Mind the Regularized GAP, for Human Action Classification and Semi-supervised Localization based on Visual Saliency

Marc Moreaux[1,2,3], Natalia Lyubova[1], Isabelle Ferrané[2] and Frederic Lerasle[3]
[1]*Softbank Robotics Europe, 43 rue du colonel Pierre Avia, Paris, France*
[2]*CNRS, LAAS, Univ. de Toulouse, Toulouse, France*
[3]*IRIT, Univ. de Toulouse, Toulouse, France*

Keywords: Semi-supervised Class Localization, Image Classification, Class Saliency, Global Average Pooling.

Abstract: This work addresses the issue of image classification and localization of human actions based on visual data acquired from RGB sensors. Our approach is inspired by the success of deep learning in image classification. In this paper, we describe our method and how the concept of Global Average Pooling (GAP) applies in the context of semi-supervised class localization. We benchmark it with respect to Class Activation Mapping initiated in (Zhou et al., 2016), propose a regularization over the GAP maps to enhance the results, and study whether a combination of these two ideas can result in a better classification accuracy. The models are trained and tested on the Stanford 40 Action dataset (Yao et al., 2011) describing people performing 40 different actions such as *drinking*, *cooking* or *watching TV*. Compared to the aforementioned baseline, our model improves the classification accuracy by 5.3 percent points, achieves a localization accuracy of 50.3%, and drastically diminishes the computation needed to retrieve the class saliency from the base convolutional model.

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

1 INTRODUCTION

Nowadays, as intelligent systems are getting more and more deeply involved in our everyday life, machine vision becomes incredibly important. Intelligent systems could greatly benefit from an ability to perceive the human environment and its major actors, allowing them to better understand what is happening around them. A lot of work has been done in automatic image labeling, namely ”image classification”, and in automatic estimation of the position of a class in an image, namely ”class localization” (LeCun et al., 2015) and it can be applied in the context of human action classification and localization (see Figure 1). In this paper, we consider that most of the proposed architectures made an extensive use of supervision in the training process when localization could have been inferred from a lower amount of information.

Since 2006, deep learning has increasingly grown in use to become the most successful approach in image classification and localization. A vast majority of networks used in this field are composed by a stack of *Convolutional Neural Network* (CNN) layers, followed by one or several *Fully Connected layers* (FC), also referred as *Dense layer*, resulting in a prediction vector. More recently, the Global Average Pooling (GAP) method has been used at the last layers of several networks (He et al., 2016; Zhou et al., 2016) to perform classification and have opened the possibility to perform semi-supervised-localization, which is defined here as inferring a class localization without training on localization data but only on labels[1],

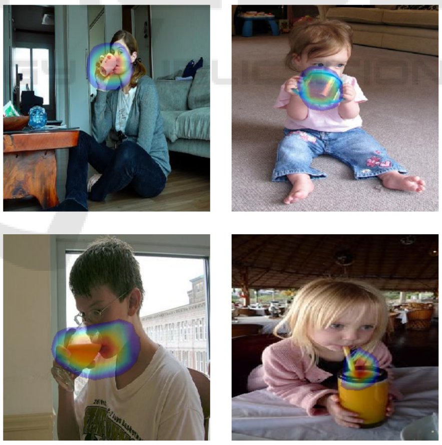


Figure 1: Examples of drinking action localization and saliency retrieved with our approach Inception-GAP5-L1 (see Section 3).

[1] In contrast with weakly-supervised-localization learning which uses a reduced amount of data to train.

Moreaux, M., Lyubova, N., Ferrané, I. and Lerasle, F.
Mind the Regularized GAP, for Human Action Classification and Semi-supervised Localization based on Visual Saliency.
DOI: 10.5220/0006548303070314
In *Proceedings of the 13th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2018) - Volume 4: VISAPP*, pages 307-314
ISBN: 978-989-758-290-5

Copyright © 2018 by SCITEPRESS – Science and Technology Publications, Lda. All rights reserved

hence, without a need of extensive localization annotation. This kind of approach is interesting as it is costly to have human annotators drawing bounding boxes around objects in dense datasets.

