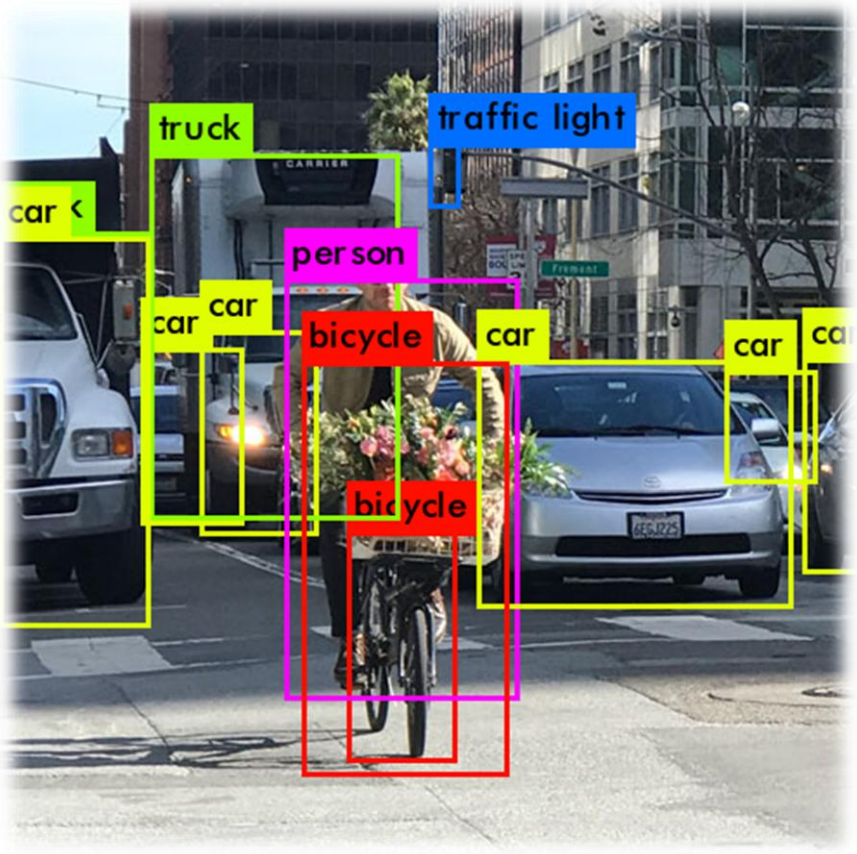
Contexto

FECHE
CRUZAMENTO

Gerenciar ativos de rede

Automatizar identificação de elementos de rede e árvores




truck
traffic light
car
person
car
car
bicycle
car
car
bicycle

TensorFlow

Classificação de imagens

Classificar

Gato

Classes

Classificar

Gato

Classificar

e localizar

Gato

Classes

Classificar

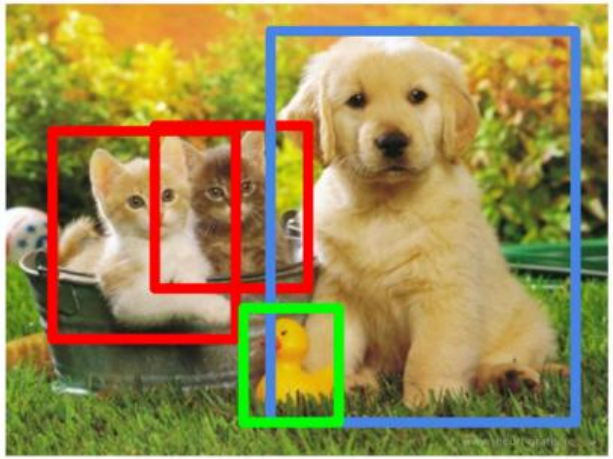
Gato

Classificar

e localizar

Gato

Detectar múltiplos objetos

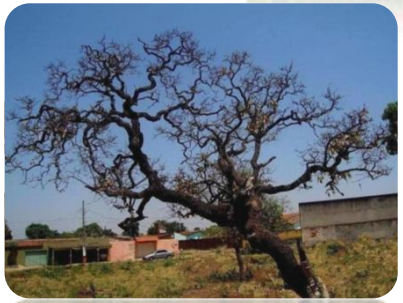


Gato, cachorro, pato

Classes

Classificação de espécies

No mínimo 20 imagens para cada classe



COQUEIRO


coqueiro-grande (24).jpg


coqueiro (19).jpg


coqueiro (28).jpg

MANGUEIRA


mangueira-grande (9).jpg


mangueira (10).jpg


mangueira-grande (10).jpg

OITI


oiti-grande


oiti (33).jpg


oiti (31).jpg

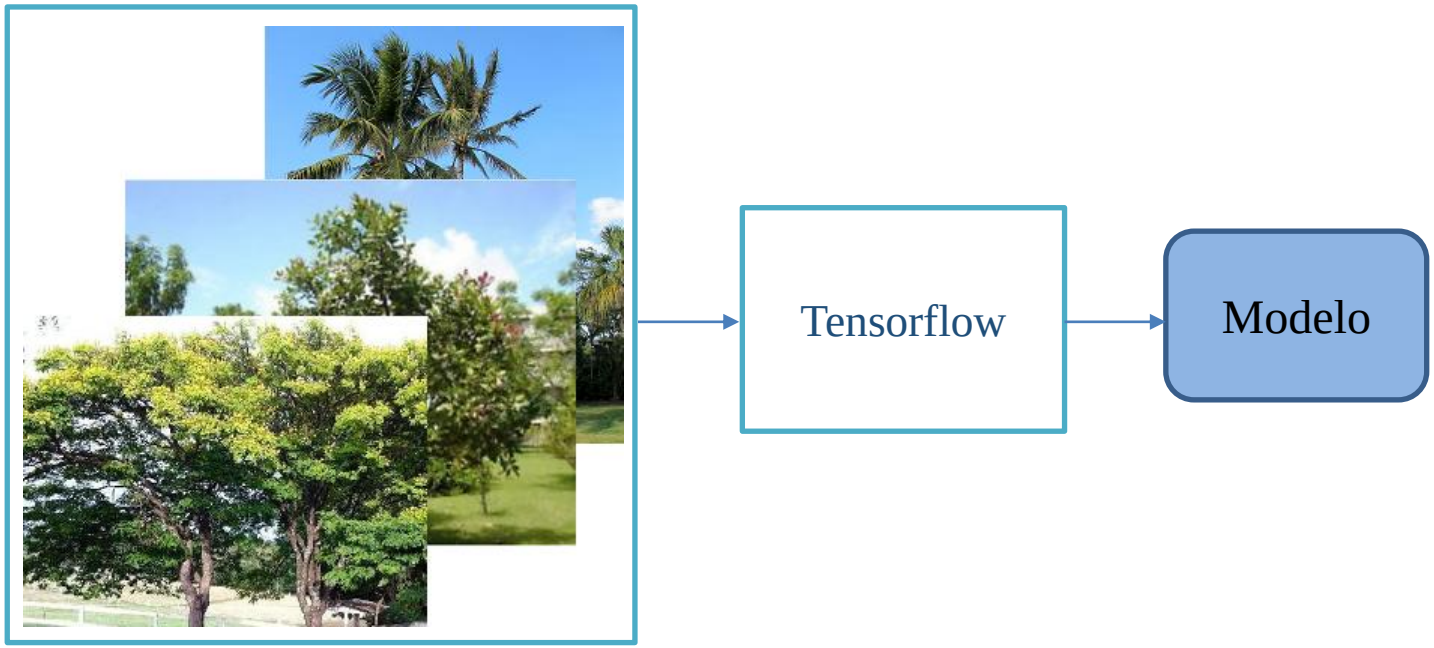
SIBIPIRUNA


sibipiruna-grande (39).jpg


sibipiruna-grande (25).jpg


sibipiruna (36).jpg

Treinar o modelo



Treinamento

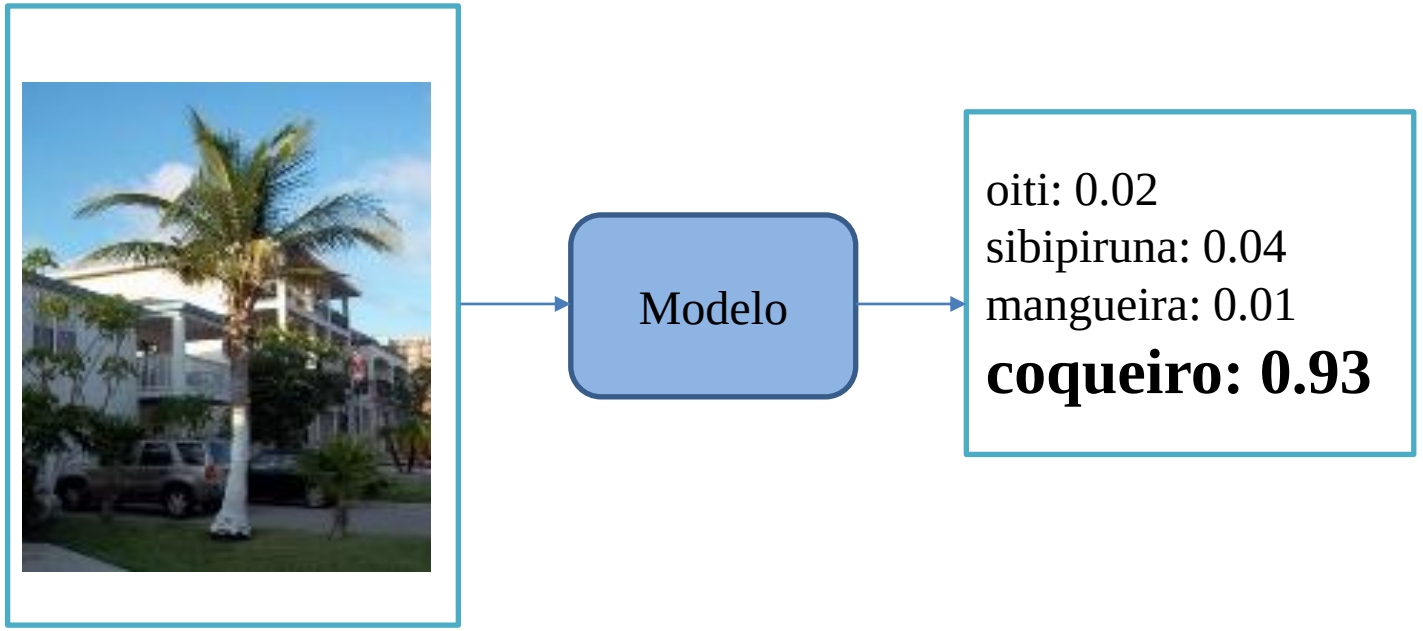
https://github.com/tensorflow/hub/raw/master/examples/image_retraining/retrain.py

