Escola Tècnica Superior d'Enginyeria de Telecomunicació de Barcelona

UNIVERSITAT POLITÈCNICA DE CATALUNYA

Fine-tuning a Convolutional Network for Cultural Event Recognition

Degree's Thesis
Audiovisual Systems Engineering

Author: Andrea Calafell Orós
Advisors: Xavier Giró-i-Nieto, Amaia Salvador and Matthias Zeppelzauer

Universitat Politècnica de Catalunya (UPC)
2014 - 2015

Abstract

This thesis explores good practices for improving the performance of an existing convnet trained with a dataset of clean data when an additional dataset of noisy data is available.

We develop techniques to clean the noisy data with the help of the clean one, a family of solutions that we will refer to as *denoising*, and then we explore the best sorting of the clean and noisy datasets during the fine-tuning of a convnet.

Then we study strategies to select the subset of images of the clean data that will improve the classification performance, a practice we will efer to as *fracking*.

Next, we determine how many layers are actually better to fine-tune in our convnet, given our amount of data.

And finally, we compare the classic convnet architecture where a single network is fine-tuned to solve a multi-class problem with the case of fine-tuning a convnet for binary classification for each considered class.

Resum

Aquesta tesis explora diverses pràctiques per millorar el rendiment d'una convnet entrenada amb un dataset que conté dades netes, quan tenim disponible un dataset addicional amb dades sorolloses.

Desenvolupem tècniques per netejar les dades sorolloses amb l'ajuda de les netes, una família de solucions a les que ens referirem com *denoising*, i després explorem la millor manera d'ordenar el dataset net i el sorollós durant l'afinació d'una convnet.

Després, estudiem estratègies per seleccionar un conjunt d'imatges del dataset net per tal de millorar el rendiment de la convnet, una pràctica a la que ens referirem com a *fracking*.

A continuació, determinem quantes capes és millor modificar durant l'afinació en la nostre xarxa, donada la nostre quantitat d'imatges.

I finalment, comparem l'estructura clàssica d'una convnet, on una xarxa es afinada per a resoldre un problema de varies classes, amb el cas on afinem una xarxa per fer una classificació binaria per cada classe.

Resumen

Esta tesis explora varias prácticas para mejorar el rendimiento de una convnet entrenada con un dataset que contiene datos limpios, cuando tenemos disponible un dataset adicional con datos ruidosos.

Desarrollamos técnicas para limpiar los datos ruidosos con ayuda de los limpios, una familia de soluciones a las que nos referiremos como *denoising*, y después exploramos la mejor manera de ordenar el dataset limpio y el ruidoso durante la afinación de una convnet.

Después, estudiamos estrategias para seleccionar un conjunto de imágenes del dataset limpio con tal de mejorar el rendimiento de la convnet, una práctica a la que nos referiremos como *fracking*.

A continuación, determinamos cuantas capas es mejor modificar durante la afinación en nuestra red, dada nuestra cantidad de imágenes.

Finalmente, comparamos la estructura clásica de una convnet, donde una red es afinada para resolver un problema de varias clases, con el caso donde afinamos una red para hacer una clasificación binaria para cada clase.

Acknowledgements

First of all, I want to thank my tutor, Xavier Giro-i-Nieto, for making possible my collaboration in this project, as well as his patience when guiding me and teaching me, and most of all, for helping me whenever he could.

I would also like to thank professor Matthias Zeppelzauer, Amaia Salvador and Daniel Manchón their work in project which made possible that we could develop all the solutions presented in this report. A special thank to Amaia for helping me in all my doubts, and Matthias, for becoming my co-tutor and guiding me during the course.

Revision history and approval record

Revision	Date	Purpose
0	15/06/2015	Document creation
1	7/07/2015	Document revision
2	8/07/2015	Document revision
3	10/07/2015	Document approbation

DOCUMENT DISTRIBUTION LIST

Name	e-mail
Andrea Calafell	acal48090@alu-etsetb.upc.edu
Xavier Giró i Nieto	xavier.giro@upc.edu
Matthias Zeppelzauer	matthias.zeppelzauer@gmail.com

Written by:		Reviewed and approved by:		Reviewed and approved by:	
Date	8/07/2015	**Date**	10/07/2015	**Date**	10/07/2015
Name	Andrea Calafell	**Name**	Xavier Giró i Nieto	**Name**	Matthias Zeppelzauer
Position	Project Author	**Position**	Project Supervisor	**Position**	Project Supervisor

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Statement of purpose

Cultural heritage is broadly considered a value to be preserved through generations. Every society has created through years collective cultural events celebrated with certain temporal periodicity, commonly yearly. These festivities may widely spread geographically, like the Chinese New Year's or Indian Holi Festival, or much more localized like the Carnival in Rio de Janeiro or the Castellers in Catalonia.

However, as in any classic multimedia retrieval problem, while the acquisition and storage of visual content is a popular practice among event attendees, their proper annotation is not. Only a minority of photo and video uploaders will add the simplest form of annotation, a tag or a title, while most users will just store their visual content with no further processing. So, the goal of cultural event recognition is not only to find images with similar content, but further to find images that are semantically related to a particular type of event.

In our work, we addressed the cultural event recognition problem in photos by classifying the visual features extracted from convolutional neural networks (convnets). These convnets require to be trained with a large amount of labeled images describing the problem that must be solved. But obtaining *clean data* is expensive and requires a big human effort to manually check all the images and label them. On the other hand, downloading *noisy data* from the Internet in an unsupervised fashion is easier and cheaper. This thesis explores good practices for improving the performance of an existing convnet trained with a dataset of clean data when an additional dataset of noisy data is available. In particular, this work focuses on the clean dataset for cultural event recognition provided in the ChaLearn Challenge 2015 [3], which was augmented with a noisy dataset downloaded from Flickr.

In particular, the main contributions of this project are:

- Find the best sorting of the clean and noisy datasets during the fine-tuning of a convnet; which is a process of adjusting the parameters of a pre-trained convnet to adapt it to a new visual classification problem.
- Develop techniques to clean the noisy dataset with the help of the clean one, a family of solutions that we will refer to as *denoising*.
- Explore strategies to select the subset of images that will improve the classification performance, a practice we will refer to as *fracking*
- Determine how many layers are actually better to fine-tune in our convnet, given our amount of data.
- Compare the classic convnet architecture where a single network is fine-tuned to solve a multi-class problem with the case of fine-tuning a convnet for binary classification for each considered class.

The baseline of this project was our participation in the CVPR ChaLearn: Looking at people Challenge in 2015 [3]. Our team composed of researchers from Universitat Politècnica de Catalunya and St. Poelten University of Applied Sciences participated in the Cultural Event Classification task and achieved the second place. The submission of those results occurred during the beginning of my thesis, so the work presented in this report was posterior. We plan to submit the presented improvements in the next edition of the challenge in September 2015.

1.2 Requirements and specifications

This project has been developed as a tool that could be used for other students or developers in the future to participate in upcoming challenges by recycling or improving it .

The requirements of this project are the following:

- Improve the results obtained in our submission to the first edition ChaLearn Challenge on Cultural Event Classification in the associated workshop at the IEEE International Conference on Computer Vision (CVPR) 2015.
- Find a method to exploit the noisy dataset collected from Flickr to improve the classification performance of the clean dataset, a task in which we failed in the first submission to the ChaLearn Challenge.
- Prepare the participation in the second edition ChaLearn Challenge on Cultural Event Classification at the associated workshop at the IEEE International Conference on Computer Vision (ICCV) 2015.
- Evaluate the results and make a comparative study between them.

The specifications are the following

- Use the software platform *Caffe*[14] as the basic deep learning framework for development.
- Developed in Python and Matlab.

1.3 Methods and procedures

The baseline of this project is the solution presented in the ChaLearn Challenge 2015 on Cultural Event Classification [27]. This solution first extracted visual features from the last three fully connected layers of both CaffeNet (pre-trained with ImageNet) and a fine tuned version for the ChaLearn challenge. Then proposed a late fusion strategy that trains a separate low-level SVM on each of the extracted neural codes. The class predictions of the low-level SVMs form the input to a higher level SVM, which gives the final event scores. The solution achieved the best result by adding a temporal refinement step into the classification scheme, which is applied directly to the output of each low-level SVM. The approach penalizes high classification scores based on visual features when their time stamp does not match well an event-specific temporal

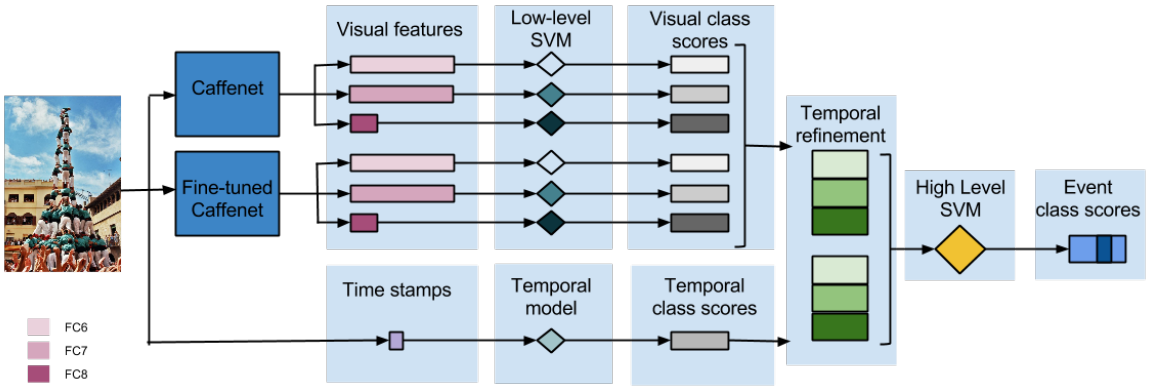


Figure 1.1: Global architecture of the solution presented in ChaLearn

distribution, learnt from the training and validation data. Figure 1.1 shows an a schematic pipeline of that solution.

An additional extension explored for the challenge submission was the addition of an external image dataset collected from Flickr, which was used to increase the number of training images for fine tuning. However, this experiment was not successful: the performance decreased when the new images were used for training. This problem is known in the literature and has been referred as cross-domain adaptation [5, 17, 10] or dataset bias [33]. For this reason, in this project we try to make this dataset useful by applying a *denoising* technique to those images before using them for training.

Moreover, we have realized that we can follow two ways when we are fine-tuning a convnet: one of them by joining all the datasets and working as it was a big one, and the other by fine-tuning first with one, and then with the next one. In this last case we analyze the best to way to order the datasets during the process.

We have also explored a method to improve the capability of a convnet to better recognize those images that are classified with a low score, referred as *fracking* by the authors in [29]. This method uses, for a given convnet model, the worst classified images in each event to train again the convnet and highlight where those images belong.

Following the lecture notes in [15], we aim to check if fine-tuning only the last layers of the convnet increases the score with respect to fine-tuning all layers with our training images. As our dataset is much smaller than the one used for training the convnet used as reference, trying to modify its learned weights of earlier layers could result in a worse perfomance.

Finally, we investigate if results improve if we perform a single convnet for each event. It means that each convnet has to recognize only one type of event, making it easier than classifying all of them.

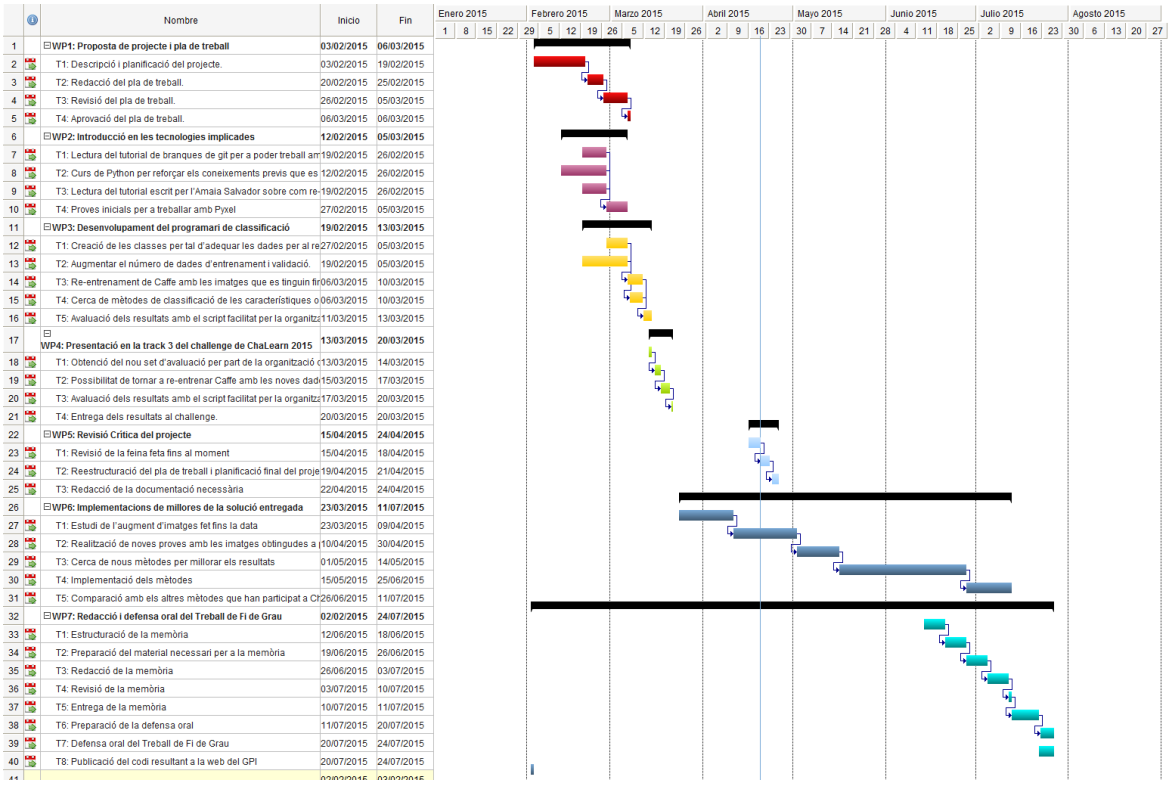
1.4 Work Plan

This project has followed the established work plan, with a few exceptions and modifications explained in the section 1.5.