Global Average Pooling (GAP), a mathematical operation performing the average of a matrix (described in Section 3), was first presented as a structural regularizer in NiN (Lin et al., 2013) and later used in GoogLeNet (Szegedy and Liu, 2015). More recently, it was used in ResNet (He et al., 2016) and GoogLeNet-GAP (Zhou et al., 2016) before a fully connected layer to perform object localization. In this latter approach, it was preferred to max-pooling to find all the discriminative parts of a class instead of the most discriminative one.

In this work, we intend to increment the classification and localization research based on the GAP methods by proposing a modified architecture and some naive regularizations. Section 2 reviews former work published on this topic. Section 3 introduces both our architecture and a naive regularization term used for localization. Section 4, describes our evaluations and the proposed network. Finally, Section 5 concludes our work.

2 RELATED WORK

In the context of visual perception, many approaches based on CNNs have been used in the last years to perform real-time object localization. Most of the successful approaches used fully-supervised learning to tackle theses problems. This section reviews the architectures that have been used first for supervised and then for weakly or semi-supervised localization in image processing in computer vision.

Fully-supervised Learning for Localization: In recent literature, many architectures propose to perform image classification and localization, at the same time, using fully-supervised learning. Models like AlexNet (Krizhevsky et al., 2012), VGGNet (Simonyan and Zisserman, 2014) and GoogLeNet (Szegedy and Liu, 2015) use a stack of convolutional layers followed by fully connected layers to predict the class instance and its location in images, using, for instance, a regression on the bounding box (Sermanet et al., 2013). Throughout time, these models competed in ILSVRC[2] (Simonyan and Zisserman, 2014) localization contest (won by (Krizhevsky et al., 2012) and (Szegedy and Liu, 2015)). Other models, like ResNet (He et al., 2016) introduced a similar approach, but with a GAP layer at the last convolutional layer of their networks, and set a new record in the ILSVRC 2014 localization contest. It is clear that, in such contest, researchers are using maximum of available resources for training their approaches, however, we would like our models to be less reliant on large amount of annotated data. This is our motivation to move towards semi-supervised learning.

Weakly and Semi Supervised Learning for Localization: Some architectures are designed to perform weakly-supervised localization, for example, the model proposed by Oquab et al. (Oquab et al., 2015) is trained in two steps. First, a traditional CNN model, finishing with a softmax layer, is trained on cropped images to learn to recognize a class based on a fixed receptive field. The weights learned at this step are frozen and the second training step consists in convolving this model to a larger image in order to produce a matrix of softmax predictions. From this matrix, a new network is learned to predict the class localization. This network includes a global max-pooling operation made to retrieve the maximum probability of a class being present in the image. We took inspiration from this work as (a1) the first part of the model is trained on images which do not include any contextual information (background removed at the cropping step) and (a2) the resulting model produces a saliency map for every class present in an image, based on a given receptive field. Even though, we consider that (b1) the two-step learning can be reduced to one step, (b2) the global max-pooling is a bottleneck operation to obtain a one-shot learning model and (b3) the model should be able to learn with a lower amount of pre-processed inputs.

These b1, b2, and b3 points have been taken into account in (Zhou et al., 2016) where the authors propose a one-shot semi-supervised method to perform image classification and localization without any annotation of localization. Their method, called ”GoogLeNet-GAP”, is a stack of CNNs ending with a large amount of convolutional units where each output map is averaged to a single value with Global Average Pooling (GAP). The resulting values are then fully connected to a softmax layer. We believe, that, because of the GAP layer being fully connected to the prediction layer, the last convolutional layer, which is used for localization, shares too much information with all the predictions resulting in an attention field broader than needed.

In our approach, we aim at developing, first, one-shot semi-supervised training for class localization as in (Zhou et al., 2016). Second, we want to reduce the attention field in our localization mechanism by removing the Dense layer following the GAP layer, in order to have an attention model similar to (Oquab et al.,

[2] Imagenet Large-Scale Visual Recognition Challenge

2015), and we refer to this modification as ”unshared GAP layer”. Third, we would like our model to decrease the computation, comparatively to (Zhou et al., 2016), for retrieving the localization of the classes. Our models (eg. ”Inception-GAP5”) will be compared to both ”GoogleLeNet-GAP”, introduced in (Zhou et al., 2016), and ”Inception-GAP-Zhou”, our implementation of the former.

