# Improving Object Classification from different domains through Guided Zoom

Aquaro Gianluca
Politecnico di Torino
S265931
s265931@studenti.polito.it

Rasicci Riccardo
Politecnico di Torino
S265942
S265942@studenti@polito.it

Torcasio Vincenzo
Politecnico di Torino
S265627
S265627@studenti.polito.it

## Abstract

*We propose a new approch to the domain adaptation problem through the technique called Guided Zoom. The problem that we faced is to recognize and classify objects indipendently from their domain combining adversarial learning with discriminative feature learning. Then we refine the prediction through guided zoom that analyze the evidence upon which a prediction is made and tries to improve it. The recent Guided Zoom techique has been used only in fine-grained classification tasks improving the accuracy and the state of the art results on the benchmarck datasets exploting the fact that in fine-grained classification it is important to focus on details and in localizing the discriminative part of the image. We try to apply the same method but in a domain adaptation task in order to see if this method brings to some improvements.*

## 1. Introduction

Domain adaptation is a field of Machine Learning where there is a model trained on a source distribution data that is used in a different target distribution data. This technique aims to reduce as much as possible the drop of performances caused by the phenomenon called dataset bias or domain shift. There are several types of Domain Adaptation approaches that depends from the information that we have on the target task, on the data and the differences in the feature space. This field is so relevant because in many application cases we are in the situation described before that consist in having training and test data belonging to different distributions or domains. Conseguently the model trained on this data is not able to generalize the features into a common feature space.

For this purpose we use an approach called Adversarial Discriminative Domain Adaptation (ADDA) that consist in learning a discriminative representation in the source domain and then learning a separate encoder that maps the target data to the same space data using an asymmetric mapping learning through a domain adversarial loss.

For the Guided Zoom implementation we use the Evidence CNN and the Decision Refinement modules.

The Evidence CNN module is a classical CNN that is trained on the evidence of the correctly classified images. The evidence represent the most rilevant part of the image for the prediction of the CNN. They are obtained using a tecnhique called contrastive excitation backpropagation (cEB) on the images of the source domain, so, with this technique we extract from the image the most relevant part for the final prediction.

The Decision Refinement module instead, is used at test time in order to correct the original prediction on the basis of the conditional class probabilities of the Conventional CNN and the Evidence CNN.

The classification is performed on the target data by the source classifier on the target encoder that share the features of both the domains. After that the saliency of the test images are given to the Evidence CNN that will classify the images again. Then we try to improve the final classfication through the use of the Decision Refinement module that corrects the prediction of the Conventional CNN on the basis of the evidence.

## 2. Related Works

### 2.1. Adversarial Discriminative Domain Adaptation

Adversarial Learning is a tecnique to face the problem of Domain Adaptatation through the minimization of an adversarial loss (i.e. minimization of domain discrepancy distance between an adversarial object and a domain discriminator) [2] . Basically it consist in fooling models through malicious input. Adversarial Learning has been used for Generative tasks using Generative Adversarial Networks (GANs), it is based in putting two networks one against each other, the generator and the discriminator. The generator is trained to produce images to confuse the discriminator which have the task to distinguish from real images. The generator is a generative model that catch the data distribution, while the discriminative model tries to distinguish between samples drawn from the generative model and the training data predicting a binary label. Both networks are trained using backpropagation on the label prediction loss in a minmax fashion, updating the generative to minimize the loss and the discriminative model to maximaze the loss.

Adversarial Discriminative Domain Adaptation uses a discriminative base model, unshared weights and standard GAN loss. The discriminative base model is choosen because it is supposed that the parameters required to generate convincing samples are irrilevant for discriminative adaptations tasks. This method like adversarial adaptive method optimize directly in discriminative space. Untying the weights allows indipendent source and target mapping, that permit to extract more specific feature to be learned. Furthermore a pre-trained source model is used as an initialization for the target representation source, because of that the target model is modified to match the source distribution. The loss is the same of GAN but instead of using the minmax loss, a generator has been trained with the standard loss function with inverted labels. This splits the optimization into two indipendent objectives, one for the generator and one for the discriminator. This objective has the same fixed-points properties as the minmax loss and provide stronger gradients to the target mapping.