1.4.1 Work Packages

- WP 1: Project propose and work plan.
- WP 2: Introduction to the involved technologies
- WP 3: Development of the classification code
- WP 4: Participation in the ChaLearn Challenge on Cultural Event Classification at CVPR 2015
- WP 5: Critical Review of the project
- WP 6: Development of improving solutions
- WP 7: Writing and presentation of the project

1.4.2 Gantt Diagram



	Nombre	Inicio	Fin
1	**WP1: Proposta de projecte i pla de treball**	**03/02/2015**	**06/03/2015**
2	T1: Descripció i planificació del projecte.	03/02/2015	19/02/2015
3	T2: Redacció del pla de treball.	20/02/2015	25/02/2015
4	T3: Revisió del pla de treball.	26/02/2015	05/03/2015
5	T4: Aprovació del pla de treball.	06/03/2015	06/03/2015
6	**WP2: Introducció en les tecnologies implicades**	**12/02/2015**	**05/03/2015**
7	T1: Lectura del tutorial de branques de git per a poder treball am	19/02/2015	26/02/2015
8	T2: Curs de Python per reforçar els coneixements previs que es	12/02/2015	26/02/2015
9	T3: Lectura del tutorial escrit per l'Amaia Salvador sobre com re-	19/02/2015	26/02/2015
10	T4: Proves inicials per a treballar amb Pyxel	27/02/2015	05/03/2015
11	**WP3: Desenvolupament del programari de classificació**	**19/02/2015**	**13/03/2015**
12	T1: Creació de les classes per tal d'adequar les dades per al re	27/02/2015	05/03/2015
13	T2: Augmentar el número de dades d'entrenament i validació.	19/02/2015	05/03/2015
14	T3: Re-entrenament de Caffe amb les imatges que es tinguin fir	06/03/2015	10/03/2015
15	T4: Cerca de mètodes de classificació de les característiques o	06/03/2015	10/03/2015
16	T5: Avaluació dels resultats amb el script facilitat per la organitz	11/03/2015	13/03/2015
17	**WP4: Presentació en la track 3 del challenge de ChaLearn 2015**	**13/03/2015**	**20/03/2015**
18	T1: Obtenció del nou set d'avaluació per part de la organització c	13/03/2015	14/03/2015
19	T2: Possibilitat de tornar a re-entrenar Caffe amb les noves dad	15/03/2015	17/03/2015
20	T3: Avaluació dels resultats amb el script facilitat per la organitz	17/03/2015	20/03/2015
21	T4: Entrega dels resultats al challenge.	20/03/2015	20/03/2015
22	**WP5: Revisió Crítica del projecte**	**15/04/2015**	**24/04/2015**
23	T1: Revisió de la feina feta fins al moment	15/04/2015	18/04/2015
24	T2: Reestructuració del pla de treball i planificació final del proje	19/04/2015	21/04/2015
25	T3: Redacció de la documentació necessària	22/04/2015	24/04/2015
26	**WP6: Implementacions de millores de la solució entregada**	**23/03/2015**	**11/07/2015**
27	T1: Estudi de l'augment d'imatges fet fins la data	23/03/2015	09/04/2015
28	T2: Realització de noves proves amb les imatges obtingudes a	10/04/2015	30/04/2015
29	T3: Cerca de nous mètodes per millorar els resultats	01/05/2015	14/05/2015
30	T4: Implementació dels mètodes	15/05/2015	25/06/2015
31	T5: Comparació amb els altres mètodes que han participat a CH	26/06/2015	11/07/2015
32	**WP7: Redacció i defensa oral del Treball de Fi de Grau**	**02/02/2015**	**24/07/2015**
33	T1: Estructuració de la memòria	12/06/2015	18/06/2015
34	T2: Preparació del material necessari per a la memòria	19/06/2015	26/06/2015
35	T3: Redacció de la memòria	26/06/2015	03/07/2015
36	T4: Revisió de la memòria	03/07/2015	10/07/2015
37	T5: Entrega de la memòria	10/07/2015	11/07/2015
38	T6: Preparació de la defensa oral	11/07/2015	20/07/2015
39	T7: Defensa oral del Treball de Fi de Grau	20/07/2015	24/07/2015
40	T8: Publicació del codi resultant a la web del GPI	20/07/2015	24/07/2015

Figure 1.2: Gantt Diagram of the Degree Thesis

1.5 Incidents and Modification

During the project we decided to increase the workload and duration of the tasks in the WP 3 at the same time as keeping the timetable. To do this, we added a task for searching methods to classify the images using the features obtained with Caffe.

The initial WP 6 consisted on participating in another scientific challenge, MediaEval 2015. However, once the challenge for this year was announced, none of the tasks was about image classification, thus we decided not to participate. Nevertheless, we thought that it was interesting to investigate why some solutions proposed for ChaLearn did not work, and implement some new solutions to improve our score. Thus, we changed the WP 6 and created new tasks and timetables, making a delay in WP 5.

Another modification that we made had to do with the main objectives of the thesis. At first, they were related to the participation to the ChaLearn challenge, but once this objective was completed, we focused on adding developing solutions to improve the results.

Chapter 2

State of the art

2.1 Convolutional Neural Networks

Deep convolutional neural networks (convnets) have recently become popular in computer vision, since they have dramatically advanced the state-of-the-art in tasks such as image classification [1], retrieval [2] or object detection [9][12].

Convnets are typically defined as a hierarchical structure of a repetitive pattern of three hidden layers: (a) a local convolutional filtering (bidimensional in the case of images), (b) a non-linear operation, (commonly Rectified Linear Units - ReLU) and (c) a spatial local pooling (typically a max operator). The resulting data structure is called a feature map and, in the case of images, they correspond to 2D signals. The higher layers in the convnet do not follow this pattern anymore but consist of fully connected (FC) layers: every value (neuron) in the fully connected layer is connected to all neurons from the previous layers through some weights. As these fully connected layers do not apply any spatial constrain anymore, they are represented as single dimensional vectors, further referred in this paper as neural codes[2].

Figure 2.1 from [2] shows an example of a convnet. Purple nodes are the input and output. Green units correspond to outputs of convolutions, red units correspond to the outputs of max pooling, and blue units correspond to the outputs of ReLU transform. Layers 6,7 and 8 are fully connected.

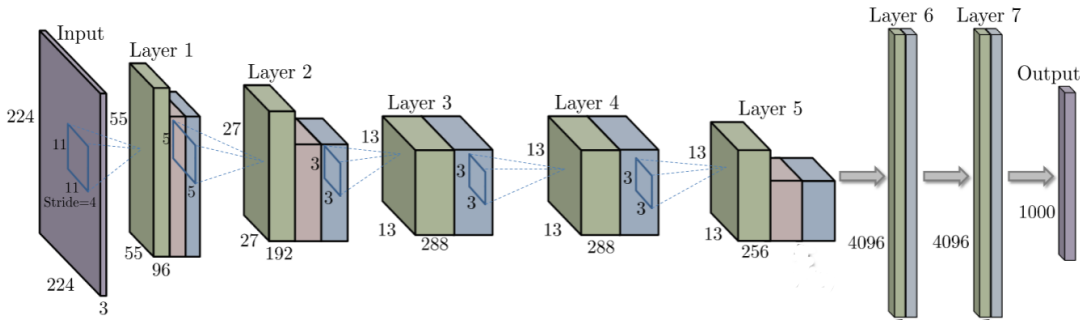


Figure 2.1: Convolutional Neural Network architecture

The amount of layers is a design parameter that, in the literature, may vary from three [20] to nineteen [30]. Some studies [36] indicate that the first layers capture finer perceptual patterns, while the deeper the level, the more complex patterns are modeled, in many cases associated to semantic concepts. However, there is no clear answer yet about how to find the optimal architecture to solve a particular visual recognition problem. The design of convnets is still mainly based on trial-and-error process and the expertise of the designer. In our work we have adopted the public implementation of *CaffeNet* [14], explained in 3.1.2

Apart from defining a convnet architecture, it is necessary to learn the parameters that govern the behaviour of the filters in each layer. These parameters are obtained through a learning

process that replaces the classic handcrafted design of visual features. This way, the visual features are optimized for the specific problems that one wants to solve. Training a convnet is achieved through backpropagation [21], a high-computational effort that has been recently boosted by the affordable costs of GPUs. In addition to the computational requirements, a large amount of annotated data is also necessary. Similarly to the strategy adopted in the design of the convnet, we have also used the publicly available filter parameters of CaffeNet [14], which had been trained for 1,000 semantic classes from the ImageNet dataset [7].

The amount of photos of annotated cultural events available in this work is much smaller than the large amount of images available in ImageNet. We have addressed the situation by fine tuning CaffeNet, that is, providing additional training data to an existing convnet which had been trained for a similar problem. This way, the network parameters are not randomly initialized, as in a training from scratch, but are already adjusted to a solution which is assumed to be similar to the desired one. Previous works [9, 12, 16] have proved that fine-tuning [13] is an efficient and valid solution to address these type of situations.

2.2 Social Event Classification

A very similar problem to Cultural Event Recognition, namely "Social Event Classification", was formulated in the MediaEval Social Event Detection benchmark in 2013 [26] [25]. The provided dataset contained 57,165 images from Instagram together with available contextual metadata (time, location and tags) provided by the API. The classification task considered a first decision level between event and non-event and, in the case of event, eight semantic classes were defined to be distinguished: concert, conference, exhibition, fashion, protest, sports, theatre/dance, other. The results over all participants showed that the classification performance strongly benefits from multimodal processing combining content and contextual information.Pure contextual processing as proposed in [31] and [11] and yielded the weakest results.

The remaining participants proposed to add visual analysis to the contextual processing. CERTH-ITI [28] combined pLSA on the 1,000 most frequent tags with a dense sampling of SIFT visual features, which were later coded with VLAD. They observed a complementary role between visual and textual modalities. Brenner and Izquierdo [4] combined textual features with the global GIST visual descriptor, which is capable of capturing the spatial composition of the scene.

The best performance in the Social Event Classification task was achieved by [23]. They combine processing of textual photo descriptions with the work from [22] for visual processing, based on bag of visual words aggregated in different fashions through events. Their results showed that visual information is the best option to discriminate between event/non-event and that textual information is more reliable to discriminate between different event types.

2.3 Convnets for Cultural Event Recognition

Our participation in the ChaLearn Challenge for Cultural Recognition in [3] has allowed us to compare our technique with other state of the art solutions in a very fair fashion. This sub-section reviews the contributions of other selected participants. It must be noticed, though, that these

papers were published during the final stage of this thesis, preventing us to incorporate some of the interesting ideas in our pipeline due to the time limitations

2.3.1 Event Recognition Using Object-Scene Convolutional Neural Networks

This proposal presented in [35] was the winning one in ChaLearn Challenge on Cultural Event Classification in CVPR 2015. The winners proposed a new architecture based on the Object-Scene Convolutional Neural Network (OS-CNN), and adapt the deep (AlexNet) [1] and very-deep (GoogLeNet) [32] networks to the task of event recognition.

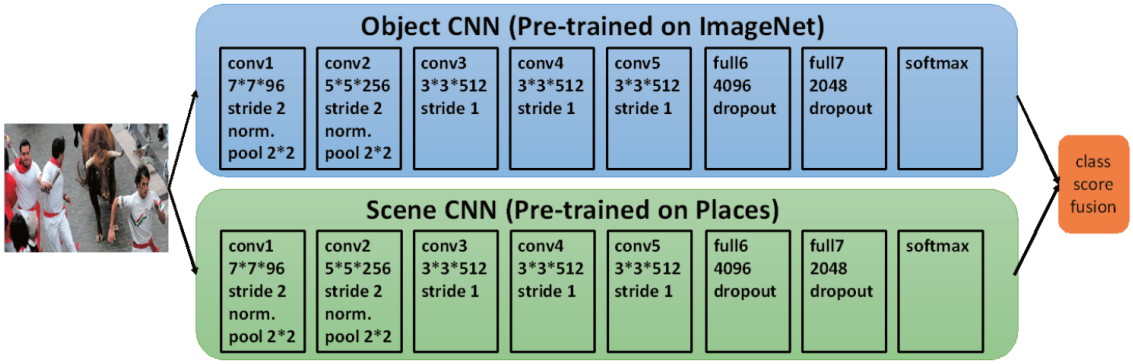


Figure 2.2: Architecture of Object-Scene Convolutional Neural Network for event recognition

The object stream, pre-trained in a large object dataset (ImageNet) [8], carries information about object depicted in the image. The scene stream, pre-trained in large scene dataset (Places) [38], captures the pattern abut scene context of this image.