3 OUR MODEL ARCHITECTURE

In our approach, the architecture is designed to perform semi-supervised learning for localization from a classification problem (see Figure 2).

The proposed architecture follows the *all-convolutional* trend observed in deep-learning and push it forward removing every dense layer present on the network. To do so, we select a deep learning architecture whose structure and training procedure is known (InceptionV3) as our *base model*, up until a desired layer and add a new convolutional layer with $c * m$ kernels (c being the amount of classes in our classification task, and m, the amount of kernels used per class) followed by a GAP layer and m sums resulting in c values, used as prediction values.

To build our model, we applied the recommendations given in (Zhou et al., 2016) for GoogLeNet (Szegedy and Liu, 2015) to InceptionV3. Hence, our architecture is composed by the initial stack of CNNs (shown by the blue parallelepipeds in Figure 2) described in (Szegedy et al., 2016) up until the layer called *Inception4e*. Following the recommendations, this layer is followed by a $[3 \times 3]$ convolutional layer of stride 1 whose resulting matrix contains saliency maps for each class (shown by orange squares in Figure 2). These maps are averaged with GAP, clustered (summing them) in c values, one per class, and fed to a softmax layer for classification.

If this model has $m = 5$ maps per layer per class, we refer to it as ”Inception-GAP5” where ”Inception” indicate that InceptionV3 is the base model. For instance, GAP1 architecture would correspond to the original approach developed in NiN (Lin et al., 2013) where they state that each of these maps *”can be easily interpreted as categories confidence maps”*. To keep track of the introduced names of the models, Table 1 gives a short description of them.

Formally, the last layers of the network are defined as follows : lets $f_k(x,y)$ be the activation of unit k with $\{k \in \mathbb{N} : k < m \cdot c\}$ in the last convolutional layer at the spatial location (x,y). Then, the GAP vector $\mathbf{g}$ with $\mathbf{g} \in \mathbb{R}^{\mathbf{m \cdot c}}$ is defined by :

$$g_k = \frac{\sum_{x=0}^{X} \sum_{y=0}^{Y} f_k(x,y)}{X \cdot Y} \quad (1)$$

Where X and Y are the sizes of the preceding convolutional layers. Then, we cluster $\mathbf{g}$ in a vector $\mathbf{g}' \in \mathbb{R}^c$.

$$g'^c = \sum_m g_k \quad (2)$$

This vector is fed to a softmax function to compute the class predictions p^c.

$$p^c = \text{softmax}^c(\mathbf{g}') \quad (3)$$

The activation $f_k(x,y)$ we chose is Rectified Linear Unit (ReLu) (Nair and Hinton, 2010).

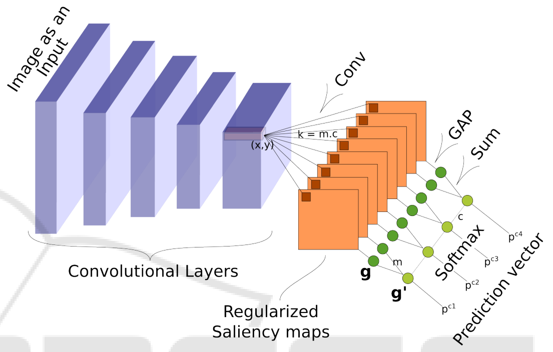


Figure 2: The architecture of the proposed model with two maps per class ($m = 2$) and four classes ($c = 4$) as an example.

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

3.1 Class Activation Mapping

In (Zhou et al., 2016), the authors described a procedure to retrieve some regions of interest with a method they call *class activation mapping*. This method is re-adapted to our architecture and described below, yet, to fairly compare each other results, the aforementioned architecture is slightly modified and re-implemented in ”Inception-GAP-Zhou”. In this model, the GoogLeNet (Szegedy and Liu, 2015) they used is swapped with the InceptionV3 base which is followed by a convolutional layer composed of 1024 units whose outputs are averaged with GAP, then fully connected to the predictions and then transformed into predictions with a softmax layer.