---

Teste

https://github.com/tensorflow/tensorflow/raw/master/tensorflow/examples/label_image/label_image.py

Classificar as árvores



Coqueiro: 70,11%

Mangueira: 15,37%

Sibipiruna: 14,20%

Oiti: 0,29%

Mangueira: 99,65%

Oiti: 0,19%

Sibipiruna: 0,09%

Coqueiro: 0,05%

Sibipiruna: 99,73%

Coqueiro: 0,17%

Oiti: 0,059%

Mangueira: 0,031%

Mangueira: 78,48

Oiti: 21,11%

Sibipiruna: 0,29%

Coqueiro: 0,11%

Detecção de objetos

Instalação


```
conda create -n nome_seu_ambiente python=3.6
conda activate nome_seu_ambiente
```

CPU

```
conda install tensorflow
```

GPU

- GPU compatível
- Drivers necessários

```
conda install tensorflow-gpu
```

CPU

```
conda install tensorflow
```

GPU

- GPU compatível
- Drivers necessários

Google Cloud

```
conda install tensorflow-gpu
```

Pacotes necessários

```
conda install pillow lxml jupyter matplotlib
  opencv contextlib2 Cython pandas
```

Pacotes necessários

```
conda install pillow lxml jupyter matplotlib
  opencv contextlib2 Cython pandas
```

Testar a instalação

```
python
import tensorflow as tf
hello = tf.constant('Hello, TensorFlow!')
sess = tf.Session()
print(sess.run(hello))
```

https://github.com/tensorflow/models/

Models and examples built with TensorFlow

- Setar variáveis de ambiente
- Baixar e compilar a biblioteca protobuf
- Setar PATH do python

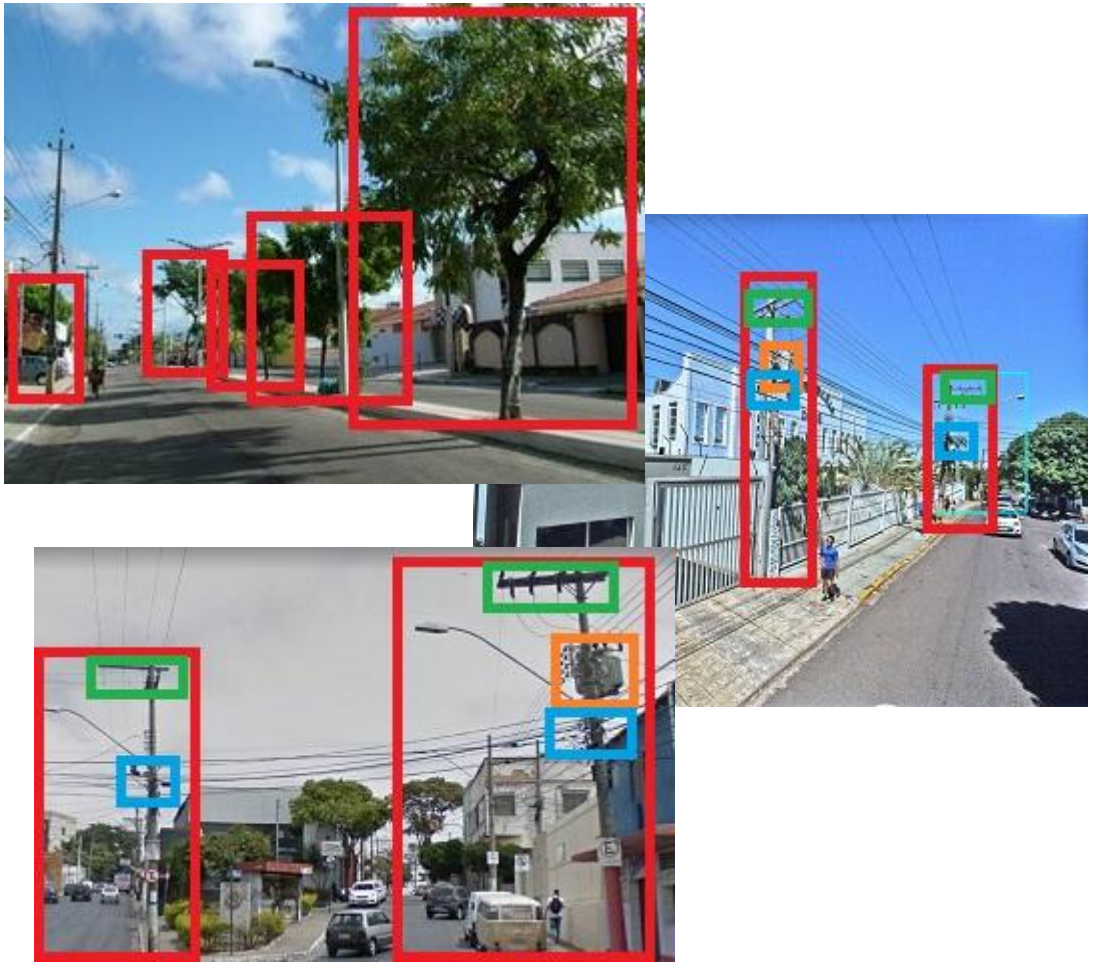
Rodar os testes

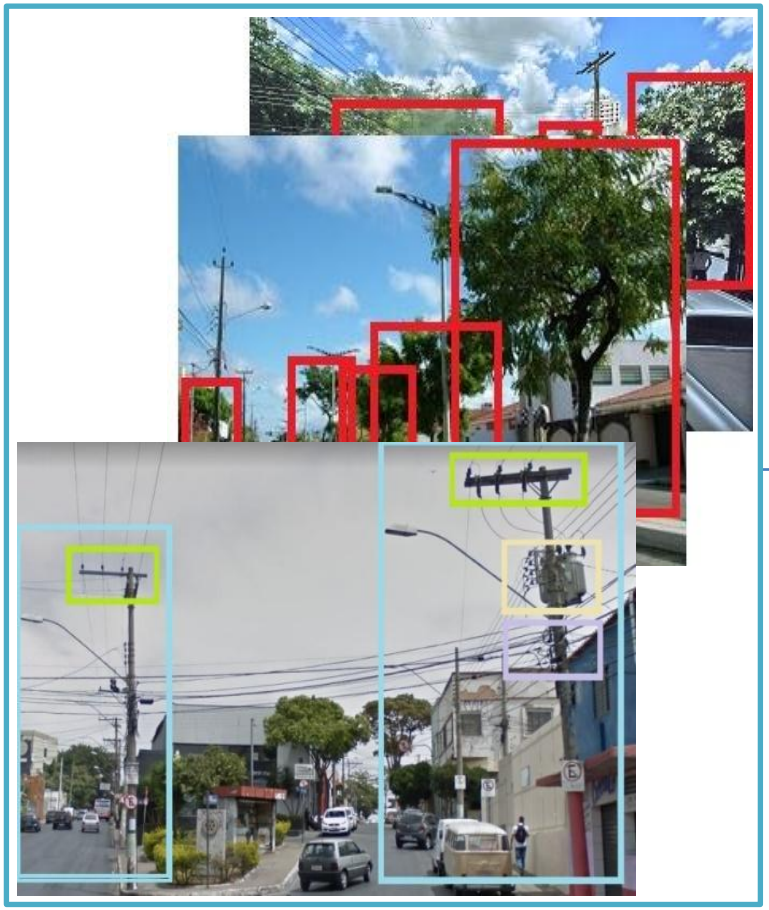
```
python object_detection/builders/model_builder_test.py
```

Treinamento

No mínimo 100 imagens para cada objeto

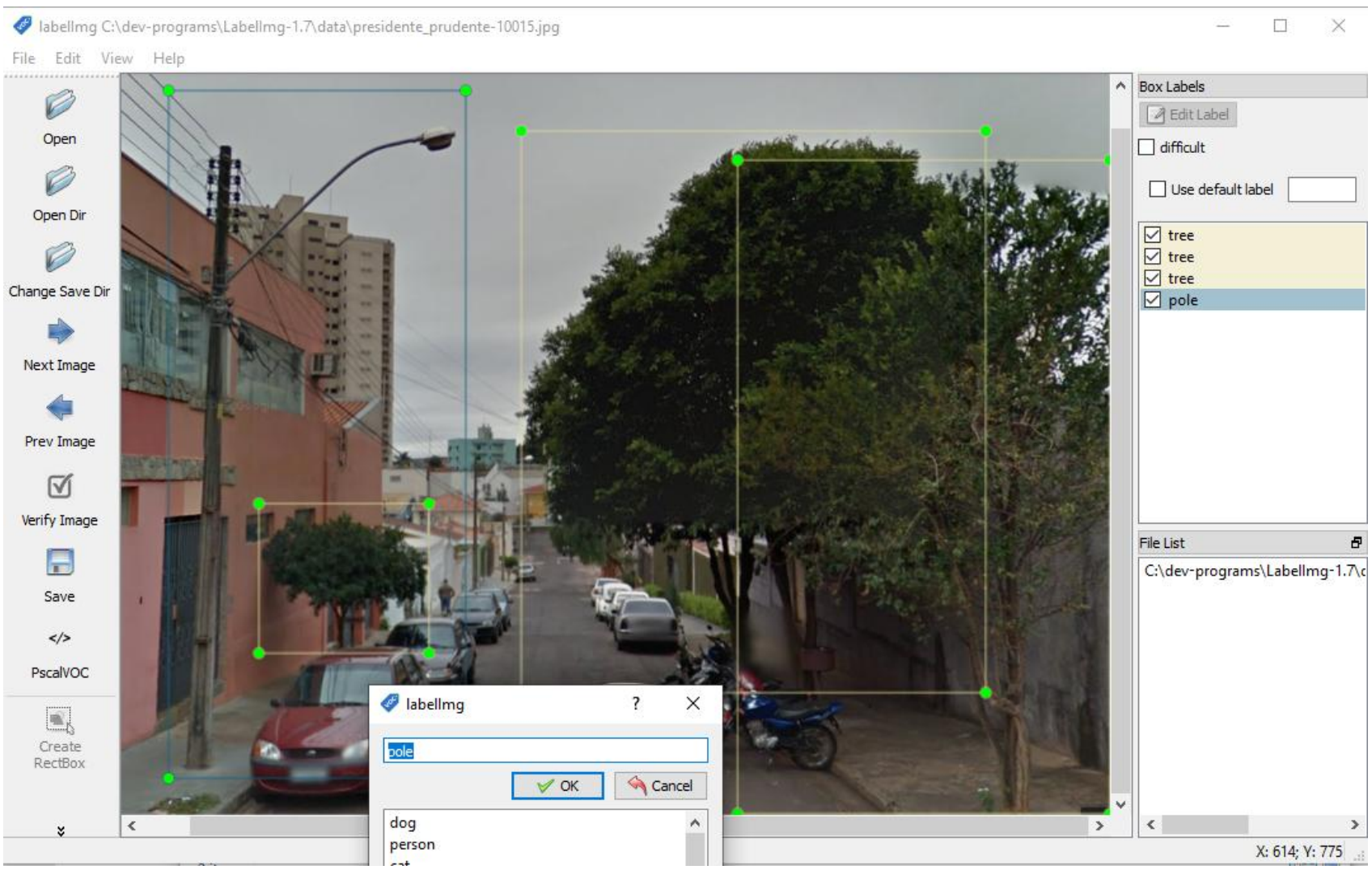
- workspace
  - images
    - 90% train
    - 10% test
  - annotations
  - pre-trained-model
  - training
  - exported-models
  - tests
    - images
    - results





