

Міністерство освіти та науки України
Львівський національний університет імені Івана Франка

Звіт

Про виконання лабораторної роботи №2

“MaskRNN”

Виконав: студент групи Фес-31 Сало Остап
Перевірив: Сінкевич О.О.

Мета: реалізувати модель для детектингу об'єктів використовуючи transfer learning.

Замітка: Я використовував MASKRNN з таких причин :

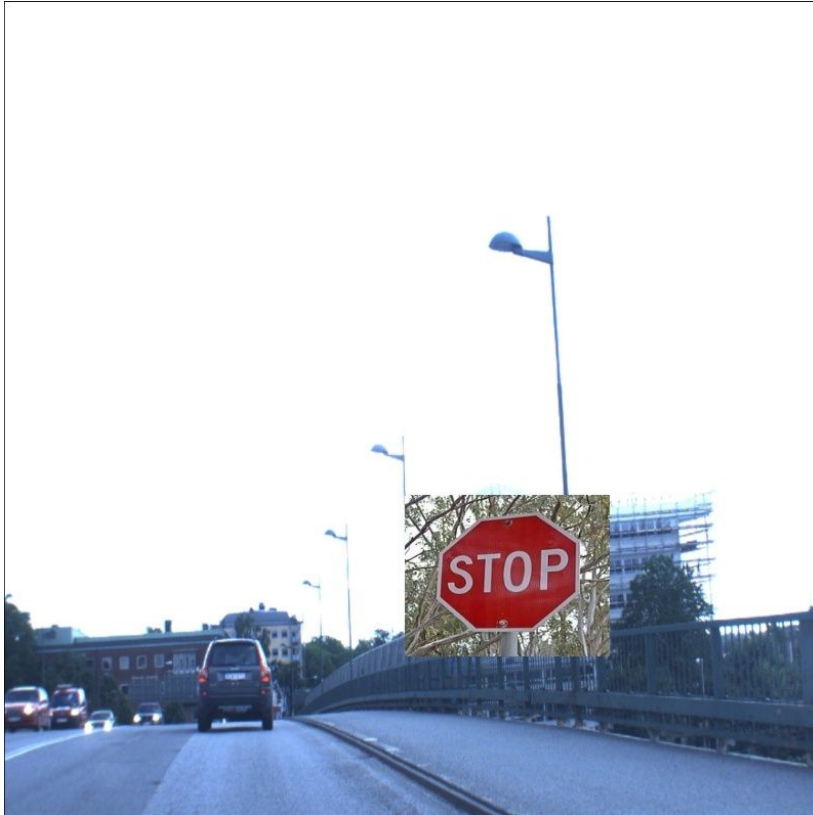
1. MaskRnn найсучасніша з сімейства RCNN
2. MaskRnn не має проблем з еалізацією в реальному часі.

Реалізація:

В якості об'єкта я взяв дорожній знак STOP.

Дані я створював власноруч за допомогою датасета звичайної дороги та 50 фотографії знака STOP, та наклав один на одний поступово створюючи xml файл для інформації про розташування об'єкта.

Ось результати реалізації :



З такими даними була навчена модель.

y%20https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fpeopleapi.v3%2Fme%3FpersonOnly&response_type=code

Enter your authorization code:

.....

Mounted at /content/drive

```
] : !git clone https://github.com/matterport/Mask_RCNN.git
```

Cloning into 'Mask_RCNN'...

remote: Enumerating objects: 956, done.

remote: Total 956 (delta 0), reused 0 (delta 0), pack-reused 956

Receiving objects: 100% (956/956), 111.90 MiB | 33.93 MiB/s, done.

Resolving deltas: 100% (571/571), done.

```
] : cd Mask_RCNN
```

/content/Mask_RCNN

```
] : !python setup.py install
```

WARNING:root:Fail load requirements file, so using default ones.

running install

Для початку необхідно встановити репозиторій MaskRnn.

y%20https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fpeopleapi.v3%2Fme%3FpersonOnly&response_type=code

Enter your authorization code:

.....