The procedure they called ”*Class Activation Mappings*” (CAM) aim to indicate ”*the discriminative image regions used by the CNN to identify*” action classes. In our model, we adapt this equation such that a CAM becomes, for each class c, and each spatial localization (x,y), the sum of the n activation functions.

$$CAM^c(x,y) = \sum_{k=(c-1)m}^{c.m-1} f_k(x,y) \quad (4)$$

Table 1: Description of the models evaluated and compared in this work.

Model name	Description
Inception-GAP5	Our GAP architecture with 5 maps per neurons built on top of InceptionV3
Inception-GAP5-L1	Inception-GAP5 whose GAP layer has a L1 penalty on its outputs
Inception-GAP5-L2	Inception-GAP5 whose GAP layer has a L2 penalty on its outputs
Inception-GAP-Zhou	Gap method, as proposed in (Zhou et al., 2016), build on top of InceptionV3
Inception-GAP-Zhou-L1	Inception-GAP-Zhou whose GAP layer has a L1 penalty on its outputs
Inception-GAP-Zhou-L2	Inception-GAP-Zhou whose GAP layer has a L2 penalty on its outputs
GoogLeNet-GAP-Zhou	Architecture proposed in (Zhou et al., 2016)

Retrieving CAM^c: In comparison to the localization method introduced in (Zhou et al., 2016), the amount of operations used to retrieve these CAM maps from the computational graph, is significantly reduced. Due to the design of InceptionV3, used as a base model in our approach, the default input image resolution is 299×299 pixels (Szegedy et al., 2016). In our architecture, we cut InceptionV3 to the *Inception2-5* layer, resulting in a matrix with a shape of $17 \times 17 \times k$ kernels. Therefore, the amount of operations needed to retrieve the CAM are :

- For Inception-GAP5, the CAM method relies on summing 5 convolutional kernel outputs through the 17 x and 17 y coordinates, resulting in $5 \times 17 \times 17 = 1445$ sums.
- For Inception-GAP-Zhou (Sec.4), the CAM method is the result of (a) weighting all the $k = 1024$ convolutional kernel outputs (of size 17×17) with its corresponding weights on the dense layer, resulting in $(1024 \times 17 \times 17) = 295,936$ multiplications and (b) summing these 1024 maps through each of the 17 x and 17 y coordinates, resulting in $(1024 \times 17 \times 17) = 295,936$ sums.

In this sense, GAP5 is more computationally efficient than GAP-Zhou.

3.2 Regularization

The softmax operator, or normalized exponential as mentioned in (Bishop, 2006), forces the exponentials of the activations (g'^c in our case) to be properly separated, yet it does not constrain the class activations to be centered on some particular value. In our model, we want the $f_k(x,y)$ activations to provide us with insights on the probability of a class to be present at a given position. Therefore, we add a regularization term on the $f_k(x,y)$ values.

As in (Raina et al., 2007), where the authors constrain the activations with a L1 regularization, we propose to force the last convolutional layer of our model to be sparse. Such property should help in having a clear visualization of the CAMs and rendering whether a class is present or not at a given spatial location (x,y). To render this property, we introduce either a L1 or a L2 regularization term to the outputs of the last convolutional layer produced by $f_k(z(x,y))$, with z being the input to last convolutional layer.

The L1 regularization, applied to our last convolutional layer, whose k kernels are weighted by W_k and followed by their ReLu activation, is as follows :

$$\begin{aligned} L_1 &= \alpha \sum_k \cdot \left| \sum_{x,y} f_k(z(x,y)) \right| \\ &= \alpha \sum_k \cdot \left| \sum_{x,y} \max(0, W_k * z(x,y)) \right| \end{aligned} \tag{5}$$

Whereas the L2 activity regularization is :

$$\begin{aligned} L_2 &= \alpha \sum_k \sqrt{\sum_{x,y} f_k(z(x,y))^2} \\ &= \alpha \sum_k \sqrt{\sum_{x,y} \max\left(0, (W_k * z(x,y))^2\right)} \end{aligned} \tag{6}$$

With the ReLU activation, in both cases the W_k weights are penalized only if the kernel k returns a map which sum is above zero.