Tensorflow

Modelo

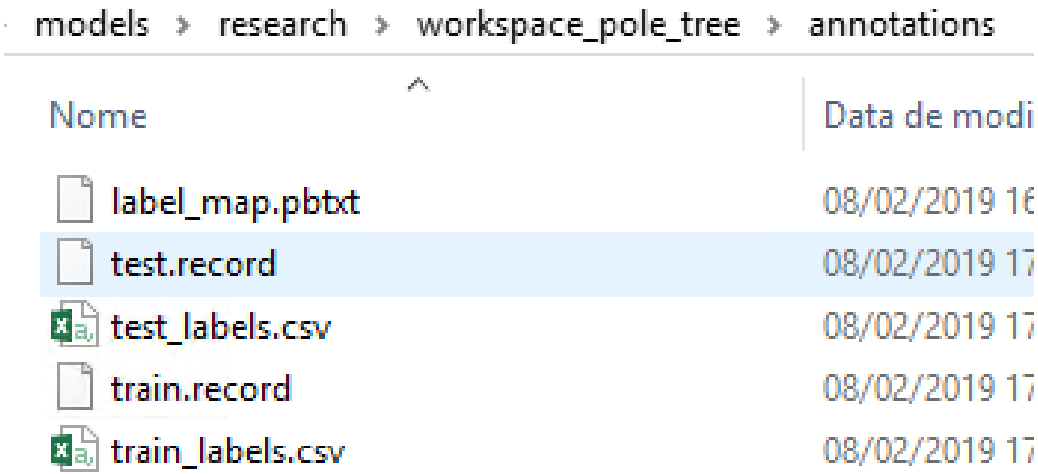

labelImg C:\dev-programs\LabelImg-1.7\data\presidente_prudente-10015.jpg
File
Edit
View
Help
Open
Open Dir
Change Save Dir
Next Image
Prev Image
Verify Image
Save
PscalVOC
Create RectBox
Box Labels
Edit Label
difficult
Use default label
tree
tree
tree
pole
File List
C:\dev-programs\LabelImg-1.7\c
labelImg
pole
OK
Cancel
dog
person
X: 614; Y: 775

```
<annotation>
    <folder>Imagens</folder>
    <filename>presidente_prudente-10015.jpg</filename>
    <path>C:\development\image_label\Imagens\presidente_prudente-10015.jpg</path>
    <source>
        <database>Unknown</database>
    </source>
    <size>
        <width>1440</width>
        <height>900</height>
        <depth>3</depth>
    </size>
    <segmented>0</segmented>
    <object>
        <name>poste</name>
        <pose>Unspecified</pose>
        <truncated>0</truncated>
        <difficult>0</difficult>
        <bndbox>
            <xmin>374</xmin>
            <ymin>103</ymin>
            <xmax>705</xmax>
            <ymax>833</ymax>
        </bndbox>
    </object>
    <object>
        <name>tree</name>
```

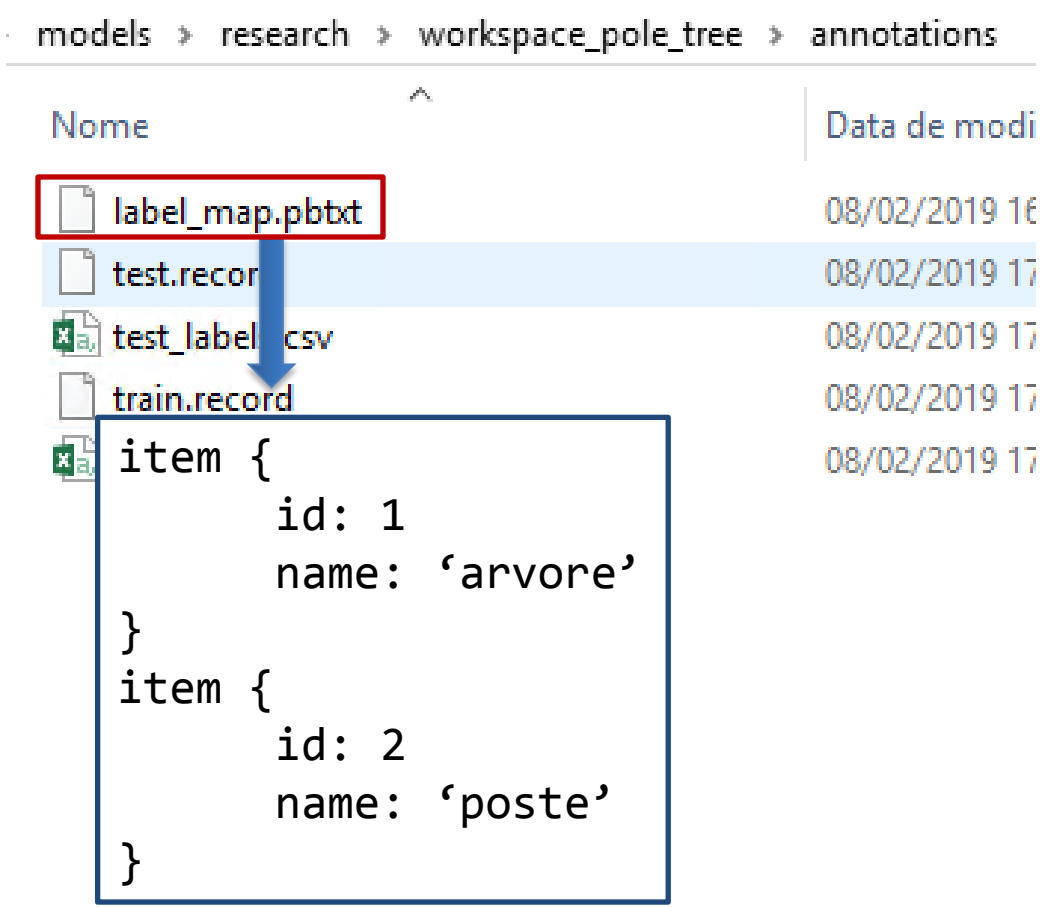
```
python scripts/preprocessing/xml_to_csv.py
  [parametros]
```

```
python scripts/preprocessing/xml_to_csv.py
  [parametros]
```

```
python scripts/preprocessing/generate_tfrecord.py
  [parametros]
```



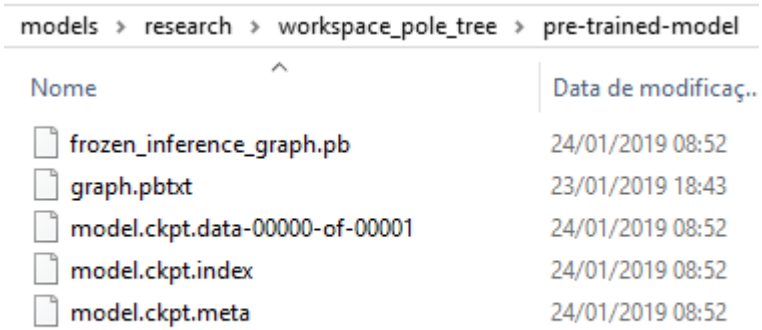
- workspace
  - images
    - train
    - test
  - annotations
  - pre-trained-model
  - training
  - exported-models
  - tests
    - images
    - results



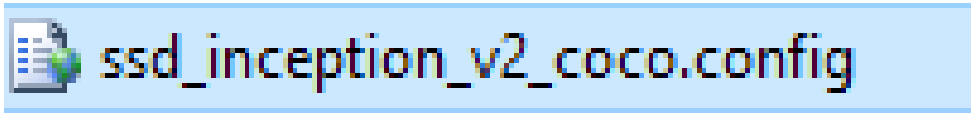
COCO-trained models

Model name	Speed (ms)	COCO mAP[^1]	Outputs
ssd_mobilenet_v1_coco	30	21	Boxes
ssd_mobilenet_v1_0.75_depth_coco ☆	26	18	Boxes
ssd_mobilenet_v1_quantized_coco ☆	29	18	Boxes
ssd_mobilenet_v1_0.75_depth_quantized_coco ☆	29	16	Boxes
ssd_mobilenet_v1_ppn_coco ☆	26	20	Boxes
ssd_mobilenet_v1_fpn_coco ☆	56	32	Boxes
ssd_resnet_50_fpn_coco ☆	76	35	Boxes
ssd_mobilenet_v2_coco	31	22	Boxes
ssd_mobilenet_v2_quantized_coco	29	22	Boxes
ssdlite_mobilenet_v2_coco	27	22	Boxes
ssd_inception_v2_coco	42	24	Boxes
faster_rcnn_inception_v2_coco	58	28	Boxes
faster_rcnn_resnet50_coco	89	30	Boxes
faster_rcnn_resnet50_lowproposals_coco	64		Boxes