- **Object Net:** They choose the Clarifai network architecture and use the pre-trained model from VGG group [6]. Then, they fine-tune the model parameters for the task of event recognition on the training dataset provided by ChaLearn.
- **Scene Net:** They use the pre-trained model in Places dataset, which chose the AlexNet architecture. Then, they fine-tune the model parameters on the training dataset provided by ChaLearn.
- **Final Score:** The score from multiple object and scene nets are combined using late fusion

2.4 Recognizing Cultural Events in Images: a Study of Image Categorization Models

The method presented in [18] is based on Least-Squares Support Vector Machines (LSSVM) and consider three types of local features: SIFT, Color and CNN. They use a combination of Spatial Pyramid Matching (SPM) [19] using as a features SIFT and Color, and a RMP using cNN. Both SPM and RMP are LSSVM:

- **Spatial Pyramid Matching (SPM) [19]:** is a popular approach for image categorization. It works by partitioning an image into increasingly fine sub-regions and aggregating local features found inside each sub-region. They use a SPM model with three levels, as shown in the figure 2.3.

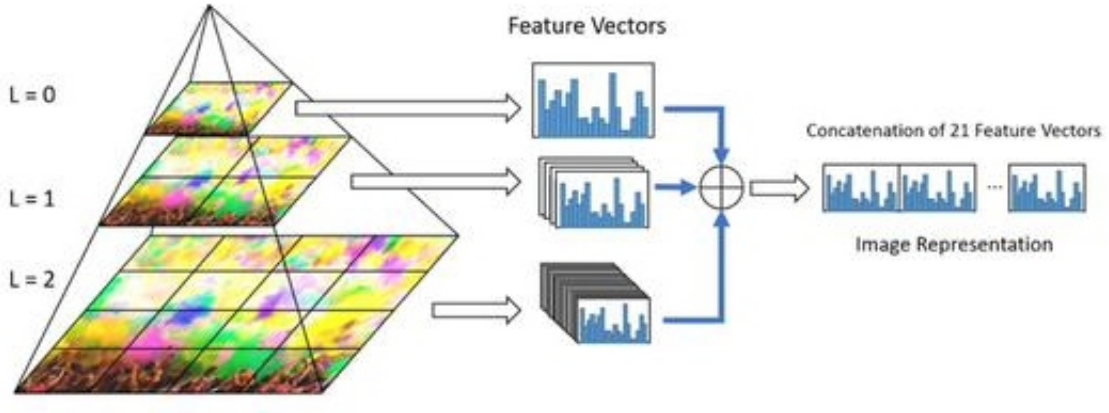


Figure 2.3: Spatial Pyramid Matching structure

- **Regularized Max Pooling (RMP):** it works by partitioning an image into multiple regions and aggregating local features computed for each region, as in the case of SPM. However, RMP does not rely on rigid geometric correspondence of grid division. In RMP, the grid division has limited discriminative power for recognizing semantic category with huge variance in location or large deformation. The region can be considered as a parts at different locations and scales.

2.5 Cultural Event Recognition by Subregion Classification with Convolutinal Neural Network

The technique [24] depicted in the Figure **??** detects regions of the image with different sizes and classifies those regions individually using convolutional neural networks. Finally, the class probabilities of each region are gathered together to produce the final outuput for the query image.

By extracting distinctive and meaningful regions from an image, they assume that the discriminant features are at the object levels. The possible object localizations are extracted via selective search [34], which combines an exhaustive search and segmentation. They exclude from the extracted candidate regions those that are too small, too tall or wide regions. Then they apply the CNN to classify these regions. Before gathering the classification results of each region, the image regions whose class probability distribution have high entropy are discarded, because they are considered to contain not much discriminant information. Finally, the classification probabilities are combined by average.

Chapter 3

Methodology

3.1 Baseline

The baseline result considered in this thesis was our submission to the ChaLearn Cultural Event challenge in CVPR 2015 [27]. I participated in this first submission during the early stages of my thesis, not as a main author, was as an exercise to become familiar with the tools and datasets considered in the problem. This participation provided me by the sufficient level of autonomy to proceed with the advances reported in the later sections, which represent the core of my work and contributions.

3.1.1 Datasets

Two datasets were used during the development of my work. A first one provided by the organizers of the ChaLearn challenge, and a second one created by our team at UPC.

3.1.1.1 ChaLearn

The organizers of the ChaLearn Challenge provided a dataset cultural events and their corresponding labels. The dataset was already divided in the three classic partitions to developed a supervised learning experimentation. The first partition is the *traininig* dataset, which contains 5,875 images, the second one is the *validation* dataset containing 2,332 images and, finally, the *test* dataset includes 3,569 images. Labels were also provided for the training and validation datasets, not for the test one to ensure a fair evaluation of the results. These images were classified between 50 cultural events, from which are also provided the country where the event is original from. The dataset was created by downloading photos from *Google Images* and *Bing* search engines, which were carefully labeled by organisers to create a very clean dataset.

3.1.1.2 Flickr

As previous works[16][1][36] have reported gains when applying some sort of data augmentation strategy, we considered that our results may improve by adding additional training data downloaded from Flickr.

Flickr is a photo repository with a public API that allows to query its large database of photos and filter the obtained results by tags, textual data search and geographical location. For the challenge 3 sets of images from Flickr[1] were generated, each of them introducing a higher degree of refinement:

[1]More details in Section 5.2 of [27], included in the Annexes

1. **90k set:** Around 90.000 photos retrieved by matching the provided event title on the Flickr *tags* and *content* metadata fields.

2. **21k set:** The query from the 90k set as combined with a GPS filtering based on the provided country.

3. **9k set:** The query from the 21k set was further filtered with manually selected terms from the Wikipedia articles related to the event. In addition, the Flickr query also toggled on an *interestingness* flag which improved the diversity of images in terms of users and dates. Furthermore, as explained in the Section 3 of the paper [27], a temporal model for each event was created. These temporal models were also used to improve the likelihood that a downloaded photo actually belonged to a certain event. For each image the day of capture was extracted from its metadata and retrieved the score from the temporal model of its class. Then, the score was threshold to remove the items that are unlikely from the temporal model.

The *Flickr dataset* used during this project corresponds to 9k set filtered with a threshold of 0.9, which result in 4,068 images.

3.1.2 Fine-tuning CaffeNet

Caffe[14] is a deep learning framework from Berkeley Vision and Learning Center (BVLC) that allows to extract features from a convolutional neural network. Caffe includes the *CaffeNet* pre-trained convnet, which was inspired by AlexNet [1]. Its architecture is composed by 5 convolutional layers and 3 fully connected layers, whose parameters were trained on 1,200,000 ImageNet images and 1,000 object classes.

As we introduced in 2.1, fine-tuning is the process adapting an already learned model to a novel classification model. Starting from an existing model allows a faster and more stable convergence than if starting with random values.

There are a few steps to follow to fine-tune a convnet with Caffe[2]:

1. **Preparation of the data:** Caffe needs a training and a validation dataset, which are read from two different text files. These text files must contain, for each image, the path where it is located and its semantic class. The train dataset is used to learn the weights in each layer, while the validation dataset is used to compute the accuracy and provide intuition about the evolution of the training process.

2. **Definition of the new architecture:** A new architecture adapted to the novel classification problem must be create in a file. The file defines parameters such as: number of layers, filters, paths to the input data, etc.

   The best practice to create such file is copying the one defining CaffeNet and change the parameters to fit our needs. The main parameters to be changed are:

   - the *source parameters* to point it to our dataset files

[2] `http://caffe.berkeleyvision.org/gathered/examples/finetune_flickr_style.html`

- the *blobs_lr*: is a specific multiplier that is applied to the base learning rate. If we do not want to change the weights of a layer, this parameter should be 0, but if we want to tune the weight of a layer we do not have to modify the original.
- If we want to completely change a layer we have to rename it and modify it as we want. For example change the *num_output* to specify a new size for the output vector. The instances where the old name appeared in the files have to be changed too. During training this layer will be detected as a new one. Thereby, the weights of this layer will be initialized from a random values instead of the weights from CaffeNet.

  In our case, for example, the only layer that we changed was the last one, because we wanted to change the output vector size from 1000 to 50, which corresponds to the number of events in our dataset.

3. **Definition of the training parameters:** The *solver* is the file that specifies the training parameters, such as: the number of iterations, the testing interval, the base learning rate, or if we are running in GPU or in CPU. It is necessary to change the *net parameter* to point it to the network file that we have created before, and the *snapshot_prefix* which points to the folder where we want to save the model.

   It is important to know that the number of epochs, which corresponds to the number of times each image passes through the network, are fixed with the parameters that we define in the solver and in the network:

$$epochs = \frac{num\_iterations \times batch\_size}{training\_samples}$$

   where:

   - *num_iterations* is the number of times that a random set of images with the size defined in the *batch_size* passes through the network.
   - *batch_size* is the number of images that go through the network at once.
   - *training_samples* is the number of images used for training.

4. **Fine-tune:** Finally we can fine-tune the network. We need to specify where the solver is, and where our reference model is, for example we can use CaffeNet.

Once we have our fine-tuned network, we can classify our images. The classification output is a prediction, generated by our trained network, which corresponds to the score given to the image for each one of the 50 classes (events).

3.2 Dataset ordering during fine-tuning

As presented in section 3.1.1, we have different two datasets: the one provided by ChaLearn and the other one downloaded from Flickr. We explored different schemes to use them both, which are summarized in Figure3.1:

- **Joint method:** We can join the two datasets as if it was a single one. To do this, we only have to join all the train and all the validation text files prepared for Caffe as we explained in 1. Then we can fine-tune using CaffeNet as a reference model, which provides the initialization weights, as explained in Section 3.1.2. This was the approach used in our baseline solution [27]

- **Sequential method:** The two datasets are used separately in a sequential scheme. To do this, we initially use only one dataset, starting from the reference model of CaffeNet. Then, we fine-tune with the other dataset, but instead of using CaffeNet as a reference model, we use the one re-trained with the first dataset. If we want to fine-tune another dataset, we should start from the one trained with the second dataset, and so on. In our case, where two datasets are considered (clean and noisy), this method can be applied in two different configurations, depending on the sorting.

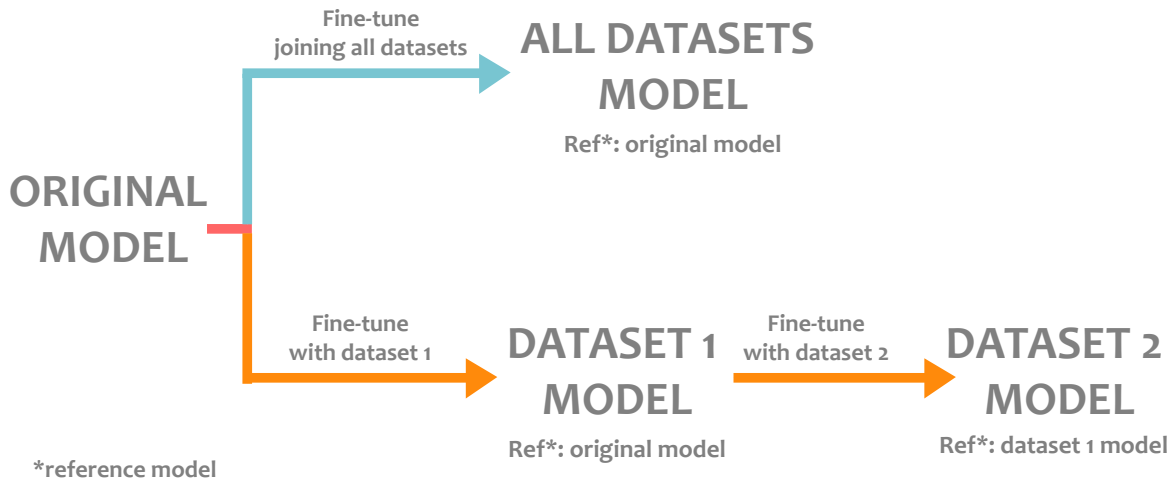


Figure 3.1: Two options to combine two datasets for fine-tunning the original model (trained on ImageNet) with data from cultural events

It is important to notice that during the fine-tunning process, Caffe will select a random set of images, defined by the *batch_size*, to pass through the network. This means than in *Joint method* will show to the network images from both datasets indistinctively, while the weights in the *sequential method* will be more adjusted to the second dataset.

The experiments on these configurations can be found in Section 4.4.