The loss function of our model is the same as (Szegedy et al., 2016), namely, the cross-entropy $l = -\sum_k Y_k \log(p_k)$ with Y_k the one hot vector class corresponding to a sample X_k. The regularization term described above is added to this loss and the alpha term weights the importance of the regularization with respect to the categorical cross entropy. After evaluation, we tuned empirically α to be equal to 10^{-7} for both the L1 and the L2 regularization terms.

3.3 Implementation

This work was implemented on Keras[3] back-ended with Tensorflow (Abadi, 2015). To implement our model, we used InceptionV3, available in Keras and pre-trained on ImageNet (Krizhevsky et al., 2012). As in InceptionV3, our models are trained and tested with images resized to 299×299 RGB pixels.

Nadam was chosen to train our model because of its fast convergence speed. The parameters used in

[3] CHOLLET, Francois. Keras (2015). http://keras.io.

SCITEPRESS
SCIENCE AND TECHNOLOGY PUBLICATIONS

our approach are those proposed by default in Keras, except for the learning rate, which was decayed every second epoch, as in (Szegedy et al., 2016).

During the first 10 epochs, the weights of inceptionV3 are fixed in such way that the GAP layer is initialized with respect to the pre-trained network. Afterwards, all the weights in the model are subjected to optimization. We empirically fix the maximum amount of epochs to be 125 (when the loss stopped decreasing) and report in the following Section the results obtained for each model we trained, with and without regularization. The results are achieved by the combination of weights scoring the lowest loss on the validation set.

4 ACTION DATASET AND ASSOCIATED EVALUATIONS

This section presents both quantitative and qualitative results obtained with our models applied on *"The Stanford 40 Action"* (Yao et al., 2011) dataset. Inception-GAP5 built on top of InceptionV3 with 5 maps per class was preferred to Inception-GAP10 as we noticed that increasing the amount *m* of maps per class (10 instead of 5) did not improve results in our classification task. Inception-GAP5 achieved an accuracy of 75.9% when Inception-GAP10 scored 1.3 points lower and Inception-GAP5-L1 achieved an accuracy of 75.5% when Inception-GAP10-L1 scored 0.7 point more. For fair comparison, we also implemented the method proposed in (Zhou et al., 2016) on-top of InceptionV3 and trained it using the same optimizer as the one used for our model, referred as Inception-GAP-Zhou.

First the dataset is presented then comes the comparison of two one-shot and semi-supervised training methods, one based on a fully shared GAP layer (GoogleNet-GAP-Zhou (Zhou et al., 2016) and our implementation of Inception-GAP-Zhou) and the other based on an unshared GAP Layer (our Inception-GAP5). Section 4.3 is a quantitative evaluation of the regularization introduced in Section 3.2 and Section 4.4 assesses the localization abilities of some of the models used up until then.

Hereafter, 5 metrics are used : accuracy, precision, recall, Mean average Precision (MaP), and Intersection over Union (IoU). Precision and recall, are computed such that we only consider a label to be true if the probability of its prediction is over 50% (as in the Keras1 implementation). This 50% threshold probability acts as a measure based on the confidence of the model. Along with these metrics, we compute the Mean average Precision which also reflects how confident a model is towards its predictions. The higher the MaP score is, the more confidence we can have on the ranked predictions of the model. Finally, we use Intersection over Union, which is a common localization metric in the literature, to evaluate the localization abilities of our model. The IoU is defined as the fraction of the overlap area of ground truth bounding-box with the predicted bounding-box over the area of their union. To be considered as correctly localized, the IoU of a predicted bounding-box should be over 0.5.

4.1 Action 40 Dataset

The Stanford 40 Action (Yao et al., 2011) dataset has been used to perform training and testing of the networks. This dataset is composed of 9532 images (4000 used for training, and 5532 for testing) of people performing one of 40 actions such as *drinking*, *cooking*, *reading*, *phoning*, or *brushing teeth*. We split the test images into two subsets : one with 3532 images used for validation and 2000 (50 images per class) for the test stage. In the dataset, all images are provided with a class label and a bounding box around the person performing the corresponding action.

4.2 Comparing Inception-GAP5 and Inception-GAP-Zhou

This section describes our comparison of Inception-GAP5 and Inception-GAP-Zhou, and reports the results given in (Zhou et al., 2016) with GoogLeNet-GAP. In Table 2, our model shows better performance than Inception-GAP-Zhou with respect to the the first three of metrics aforementioned (accuracy, precision and recall), meaning that Inception-GAP5 is better at classifying the dataset and that its classification is more reliable.

