Curso Inteligência Artificial: do Zero ao Infinito

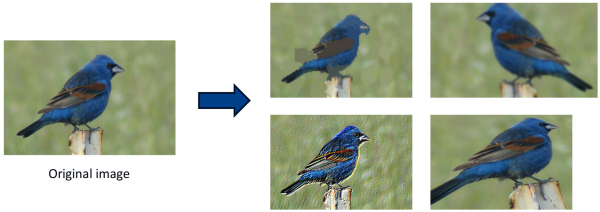
Object Detection Data Augmentation

Universidade Federal de Mato Grosso

Agenda

Object Detection Data Augmentation

O que é data augmentation?



Cria um novo conjunto de dados modificando as imagens existentes

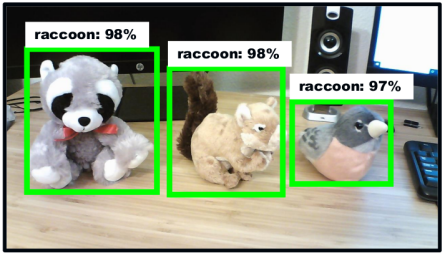
Object Detection Data Augmentation

Por que usar data augmentation?

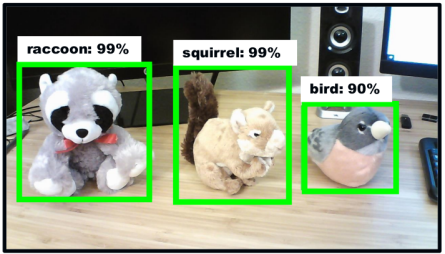
- Aumenta o tamanho do dataset
  - Previne overfitting
  - Aumenta a robustez das *features* extraídas
  - Aumenta a variedade dos ambientes visuais
- Resolve o desbalanceamento entre as classes
  - Desbalancemaneto de classes causa um enviesamento do modelo à classe dominante

Object Detection Data Augmentation

Model biased towards racoons



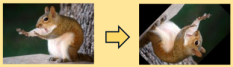
Unbiased model



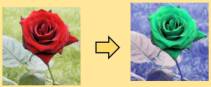
Object Detection Data Augmentation

Métodos para Data Augmentation

Basic image manipulation
(geometric, color space, kernel filters)



Rotation

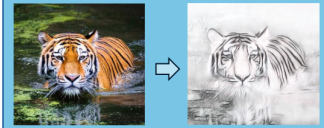


HSV shift

Dropout filter

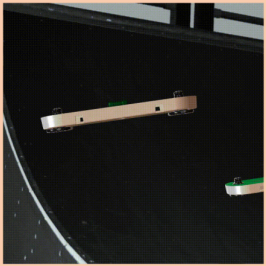
Neural networks
(GANs, Neural Style Transfer)

GAN examples



Neural Style Transfer

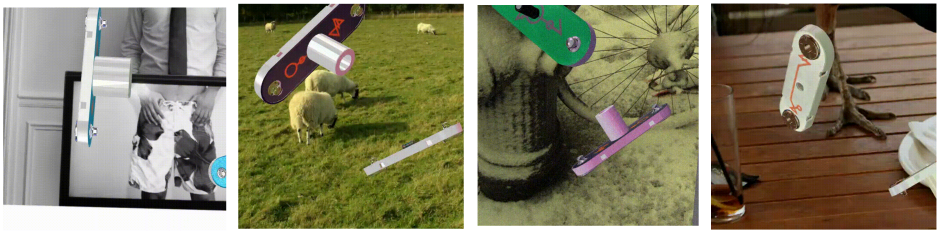
Synthetic image generation



Animation courtesy of Synapse Product Development, Inc.

Object Detection Data Augmentation

Synthetic Data Augmentation



Fonte: https://blog.synapse.com/post/automation-ar-and-jobs-a-demonstration.

Object Detection Data Augmentation

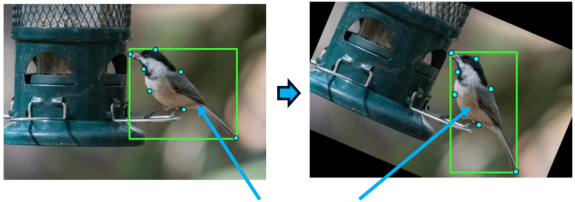
Desafios

Data Augmentation para **Object Detection** é mais desafiador que **Image Classification**:

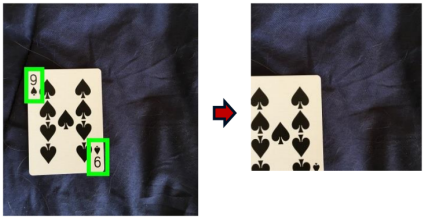
- Não transforma apenas a imagem, mas também as coordenadas do *bounding box*.
- Precisa garantir que a transformacão não destruirá o *bounding box*.