- workspace
  - images
    - train
    - test
  - annotations
  - pre-trained-model
  - training
  - exported-models
  - tests
    - images
    - results

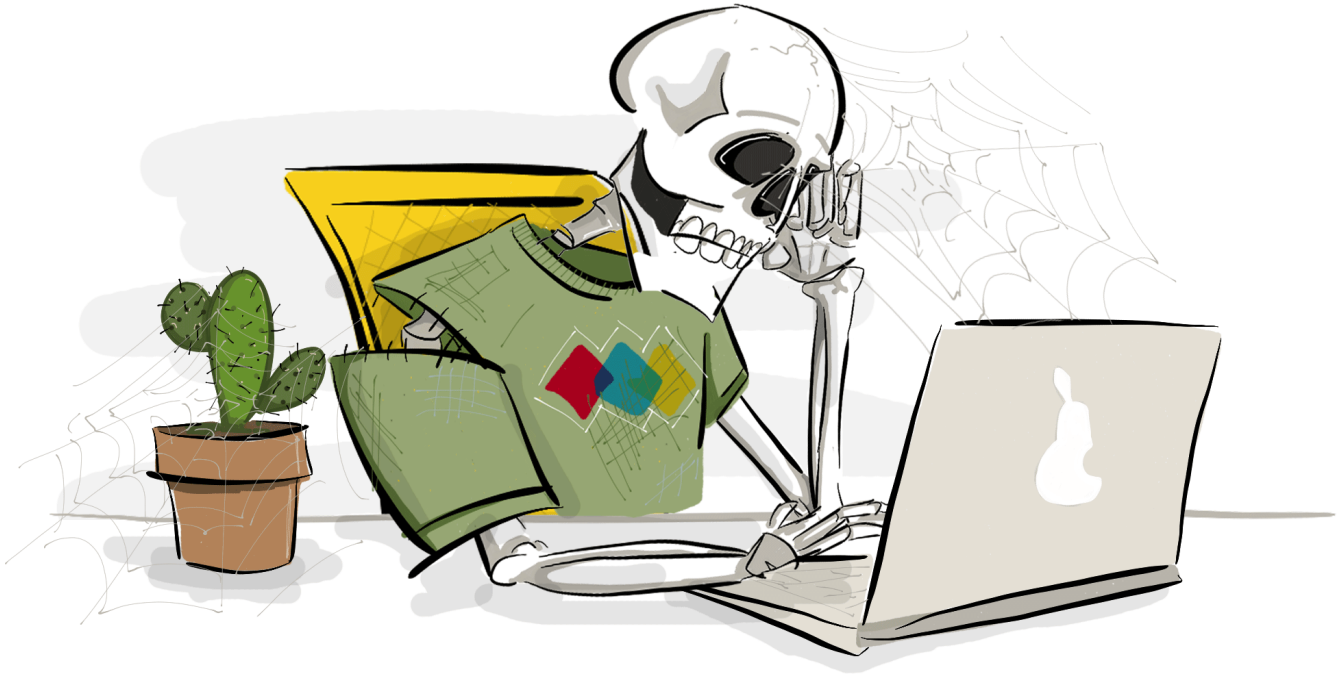


- workspace
  - images
    - train
    - test
  - annotations
  - pre-trained-model
  - training
  - exported-models
  - tests
    - images
    - results



```
python object_detection\train.py [parametros]
```

AGUARDANDO...



O TREINAMENTO DA MINHA REDE NEURAL

Google Cloud

2 GPUs NVIDIA Tesla K80

52 GB de memória

1162 imagens (até 4MB)

60 mil passos

Google Cloud

2 GPUs NVIDIA Tesla K80

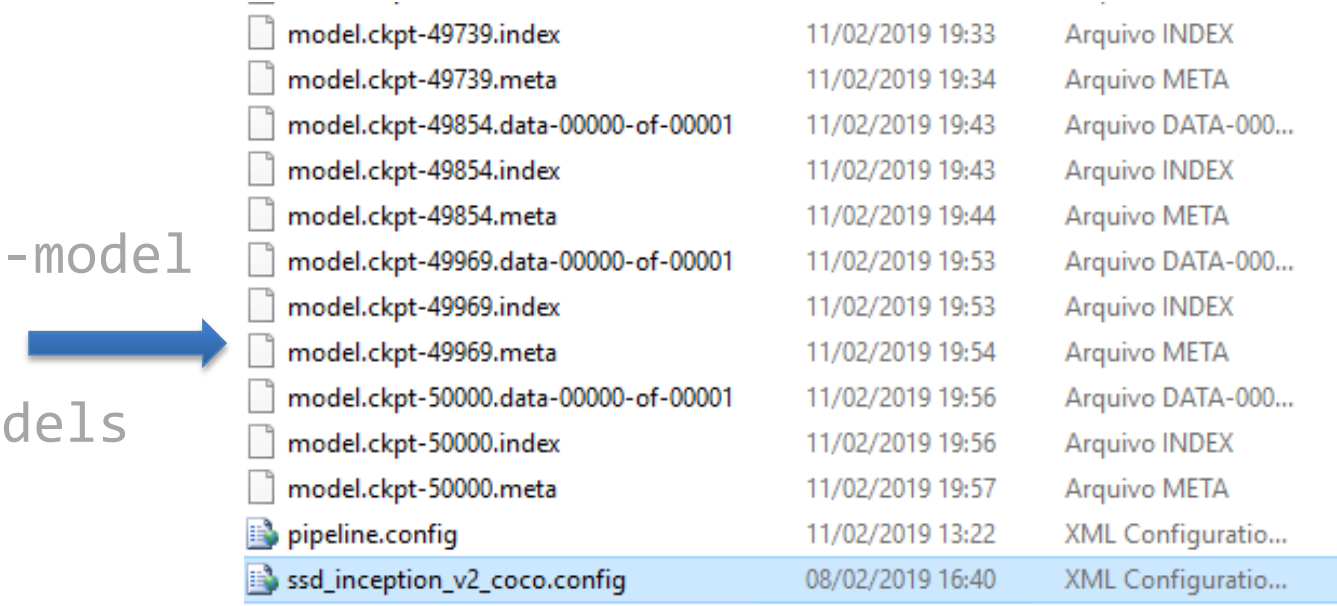
52 GB de memória

1162 imagens (até 4MB)

60 mil passos

91 horas e 40 minutos

- workspace
  - images
    - train
    - test
  - annotations
  - pre-trained-model
  - training
  - exported-models
  - tests
    - images
    - results

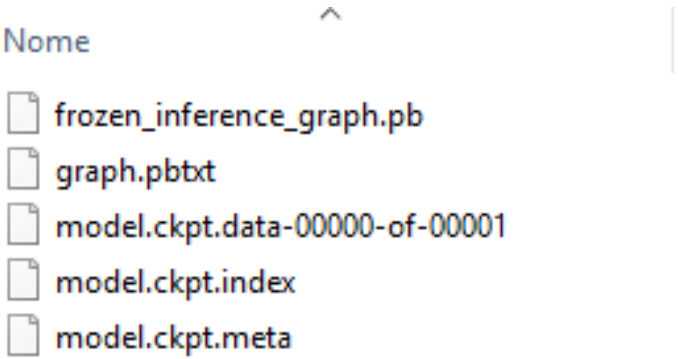


model.ckpt-49739.index	11/02/2019 19:33	Arquivo INDEX
model.ckpt-49739.meta	11/02/2019 19:34	Arquivo META
model.ckpt-49854.data-00000-of-00001	11/02/2019 19:43	Arquivo DATA-000...
model.ckpt-49854.index	11/02/2019 19:43	Arquivo INDEX
model.ckpt-49854.meta	11/02/2019 19:44	Arquivo META
model.ckpt-49969.data-00000-of-00001	11/02/2019 19:53	Arquivo DATA-000...
model.ckpt-49969.index	11/02/2019 19:53	Arquivo INDEX
model.ckpt-49969.meta	11/02/2019 19:54	Arquivo META
model.ckpt-50000.data-00000-of-00001	11/02/2019 19:56	Arquivo DATA-000...
model.ckpt-50000.index	11/02/2019 19:56	Arquivo INDEX
model.ckpt-50000.meta	11/02/2019 19:57	Arquivo META
pipeline.config	11/02/2019 13:22	XML Configuratio...
ssd_inception_v2_coco.config	08/02/2019 16:40	XML Configuratio...

Exportar modelo

```
python object_detection\export_inference_graph.py
  [parâmetros]
```

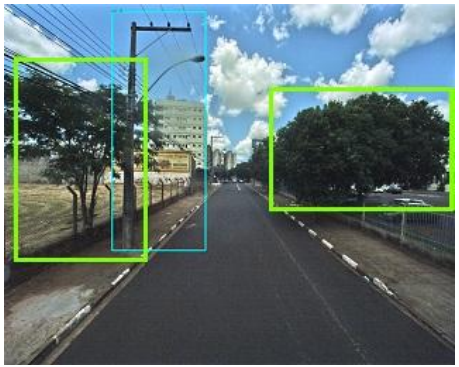
- workspace
  - images
    - train
    - test
  - annotations
  - pre-trained-model
  - training
  - exported-models
  - tests
    - images
    - results

