DLDR: Deep Linear Discriminative Retrieval for cultural event classification from a single image

September 15, 2015

1 Team details

- Team name: CVL_ETHZ
- Team leader name: Rasmus Rothe
- Team leader address, phone number and email:
  Office ETF D115, Sternwartstrasse 7, 8092 Zurich, Switzerland
  +41 (0) 78 633 02 22
  rrothe@vision.ee.ethz.ch
- Rest of the team members: Radu Timofte, Luc Van Gool
- Team website URL (if any): `www.vision.ee.ethz.ch`
- Affiliation: Computer Vision Laboratory, D-ITET, ETH Zurich

2 Contribution details

The proposed method Deep Linear Discriminative Retrieval (DLDR) for cultural event classification from a single image extracts CNN features at 4 different scales from the test image. For classification both the raw as well as LDA-projected features are considered. Additionally the flipped representation is used. Classification is performed through the Iterative Nearest Neighbors Classifier (INNC), introduced by Timofte and Van Gool [1]. We combine the predictions of two CNNs with the VGG-16 [2] architecture which were pre-trained on different datasets (Places and Imagenet). Both networks are fine-tuned on an augmented version of the provided ChaLearn dataset.

3 Data Preprocessing

We train two separate CNNs on the provided dataset. Both share the same architecture VGG-16 [2]. We adapt the output layer of the network to have 100 neurons, one for each class. One of the networks was pre-trained on Imagenet, the other one on the Places dataset. The training data, consisting of the provided training and validation set, was randomly split into 90% used for training and 10% for testing. We kept the distribution of classes the same in both sets. The original images are augmented by adding 10 random crops from each image to the training set. Each random crop has at least half the side length of the original image.

Before extracting features, each image is resized to a quadratic shape, as this is what the CNN was trained for. Inspired by [3] we extract CNN features in a pyramidal fashion. Specifically we extract features at 4 scales. In the first level we extract features over the entire image, in the second, third, and fourth level we extract 2x2,3x3, and 4x4 regions, respectively. The regions overlap with 50%. Each region we scale up to 256×256. Then we extract the last feature layer (fc7, 4096 dimension) for 10 different crops at a size of 224×224 in each corner and the center of the image. We do the same for the flipped version of the image. The features of these 10 crops are then averaged to give the final feature representation. This results in $1^2 + 2^2 + 3^2 + 4^2 = 30$ feature representations of 4096 dimensions.

We now learn a separate Linear Discriminant Analysis (LDA) projection for each of the 4 layers. We then concatenate the LDA projected features to form a feature vector of $30 * 99 = 2970$ dimensions, representation R1. Additionally we construct a flipped representation of R1 by horizontally flipping the local representation for the 2nd, 3rd and 4th layer, called R2. Note that R2 is just a permutation of the features of R1. The LDA helps to not only reduce the dimensionality but also to embed some discriminativeness into the features. A third representation is constructed by averaging the raw CNN features of each layer separately and then taking the average over the 4 layers. In contrast to the LDA representation, this representation, R3, is high dimensional (4096 dimensions) and thus can capture the saddle details learned by the CNN.

4 Classification details

For classification we use the Iterative Nearest Neighbors Classifier (INNC) of Timofte and Van Gool [1]. For each test sample we obtain an INN representation over the training set. This sparse weight matrix is then used for classification. The probability for a given test sample for a class is the sum of the weights of all training samples of that class. As the representation is quite sparse and often results in less non-zero weights than classes, many classes will have a probability of 0. To overcome this issue we extend the formulation of INNC by additionally spreading the weights also to the nearest neighbors of the training samples, with some exponential decay. This helps to increase retrieval performance especially

on difficult samples.

INNC is applied to the feature representations separately:

1. R1 in the training set and R1 at testing
2. R1 in the training set and R2 at testing
3. R1 and R2 in the training set and R1 at testing
4. R3

Note that if we would have had R2 in the training set and R1 at testing, this would be the same as 2) as R1 and R2 just differ by permutation. We obtain those predictions for both networks, resulting in 8 predictions in total which are averaged to give the final prediction score.

5 Global Method Description

We use the pre-trained network on Imagenet from Oxford [1] and the pre-trained network on Places from the Shenzhen Institutes of Advanced Technology [2].

The qualitative evaluation revealed that the proposed method successfully discriminates between the cultural events. The failure cases are mostly for similar events (i.e. there are several carnival events) or in cases where the images contain nothing specific to the event (i.e. just a face). We consider to submit a paper to the ChaLearn workshop describing our approach [4] in depth. This would also include comparisons with other methods. The novelty of our proposed method lies in using the LDA-projected representation of CNN features at different scales, to improve classification accuracy. Furthermore we extend the formulation of INNC with weight-spreading to be able to deal with retrieval for a larger number of classes.

6 Other details

The pipeline is written in Matlab. The CNNs are trained on Nvidia Tesla K40C GPUs using the Caffe framework. The machine used for calculating the LDA projections and classification has 128 GB of memory.

The proposed method was developed over about 8 weeks with an involvement of 40% of the main author's working time. Developing and implementing took roughly 6 weeks and the last 2 weeks were used for validation and fine-tuning.

Training each of the two CNNs took about 30 hours. At test time extracting the features over all training and testing images over all 4 layers (30*10=300 extractions per image) took around 100 hours. Calculating the LDA projections and classification took around 3 hours.

[1] `http://www.robots.ox.ac.uk/~vgg/software/very_deep/caffe/VGG_ILSVRC_16_layers.caffemodel`

[2] `http://mmlab.siat.ac.cn/Places205-VGGNet/siat_scene_vgg_16.caffemodel`

We enjoyed taking part in the challenge and consider to participate again. It would be great if the online evaluation script was speeded up next time. The offline evaluation in Matlab took less than 1s vs. 90 minutes for the online version. Thanks for organizing!

References

[1] Radu Timofte and Luc Van Gool, "Iterative nearest neighbors for classification and dimensionality reduction," in *CVPR*, 2012.

[2] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv:1409.1556*, 2014.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," in *ECCV*. 2014.

[4] Rasmus Rothe, Radu Timofte, and Luc Van Gool, "Dldr: Deep linear discriminative retrieval for cultural event classification from a single image," in *ICCV Workshops*. 2015.