Our approach in the implementation of the Adversarial Discriminative Domain Adaptation has followed faithfully the paper without changing its structure or apporting any modifications.

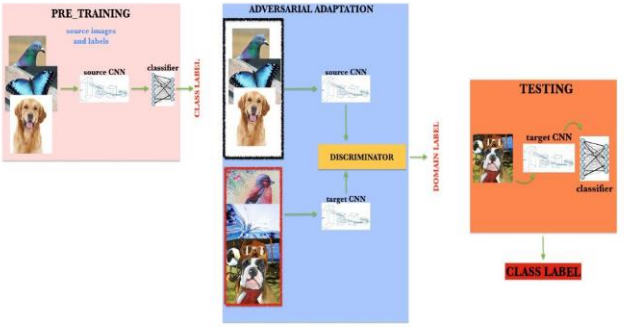


Figure 1: Overview of Adversarial Discriminative Domain Adaptation. The method can be divided in three step. The pre-training that consist in training the source CNN and the source classifier with the source domain. The Adversarial Adaptation learn a target encoder CNN such that a discriminator that see source and target images can't reliably predict the domain label. The testing where target images are mapped with the target encode to the common feature space and classified by the source classifier.

## 2.2. Guided Zoom

Guided Zoom uses the evidence to make a preliminary decision at test time and by comparing it with a reference pool of evidence and prediction pairs [1]. The pool is composed of evidence and prediction for corrected classified training examples. The main goal of the Guided Zoom is to improve the classification comparing the evidence of the refined class prediction with the training evidence of that class and to the evidence of the other candidate top classes. By examining the evidence the Guided Zoom can achieve state-of-the-art results. This is done thorugh the Evidence CNN and the Decision Refinement.

The Conventional CNN is a net that is trained for image classification and gives as output the class conditional probabilities upon which a prediction is made.

Evidence CNN takes as input for training the evidence of the correctly classified images on the Conventional CNN. The evidence are generated using a top-down spatial grounding technique: contrastive Excitation Backprop (cEB). cEB passing recursively top-down signals layer by layer computes the class-specific discriminative saliency maps. Then an adversarial erasing is performed deleting progressively the evidence with a black patch on the previous most salient evidence to encourage a highlight of the next salient evidence.

The Decision Refinement consist in changing the prediction on the base of the evidence, for example if the second-top predicted class is more coherent with the reference pool (evidence, prediction) the refined prediction will be inclined towards this class. At test time in fact, the evidence upon which a prediction is made is analyzed examining the consistency of the test image (evidence, prediction) with the reference pool that is used to train the Evidence CNN. The visual evidence is used for prediction refinement and the refinement will be inclined toward the top-k classes proportionally to the coherence with the reference pool.

Our approch is slighlty different in some implementative choise due to our limited resource and the lack of computational power and memory.

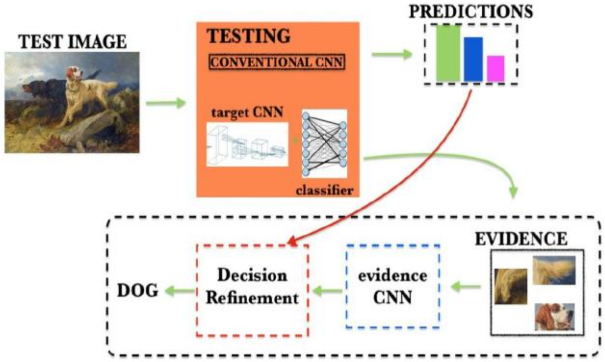


Figure 2: Architecture of Guided Zoom. The Conventional CNN gives as output the conditional class probabilities. The pool evidence, prediction for the current image is generated and passes through the Evidence CNN which will outputs his own conditional class probabilities. The predictions of the two nets are evaluated by the decion refinement module that will eventually refine the original prediction on the basis of the evidence.