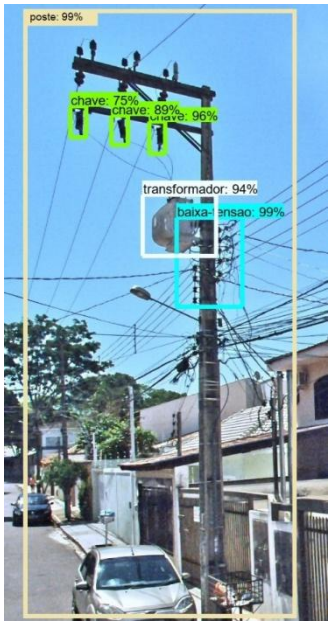


Testar

```
python object_detection\object_detection.py
```

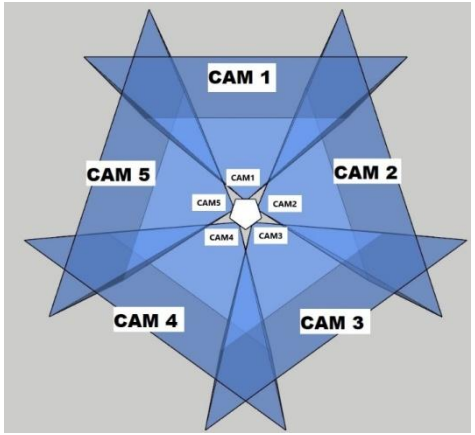
- workspace
  - images
    - train
    - test
  - annotations
  - pre-trained-model
  - training
  - exported-models
  - tests
    - images
    - results



