

Lesson 2

Saturday, 18 January 2020

13:30

9. Optimizations on the Pre-Trained Models

- Lower the precision: this means less memory and less compute power needed
- Layers fusion: fuse different operations into one single operation

Lower precision drawback: less accuracy

Improvements achieved by using the Model Optimizer in OpenVino

10. Choosing the Right Model for Your App

Search a model that solves a similar task with respect to the one we want to solve. Test and compare the results between different models that solve the similar tasks.

Some processing could be needed in order to use a model that is suited for the different task.

Also some processing could bring to better results, provided that we are able to make it efficient.

11. Pre-processing Inputs

Needed in order to feed the input data into the model we want to use, so the work will vary based on the model we'll use.

- Colors channels order is one of the things to look at: OpenVINO and OpenCV for example use the BGR format, while other libraries can use other formats like RGB. So colors channels order conversion may be needed.
- Images size
- Inputs order (if the model accepts the image size before and then the number of channels, or if it accepts the width first and the height then, etc.)
- Normalization: some models need the pixel values between 0 and 1, others want them between -1 and 1 and so on

OpenCV library has a lot of functions for image preprocessing: cv2.imread reads images in BGR format, cv2.resize resize them.

12. Excercise

Notes:

cv2.imread output: (H, W, C)

Human Pose Estimation: [human-pose-estimation-0001](#)

Model Input: BxCxHxW = 1x3x256x456

Excercise input image:

Shape: (750, 1000, 3) = (H, W, C)

Preprocessing:

Resize: (750, 1000, 3) -> (256, 456, 3)

Transpose: (256, 456, 3) -> (3, 256, 456)

Reshape: (3, 256, 456) -> (1, 3, 256, 456)

Text Detection: [text-detection-0004](#)

Model Input: BxCxHxW = 1x3x768x1280

Excercise input image:

Shape: (667, 1000, 3) = (H, W, C)

Preprocessing:

Resize: (667, 1000, 3) -> (768, 1280, 3)

Transpose: (768, 1280, 3) -> (3, 768, 1280)

Reshape: (3, 768, 1280) -> (1, 3, 768, 1280)

Determining Car Type & Color: [vehicle-attributes-recognition-barrier-0039](#)

Model Input: BxCxHxW = 1x3x72x72

Excercise input image:

Shape: (562, 1000, 3) = (H, W, C)

Preprocessing:

Resize: (562, 1000, 3) -> (72, 72, 3)

Transpose: (72, 72, 3) -> (3, 72, 72)

Reshape: (3, 72, 72) -> (1, 3, 72, 27)