3.3 Denoising a weakly labeled dataset

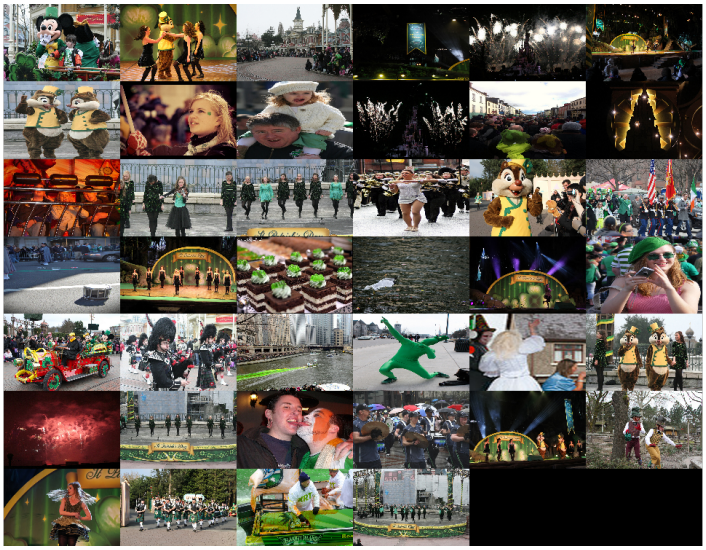
A visual inspection of some images from the clean (ChaLearn) and noisy (Flickr) datasets clearly shows the differences between both sets. As is can be observed in Figures 3.2 and 3.3, the Flickr images are much more diverse than the ones manually curated by the ChaLearn organizers. We hypothesize that this diversity in quality was one of the reasons why in our baseline experiments adding images from Flickr to fine-tuning produced a strong degradation of the performance [27].

(a) From *validation dataset*

(b) From *Flickr dataset*

Figure 3.2: Mosaic of Queens Day images

(a) From *validation dataset*



(b) From *Flickr dataset*

Figure 3.3: Mosaic of St.Patrick Day images

Based on this hypothesis, we decided to explore strategies to remove those images from the Flickr dataset which were "too different" from the ones provided by the clean ChaLearn dataset, so we called this method *denoising*, as also suggested in [5]. Our strategy consists on using a model only trained with the ChaLearn images to discard the Flickr images which are badly classified. Thus, those images that are very different from those used to train are discarded, and the remaining images are added to the training dataset.

In particular, the adopted denoising process follows these steps:

1. Classify all images in the Flickr dataset with a convnet fine-tuned with ChaLearn images.
2. For each class, generate a ranked list of the images labeled with this class according to their ground truth. The sorting of the images is based on classification score provided by the ChaLearn convnet.
3. Decide a percentage of top ranked images of each ranked list which are to be kept in the filtered list. We ran experiments with different percentage values.
4. Build a new filtered set of Flickr images that will be combined by the ChaLearn ones following one of the schemes presented in Section 3.2.

The results of this technique are reported in Section 4.4.

3.4 Fracking the training dataset

The easiness or difficulty of the samples used to train the classifiers may have an impact in the performance of the learned model. We refer to *fracking* [29] to the process of selecting those samples which will drive to a superior performance of our convnet. As suggested in [29], we will select these samples through an iterative process that will show multiple times to the network those samples which are wrongly classified. By doing this, we show multiple times to our networks those images that cause most confusion to them, with the goal that it will learn better decision boundaries between the classes.

Only those images which are not correctly classified will be re-introduced to the network for a limited amount of iterations.

The detailed steps for our fracking approach are the following:

1. Classify the training set of ChaLearn images with a convnet which was fine-tuned with those same images.
2. For each class, consider the ground truth labels to tag each image as *positive* or *negative* depending whether the prediction was correct or incorrect, respectively.
3. For each class, generate a ranked list of positive and negative images based on the prediction score assigned by the convnet.
4. Select a predefined percentage of bottom-ranked images from the *positive* list of each class. These images correspond to the worst classified belonging to the class, and we want to "remind" the convnet that they belong to the class.
5. Select another predefined percentage of top-ranked images from the *negative* list of each class. These images correspond to the best classified images that do not belong to the predicted class.
6. Build a new reduced set of training images to fine-tune again the convnet. In this case, the fine-tuning ordering must follow a *sequential method* as introduced in Section 4.5.

3.5 Fine-tuning deeper layers only

Recent visualization works on convnet layers [37] have pointed out that early layers tend to capture low-level perceptual features such as colors, textures, or contours. These basic visual primitives are common to most computer vision tasks, while deeper layers tend to adjust more specifically to the semantic classes under consideration.

Taking this into account, we realized that in our baseline system [27] we were fine-tuning all layers, also the early ones trained with 1.2 million images from ImageNet. If we assume that the low-level primitives for ImageNet and ChaLearn are the same or very similar, it is not necessary to fine-tune these early layers with our small set of 5,875 images from the ChaLearn training dataset. So, as suggested in [15], we decided to assess the gains of just fine tunning the deeper layer only.

In the Caffe software platform, we fixed the weights of the layers we did not want to fine-tune by setting the *blobs_lr* parameter to 0 in our network file, as explained in section 3.1.2.

The results of this strategy are reported in Section 4.6.

3.6 Ensemble of binary classifiers

The last strategy to explore was inspired by our SVM classifiers used in our baseline system [27] which, instead of adopting a multi-class form, they were implemented as a collection of one-vs-the rest binary classifiers, one for each cultural event. We hypothesized that the classic multi-class architecture of the soft-max classifier built on top of the last layer of the convnet may also benefit from a reformulation as a binary classifier.

So we tried is to fine-tune 50 different convnets, one for each class, instead of the single convnet for the 50 cultural events. This method is much slower than the others, but allows each convnet to train the layers to distinguish one event. It means that all weights in the network are specialized into the single task of identifying a single event. Finally, each image in the test dataset would obtain a classification score for each class, which corresponds to the *positive* class of each specialized convnet for each event.

The implementation of this experiment required the creation of one training file and one validation file for each event class. These files have to have only two classes, one class with the images that belong to the *positive* class, and a second *negative* class with all the other images.

The results of this technique can be found in Section 4.7.

Chapter 4

Results

This chapter presents the results obtained with the techniques presented in Chapter 3 to improve the baseline system described in Section 3.1.

4.1 Experimental setup

The datasets presented in Section 3.1.1 can be summarized as follows:

- **Training ChaLearn**: 5,875 images provided by ChaLearn organizers
- **Validation ChaLearn**: 2,332 images provided by ChaLearn organizers
- **Flickr dataset**: 4,068 images obtained from Flickr

We adjusted some parameters in Caffe to adapt it to our needs. We decided to set to constant all the learning rate parameters and use the same number of epochs in all the fine-tuning processes. This way it was possible to compare between the experiments, This means that we only modify the *num_iterations* in the solver file depending on the *training_samples*, as we see in equation 3. The chosen number of epochs is 270 to be able to compare it with a previous test of the fine-tuned convnet performed for the challenge. More details about the training setup were presented in Section 3.1.2

During the experimentation we have also developed a tool to visualize the classification of the images. This tool give us a mosaic of a certain number of images, sorted by the score, with a green frame in the images that belong to the class that we are visualizing, or a red frame in the images that do not belong. This tool has been used to generate the visualization of results presented in this Chapter.

4.2 Evaluation metric

The evaluation of the proposed techniques was based on the metric and evaluation script provided by the organizers of ChaLearn. This script needs as an input one text file for each event, which must contain a ranked list of all the images used for evaluation. As an output it provides a score which corresponds to the mean Average Precision (mAP) of all the classes.

The mean average precision of the system is obtained by computing the mean of the average precisions (AP) of the Q classes, as as presented in Equation 4.1.

$$MAP = \frac{1}{|Q|} AP(q) \tag{4.1}$$

At the same time, the average precision for a class by averaging the precision at position k ($P(k)$) of all images in the ranked list which actually belong to the class, which define the set R. The Precision at position k ($P(k)$) is defined as the amount of elements in R between positions 1 and k in the ranked list, divided by k. Equation 4.2 formulates this metric.

$$AP = \frac{1}{|R|}\sum_{k=1}^{N} P(k)\cdot rel(k),\ rel(k) = \begin{Bmatrix} rel(k) = 1 \Leftrightarrow k \in R \\ rel(k) = 0 \Leftrightarrow k \notin R \end{Bmatrix} \tag{4.2}$$

4.3 Dataset biases

The first set of experiments aim at measuring the difference between the ChaLearn and Flickr datasets considered in this work. This study is based on the classification performance obtained in the reference CaffeNet network when fine-tuned with each of the two datasets separately.

Table 4.1 contains the results of the experiments. Results clearly show how fine-tunning with only one of the two datasets greatly biases the classification results despite both datasets depicting the same classes.

	Chalearn Train	**70% Flickr**
ChaLearn Validation	0.61365	0.31929
30% Flickr	0.25686	0.44305

Table 4.1: Dataset bias when fine-tunning with one or two datasets

A qualitative example of the bias of the models can be appreciated by comparing Figure 4.1a with Figure 4.1b, both of them generated with the convnet fine-tuned with ChaLearn only. While the top ranked ChaLearn images from Figure 4.1a are mostly correctly classified (i.e. those that have the green frame, and are in the firsts positions), the Flickr images with highest scores in Figure 4.1b present a high ratio of red frames, indicating that they belong to another event.

(a) From *validation ChaLearn*

(b) From *Flickr dataset*

Figure 4.1: Qualitative results for the Harbin Ice and Snow Festival with fine-tunning with ChaLearn

4.4 Denoising the Flickr dataset

The clear difference between the ChaLearn and Flickr datasets measured in Section 4.3 motivates the main goal of this section, which is being able to use Flickr images to improve our baseline performance on ChaLearn images, which is our target dataset.

The denoising technique presented in Section 3.3 aims at filtering the list of the Flickr images to improve the fine-tuning for ChaLearn data. As a reminder to the reader, this technique was based on ranking the Flickr images for each event based on their classification score with the convnet fine-tuned with ChaLearn. The amount of images to include in the filtered list depends on a predefined percentage of top images to consider, so our results are provided depending on this value.

As previously introduced in Section 4.4, there exist three possibilities to combine the two available clean (ChaLearn) and noisy (Flickr) datasets: a first option with a *joint* fine-tuning or the two configurations that results from a *sequential* scheme, depending on the order of the datasets.

4.4.1 Joint fine-tuning of the clean and noisy datasets

Figure 4.2 depicts the results of classification the ChaLearn validation set with a convnet fine-tuned jointly with all ChaLearn training set plus a percentage of the Flickr dataset. Results should be compared with the 0.61365 shown in Table 4.1, which corresponds to the results when no Flickr data is considered at all.

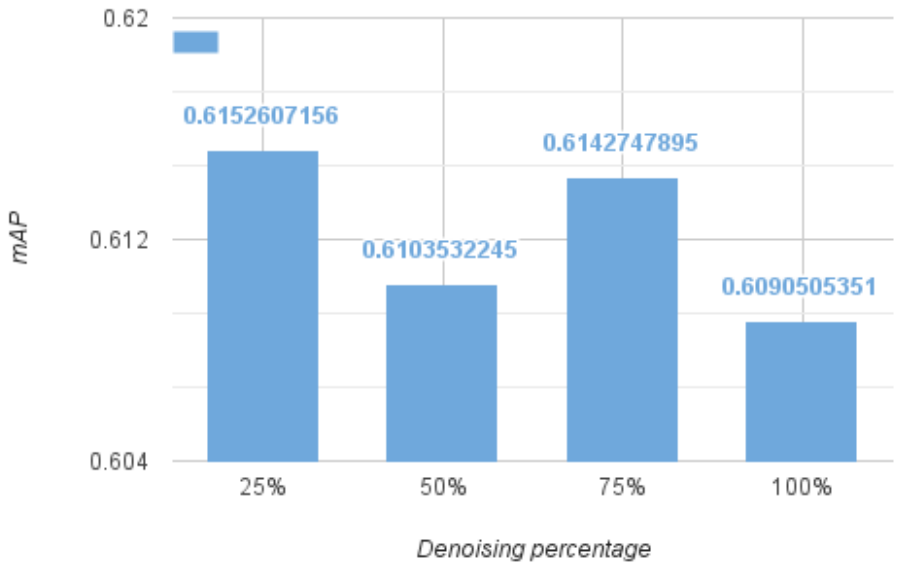


Figure 4.2: Results of fine-tuning joining as a training dataset the *Training ChaLearn* and the denoised *Flickr dataset*

We observe that in this case there is a slight improvement by the fact of using a filtered list of Flickr images, being the best result the case where only the top 25% ranked images are considered.

However, no strong conclusions should be drawn from these results. The figures are very similar between them and the small variations in the scores may be due to the randomly chosen sets of images that the network uses to train in each iteration, or the random weight values that takes in the initialization in the last layer.

4.4.2 Sequential fine-tuning of clean and noisy dataset

The first sequential scheme that was tested by fine-tuning with the clean dataset (ChaLearn) first and the noisy one (Flickr) in a second stage. This was actually the approach adopted in our baseline approach [27].

The results presented in Figure 4.3 indicate this is actually a very bad solution because the more noisy images are introduced, the worse the performance is. Not even the case of 25% improves the 0.61365 obtained with no Flickr images at all. As a conclusion, this configuration should be avoided.