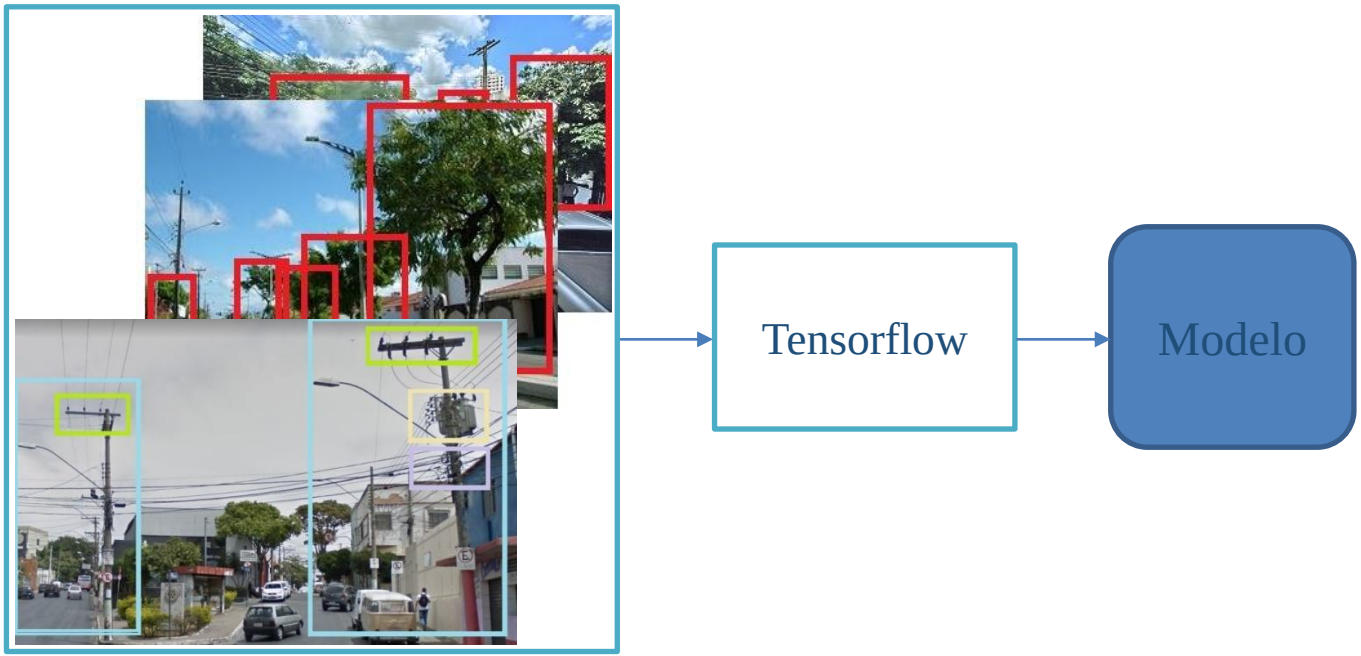


Detecção de árvores e equipamentos

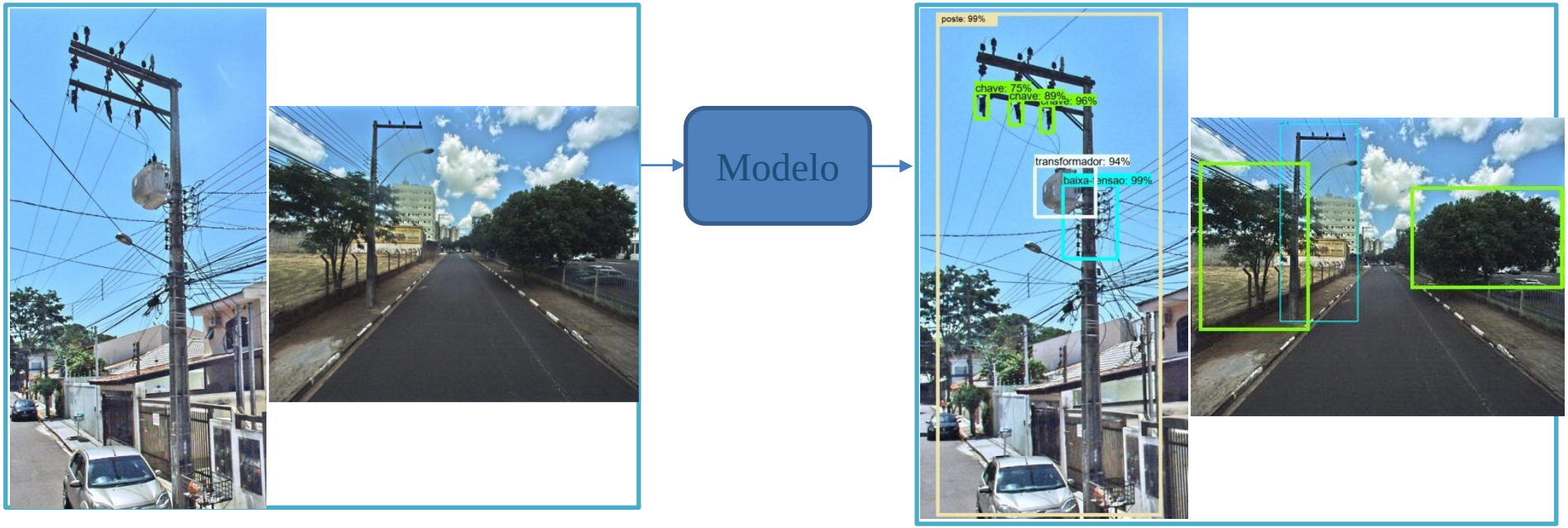
Sistema de mapeamento terrestre móvel

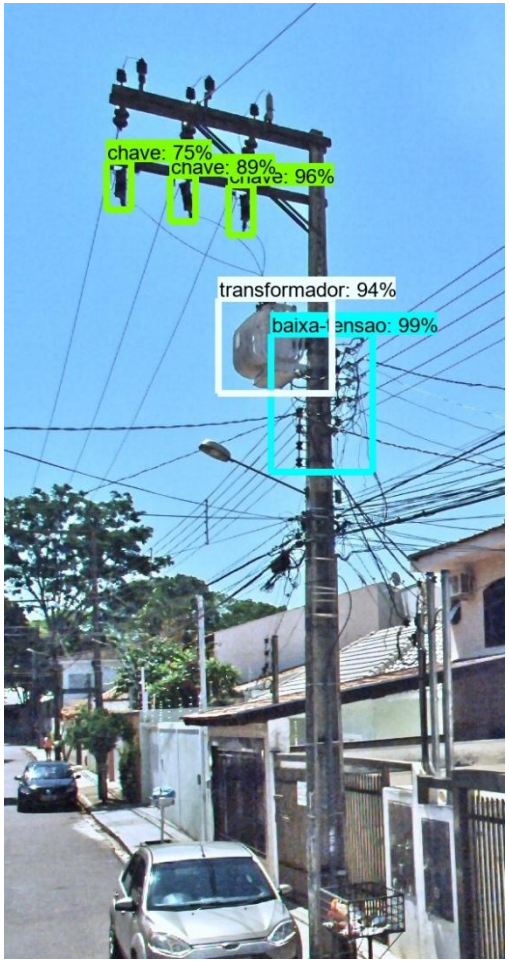

CAM 1
CAM 5
CAM 2
CAM 4
CAM 3
CAM1
CAM5
CAM2
CAM4
CAM3

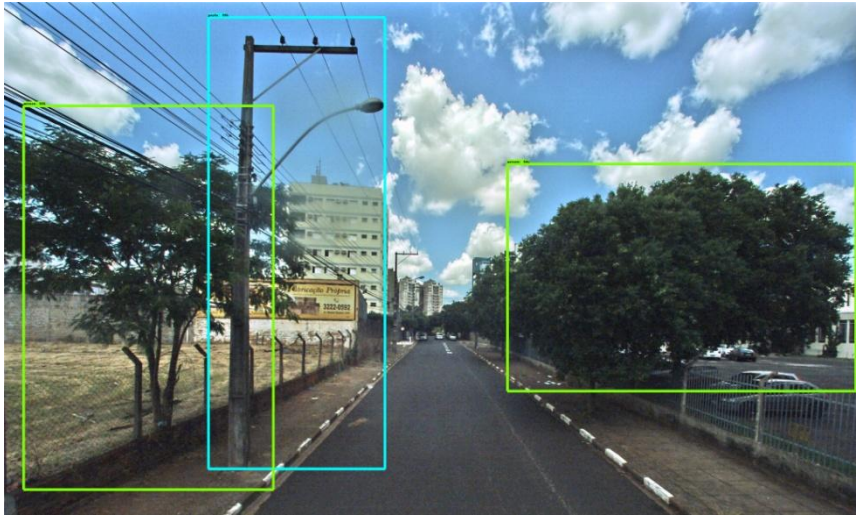
I - Treinar o modelo

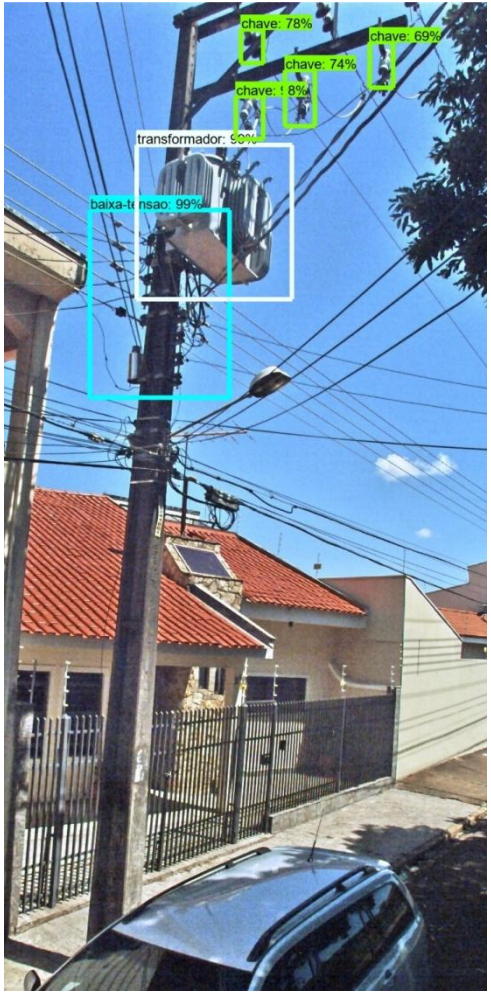


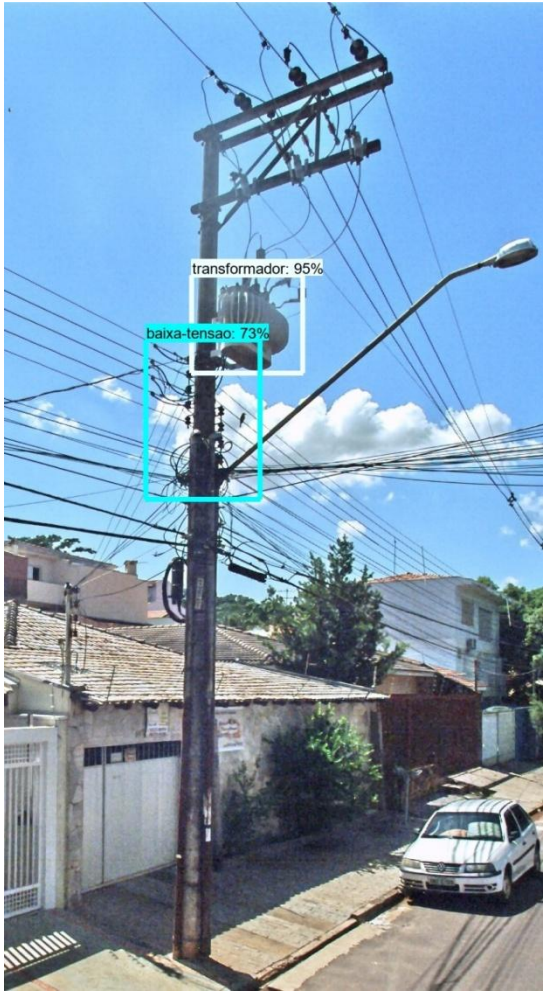
II – Identificar os objetos

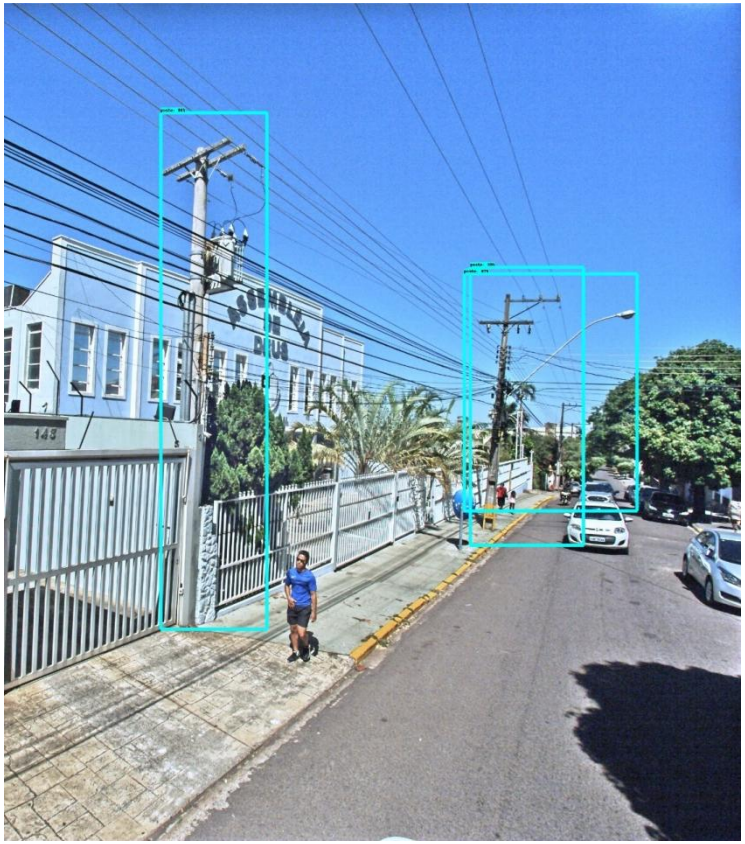



chave: 75%
chave: 89%
transformador: 94%
baixa-tensao: 99%




chave: 78%
chave: 69%
chave: 74%
chave: 1 8%
transformador: 99%
baixa-tensao: 99%


transformador: 95%
baixa-tensao: 73%


143

Também erra!

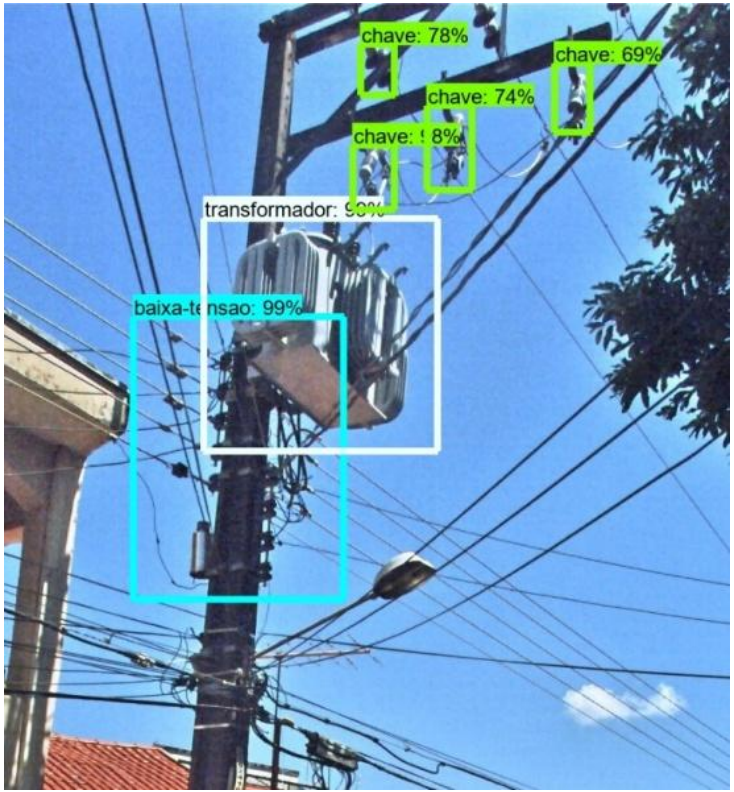
Treinamento

OBJETO	NÚMERO DE IMAGENS
Árvores	512
Poste	650
Chave	221
Transformador	317
Cabo	932

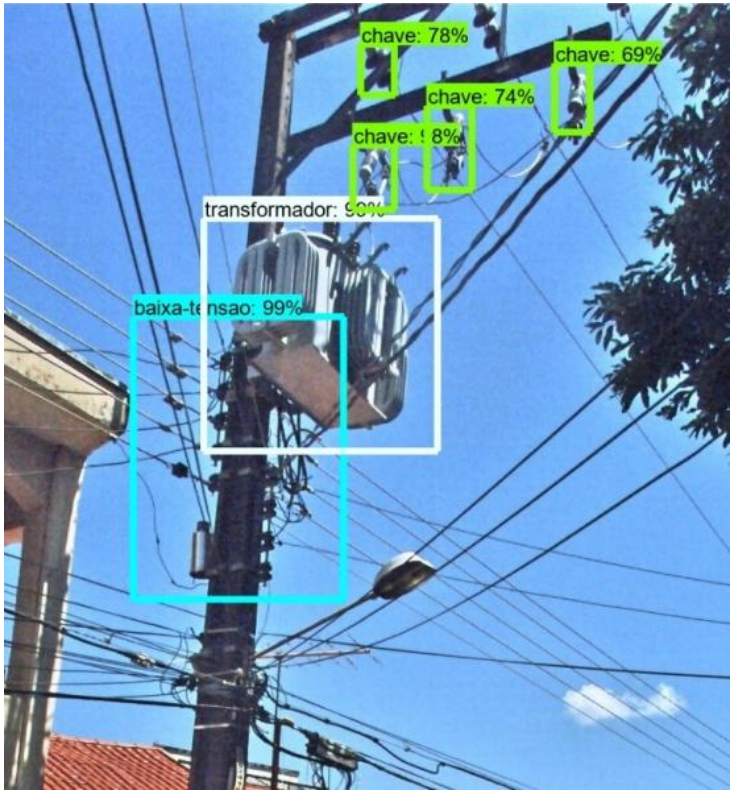
Avaliação

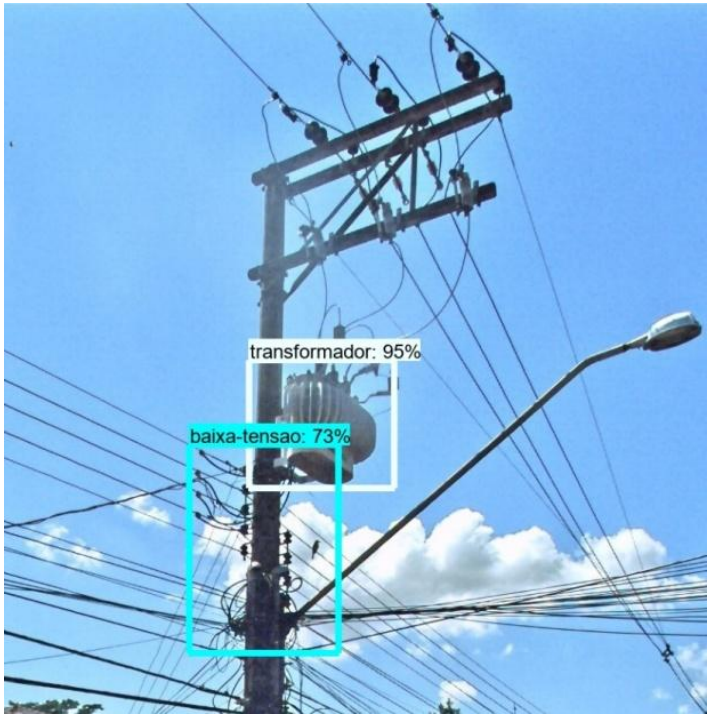
- Teste em 214.400 imagens
- Inspeção manual de amostra com 13.509 imagens (6,3% do total)