### 2.3 Grad-CAM

Grad-CAM uses the gradient information flowing into the last convolutional layer of the CNN to assign importance values to each neuron for a particular decision of interest [3]. This technique is fairly general and can be used to explain activations in any layer of a deep network. In order to obtain the class-discriminative localization map Grad-CAM $L^{c}_{Grad\text{-}CAM} \in \mathbb{R}^{uxv}$ of width $u$ and height $v$ for any class $c$, $y^c$ (before the softmax), with respect to feature map activations $A^k$ of a convolutional layer. These gradients flowing back are global-average-pooled over the width and height dimensions to obtain the neuron importance weights of $\alpha^{c}_{k}$ .

$$\alpha_k^c = \overbrace{\frac{1}{Z}\sum_i \sum_j}^{\text{global average pooling}} \qquad \underbrace{\frac{\partial y^c}{\partial A_{ij}^k}}_{\text{gradients via backprop}}$$

During the computation of $\alpha^{c}_{k}$ while backpropagating gradients with respect to activations, the exact computations amounts to successive matrix products of the weight matrices and the gradient with respect to acrivation functions till the final convolutional layer that the gradients are being propagated to.

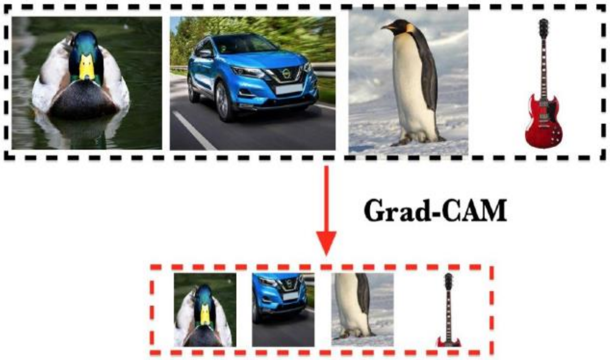


Figure 3: Practical use of the Grad-CAM method on our dataset. From the picture we can see how the method find the most important evidence for the prediction.

## 3. Methods

We now present the implementation of the architecture for the image classification between domains. Our architecture can be divided in two sets. In the first we implement the ADDA training the source CNN, the target CNN and the Discriminator, in the second we describe the implementation of the Guided Zoom defining the Evidence CNN and the Decision Refinement module.

In the following we will analyze in details all the components of our architecure.

### 3.1 Source CNN (Pre-Train)

The first component of the architecture is the Source CNN, which is composed by two elements, the Source Encoder and the Source Classifier. We defined the Source Encoder as the full net except for the last layer which has been eliminated manually. The Source Classifier instead is the last layer than we eliminated from the Source Encoder, it is defined and instantiated separately. The Source Classifier is a one fully connected layer, conseguently its forward function consist only in forwarding this layer. Then we train both the Encoder and the Classifier on the source data in order to train the classifier for the final classification.

### 3.2 Target CNN (Adversarial Adaptation)

For the Adaptation we have implemented a Target Encoder and a Discriminator. The Target Encoder has the same structure of the Source Encoder, so it does not present the last layer. We have decided to define the Discriminator as a Fully Connected Layer with three layers that have dimensions: 4096x500, 500x500, 500x2. Each layer is separated by a ReLu activation function. We have two output neurons because the Discriminator have to predict the domain label.

In this phase we train the Discriminator and the Target Encoder, the Discriminator receives both the Source and Target images while the Target Encoder receives only the Target images. For the training of the Discriminator we firstly extract and concatenate the features outputs from the Source and Target Encoder and we make the prediction on them. Then we prepare real and fake labels and compute the loss of the Discriminator on the basis of the prediction and the labels. Doing in this way we are confusing the Discriminator because it can't surely predict the domain since we are passing it false labels. The Target Encoder is trained with this Discriminator so the extracted features will be shared with the features of the source data.