Mounted at /content/drive

```
] : !git clone https://github.com/matterport/Mask_RCNN.git
```

Cloning into 'Mask_RCNN'...

remote: Enumerating objects: 956, done.

remote: Total 956 (delta 0), reused 0 (delta 0), pack-reused 956

Receiving objects: 100% (956/956), 111.90 MiB | 33.93 MiB/s, done.

Resolving deltas: 100% (571/571), done.

```
] : cd Mask_RCNN
```

/content/Mask_RCNN

```
] : !python setup.py install
```

WARNING:root:Fail load requirements file, so using default ones.

running install

Імпорт бібліотек.



Завантаження датасету.

```

# class that defines and loads the kangaroo dataset
class KangarooDataset(Dataset):
    # load the dataset definitions
    def load_dataset(self, dataset_dir, is_train=True):
        # define one class
        self.add_class("dataset", 1, "stop")
        # define data locations
        images_dir = dataset_dir + '/images/'
        annotations_dir = dataset_dir + '/annotation/'
        # find all images
        for filename in listdir(images_dir):
            # extract image id
            image_id = filename[:-4]
            # skip bad images
            if image_id in ['90']:
                continue
            # skip all images after 150 if we are building the train set
            if is_train and int(image_id) >= 210:
                continue
            # skip all images before 150 if we are building the test/val set
            if not is_train and int(image_id) < 210:
                continue
            img_path = images_dir + filename
            ann_path = annotations_dir + image_id + '.xml'
            # add to dataset
            self.add_image('dataset', image_id=image_id, path=img_path, annotation=a
nn_path)

# extract bounding boxes from an annotation file
def extract_boxes(self, filename):
    # load and parse the file
    tree = ElementTree.parse(filename)
    # get the root of the document
    root = tree.getroot()
    # extract each bounding box
    boxes = list()
    for box in root.findall('.//bndbox'):
        xmin = int(box.find('xmin').text)
        ymin = int(box.find('ymin').text)
        xmax = int(box.find('xmax').text)
        ymax = int(box.find('ymax').text)
        coors = [xmin, ymin, xmax, ymax]
        boxes.append(coors)
    # extract image dimensions
    width = int(root.find('.//size/width').text)
    height = int(root.find('.//size/height').text)
    return boxes, width, height

# load the masks for an image
def load_mask(self, image_id):
    # get dataset

```

Клас для parse даних для моделі.

```

-----
Configurations:
BACKBONE                resnet101
BACKBONE_STRIDES        [4, 8, 16, 32, 64]
BATCH_SIZE              2
BBOX_STD_DEV            [0.1 0.1 0.2 0.2]
COMPUTE_BACKBONE_SHAPE  None
DETECTION_MAX_INSTANCES 100
DETECTION_MIN_CONFIDENCE 0.7
DETECTION_NMS_THRESHOLD 0.3
FPN_CLASSIF_FC_LAYERS_SIZE 1024
GPU_COUNT               1
GRADIENT_CLIP_NORM      5.0
IMAGES_PER_GPU          2
IMAGE_CHANNEL_COUNT     3
IMAGE_MAX_DIM           1024
IMAGE_META_SIZE         14
IMAGE_MIN_DIM           800
IMAGE_MIN_SCALE         0
IMAGE_RESIZE_MODE       square
IMAGE_SHAPE             [1024 1024   3]
LEARNING_MOMENTUM       0.9
LEARNING_RATE           0.001
LOSS_WEIGHTS            {'rpn_class_loss': 1.0, 'rpn_bbox_loss': 1.0, 'mrcnn_
1.0, 'mrcnn_bbox_loss': 1.0, 'mrcnn_mask_loss': 1.0}
MASK_POOL_SIZE          14
MASK_SHAPE              [28, 28]
MAX_GT_INSTANCES        100
MEAN_PIXEL              [123.7 116.8 103.9]
MINI_MASK_SHAPE         (56, 56)
NAME                    kangaroo_cfg
NUM_CLASSES              2
POOL_SIZE               7
POST_NMS_ROIS_INFERENCE 1000
POST_NMS_ROIS_TRAINING  2000
PRE_NMS_LIMIT           6000
ROI_POSITIVE_RATIO      0.33
RPN_ANCHOR_RATIOS       [0.5, 1, 2]
RPN_ANCHOR_SCALES       (32, 64, 128, 256, 512)
RPN_ANCHOR_STRIDE       1
RPN_BBOX_STD_DEV        [0.1 0.1 0.2 0.2]
RPN_NMS_THRESHOLD       0.7
RPN_TRAIN_ANCHORS_PER_IMAGE 256
STEPS_PER_EPOCH         131
TOP_DOWN_PYRAMID_SIZE   256
TRAIN_BN                False
TRAIN_ROIS_PER_IMAGE     200
USE_MINI_MASK           True
USE_RPN_ROIS            True
VALIDATION_STEPS        50
WEIGHT_DECAY            0.0001