Precisão



Recall


chave: 78%
chave: 69%
chave: 74%
transformador:
baixa-tensao: 99%


transformador: 95%
baixa-tensao: 73%

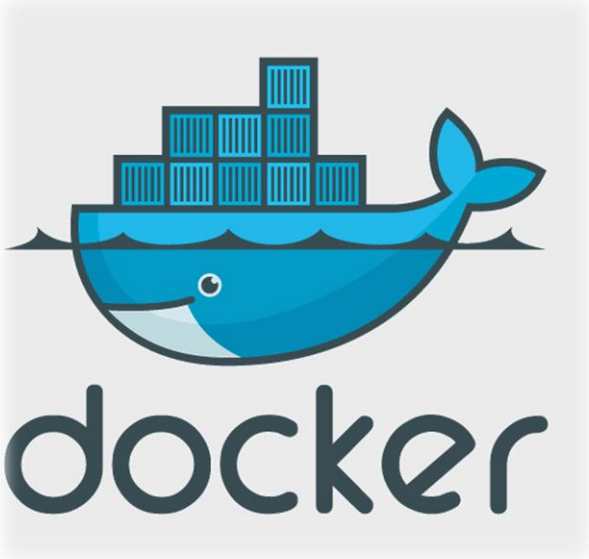
Resultados

OBJETO	Nº DETECÇÕES	PRECISÃO	RECALL
Árvores	554	93,32%	93,15%
Poste	373	91,69%	90%
Cabos	394	100%	74,17%
Chave	28	92,85%	71,79%
Transformador	33	78,79%	70,27%
TOTAL	**1382**	**93,27%**	**86,91%**

Desafios

Ambiente

- Funcionou para classificação
- Pesado



Ambiente

Instalação... 

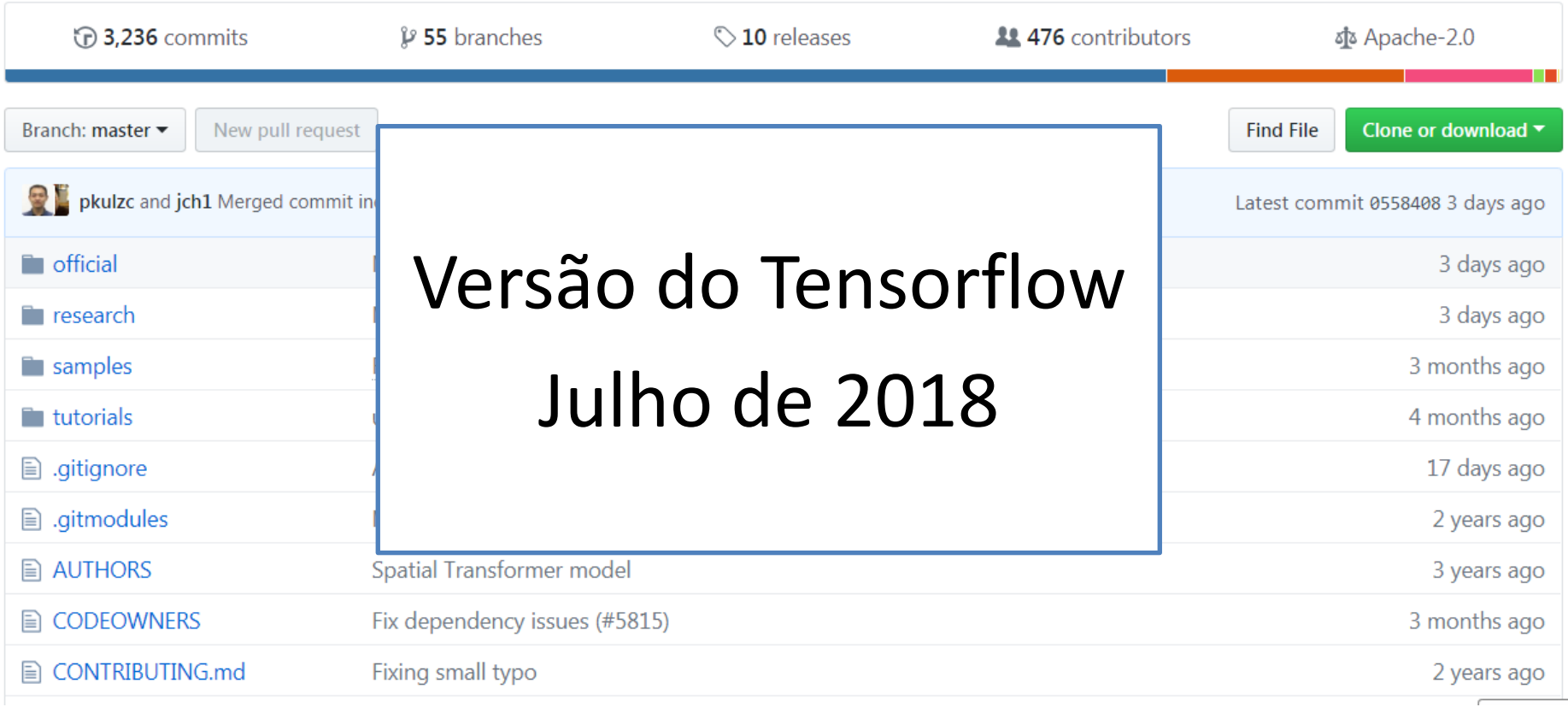
Gerar o modelo...

Instalação...

Gerar o modelo...