Table 2: Comparison of our architecture (Inception-GAP5) with respect to both the original GoogLeNet-GAP (Zhou et al., 2016) and its variant Inception-GAP-Zhou evaluated on Stanford Action 40 dataset. (*Acc.* stands for *Accuracy*).

Model name	Acc.	Precision	Recall
Inception-GAP5	**75.9**%	**80.1**%	**74.2**%
Inception-GAP-Zhou	73.7%	75.7%	72.8%
GoogLeNet-GAP-Zhou	70.6%	-	-

4.3 Impact of Regularization on GAP Models

This section presents the impact of L1 and L2 regularization terms on both Inception-GAP5 and Inception-GAP-Zhou.

One of the expected behaviors mentioned in Section 3.2 is to observe a sharper class separation by forcing the activations of the GAP maps, and therefore the activations of the softmax, to be close to zero. Such effect is seen in Table 3, where we observe the precision of both architectures increasing when applied the regularization term. Even though such phenomenon could result in an accuracy drop, this trend is not observed here. The accuracy and the precision of the Inception-GAP-Zhou model and its L1-regularized counterpart (Inception-GAP-Zhou-L1) both improved gaining 2.1 points in accuracy, 7.1 points in precision and 9.9 points in its Mean average Precision, whereas Inception-GAP5 only dropped by 0.4 points in accuracy, and gained 8.7 points in precision and 14.3 points in its Mean average Precision with the L1 regularization. Such results clearly demonstrate the benefit of L1 regularization on classification.

Table 3: The impact of L1 and L2 regularization terms evaluated on Inception-GAP5 and Inception-GAP-Zhou architectures. (*Acc.* stands for *Accuracy*, *Prec.* for *Precision* and *MaP.* for *Mean Average Precision*).

Inception-...	Acc.	Prec.	Recall	MaP.
GAP5	**75.9**%	80.1%	**74.2**%	63.8%
GAP5-L1	75.5%	**88.8**%	63.5%	**78.1**%
GAP5-L2	73.5%	88.1%	61.1%	77.0%
GAP-Zhou	73.7%	75.7%	**72.8**%	66.6%
GAP-Zhou-L1	**75.8**%	**82.8**%	70.5%	**76.5**%
GAP-Zhou-L2	73.5%	77.3%	71.2%	72.4%

The effect of regularization is also visible on the Class Activation Maps (Figure 3) where we plot the 40 action CAMs corresponding to the processing of the same image by each of six different models. We observe the absolute values of the CAMs, ranging from zero, up to the maximum value observed in all the CAMs for the model and image ($\max(f_k(x,y))$). In other terms, all CAMs are divided by the same maximum value of the observed CAMs for an image and a model.

Similar effects appears in both Inception-GAP5 and Inception-GAP-Zhou when applying the same regularization. When L1 regularization is applied (Figure 3b and 3e), all the neuron activations producing the CAMs becomes close to zero except the neuron recognizing a class, here the class is *"playing-guitar"*. We believe these activations are closer to reality as absent classes do not activate their neurons. The use of L2 regularization (Figure 3c and 3f) results in CAMs that are not sparse, that magnifies the receptive fields of a neuron and do not discriminate classes as properly as L1 regularization does.

4.4 Evaluation of Action Localization

The localization properties of our model are evaluated and shown in Table 4 and Figure 4. On the one hand, the table reports the results of the center of mass of the CAMs being in the ground-truth bounding-box, on the other hand, the Figure 4 reports how accurate we are in defining a bounding box around a class. To draw a bounding box around the predicted class, we threshold the CAM of the prediction by a given percentage of its maximum value and consider the bounding box to be smallest rectangle surrounding all these points (as in (Zhou et al., 2016)).