### 3.3 Evidence CNN (Train Evidence)

The next step in our architecture need the definiton of the Evidence CNN. The Evidence CNN has the same structure of the Conventional CNN that is the combination of the Target Encoder and the Source Classifier, so for simplicity, the Evidence CNN is divided in Encoder and Classifier as the others net. In our case the Evidence CNN is trained on all the evidence of the train images of the Conventional CNN and not only on the correctly classified images of the Conventional.

That is because we have used the ImageFolder class to read the dataset and this class does not allow us to keep the name of the image that the Conventional has classified correctly. So assuming that the Conventional will classify in the right way the great part of the training images we have decided to train the Evidence on all the evidence of the training set. Since the Conventional is trained on the Source data also the Evidence will be trained on them. Furthermore in our implementation we train the Evidence CNN only on the most discriminative evidence of the images because we don't perform adversarial erasing since we don't have the practical instruments to perform this kind of operation. The most important patch for the prediction of a certain class on which the Evidence CNN will be trained is obtained through the Grad-CAM method instead of the contrastive Excitation Backpropagation. We have chosen this method because it is very similar to the cEB approach but it is more intuitive and effective.

### 3.4 Decision Refinement

For the implementation of the Decision Refinement it was necessary compute the conditional class probabilities of each image for Conventional and Evidence CNN. For doing this we have used the softmax function on the output of the net. In this way we have obtained the probabilities for all the labels that we are considering. Suddenly we have decided to consider only the 3-top class for the prediction and we have saved in two different structures (both ordered in ascending value of labels) the pairs probability, label.

For the implementation of the Decision Refinement module we have adopted only one strategy to correct the initial prediction that is based on computing the mean between the probabilities of the Conventional and the Evidence for the three top classes. The label that will have the greater value of probability will be the final prediction. For example, if the Conventional gives as output the following probabilities [0.5, 0.4, 0.1] for the labels [15, 18, 33] and the Evidence gives as output the probabilities [0.3, 0.6, 0.1] for the labels [15, 18, 33] (because we have sorted them in ascending order of label), the Decision Refinement computes the mean between the probabilities that is [0.4, 0.5, 0.1] for labels [15, 18, 33]. So the final predicted label will be the label number 18 instead of the label 15 as predicted by the Conventional CNN.

### 3.5 Test

At test time each image pass through the Conventional and Evidence CNN. From the output of the Conventional we take the conditional class probalities of the top-3 classes, suddenly for each test image we compute the saliency through Grad-CAM, obtaining the corresponding evidence that will go through the Evidence CNN. At this point, also the Evidence CNN will give as output the three conditional class probabilites of the top-3 classes on the basis of the evidence. These probabilities will be analyzed by the Decision Refinement that computes the mean. The final class will be the label that have the highest value of probability between the Conventional and the Evidence CNN.

## 4. Experiments

All our experiments have been conducted on Google Colab and the dataset was uploaded on Google Drive. Since on Google Colab we have several limitations like only one GPU is available at time, the memory is restricted, the hours of computation are limited, our experiments are not completely performing and satisfactory.

### 4.1 Data

The Dataset that we have choosen is DomainNet. This dataset is composed by over 0.6 Millions of images divided between six different domains that are: Real, Painting, Sketch, ClipArt, Quickdraw and Infograph. DomainNet is composed by 345 classes, the dataset contains a large variety of categories, such as animals, common and particular objects.

