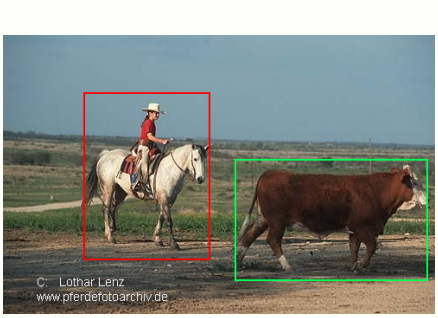
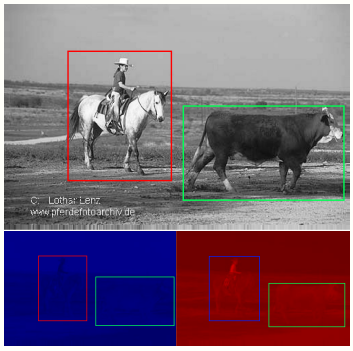
Object Detection in the DCT Domain: is Luminance the Solution ?

Introduction

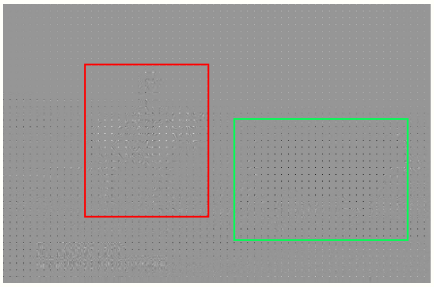
- While efficient, most of the current deep-learning detection methods rely on RGB inputs
- In order to be stored and transferred efficiently, data must be compressed, hence to run detection, data must be decompressed first
- Industrial applications require detection systems that have low computational and bandwidth requirements
- Recent work has shown that classification can be done on compressed data [1]
- JPEG encoding is a lossy compression algorithm that relies on the fact that the human eye is more sensitive to the luminance to improve the compression ratio by discarding information from the other image's components.
- *The objective*: explore the possibility to run object detection using the DCT representation of images as well as only the "most important" part of images information (luminance) in order to lower computational and bandwidth requirements



(a) Original RGB image



(b) YC_bC_r representation of the data



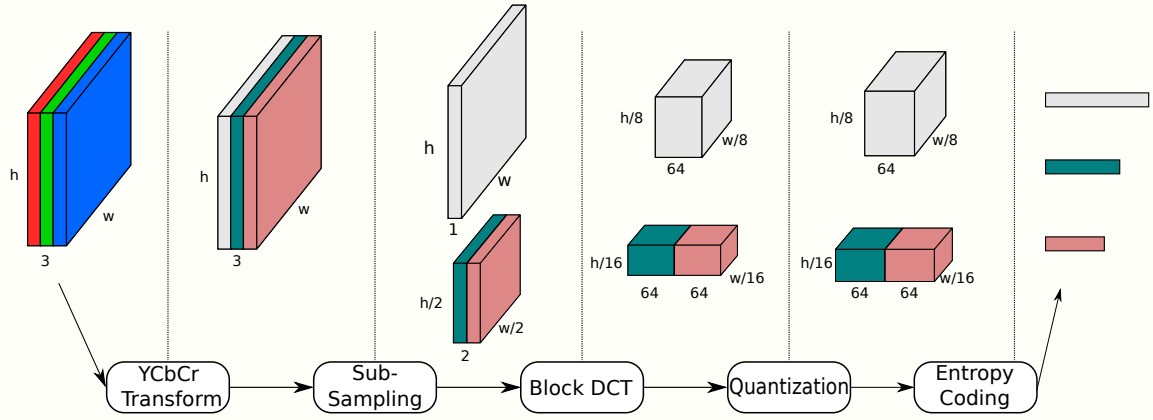
(c) Y DCT representation of the data

Figure: The different representations of the original data (a). In this article we use (c) as input for our networks.

JPEG Compression

The JPEG compression steps are as follow:

- First the RGB image is converted to the YC_bC_r domain
- Second, some/all of the components are subsampled (usually C_bC_r by 1/2 as the human eye is less sensitive to them)
- Third, a block-wise DCT is computed over the image
- The DCT blocks are quantized to further compress the image by removing frequencies to which, again, the human eye is less sensitive
- Finally, RLE/Huffman entropy coding is used to get the final representation of the compressed data



