

Object recognition in the wild using Convolutional Neural Networks

Practical Work 05 – Transfer learning, part 2

Class A:

Professor: Stephan Robert
Assistants: Benoît Hohl, Shabnam Ataei

Class B:

Professor: Andres Perez-Urbe
Assistants: Hector-Fabio Satizabal, Simon Walther

Emails : prenom.nom@heig-vd.ch

Goals:

- Learn to create your own dataset and use data augmentation techniques
- Train and evaluate CNN models using a transfer learning approach
- Test your solution in the real-world
- Learn how to verify the relevant working of a CNN and analyze its potential limits and errors
- Prepare a complete report of the whole experiment

1. Introduction

In this practical work you will go through all the steps required to create an object classification application: dataset creation, data exploration, data augmentation, preprocessing, model selection and performance evaluation. You will need to choose the classes that you want in your dataset (e.g: book, pen, key, person, ...) and you will then create your dataset by gathering photos for each of those classes. At the end, you will be able to test your trained model embedding it on a smartphone.

You will also discover transfer learning by using an already trained MobileNetV2 model and adding your own layers on top of it.

To help you understand what region of the images is being mainly used by the neural network-based classifier to achieve a prediction you will visualize the images and a sort of “heat-map” called the Class Activation Map (CAM).

Report

Since this is a very complete experiment including a real-world test of your application, it merits a complete report including all standard sections:

1. **Introduction:** describe the context of your application and its potential uses, briefly describe how you are going to proceed (methodology), that is, what data are you going to collect (your own pictures of.... , maybe extra-pictures collected from the web, etc), what methods are you going to use (e.g., CNNs, transfer learning) .
2. **The problem:** describe what are the classes you are learning to detect using a CNN, describe the database you collected, show the number of images per class or a histogram of classes (is it a balanced or an unbalanced dataset? Small or big Data ?)

and provide some examples showing the intra-class diversity and the apparent difficulty for providing a solution (inter-class similarity).

3. **Data preparation:** describe the pre-processing steps you needed (e.g., resizing, normalization, filtering of collected images, perhaps you discarded some data). Describe the train, validation and test datasets split.
4. **Model creation:** describe how did you proceed to come up with a final model (model selection methodology, hyper-parameter exploration, cross-validation)
 - a. What hyperparameters did you choose (nb epochs, optimizer, learning rate, ...) ?
 - b. What is the architecture of your final model ? How many trainable parameters does it have?
 - c. How did you perform the transfer learning ? Explain why did you use transfer learning, e.g., what is the advantage of using transfer learning for this problem and why it might help ?
5. **Results:** describe the experiments you performed with the model both off-line (running in your notebooks with your own-collected images) and on-line (running in the smartphone and processing images captured in the “wild”).
 - a. Provide your plots and confusion matrices
 - b. Provide the f-score you obtain for each of your classes.
 - c. Provide the results you have after evaluating your model on the test set. Comment if the performance on the test set is close to the validation performance. What about the performance of the system in the real world ?
 - d. Present an analysis of the relevance of the trained system using the Class Activation Map methods (grad-cam)
 - e. Provide some of your misclassified images (test set and real-world tests) and comment those errors.
 - f. Based on your results how could you improve your dataset ?
 - g. Observe which classes are confused. Does it surprise you? In your opinion, what can cause those confusions ? What happens when you use your embedded system to recognize objects that don't belong to any classes in your dataset ? How does your system work if your object is placed in a different background ?
6. **Conclusions:** finalize your report with some conclusions, summarize your results, mention the limits of your system and potential future work.