```

```

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensor
py:541: The name tf.placeholder is deprecated. Please use tf.compat.v1.placeholder :

```

Параметри навчання.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/callbacks.py:1125: The name tf.summary.FileWriter is deprecated. Please use tf.compat.v1.summary.FileWriter instead.

Epoch 1/5

131/131 [=====] - 389s 3s/step - loss: 0.7568 - rpn_class_loss: 0.0059 - rpn_bbox_loss: 0.0972 - mrcnn_class_loss: 0.0406 - mrcnn_bbox_loss: 0.3006 - mrcnn_mask_loss: 0.3125 - val_loss: 0.3748 - val_rpn_class_loss: 0.0051 - val_rpn_bbox_loss: 0.1000 - val_mrcnn_class_loss: 0.0033 - val_mrcnn_bbox_loss: 0.0880 - val_mrcnn_mask_loss: 0.1784
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/callbacks.py:1265: The name tf.Summary is deprecated. Please use tf.compat.v1.Summary instead.

Epoch 2/5

131/131 [=====] - 328s 3s/step - loss: 0.3527 - rpn_class_loss: 0.0035 - rpn_bbox_loss: 0.0846 - mrcnn_class_loss: 0.0059 - mrcnn_bbox_loss: 0.0858 - mrcnn_mask_loss: 0.1729 - val_loss: 0.3603 - val_rpn_class_loss: 0.0029 - val_rpn_bbox_loss: 0.1103 - val_mrcnn_class_loss: 0.0029 - val_mrcnn_bbox_loss: 0.0792 - val_mrcnn_mask_loss: 0.1650

Epoch 3/5

131/131 [=====] - 328s 3s/step - loss: 0.3257 - rpn_class_loss: 0.0026 - rpn_bbox_loss: 0.0792 - mrcnn_class_loss: 0.0036 - mrcnn_bbox_loss: 0.0674 - mrcnn_mask_loss: 0.1729 - val_loss: 0.3652 - val_rpn_class_loss: 0.0026 - val_rpn_bbox_loss: 0.1081 - val_mrcnn_class_loss: 0.0033 - val_mrcnn_bbox_loss: 0.0910 - val_mrcnn_mask_loss: 0.1602

Epoch 4/5

131/131 [=====] - 328s 3s/step - loss: 0.2919 - rpn_class_loss: 0.0021 - rpn_bbox_loss: 0.0707 - mrcnn_class_loss: 0.0043 - mrcnn_bbox_loss: 0.0596 - mrcnn_mask_loss: 0.1552 - val_loss: 0.3697 - val_rpn_class_loss: 0.0018 - val_rpn_bbox_loss: 0.1096 - val_mrcnn_class_loss: 0.0021 - val_mrcnn_bbox_loss: 0.0803 - val_mrcnn_mask_loss: 0.1759

Epoch 5/5

131/131 [=====] - 328s 3s/step - loss: 0.2924 - rpn_class_loss: 0.0016 - rpn_bbox_loss: 0.0772 - mrcnn_class_loss: 0.0042 - mrcnn_bbox_loss: 0.0529 - mrcnn_mask_loss: 0.1565 - val_loss: 0.3414 - val_rpn_class_loss: 0.0021 - val_rpn_bbox_loss: 0.1099 - val_mrcnn_class_loss: 0.0016 - val_mrcnn_bbox_loss: 0.0535 - val_mrcnn_mask_loss: 0.1744

Результати навчання (На тестовому датасеті з 300 зображень)



Результати прогнозування.

Висновок : В цій лабораторній роботі я ознайомився пре - тренною моделю MaskRnn