https://github.com/tensorflow/models/

Models and examples built with TensorFlow

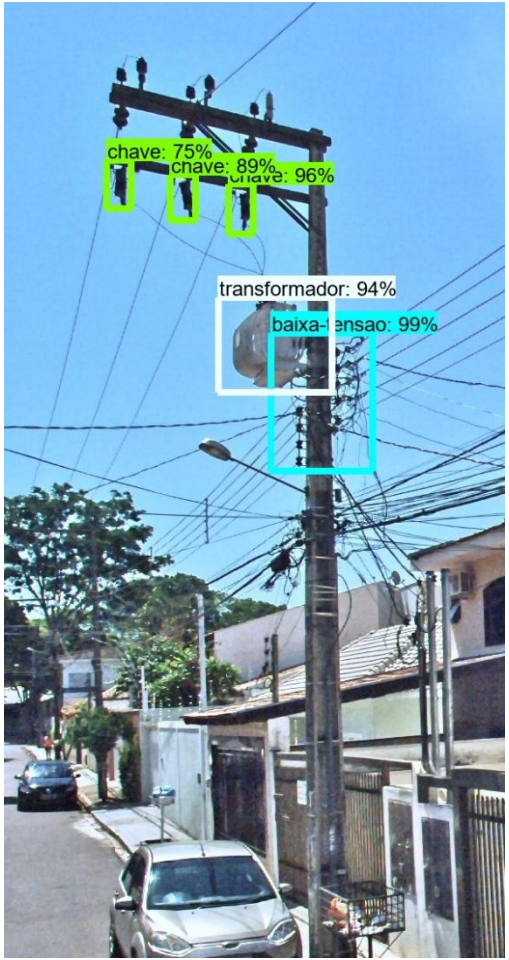


Versão do Tensorflow

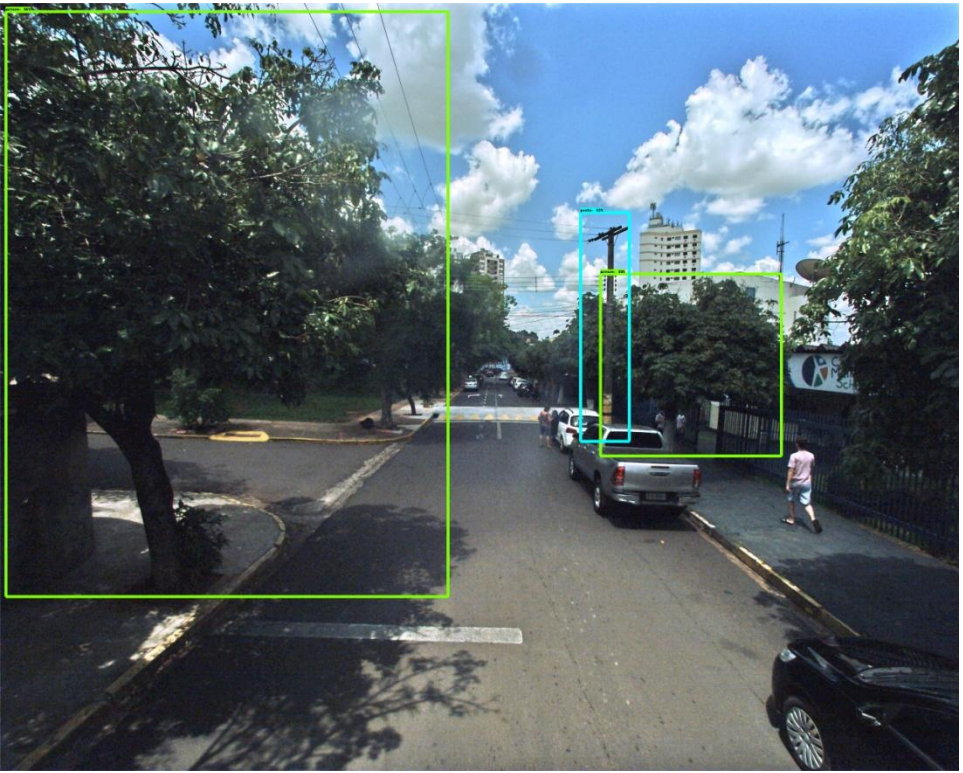
Julho de 2018

Rede neural

Objetos pequenos



Solução: recorte do poste



Rede neural

Localização de cabos

Rede neural

Localização de cabos

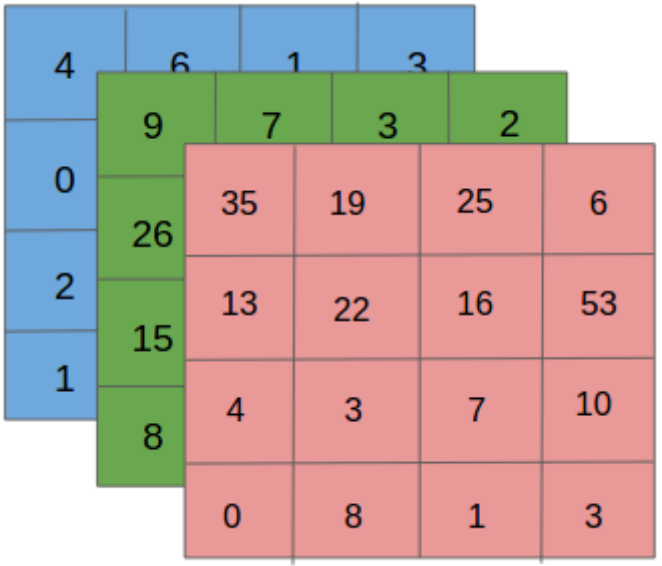
Rede neural

Área rural

X

Área urbana

Como funciona?


4
6
1
3
9
7
3
2
0
35
19
25
6
26
2
13
22
16
53
15
1
4
3
7
10
8
0
8
1
3

0	0	0	0	0	0	30
0	0	0	0	50	50	50
0	0	0	20	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0

Representação numérica de uma forma

0	0	0	0	0	0	30
0	0	0	0	50	50	50
0	0	0	20	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0

*

Filtro

0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

50*30 + 20*30 + 50*30 + 50*30 + 50*30 = 6600

0	0	0	0	0	0	0
0	40	0	0	0	0	0
40	0	40	0	0	0	0
40	20	0	0	0	0	0
0	50	0	0	0	0	0
0	0	50	0	0	0	0
25	25	0	50	0	0	0

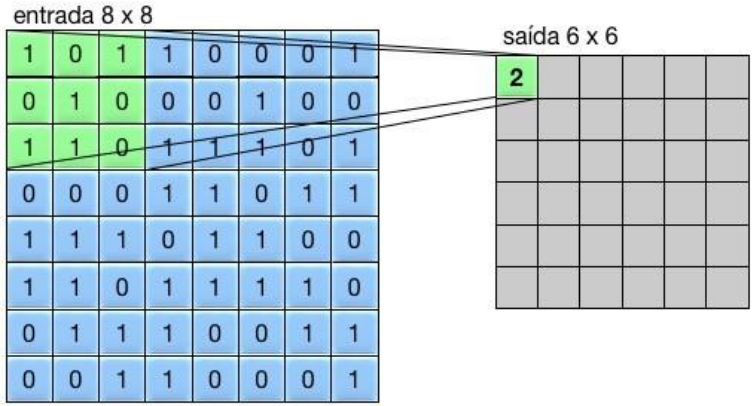
*

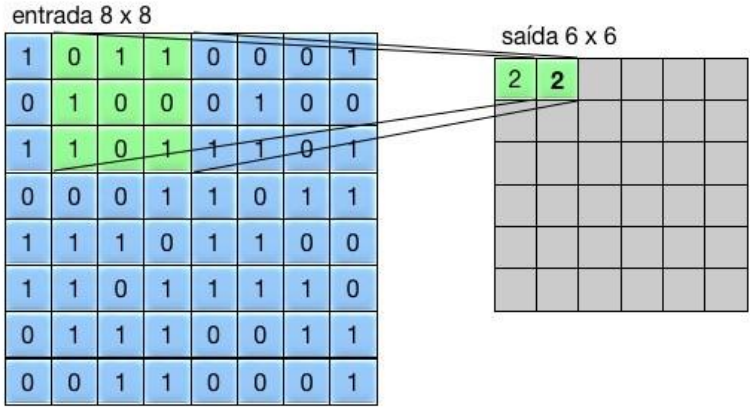
0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

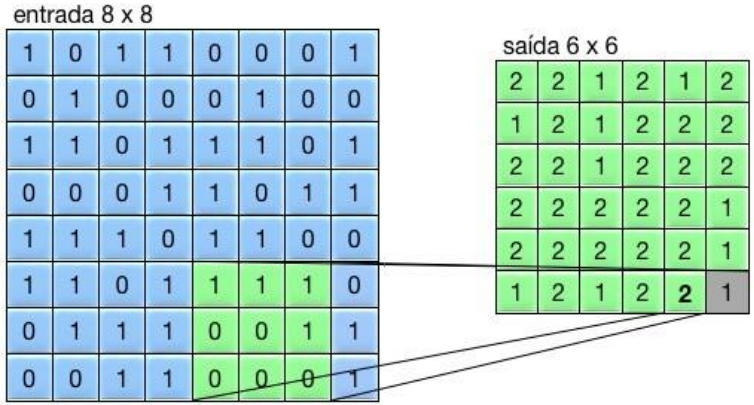
Multiplicação e soma = zero

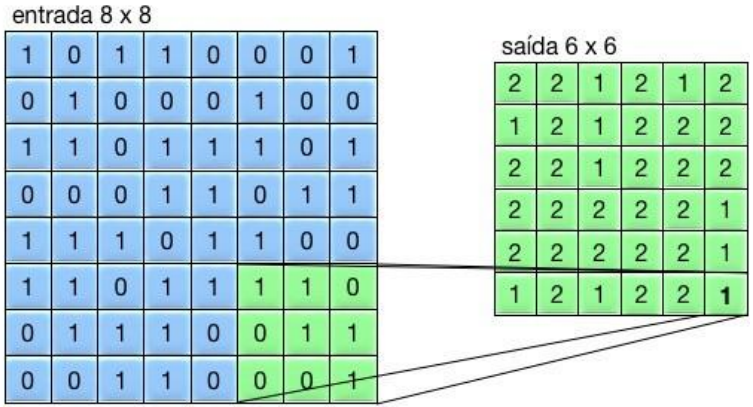
filtro 3 x 3

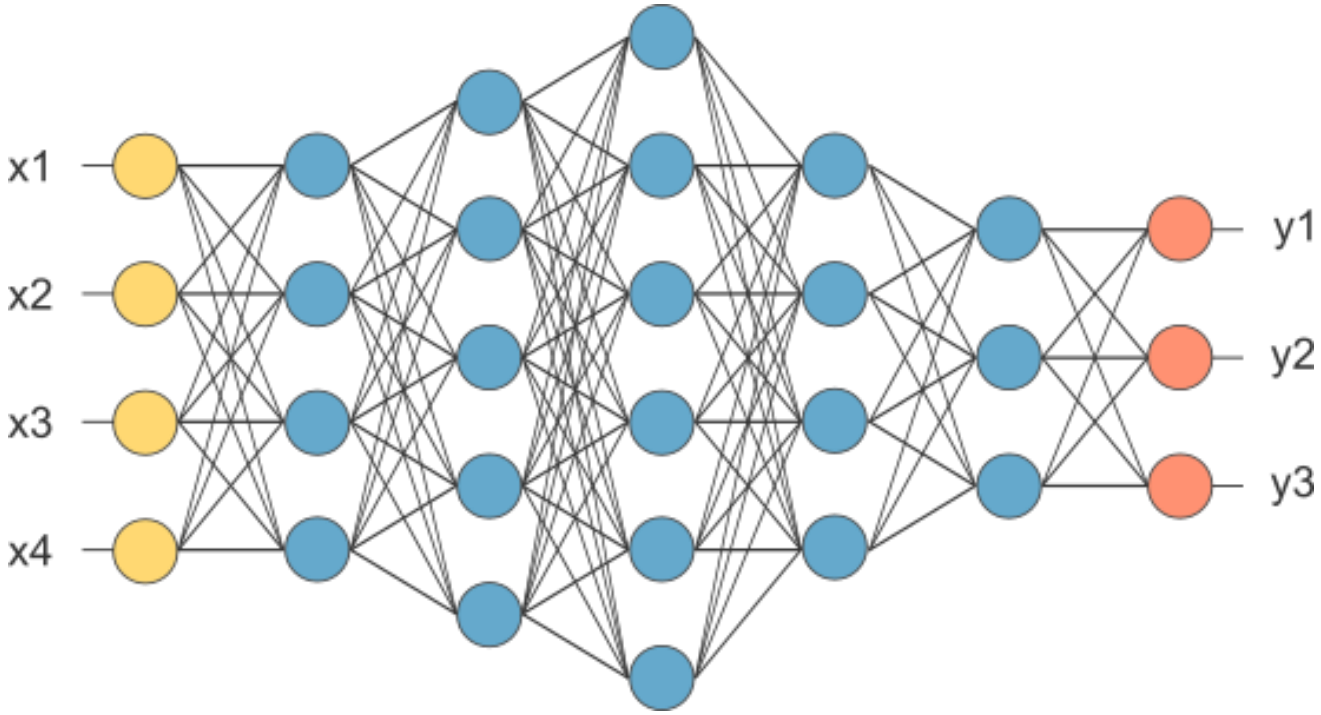
1	0	1
0	1	0
1	0	1










x1
x2
x3
x4
y1
y2
y3

Referências

Tutorial Tensorflow object detection:

https://tensorflow-object-detection-api-tutorial.readthedocs.io

https://www.linkedin.com/pulse/o-que-%C3%A9-um-rede-neural-convolucional-alex-fernandes-mansano

https://imasters.com.br/back-end/classificacao-de-imagens-com-deep-learning-e-tensorflow

https://www.tensorflow.org/hub/tutorials/image_retraining

Tamara Mendes

Mestre em Ciência da Computação

Analista de Software na CONCERT Technologies

tammends@gmail.com

http://br.linkedin.com/in/tamaramendes

CONCERT TECHNOLOGIES

juntos somos INQUIETOS