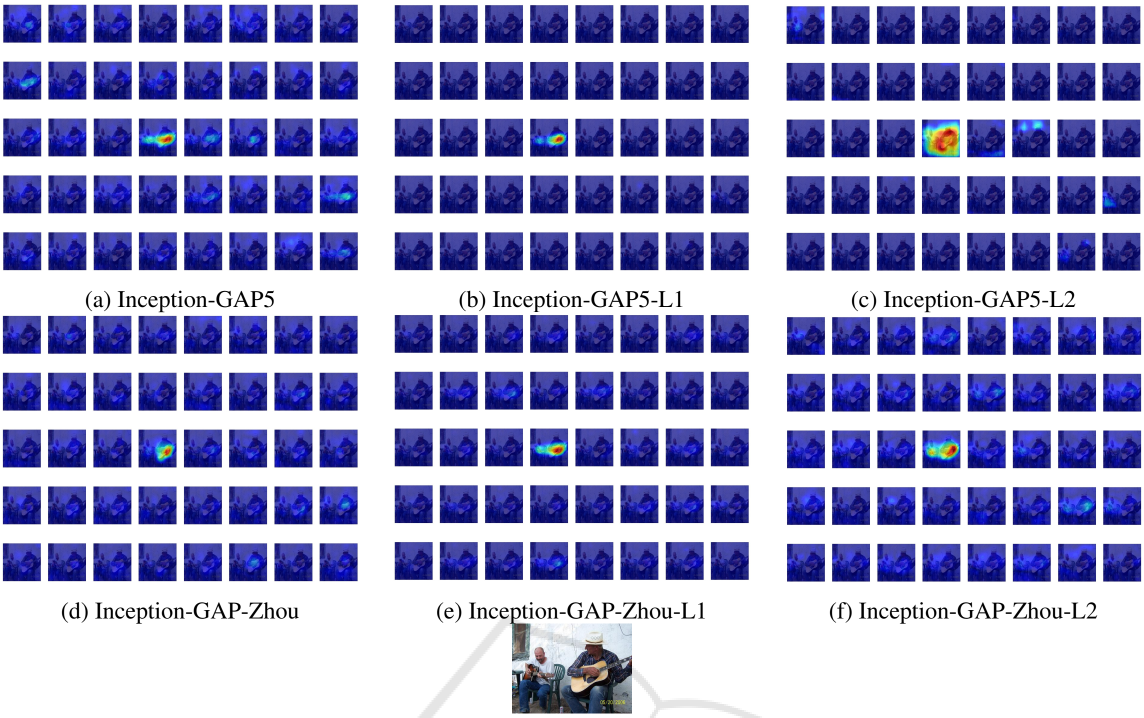
It is important to observe that, in the Stanford 40 Action (Yao et al., 2011) dataset, the discriminative parts of an action are mostly located next to the human performing the action (e.g. the *fishing* action is mostly determined and localized, with our method, by the presence of a fishing rod; instead of a person). Yet, as mentioned in Section 4.1, the ground-truth bounding-boxes provided are surrounding the person performing the action, and not the action, hence, our weakly supervised localization is penalized by its focus on the object rather than on the human.

The accuracies reported in Table 4, show that our model without regularization, called Inception-GAP5, which is based on unshared GAP layer, performs better than Inception-GAP-Zhou, and based on a shared GAP layer, by 6.3 points on prediction knowing the class ground-truth and 2.8 points not knowing it. Interestingly, the L1 regularized versions of Inception-GAP5 and Inception-GAP-Zhou do not show the same results. Yet, we believe that in both cases the regularization term constrained the last convolutional layer to be more attentive to discriminative elements due to the sparse constraints on the weights. In the case of Inception-GAP5, the networks is forced to be attentive to the object rather than the human, while, in the case of Inception-GAP-Zhou, the network is constrained to be more attentive on the human performing the action.

Table 4: Class localization accuracies. The second column shows the evaluation results when knowing the ground truth class and the third column considers the class predicted by the model.

Inception-...	based on the ground truth	based on the prediction only
GAP5	**72.8**%	**50.3**%
GAP5-L1	65.1%	46.6%
GAP5-L2	70.3%	49.6%
GAP-Zhou	66.5%	**47.5**%
GAP-Zhou-L1	**68.1**%	**47.6**%
GAP-Zhou-L2	67.4%	45.8%

SCIENCE AND TECHNOLOGY PUBLICATIONS



(a) Inception-GAP5

(b) Inception-GAP5-L1

(c) Inception-GAP5-L2

(d) Inception-GAP-Zhou

(e) Inception-GAP-Zhou-L1

(f) Inception-GAP-Zhou-L2

(g) Example

Figure 3: Visualization of the CAMs obtained with Inception-GAP5 (Figure 3a, 3b and 3c) and Inception-GAP-Zhou (Figure 3d, 3e and 3f) with and without L1 or L2 regularization terms to an image of people playing guitar (Figure 3g). Each map corresponds to one of the 40 classes. The mostly activated class is *"people playing guitar"*.