Object Detection Data Augmentation

Desafios



Keypoints and bounding boxes rotate with image



Destroying bounding boxes with cropping

Object Detection Data Augmentation

Ferramentas

Open-source GitHub libraries: Imgaug, Augmentator, Albumentations

- Pros: Free, well-documented, good examples
- Cons: May become unavailable or stop being maintained

OpenCV has image manipulation functions for manually creating augmentations

- Pros: Free, tight and flexible control over augmentation parameters
- Cons: Have to write everything yourself

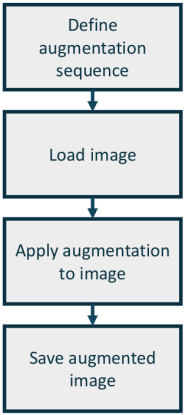
TensorFlow Object Detection API provides augmentation options to automatically use during preprocessing

- Pros: Free, easy to use (zero-effort)
- Cons: Not many operations available, may stack poorly with existing augmentations in dataset

Object Detection Data Augmentation

Code Example Using Imgaug

Basic augmentation process




```python
import imgaug as ia
from imgaug import augmenters as iaa
import cv2

#---- Define augmentation ----#
seq1 = iaa.Sequential([
    iaa.Fliplr(0.5),                                # Horizontal flip 50% of images
    iaa.Crop(percent=(0, 0.10)),                    # Crop all images between 0% to 10%
    iaa.GaussianBlur(sigma=(0, 0.5)),               # Add slight blur to images
    iaa.Multiply((0.8, 1.2), per_channel=0.2),  # Slightly brighten or darken images
    iaa.Affine(
        scale={"x": (0.8, 1.2), "y": (0.8,1.2)},                # Resize image
        translate_percent={"x": (-0.2, 0.2), "y": (-0.2, 0.2)}, # Translate image
        rotate=(-15, 15)                                        # Rotate image
        )
    ])

#---- Load image ----#
img1_bgr = cv2.imread('images/squirrel-780.jpg')  # Load image with OpenCV
img1 = cv2.cvtColor(img1_bgr, cv2.COLOR_BGR2RGB)    # Re-color to RGB from BGR

#---- Augment image ----#
img_aug1 = seq1(images=[img1])[0]   # Apply augmentation to image

#---- Save image ----#
img_aug_bgr1 = cv2.cvtColor(img_aug1, cv2.COLOR_RGB2BGR)    # Re-color to BGR from RGB
cv2.imwrite('outputs/basic1.jpg',img_aug_bgr1)              # Save image to disk
```

Object Detection Data Augmentation

Example

Object Detection Data Augmentation

Example

Augmentation Method	Top-1 Accuracy (%)	Top-5 Accuracy (%)
Baseline	48.13 ± 0.42	64.50 ± 0.65
Flipping	49.73 ± 1.13	67.36 ± 1.38
Rotating	50.80 ± 0.63	69.41 ± 0.48
Cropping	**61.95 ± 1.01**	**79.10 ± 0.80**
Color Jittering	49.57 ± 0.53	67.18 ± 0.42
Edge Enhancement	49.29 ± 1.16	66.49 ± 0.84
Fancy PCA	49.41 ± 0.84	67.54 ± 1.01