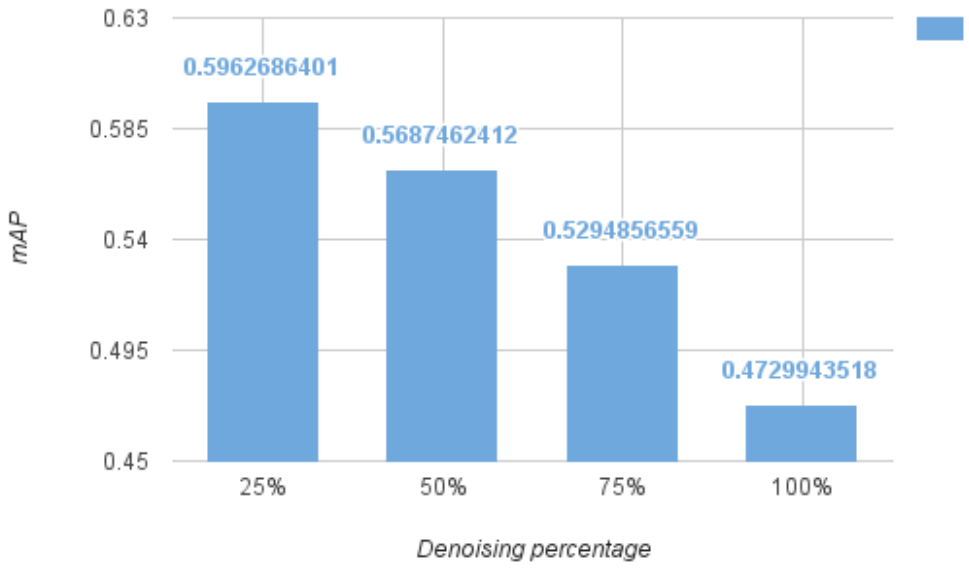


Figure 4.3: Results of fine-tuning using as a first training dataset the *training ChaLearn* and as a second the denoised *Flickr dataset*

4.4.3 Sequential fine-tuning of noisy and clean dataset

Finally, we performed one last experiment changing the order of the datasets in the fine-tuning.

The results depicted in Figure 4.4 explain a completely different story from the previous section. Not only using the top 25% Flickr images increases the 0.61365 achieved with ChaLearn images only, but not filtering Flickr images at all even increases the performance.

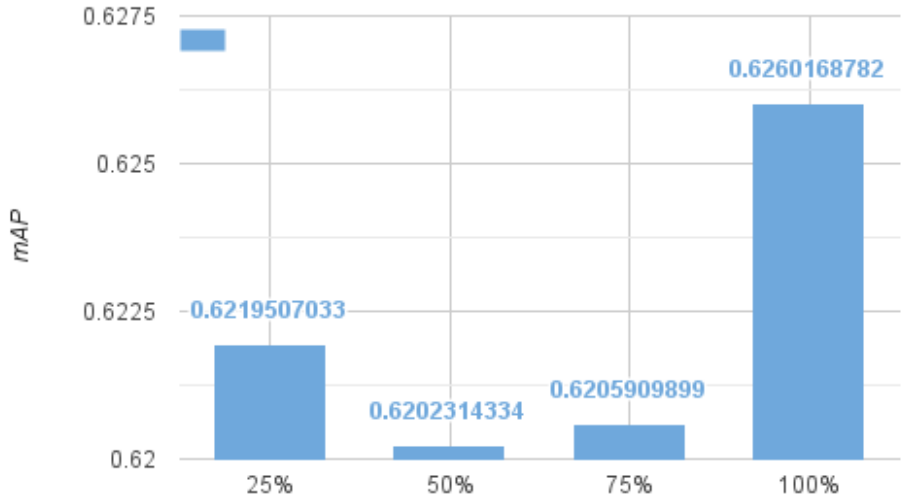


Figure 4.4: Results of fine-tuning using as a first training dataset the denoised *Flickr dataset* and as a second the *training ChaLearn*

These results were actually surprising to us because they indicate that the problem was not actually in filtering the noisy Fkickr images, but on how we were using them. Instead of using them a posteriori to the ChaLearn data, we should use them a priori. This result indicates that noisy data can be beneficial because, despite noisy, it is still more similar to the classification problem of cultural event recognition than the data that was used to train CaffeNet from scratch. While the original CaffeNet weights were optimized by the objects in the ImageNet dataset, our noisy Flickr dataset initiates an adaptation of the network to the domain of cultural event recognition. This preliminary adaptation is finally adjusted to the test data by the ChaLearn training data, when used in the later stage of the sequential fine-tuning scheme.

4.5 Fracking positive and negative images from ChaLearn

After the experiments with the different datasets presented in Sections 4.3 and 4.4, we also addressed a strategy aimed at a better use of the available training data.

As we explained in section 3.4, this method iteratively fine-tunes the network with a list training samples which were wrongly classified in the previous iteration. Similarly to *denoising*, the amount of images in the list will be set with a percentage. In this case we have a percentage for fracking positive samples, and another one for the negative ones. These percentages were chosen trying to balance the total amount of *positive* and *negative* images.

The scores shown in Table 4.2 are slightly higher than the 0.61365 obtained with the whole ChaLearn training dataset. This means that the network actually succeeds in improving his performance by learning from its own mistakes.

Positive percentage	**Negative percentage**	**Amount of fracking images**	**Test set**	**mAP**
10%	0,5%	1606	Validation ChaLearn	0.61694
20%	2%	3959	Validation ChaLearn	0.62268
15%	3%	4446	Validation ChaLearn	0.62221

Table 4.2: Results of fine-tuning using fracking

4.6 Fine-tunning FC6, FC7 and FC8 only

As introduced in the Section 3.5 results may improve if we only modify the deeper layers while keeping the weights for the first layers.

We have performed three experiments with the last fully connected layers FC6, FC7 and FC8, always using the ChaLearn training dataset for fine-tuning. In order to obtain more robust conclusions, we have also included in the study the 30% Flickr as a test dataset.

The results contained in Table 4.3 support the expectations, with a gain of more than 3 points for the ChaLearn case when comparing the fine-tuning of only the three last layers with the fine-tunning of all layers. Notice also that fine-tuning only FC8 is not flexible enough to transform the object classification task defined with ImageNet to the cultural event task targeted by our experiments.

Fine-tuned layers	**Test set**	**mAP**
fc678	Validation ChaLearn	0.6434
fc678	30% Flickr dataset	0.2748
fc78	Validation ChaLearn	0.6298
fc78	30% Flickr dataset	0.2721
fc8	Validation ChaLearn	0.5757
fc8	30% Flickr dataset	0.2511
All	Validation ChaLearn	0.6137
All	30% Flickr dataset	0.2569

Table 4.3: Results of only fine-tuning the deeper layers

The effects of fine-tunning the last three layers only can also be qualitatively observed in Figure 4.5. We can observe that in the second case there are a few more images well classified, which corresponds to the expected results with the increase of the score.

Given the good results obtained with the fine-tuning of only the fully connected layers, this finding was added in some of the most promising configurations found in the previous sections of this chapter. The results obtained are reported in Table 4.4, where it is clear that the gain is consistent for all configurations.

Fine-tuned layers	**Improvement**	**mAP**
all	None	0.6137
fc678	None	0.6434
all	Joint Chalearn + Flickr (top 25%)	0.6152
fc678	Joint Chalearn + Flickr (top 25%)	0.6467
all	20-2% fracking	0.6227
fc678	20-2% fracking	0.6486
all	Flickr+Chalearn	0.6260
fc678	Flickr+Chalearn	0.6492
fc678	Fusion	0.6532

Table 4.4: Results of only fine-tuning the deeper layers

The last row in table 4.4 corresponds to a configuration that combines all the lessons learned in a single experiment, In particular, it consists on applying *fracking* with the sequential scheme for fine-tuning of noisy (Flickr) + clean (ChaLearn) datasets.

4.7 Ensemble of event detectors

At the later stages of this thesis, we decided to try another architecture based on an ensemble binary classifiers instead of a single convnet to predict all classes. This approach is detailed in Section 3.6.

Initially we expected bad results for the experiment because the resulting networks would be trained with highly unbalanced datasets. For example in our case, for each convnet that we are fine-tuning, we specify only two classes, one class with the images that belong to the class, that we call positive, and a second with the all the other images, which would be the negatives and will have a higher amount of images. So, theoretically, when the fine-tuning is running, it will see more negative images than positive and as a result the convnet could be biased towards classifying most images as a negative. Considering our dataset, this would correspond that each binary convnet would be trained with around 100-150 positive images and 5,700 negative images.

Surprisingly, the results reported in Table 4.5 are the best obtained for all configurations tested in this thesis. The results are obtained with the fine-tuning of only the fully connected layers in each network. According to our experiments, this simple change in the architecture introduces a very significant gain of almost 7 points.

Validation dataset	**mAP**
One multi-class convnet (baseline)	0.6137
Fusion of all previous improvements	0.6532
Ensemble of binary convnets	0.67060

Table 4.5: Results of ensemble of binary

We could not extend the experimentation and analysis of this research line due to the time constraints of the thesis. Nevertheless, we plan to pursue this research line in the future to try to understand the reasons of this unexpected result.

(a) Results of fine-tuning all layers

(b) Results of fine-tuning the last three layers

Figure 4.5: Classification of Castellers images

Chapter 5

Budget

This project has been developed using the resources provided by Image Processing Group of UPC, and as it is a comparative study, there are not maintenance costs.

Thus, the main costs of this projects comes from the salary of the researches and the time spent in it. I consider that my position as well as the one of the Phd student supervising me was of junior engineer, while the two professors who were advising me had a wage/hour of a senior engineer. I will consider that the total duration of the project was of 25 weeks, as depicted in the Gantt diagram in Figure 1.2.

	Amount	**Wage/hour**	**Dedication**	**Total**
Junior engineer	1	8,00 €/h	30 h/week	6,000 €
Junior engineer	1	8,00 €/h	4 h/week	800 €
Senior engineer	2	20,00 €/h	4 h/week	4,000 €
			Total	10,800 €

Table 5.1: Budget of the project

Chapter 6

Conclusions

The main objective of this project was to improve the baseline configuration we submitted in the ChaLearn Challenge of Culural Event Recognition at CVPR 2015. Our submission was awarded with the second prize in a recognized scientific venue and our work highlighted in the front page of our ETSETB TelecomBCN school [1]. Compared to that baseline, the contributions reported in this thesis have improved performance in around 5-7 points. For this reason, I consider that the main goal of the thesis has been accomplished and I am already looking forward to submit our new results in the next edition of the challenge in September 2015.

Reaching this point though has not been a straight line but full of turns and dead ends. At first, we wanted to take advantage of the Flickr images analyzing why they did not help. We realized that the Flickr images are noisier than the datasets provided by ChaLearn, so we tried a denoising method. We were able to obtain a cleaner *Flickr dataset* by filtering it with the model obtained from ChaLearn images. Unfortunately, the performance did not improve. But thanks to those experiments, we noticed the importance of the method you follow and the order during the fine-tuning, and the *Flickr dataset* finally helped us to improve the score by swapping the order in which we were using the clean and the noisy dataset.

Then, we tried improving the discriminative capability of the network applying *fracking*. In this case we reached some gain than in *denoising*, but it still was very little.

Furthermore, we wanted to find the optimum number of layers to fine-tune, so we tried some combinations with the later layers, and we realized that fine-tuning only the last three layers gave the best results. Thus we concluded that, for the first layers, it is better to keep the the weights learned from a very large dataset such as ImageNet.

As an unexpected result, we realize that fine-tuning one convnet for each class increases the score, at least with our type of data. We plan to study this result with more attention in the future, as this was an experiment run during the later stage of this thesis.

As a future work, we want to mix our solutions with a fine-tuned network with PLACES, the model used by the winners of the challenge, and compare the results with the ones obtained in this project. Additionally, we want to repeat the *ensemble of binary* experiment, trying to balance the data with some data augmentation, or by balancing the batches.

[1] `https://www.etsetb.upc.edu/mason-share/notif/2806.html`

Chapter 7

Appendices

As appendices we can found our paper presented in the CVPR 2015 [27], and the diploma of the 2nd prize received.

Cultural Event Recognition with Visual ConvNets and Temporal Models

Amaia Salvador*, Matthias Zeppelzauer**, Daniel Manchón-Vizuete*, Andrea Calafell*, and Xavier Giró-i-Nieto*

*Universitat Politecnica de Catalunya (UPC), Barcelona, Catalonia/Spain, amaia.salvador@upc.edu
**St. Poelten University of Applied Sciences, St. Poelten, Austria, matthias.zeppelzauer@fhstp.ac.at

Abstract

This paper presents our contribution to the ChaLearn Challenge 2015 on Cultural Event Classification. The challenge in this task is to automatically classify images from 50 different cultural events. Our solution is based on the combination of visual features extracted from convolutional neural networks with temporal information using a hierarchical classifier scheme. We extract visual features from the last three fully connected layers of both CaffeNet (pre-trained with ImageNet) and our fine tuned version for the ChaLearn challenge. We propose a late fusion strategy that trains a separate low-level SVM on each of the extracted neural codes. The class predictions of the low-level SVMs form the input to a higher level SVM, which gives the final event scores. We achieve our best result by adding a temporal refinement step into our classification scheme, which is applied directly to the output of each low-level SVM. Our approach penalizes high classification scores based on visual features when their time stamp does not match well an event-specific temporal distribution learned from the training and validation data. Our system achieved the second best result in the ChaLearn Challenge 2015 on Cultural Event Classification with a mean average precision of 0.767 on the test set.

1. Motivation

Cultural heritage is broadly considered a value to be preserved through generations. From small town museums to worldwide organizations like UNESCO, all of them aim at keeping, studying and promoting the value of culture. Their professionals are traditionally interested in accessing large amounts of multimedia data in rich queries which can benefit from image processing techniques. For example, one of the first visual search engines ever, IBM's QBIC [9], was showcased for painting retrieval from the Hermitage Museum in Saint Petersburg (Russia).