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

The same conclusions may be drawn from Figure 4 where we test the IoU localization metric with different threshold values. In the best case, Inception-GAP5 is better than Inception-GAP-Zhou by 2.7 points, the regularized version of Inception-GAP-Zhou is better than its non-regularized counterpart by 3 points, and that the regularized version of Inception-GAP5 is worse than its non-regularized counter-part, by 2.25 points. Here also, we explain these differences by the models being more attentive to discriminative elements - which may lead to a detection outside of the ground truth bounding box.

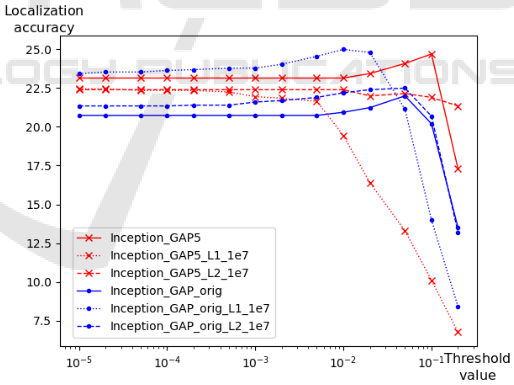


Figure 4: Percentage of correctly localized images based on the IoU metric depending on the threshold value selected to extract our prediction bounding-box.

5 CONCLUSION AND FUTURE WORK

This work presented semi-supervised image classification and localization on RGB images using an unshared GAP layers. Based on evaluations, we improve upon existing approaches in terms of the performance for both image classification and localization, in the context of human action localization, we hypothesize that the increased performance is due to the unshared GAP layer and to the reduced attention field in the model, which makes our model similar to (Oquab et al., 2015). This increased performance also exists even though the amount of parameters is reduced and the visualization method needs less computation.

As our next step, we are going to asses this modification on a larger dataset such as Imagenet, then, explore the use of shallower models (models with less convolutional layers) to tackle this problem. We will explore whether, in the context of shallower models, increasing the amount *m* of maps per neuron shows benefits. We have strong assumptions that a wider GAP layer will perform better than a narrower one in the context of shallow neural network.

In the context of human action localization, it will also be interesting to generate a prediction based on a coherence of several consecutive frames rather than on a single frame. Our future work will consider both an aspect of time and coherence between consecutive images and an associated audio coherence where we consider to transpose it from the semi-supervised spatial localization in images to semi-supervised temporal localization in audio events.

REFERENCES

Abadi, M. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Bishop, C. M. (2006). *Pattern recognition*. Machine Learning, 2006, vol. 128, p. 1-58.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778.

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

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436–444.

Lin, M., Chen, Q., and Yan, S. (2013). Network in network. *CoRR*, abs/1312.4400.

Nair, V. and Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814.

Oquab, M., Bottou, L., Laptev, I., and Sivic, J. (2015). Is object localization for free?-weakly-supervised learning with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 685–694.

Raina, R., Battle, A., Lee, H., Packer, B., and Ng, A. Y. (2007). Self-taught learning: transfer learning from unlabeled data. In *Proceedings of the 24th international conference on Machine learning*, pages 759–766. ACM.

Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., and LeCun, Y. (2013). Overfeat: Integrated recognition, localization and detection using convolutional networks. *CoRR*, abs/1312.6229.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556.

Szegedy, C. and Liu, W. (2015). Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826.

Yao, B., Jiang, X., Khosla, A., Lin, A. L., Guibas, L., and Fei-Fei, L. (2011). Human action recognition by learning bases of action attributes and parts. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1331–1338. IEEE.

Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., and Torralba, A. (2016). Learning deep features for discriminative localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2921–2929.

SCIENCE AND TECHNOLOGY PUBLICATIONS