Object Detection Data Augmentation

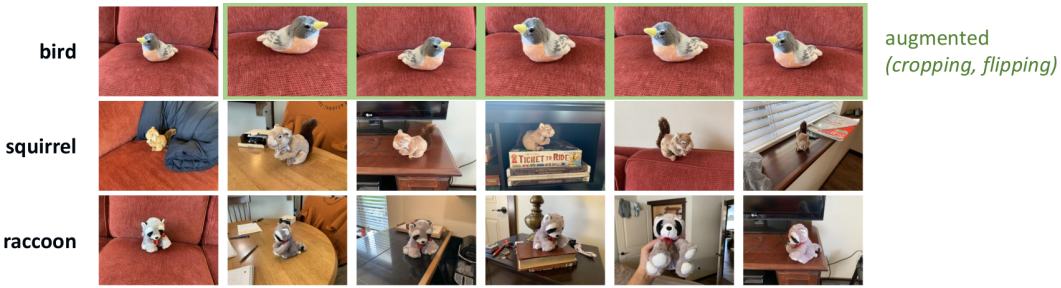
Experimento

Experiment to demonstrate how augmentation can improve performance

- 3-class model: Bird, squirrel, and raccoon (stuffed animals)
- Start with unbalanced dataset: 5 bird images, 30 squirrel images, 30 raccoon images
- Use augmentation to create 25 more bird images
- Train models using unbalanced and augmented dataset, compare performance

Object Detection Data Augmentation

Experimento



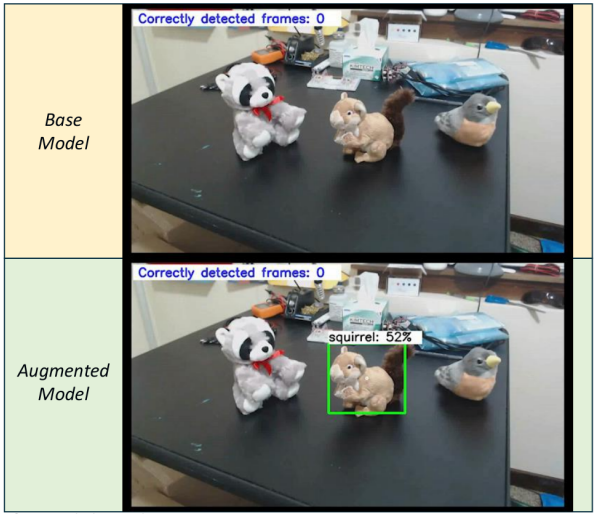
Object Detection Data Augmentation

Experimento

Model	mAP Score (MSCOCO AP50)	Correct Frames
Base	77.87%	18
Augmented	89.47%	67

Model: yolov3-tiny
Framework: darknet
Epochs: 6000

Augmented model is better at detecting birds!



Consideracões

- When to Perform Augmentation?
  - Augment images before training
    - ⋆ Reduces training pipeline overhead (speeds up training)
    - ⋆ Can take considerable storage space
  - Augment images during training
    - ⋆ Requires more processing power (slows down training)
    - ⋆ Reduces and simplifies storage requirements
    - ⋆ Increases statistical variation, better improvement in generalization
- Which Dataset to Augment?
  - Train: Yes ? allows network to learn robust features
  - Validation: No ? Unless planning to tweak hyperparameters during training
  - Test: No ? Test images should be from real-world conditions

Consideracões

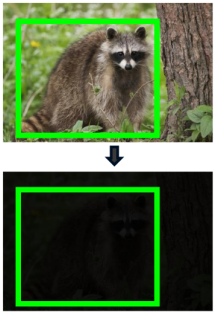
- Overall number of images
  - Generally, more is better, but there?s an upper limit to effective dataset sizes
  - Dependent on application ? if used in wide variety of visual conditions, more images are needed
  - Pete Warden rule of thumb: 1000 images per class if training from scratch
- Ratio of augmented to original images
  - If only 50 images are used to create 5000 augmented images, model will be heavily biased towards 50 original images
  - Ratio between 5:1 and 10:1 works well
- Resolution of images
  - Resolution should be similar to resolution of camera/video/images that will be used in application
  - Higher resolution requires much more memory during training

Object Detection Data Augmentation

Limitacões

Augmentations must be label-preserving

- Unsafe transformations can destroy label data



Augmentations must be dataset aware

- Transformation can change label or be nonsensical

Can't overcome a limited dataset

- Can't "create" new classes from other classes

Conclusões

- Data augmentation can increase model accuracy with minimal effort
- Random background augmentation technique is very effective for static 2D objects
- Need to consider when and where to apply augmentation, and how much
- Data augmentation has some limitations

Referencias

- Practical Image Data Augmentation Methods for Training Deep Learning Object Detection Models, a Presentation from EJ Technology Consultants
  - https://www.edge-ai-vision.com/2021/01/practical-image-data-augmentation-methods-for-training-deep-learning-object-

Curso Inteligência Artificial: do Zero ao Infinito

Object Detection Data Augmentation

Universidade Federal de Mato Grosso