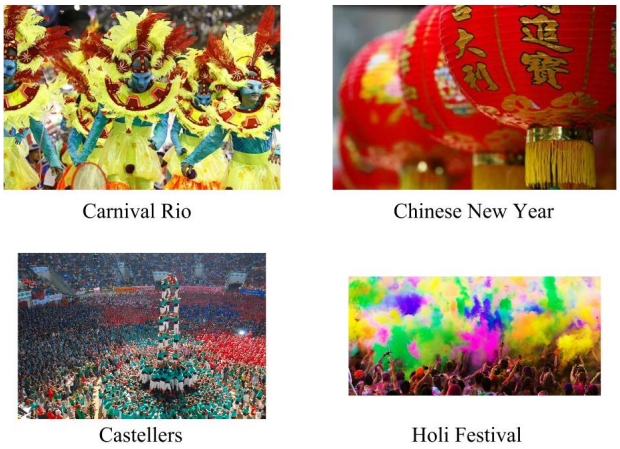


Figure 1. Examples of images depicting cultural events.

A cultural expression which is typically not found in a museum are social events. Every society has created through years collective cultural events celebrated with certain temporal periodicity, commonly yearly. These festivities may widely spread geographically, like the Chinese New Year's or Indian Holi Festival, or much more localized like the Carnival in Rio de Janeiro or the *Castellers* (human towers) in Catalonia. An image example for each of these four cultural events is presented in Figure 1. All of them have a deep cultural and identity nature that motivates a large amount of people to repeat very particular behavioral patterns.

The study and promotion of such events has also benefited from the technological advances that have popularized the acquisition, storage and distribution of large amounts of multimedia data. Cultural events across the globe are at the tip of a click, improving both the access of culture lovers to rich visual documents, but also their touristic power or even exportation to new geographical areas.

However, as in any classic multimedia retrieval problem,

while the acquisition and storage of visual content is a popular practice among event attendees, their proper annotation is not. While both personal collections and public repositories contain a growing amount of visual data about cultural events, most of it is not easily available due to the almost non-existent semantic metadata. Only a minority of photo and video uploaders will add the simplest form of annotation, a tag or a title, while most users will just store their visual content with no further processing. Current solutions will mostly rely in on temporal and geolocation metadata attached by the capture devices, but also these sources are unreliable for different reasons, such as erroneous set up of the internal clock of the cameras, or the metadata removal policy applied in many photo sharing sites to guarantee privacy.

Cultural event recognition is a challenging retrieval task because of its strong semantic dimension. The goal of cultural event recognition is not only to find images with similar content, but further to find images that are semantically related to a particular type of event. Images of the same cultural event may also be visually different. Thus, major research questions in this context are, (i) if content-based features are able to represent the cultural dimension of an event and (ii) if robust visual models for cultural events can be learned from a given set of images.

In our work, we addressed the cultural event recognition problem in photos by combining the visual features extracted from convolutional neural networks (convnets) with metadata (time stamps) of the photos in the hierarchical fusion scheme shown in Figure 2. The main contributions of our paper are:

- Late fusion of the neural codes from both the fine-tuned and non-fine-tuned fully connected layers of the CaffeNet [15] convnet.
- Generation of spline-based temporal models for cultural events based on photo metadata crawled from the web.
- Temporal event modeling to refine visual-based classification as well as noisy data augmentation.

This paper is structured as follows. Section 2 overviews the related work, especially in the field of social event detection and classification. Section 3 describes a temporal modeling of the cultural events which has been applied both on the image classification and data augmentation strategies presented in Section 4 and Section 5, respectively. Experiments on the ChaLearn Cultural Event Dataset [2] are reported in Section 6 and conclusions drawn in Section 7.

This work was awarded with the 2nd prize in the ChaLearn Challenge 2015 on Cultural Event Classification. Our source code, features and models are publicly available online[1].

2. Related work

The automatic event recognition on photo and video collections has been broadly addressed from a multimedia perspective, further than just the visual one. Typically, visual content is accompanied by descriptive metadata such as a time stamp from the camera or an uploading site, a geolocation from a GPS receiver or some text in terms of a tag, a title or description is available. This additional contextual data for a photo is highly informative to recognize the depicted semantics.

Previous work on social events has shown that temporal information provides strong clues for event clustering [27]. In the context of cultural event recognition, we consider temporal information a rather "asymmetric clue" where time provides an indicator to rather reject a given hypothesis than to support it. On the one hand, given a prediction (e.g. based on visual information) for a photo for a particular event, we can use temporal information, i.e. the capture date of the photo, to easily reject this hypothesis if the capture date does not coincide with the predicted event. In this case temporal information represents a *strong clue*. On the other hand, cultural events may take place at the same time. As a consequence, the coincidence of a captured date with the predicted event in this case represents just a *weak clue*. We take this "asymmetric nature" in our temporal refinement scheme (see Section 4.3) into account.

Temporal information has further been exploited for event classification by Mattive et al. [18]. The authors define a two-level hierarchy of events and sub-events which are automatically classified based on their visual information described as a Bag of Visual Words. All photos are first classified visually. Next, the authors refine the classification by enforcing temporal coherence in the classification for each event and sub-event which considerably improved the purely visual classification.

A similar approach is applied by Bossard et al. [3], exploiting temporal information to define events as a sequence of sub-events. The authors exploit the temporal ordering of photos and model events as a series of sub-events by a Hidden Markov Model (HMM) to improve the classification.

A very similar problem to Cultural Event Recognition, namely "Social Event Classification", was formulated in the MediaEval Social Event Detection benchmark in 2013 [21, 20]. The provided dataset contained 57,165 images from Instagram together with available contextual metadata (time, location and tags) provided by the API. The classification task considered a first decision level between

[1] https://imatge.upc.edu/web/resources/cultural-event-recognition-computer-vision-software

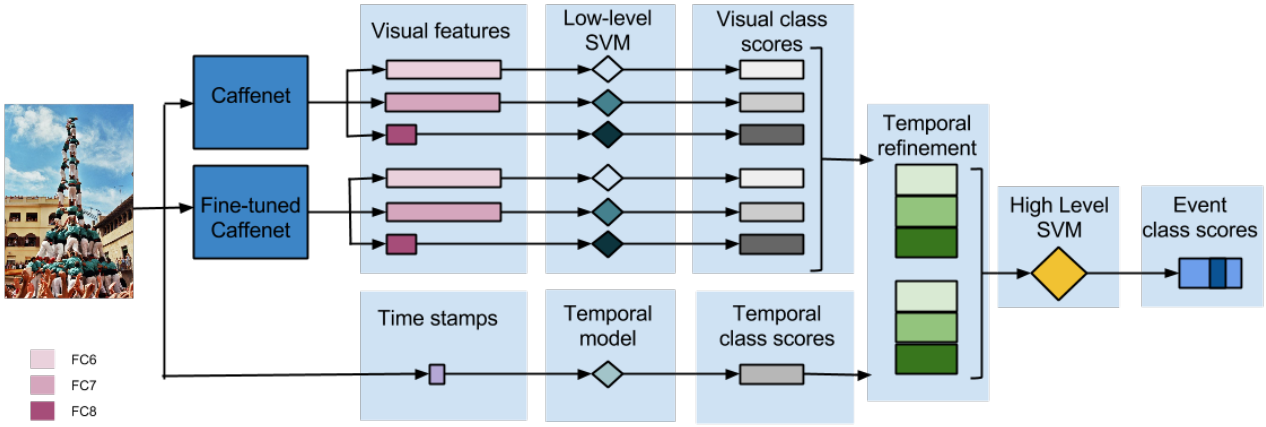


Figure 2. Global architecture of the proposed system.

event and *non-event* and, in the case of *event*, eight semantic classes were defined to be distinguished: *concert, conference, exhibition, fashion, protest, sports, theatre/dance, other*. The results over all participants showed that the classification performance strongly benefits from multimodal processing combining content and contextual information. Pure contextual processing as proposed in [26] and [11] and yielded the weakest results. The remaining participants proposed to add visual analysis to the contextual processing. CERTH-ITI [23] combined pLSA on the 1,000 most frequent tags with a dense sampling of SIFT visual features, which were later coded with VLAD. They observed a complementary role between visual and textual modalities. Brenner and Izquierdo [4] combined textual features with the global GIST visual descriptor, which is capable of capturing the spatial composition of the scene. The best performance in the Social Event Classification task was achieved by [19]. They combine processing of textual photo descriptions with the work from [18] for visual processing, based on bag of visual words aggregated in different fashions through events. Their results showed that visual information is the best option to discriminate between *event / non-event* and that textual information is more reliable to discriminate between different event types.

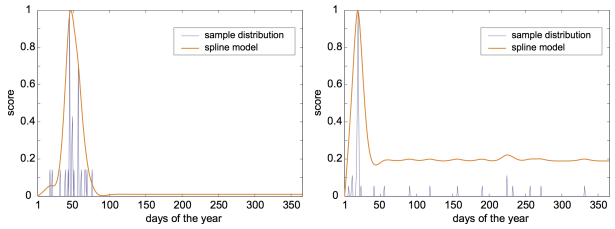
In terms of benchmarking, a popular strategy is to retrieve additional data to extend the training dataset. The authors of [22], for example, retrieved images from Flickr to build unigram language models of the requested event types and locations in order to enable a more robust matching with the user-provided query. We explored a similar approach in for cultural event recognition. Results, however showed that extending the training set this did not improve results but made them even worse.

3. Temporal models

Cultural events usually occur at a regular basis and thus have a repetitive nature. For example, “St. Patrick’s day” always takes place on March, 17, “La Tomatina” is always scheduled for the last week of August, and the “Carneval of Rio” usually takes place at some time in February and lasts for one week. More complex temporal patterns exist, for example, for cultural events coupled to the lunar calender which changes slightly each year. An example is the “Maslenitsa” event in Russia is which is scheduled for the eighth week before Eastern Orthodox Easter.

The temporal patterns associated with cultural events are a valuable clue for their recognition. A photo captured, for example, in December will very unlikely (except for erroneous date information) show a celebration of St. Patrick’s day. While temporal information alone is not sufficient to assign the correct event (many events may take place concurrently), we hypothesize that temporal information provides strong clues that can improve cultural event recognition.

To start with temporal processing, first temporal models have to be extracted from the data. Temporal models for cultural events can be either generated manually in advance or extracted automatically from metadata of related media. We propose a fully automatic approach to extract temporal models for cultural events. The input to our approach is a set of capture dates for media items that are related to a given event. Capture dates may be, for example, extracted from social media sites like Flickr or from the metadata embedded in the photos (e.g. EXIF information). In a first step, we extract the day and month of the capture dates and convert them into a number d between 1 and 365, encoding the day in the year when the photo was taken. From these numbers, we compute a temporal distribution $T(d)$ of all available capture dates. Assuming that a cultural event takes place



(a) Maslenitsa (b) Timkat

Figure 3. Temporal spline models for the "Maslenitsa" and the "Timkat" event: (a) for normally distributed data the model becomes approximately Gaussian-shaped; (b) the uncertainty of the distribution is reflected in the temporal model.

annually, it is straight-forward to model the temporal distribution with a Gaussian model. Gaussian modeling works well when a sufficient number of timestamps exists. For sparse data, however, with a few timestamps only, the distribution is likely to become non-Gaussian and thus model fitting fails in generating accurate models. Additionally, the timestamps of photos are often erroneous (or overwritten by certain applications) yielding strong deviations of the ideal distribution. To take the variability that is present in the data into account, a more flexible model is required. We model the distribution $t(d)$ by a piecewise cubic smoothing spline [7]. To generate the final model T, we evaluate the spline over the entire temporal domain and normalize it between 0 and 1. Given a photo i with a certain timestamp d_i, the fitted temporal model $T_c(d_i)$ provides a score s_c that the photo refers to the associated event c. The flexible spline model enables the modeling of sparse and non-Gaussian distributions and further to model events with more complex than annual occurrence patterns.

Figure 3 shows temporal models for two example events. The "Maslenitsa" (3(a)) takes place between mid of February and mid of March (approx. days 46-74). This corresponds well with the timestamps extracted from the related media items, resulting in a near Gaussian-shaped model. The "Timkat" event always takes place on January 19. This is accurately detected by the model, which has its peak at day 19. The photos related to this event, however, have timestamps that are distributed across the entire year. This property of the underlying data is reflected in the model, giving low but non-zero scores to photos with timestamps other than the actual event date.

Figure 4 shows the temporal models extracted from the training and validation data for all 50 classes. We observe that each model (row) exhibits one strong peak which represents the most likely date of the event. Some models contain additional smaller side-peaks learned from the training data which reflect the uncertainty contained in the training data. The events are distributed over the entire year, some events occur at the same time.