Benjamin Deguerre, Clement Chatelain, Gilles Gasso

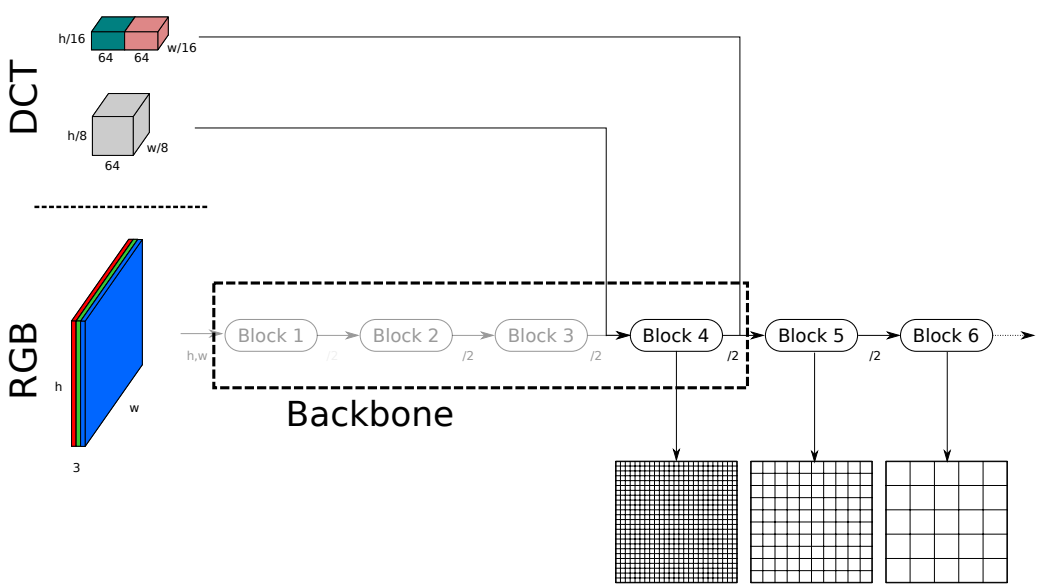
firstname.lastname@insa-rouen.fr

Laboratoire d'Informatique, du Traitement de l'Information et des Systèmes (LITIS)

Object Detection in the DCT Domain: is Luminance the Solution ?

Proposed Approach

- Use the blockwise compression to bypass (or reduce the number of) the first layers
- Use either only Y as input or Y, C_b and C_r
- We modify the SSD300 [2] to take the DCT representation as input:
  - We hard prune the first blocks of the VGG backbone in the original SSD architecture (figure below)
  - We modify the original SSD to use the ResNet50 as a backbone (more blocks are kept for feature extraction)
  - We test deconvolution versions of the networks to try to provide the smallest box predictors with C_bC_r information



Prediction results

- The VGG based architectures (hard pruning the backbone) are the fastest but lag behind when compared with the RGB accuracy
- The Resnet50 based architectures are more accurate than the VGG ones, but also slower
- The LC-RFA-Thinner model provide with the best speed/accuracy compromise
- The deconvolution based networks give the overall worst results, either with the lowest accuracy (VGG based) or the lowest speed (Resnet50 based)
- Using YC_bC_r seems to be equivalent to using Y only

Table: Detection results on the Pascal VOC 07 test set (trained on 07+12 train/val) and MS-COCO test set (trained on MS-COCO train/val). A star corresponds to best DCT results between the YC_bC_r and Y version of an architecture.

Network	Pascal VOC		MS-COCO		
	Full Input	Y only	Full Input	Y only	FPS
VGG based:					
RGB	**74.0**	-	24.5	-	100
DCT	*60.0**	59.8	14.3	*14.4**	270
DCT (Deconvolution)	53.5	-	13.5	-	**280**
ResNet50 based:					
RGB	73.1	-	**26.8**	-	105
DCT (LC-RFA)	70.7	*71.0**	*25.8**	25.2	110
DCT (LC-RFA-Thinner)	67.5	*70.2**	*25.4**	24.6	175
DCT (Deconvolution-RFA)	68.8	-	25.9	-	100

Benjamin Deguerre, Clement Chatelain, Gilles Gasso

firstname.lastname@insa-rouen.fr

Laboratoire d'Informatique, du Traitement de l'Information et des Systèmes (LITIS)

Object Detection in the DCT Domain: is Luminance the Solution ?

Future Work

- Test the impact of changes in the JPEG compression parameters
- Test transferability from trained RGB based networks to DCT based networks
- Explore the possibility to adapt this work to video compression algorithms

References

[1] Lionel Gueguen, Alex Sergeev, Ben Kadlec, Rosanne Liu, and Jason Yosinski. Faster neural networks straight from jpeg. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31, pages 3933–3944. Curran Associates, Inc., 2018.

[2] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: single shot multibox detector. *CoRR*, abs/1512.02325, 2015.

Benjamin Deguerre, Clement Chatelain, Gilles Gasso

firstname.lastname@insa-rouen.fr

Laboratoire d'Informatique, du Traitement de l'Information et des Systèmes (LITIS)