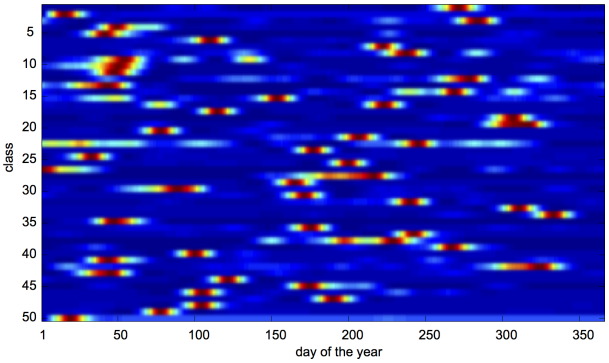


Figure 4. Automatically generated temporal models for each event class. For each event we observe a typical pattern of recording dates exhibiting one strong peak. The colors range from dark blue (0) to red (1).

The generated temporal models can be used to refine decisions made during classification (see Section 4.3) as well as for the filtering of additional data collections to reduce noise in the training data (see Section 5.2).

4. Image Classification

The automatic recognition of a cultural event from a photo is addressed in this paper with the system architecture presented in Figure 2. We propose combining the visual features obtained at the fully connected layers of two versions of the same *Caffenet* convolutional neural network: the original one and a modified version fine-tuned with photos captured at cultural events. A low-level SVM classifier is trained for each visual feature, and its scores refined with the temporal model described in Section 3. Finally, the temporally modified classification scores are fused in a final high-level SVM to obtain the final classification for a given test image.

4.1. Feature extraction

Deep convolutional neural networks (convnets) have recently become popular in computer vision, since they have dramatically advanced the state-of-the-art in tasks such as image classification [16], retrieval [1] or object detection [10, 12]

Convnets are typically defined as a hierarchical structure of a repetitive pattern of three hidden layers: (a) a local convolutional filtering (bidimensional in the case of images), (b) a non-linear operation, (commonly Rectified Linear Units - ReLU) and (c) a spatial local pooling (typically a *max* operator). The resulting data structure is called a *feature map* and, in the case of images, they correspond to 2D signals. The deepest layers in the convnet do not follow this pattern anymore but consist of *fully connected (FC)* layers: every value (neuron) in the fully connected layer is connected to all neurons from the previous layers through

some weights. As these fully connected layers do not apply any spatial constrain anymore, they are represented as single dimensional vectors, further referred in this paper as *neural codes* [1].

The amount of layers is a design parameter that, in the literature, may vary from three [17] to nineteen [24]. Some studies [28] indicate that the first layers capture finer patterns, while the deeper the level, the more complex patterns are modeled. However, there is no clear answer yet about how to find the optimal architecture to solve a particular visual recognition problem. The design of convnets is still mainly based on trial-and-error process and the expertise of the designer. In our work we have adopted the public implementation of *CaffeNet* [15], which was inspired by *AlexNet* [16]. This convnet is defined by 8 layers, being the last 3 of them fully connected. In our work we have considered the neural codes in these layers (FC6, FC7 and FC8) to visually represent the image content.

Apart from defining a convnet architecture, it is necessary to learn the parameters that govern the behaviour of the filters in each layer. These parameters are obtained through a learning process that replaces the classic handcrafted design of visual features. This way, the visual features are optimized for the specific problems that one wants to solve. Training a convnet is achieved through backpropagation, a high-computational effort that has been recently boosted by the affordable costs of GPUs. In addition to the computational requirements, a large amount of annotated data is also necessary. Similarly to the strategy adopted in the design of the convnet, we have also used the publicly available filter parameters of *CaffeNet* [15], which had been trained for 1,000 semantic classes from the ImageNet dataset [8].

The cultural event recognition dataset aimed in this paper is different from the one used to train *CaffeNet*, both in the type of images and in the classification labels. In addition, the amount of photos of annotated cultural events available in this work is much smaller than the large amount of images available in ImageNet. We have addressed the situation by also considering the possibility of fine tuning *CaffeNet*, that is, providing additional training data to an existing convnet which had been trained for a similar problem. This way, the network parameters are not randomly initialized, as in a training from scratch, but are already adjusted to a solution which is assumed to be similar to the desired one. Previous works [10, 12, 6] have proved that fine-tuning [13] is an efficient and valid solution to address these type of situations. In the experiments reported in Section 6 we have used feature vectors from both the original *CaffeNet* and its fine-tuned version.

4.2. Hierarchical fusion

The classification approach applied in our work is using the neural codes extracted from the convnets as features to train an classifier (Support Vector Machines, SVMs, in our case), as proposed in [6]. As we do not know a priori which network layer are most suitable for our task, we decide to combine several layers using a late fusion strategy.

The neural codes obtained from different networks and different layers may have strongly different dimensionality (e.g. from 4,096 to 50 in our setup). During the fusion of these features we have to take care that features with higher dimensionality do not dominate the features with lower dimensionality. Thus, we adopted a hierarchical classification scheme to late fuse the information from the different features in a balanced way [25].

At the lower level of the hierarchy we train separate multi-class SVMs (using one-against-one strategy [14]) for each type of neural code. We neglect the final predictions of the SVM and retrieve the probabilities of each sample for each class. The probabilities obtained by all lower-level SVMs form the input to the higher hierarchy level.

The higher hierarchy level consists of an SVM that takes to probabilistic output of the lower-level SVMs as input. This assures that all input features are weighted equally in the final decision step. The higher-level SVM is trained directly from the probabilities and outputs a prediction for the most likely event. Again we reject the binary prediction and retrieve the probabilities for each event as the final output.

4.3. Temporal Refinement

While visual features can easily be extracted from each image, the availability of temporal information depends on the existence of suitable metadata. Thus, temporal information must in general be considered to be a sparsely available feature. Due to its sparse nature, we propose to integrate temporal information into the classification process by refining the classifier outputs. This allows us to selectively incorporate the information only for those images where temporal information is available.

The basis for temporal refinement are the temporal models introduced in Section 3. The models T_c with $c = 1, \ldots, C$ and C the number of classes, represent for each event class c and each day of the year d, a score s representing the probability of a photo captured in a given day to belong to the event: $s = T_c(d)$. For a given image with index i, we first extract the day of the year d_i from its capture date and use it as an index to retrieve the scores from the temporal models of all event classes: $s_c = T_c(d_i)$, with $s = \{s_1, \ldots, s_C\}$.

Given a set of probabilities P_i for image i obtained from a classifier, the refinement of these probabilities is performed as follows. First, we compute the difference between the probabilities and the temporal scores: $d_i = P_i - s$. Next, we distinguish between two different cases:

(I) $d_i(c) < 0$: Negative differences mean that the probability for a given class predicted by the classifier is less

than the temporal score for this class. This case may easily happen as several events may occur at the same time as the photo was taken. The temporal models indicate that several events may be likely. Thus, the temporal information provides only a weak clue that is not discriminative. To handle this case, we decide to trust the class probabilities by the classifier and to ignore the temporal scores by setting $d = max(d, 0)$.

(II) $d_i(c) > 0$. In this case the temporal score is lower than the estimate of the classifier. Here, the temporal score provides a strong clue that indicates an inaccurate prediction of the classifier. In this case, we use the difference $d_i(c)$ to re-weight the class probability.

The weights w_i are defined as $w_i = max(d, 0) + 1$. The final re-weighting of the probabilities P_i is performed by computing $\tilde{P}_i = P_i / w_i$. In case **(I)** the temporal scores do not change the original predictions of the classifier. In case **(II)** the scores are penalized by a fraction that is proportional to the disagreement between the temporal scores and the prediction of the classifier.

5. Data Augmentation

The experiments described in Section 6 were conducted with the ChaLearn Cultural Event Recognition dataset [2], which was created by downloading photos from *Google Images* and *Bing* search engines. Previous works [16, 28, 6] have reported gains when applying some sort of data augmentation strategy.

We have explored two paths for data augmentation: artificial transformations on the test images and an extension of the training dataset by downloading additional data from Flickr.

5.1. Image transformations

A simple and classic method for data augmentation is to artificially generate transformations of the test image and fuse the classification scores obtained in each transformation. We adopted the default image transformations associated to *CaffeNet* [15], this is an horizontal mirroring and 5 crops in the input image (four corners and center). The resulting neural codes associated to each fully connected layer were fused by averaging the 10 feature vectors generated with the 10 image transformations.

5.2. External data download

We decided to extend the amount of training data to finetune our convnet, as discussed in Section 4.1. By doing this, we expected to reduce the generalization error of the learned model by having examples coming from a wider origin of sources.

The creators of the ChaLearn Cultural Event Recognition dataset [2] described each of the 50 considered events with pairs of title and geographical location; such as *Carnival Rio-Brazil*, *Obon-Japan* or *Harbin Ice and Snow Festival-China*. This information allows generating queries on other databases to obtained an additional set of labeled data.

Our chosen source for the augmented data was the *Flickr* photo repository. Its public API allows to query its large database of photos and filter the obtained results by tags, textual data search and geographical location. We generated 3 sets of images from Flickr, each of them introducing a higher degree of refinement:

90k set: Around 90,000 photos retrieved by matching the provided event title on the Flickr *tags* and *content* metadata fields.

21k set: The query from the 90k set was combined with a GPS filtering based on the provided country.

9k set: The query from the 21k set was further with manually selected terms from the Wikipedia articles related to the event. In addition, the Flickr query also toggled on an *interestingness* flag which improved the diversity of images in terms of users and dates. Otherwise, Flickr would provide a list sorted by upload date, which will probably contain many similar images from a reduced set of users.

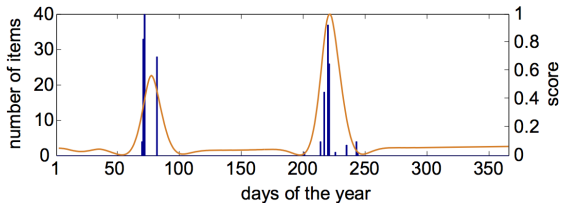
The temporal models T_c presented in Section 3 were also used to improve the likelihood that a downloaded photo actually belongs to a certain event. Given a media item i retrieved for a given event class c, we extract the day of capture d_i from its metadata and retrieve the score $s_c = T_c(d_i)$ from the respective temporal model. Next, we threshold the score to remove items that are unlikely under the temporal model. To assure a high precision of the filtered media collection, the threshold should be set to a rather high value, e.g. 0.9. Figure 5 gives two examples of media collections retrieved for particular events. We provide the distribution of capture dates with the pre-trained temporal models.

The Flickr IDs of this augmented dataset filtered by minimum temporal scores have been published in JSON format from the URL indicated in Section 1.

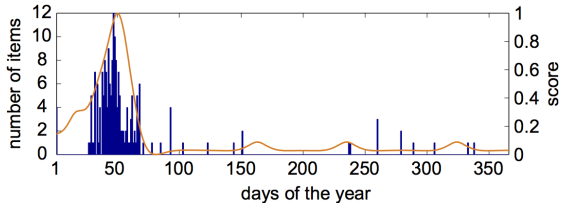
6. Experiments

6.1. Cultural Event Recognition dataset

The Cultural Event Recognition dataset [2] depicts 50 important cultural events all over the world. In all the image categories, garments, human poses, objects and context do constitute the possible cues to be exploited for recognizing the events, while preserving the inherent inter- and intra-class variability of this type of images. The dataset is divided in three partitions: 5,875 images for *training*, 2,332 for *validation* and 3,569 for *test*.



(a) Desfile de Silleteros



(b) Carnival of Venice

Figure 5. Two examples of retrieved image collections from Flickr and their temporal distribution. (a) the retrieved images match well the pre-trained temporal model. (b) the temporal distribution shows numerous outliers which are considered unlikely given the temporal model. The proposed threshold-based filtering removes those items.

6.2. Experimental setup

We employ two different convnets as input (see Section 4.1): the original *CaffeNet* trained on 1,000 Imagenet classes, and a fine-tuned version of *CaffeNet* trained during 60 epochs on the 50 classes defined in the Chalearn Cultural Recognition Dataset. Fine-tuning of the convnet was performed in two stages: in a first one the *training* partition was used to train and the *validation* partition to estimate the training loss and allow the network to learn. In a second stage, the two partitions were switched so that the network had to learn the optimal features from all the available labeled data.

From both convnets we extracted neural codes from layers FC6 and FC7 (each of 4,096 dimensions), as well as FC8 (the top layer with a softmax classifier), which has 1000 dimensions for the original *CaffeNet* and 50 for the fine-tuned network. Both feature extraction and fine tuning have been performed using the *Caffe* [15] deep learning framework.

As presented in Section 4.2, a classifier was trained for each of the 6 neural codes, in addition to the one used for late fusion. The implementation of *Libsvm* library [5] of the linear SVM was used, with parameter $C = 1$ determined by cross validation and grid search and probabilistic output switched on.

Each image was scored for each of the 50 considered cultural events and results were measured by a precision/recall curve, whose area under the curve was used to estimate the average precision (AP). Numerical results are averaged over the 50 events to obtain the mean average precision (mAP).

	FC6	FC7	FC8
Raw layer	0,6832	0,6669	0,6079
+ temporal refinement	0,6893	0,6730	0,6152

Table 1. Results on single layer raw neural codes.

Fine-tunned FC6-FC7-FC8	0,6919
+ raw FC6-FC7-FC8	0,7038
+ temporal refinement	0,7357

Table 2. Results on fine-tuned and fused multi-layer codes.

More details about the evaluation process can be found in [2].

6.3. Results on the validation dataset

A first experimentation was performed to assess the impact of temporal refinement on the default *CaffeNet*, that is, with no fine-tunning. Results in Table 1 indicate diverse performance among the fully connected layers, being FC6 the one with a highest score. Temporal refinement slightly increases the mAP consistently in all layers.

The preliminary results were further extended to compare the performance of the three neural codes (FC6, FC7 and FC8) when temporally refined and finally complemented with the features from the original *CaffeNet*. The results shown in Table 2 indicate a higher impact of temporal refinement than in the case of single layers, and an unexpected gain by adding the raw neural codes from *CaffeNet*.

Our experimentation on the additional data downloaded from Flickr was unsuccessful. The selected dataset was the 9k Flickr one with a restrictive threshold of 0.9 on the temporal score. With this procedure we selected 5,492 images, which were added as training samples for fine tuning. We compare the impact of adding this data into training only on the softmax classifier at the last layer of *CaffeNet*, obtaining a drop in the mAP from 0.5821 to 0.4547 when adding the additional images to the already fine-tuned network. We hypothesize that the visual nature of the images downloaded from Flickr differs from the one of the data crawled from Google and Bing by the creators of the ChaLearn dataset. A visual inspection on the augmented dataset did not provide any hints that could expalin this behaviour.

6.4. Results on the test dataset

The best configuration obtained with the validation dataset was used on the test dataset to participate in the ChaLearn 2015 challenge. Our submission was scored by the organizers with a mAP of $0,767$, the second best performance among the seven teams which completed the submission, out of the 42 participants who had initially registered on the challenge website.

7. Conclusions

The presented work proves the high potential of the visual information for cultural event recognition. This result is especially sounding when contrasted with many of the conclusions made in the MediaEval Social Event Detection task [20], where it was frequently observed that visual information was less reliable than contextual metadata for event clustering. This difference may be caused by the very salient and distinctive visual features that often make cultural events attractive and unique. The dominant green in Saint Patrick's parades, the vivid colors from the Holi Festivals or the skull icons from the Dia de los Muertos

In our experimentation the temporal refinement has provided modest gain. We think this may be caused by the low portion of images with available EXIF metadata, around 24% according to our estimations. In addition, we were also surprised by the loss introduced by the Flickr data augmentations. We plan to look at this problem more closely and figure out the difference between the ChaLearn dataset and ours.

Finally, it must be noticed that the quantitative values around 0.7 may be misleading, as in this dataset every image belonged to one of the 50 cultural events. Further editions of the ChaLearn challenge may also introduce the *no event* class as in MediaEval SED 2013 [21] to, this way, better reproduce a realistic scenario where the event retrieval is performed in the wild.

Acknowledgements

This work has been developed in the framework of the project BigGraph TEC2013-43935-R, funded by the Spanish Ministerio de Economa y Competitividad and the European Regional Development Fund (ERDF).

The Image Processing Group at the UPC is a SGR14 Consolidated Research Group recognized and sponsored by the Catalan Government (Generalitat de Catalunya) through its AGAUR office.

We gratefully acknowledge the support of NVIDIA Corporation with the donation of the GeoForce GTX Titan Z used in this work.

References

[1] A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky. Neural codes for image retrieval. In *Computer Vision–ECCV 2014*, pages 584–599. Springer, 2014.

[2] X. Baro, J. Gonzalez, F. Junior, M. A. Bautista, M. Oliu, I. Guyon, H. J. Escalante, and S. Escalera. Chalearn looking at people challenge 2015: Dataset and results. In *CVPR, ChaLearn Looking at People Workshop*, 2015.

[3] L. Bossard, M. Guillaumin, and L. Van. Event recognition in photo collections with a stopwatch hmm. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 1193–1200. IEEE, 2013.

[4] M. Brenner and E. Izquierdo. Multimodal detection, retrieval and classification of social events in web photo collections. In *ICMR 2014 Workshop on Social Events in Web Multimedia (SEWM)*, pages 5–10, 2014.

[5] C.-C. Chang and C.-J. Lin. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011.

[6] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *British Machine Vision Conference*, 2014.

[7] C. de Boor. *A Practical Guide to Splines*. Springer-Verlag, New York, 2d edition, 1984.

[8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

[9] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, et al. Query by image and video content: The qbic system. *Computer*, 28(9):23–32, 1995.

[10] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 580–587. IEEE, 2014.

[11] I. Gupta, K. Gautam, and K. Chandramouli. Vit @ MediaEval 2013 social event detection task: Semantic structuring of complementary information for clustering events. In *Proceedings of the MediaEval 2013 Multimedia Benchmark Workshop*, volume 1043. CEUR-WS. org, 2013.

[12] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Simultaneous detection and segmentation. In *Computer Vision–ECCV 2014*, pages 297–312. Springer, 2014.

[13] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[14] C.-W. Hsu and C.-J. Lin. A comparison of methods for multiclass support vector machines. *Neural Networks, IEEE Transactions on*, 13(2):415–425, 2002.

[15] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the ACM International Conference on Multimedia*, pages 675–678. ACM, 2014.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[17] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[18] R. Mattivi, J. Uijlings, F. G. De Natale, and N. Sebe. Exploitation of time constraints for (sub-) event recognition. In *Proceedings of the 2011 joint ACM workshop on Modeling and representing events*, pages 7–12. ACM, 2011.

[19] T. Nguyen, M.-S. Dao, R. Mattivi, and E. Sansone. Event clustering and classification from social media: Watershed-

based and kernel methods. In *MediaEval 2013 Workshop*, Barcelona, Catalonia, October 18-19 2013.

[20] G. Petkos, S. Papadopoulos, V. Mezaris, R. Troncy, P. Cimiano, T. Reuter, and Y. Kompatsiaris. Social event detection at MediaEval: a three-year retrospect of tasks and results. In *ICMR 2014 Workshop on Social Events in Web Multimedia (SEWM)*, pages 27–34, 2014.

[21] T. Reuter, S. Papadopoulos, V. Mezaris, P. Cimiano, C. de Vries, and S. Geva. Social Event Detection at MediaEval 2013: Challenges, datasets, and evaluation. In *MediaEval 2013 Workshop*, Barcelona, Catalonia, October 18-19 2013.

[22] M. Riga, G. Petkos, S. Papadopoulos, E. Schinas, and Y. Kompatsiaris. CERTH @ MediaEval 2014 Social Event Detection Task. In *MediaEval 2014 Multimedia Benchmark Workshop*, 2014.

[23] E. Schinas, G. Petkos, S. Papadopoulos, and Y. Kompatsiaris. CERTH @ MediaEval 2012 Social Event Detection Task. In *MediaEval 2012 Multimedia Benchmark Workshop*, 2012.

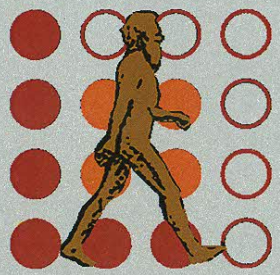
[24] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[25] C. G. Snoek, M. Worring, and A. W. Smeulders. Early versus late fusion in semantic video analysis. In *Proceedings of the 13th annual ACM international conference on Multimedia*, pages 399–402. ACM, 2005.

[26] T. Sutanto and R. Nayak. Admrg @ MediaEval 2013 social event detection. In *Proceedings of the MediaEval 2013 Multimedia Benchmark Workshop*, volume 1043. CEUR-WS. org, 2013.

[27] M. Zaharieva, M. Del Fabro, and M. Zeppelzauer. Cross-platform social event detection. *IEEE Multimedia*, 2015.

[28] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014*, pages 818–833. Springer, 2014.



CHA LEARN

HuPBA

ChaLearn Looking at People Challenge and Workshop, 2015

Computer Vision and Pattern Recognition

2nd place award in Cultural Event Recognition

Presented to

Xavier Giró i Nieto

Universitat Politècnica de Catalunya

Bibliography

[1] A.Krizhevsky, I.Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[2] A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky. Neural codes for image retrieval. In *Computer Vision–ECCV*, page 584–599, Springer, 2014.

[3] X Baro, J Gonzlez, J Fabian, MA Bautista, M Oliu, I Guyon, HJ Escalante, and S Escalers. Chalearn looking at people 2015 cvpr challenges and results: action spotting and cultural event recognition. In *CVPR, ChaLearn Looking at People workshop*, volume 1, page 4, 2015.

[4] M. Brenner and E. Izquierdo. Multimodal detection, retrieval and classification of social events in web photo collections. In *ICMR 2014 Workshop on Social Events in Web Multimedia (SEWM)*, pages 5–10, 2014.

[5] Shih-Fu Chang, Junfeng He, Yu-Gang Jiang, Akira Yanagawa, Eric Zavesky, Elie Khoury, and Chong-Wah Ngo. Columbia university/vireo-cityu/irit trecvid2008 high-level feature extraction and interactive video search. In *TRECVID*. Citeseer, 2008.

[6] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *British Machine Vision Conference*, 2014.

[7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, , and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition*, 2009.

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

[9] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Computer Vision and Pattern Recognition(CVPR), 2014 IEEE Conference on*, page 580–587, 2014.

[10] Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2066–2073. IEEE, 2012.

[11] I. Gupta, K. Gautam, and K. Chandramouli. Vit @ mediaeval 2013 social event detection task: Semantic structuring of complementary information for clustering events. In *Proceedings of the MediaEval 2013 Multimedia Benchmark Workshop*, page volume 1043, 2013.

[12] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Simultaneous detection and segmentation. In *Computer Vision and Pattern Recognition(CVPR), 2014 IEEE Conference on*, page 580–587, 2014.

[13] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. In *Science, 313(5786):504–507*, 2006.

[14] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrel. Caffe: Convolutional architecture for fast feature embedding. In *ACM Multimedia Open Source Competition*, 2014.

[15] Andrej Karpathy. Convolutional neural networks for visual recognition. In *Stanford CS class CS231n*.

[16] K.Chatfield, K.Simonyan, A.Vedaldi, and A.Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *British Machine Vision Conference*, 2014.

[17] Brian Kulis, Kate Saenko, and Trevor Darrell. What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1785–1792. IEEE, 2011.

[18] Heeyoung Kwon, Kiwon Yun, Minh Hoai, and Dimitris Samaras. Recognizing cultutral events in images: a study of image categorization models. In *Conference Computer Vision and Pattern Recognition*, 2015.

[19] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 2169–2178. IEEE, 2006.

[20] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient based learning applied to document recognition. In *Proceedings of the IEEE*, page 86(11):2278–2324, 1998.

[21] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[22] R. Mattivi, J. Uijlings, F. G. De Natale, and N. Sebe. Exploitation of time constraints for (sub-) event recognition. In *Proceedings of the 2011 joint ACM workshop on Modeling and representing events*, pages 7–12, ACM, 2011.

[23] T. Nguyen, M.-S. Dao, R. Mattivi, and E. Sansone. Event clustering and classification from social media: Watershed-based and kernel methods. In *MediaEval 2013 Workshop*, 2013.

[24] Sungheon Park and Nojun Kwak. Cultural event recognition by subregion classification with convolutional neural network. In *Conference Computer Vision and Pattern Recognition*, 2015.

[25] G. Petkos, S. Papadopoulos, V. Mezaris, R. Troncy, P. Cimiano, T. Reuter, and Y. Kompatsiaris. Social event detection at mediaeval: a three-year retrospect of tasks and results. In *ICMR 2014 Workshop on Social Events in Web Multimedia (SEWM)*, pages 27–34, 2014.

[26] T. Reuter, S. Papadopoulos, V. Mezaris, P. Cimiano, C. de Vries, and S. Geva. Social event detection at mediaeval 2013: Challenges, datasets, and evaluation. In *MediaEval 2013 Workshop*, ACM, 2013.

[27] Amaia Salvador, Matthias Zeppelzauer, Daniel Manchon-Vizuete, Andrea Calafell, and Xavier Giro-i Nieto. Cultural event recognition with visual convnets and temporal models. In *Conference Computer Vision and Pattern Recognition*, 2015.

[28] E. Schinas, G. Petkos, S. Papadopoulos, and Y. Kompatsiaris. Certh @ mediaeval 2012 social event detection task. In *Proceedings of the MediaEval 2012 Multimedia Benchmark Workshop*, 2012.

[29] Edgar Simo-Serra, Eduard Trulls, Luis Ferraz, Iasonas Kokkinos, and Francesc Moreno-Noguer. Fracking deep convolutional image descriptors. *arXiv preprint arXiv:1412.6537*, 2014.

[30] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *arXiv preprint arXiv:1409.1556*, 2014.

[31] T. Sutanto and R. Nayak. Admrg. Mediaeval 2013 social event detection. In *Proceedings of the MediaEval 2013 Multimedia Benchmark Workshop*, page volume 1043, 2013.

[32] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *arXiv preprint arXiv:1409.4842*, 2014.

[33] Antonio Torralba, Alexei Efros, et al. Unbiased look at dataset bias. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1521–1528. IEEE, 2011.

[34] Jasper RR Uijlings, Koen EA van de Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013.

[35] Limin Wang, Zhe Wang, Wenbin Du, and Yu Qiao. Object-scene convolutional neural networks for event recognition in images. In *Conference Computer Vision and Pattern Recognition*, 2015.

[36] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *Computer Vision- ECCV*, pages 818–833, Springer, 2014.

[37] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014*, pages 818–833. Springer, 2014.

[38] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. In *Advances in Neural Information Processing Systems*, pages 487–495, 2014.