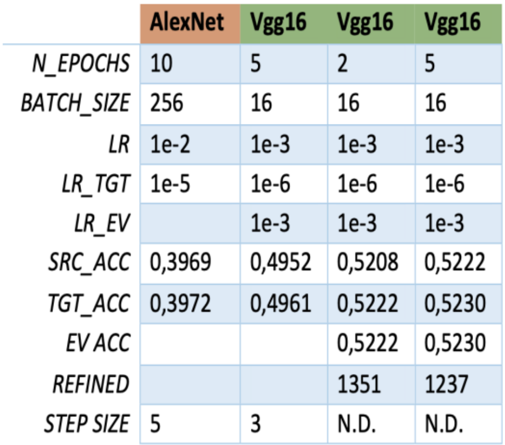
For computational reason we have considered only two domains that are Real and Painting. In our experiments we consider the Real domain as Source and the Painting as Target domain. Doing so, our training set is Real and our test set is Painting.

Again since our computational resources are limited we have taken a subset of 100 classes, for a total of 54784 training images and a total of 25760 test images.

### 4.2 Results

We have conducted several experiments from which we obtained many different results one from each other. The first try that we have done is with AlexNet, on which we implement only the Adaptation. In our experiments we have used a different Learning Rate for each network in order to see eventual improvements. On AlexNet we have used a Learning Rate equals to 0.01 on the Source and a Learning Rate of 0.000001 on the Target, with a large Batch of 256 images and 10 epochs. Since the results obtained were not satisfactory we have decided to use a more complex net that is Vgg16. Doing the experiment with Vgg we noticed that the AlexNet's parameters were not usable. In fact we have reduces the Batch until 16 images because with higher values the computational resources of Colab does not allow us to finish our evaluation. Substituting AlexNet with Vgg only on the Adaptation have increased Source accuracy from the 39.69% to 49.52% and the Target accuracy from 39.72% to 49.61%. As expected the Discriminator accuracy instead is always around the 50% which means that it can not predict reliably the domain label.

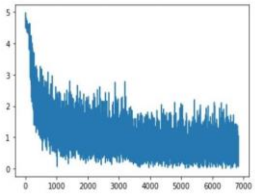
With Vgg16 we have implemented all our architecture. We have conducted two experiment, changing from one to the other only the number of epochs from 2 to 5. The common parameters are: Source Learning Rate 0.001, Target Learning Rate 0.000001, Evidence Learning Rate 0.001, Batch Size equals to 16. And we have obtained the following results considering two epochs: the source accuracy is increased from 52.08% to 52.22%. For five epochs the source accuracy goes from 52.22% to 52.30%, the total number of refinement prediction are 1351 in the first experiment against the 1237 of the second. The number of the refined prediction is 1351 and 1237 for 2 and 5 epochs so we can conclude that in some cases the prediction of the Conventional was different from the prediction of the Evidence and we have refined it. Since the accuracy of the Conventional and the Evidence are the same in both cases our technique is not very effective for our implementation, parameters and dataset.



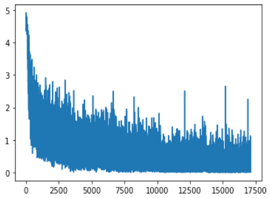
| | AlexNet | Vgg16 | Vgg16 | Vgg16 |
|---|---|---|---|---|
| N_EPOCHS | 10 | 5 | 2 | 5 |
| BATCH_SIZE | 256 | 16 | 16 | 16 |
| LR | 1e-2 | 1e-3 | 1e-3 | 1e-3 |
| LR_TGT | 1e-5 | 1e-6 | 1e-6 | 1e-6 |
| LR_EV | | 1e-3 | 1e-3 | 1e-3 |
| SRC_ACC | 0,3969 | 0,4952 | 0,5208 | 0,5222 |
| TGT_ACC | 0,3972 | 0,4961 | 0,5222 | 0,5230 |
| EV ACC | | | 0,5222 | 0,5230 |
| REFINED | | | 1351 | 1237 |
| STEP SIZE | 5 | 3 | N.D. | N.D. |

Figure 4: Summarize of all the experiments that we have done. We can see how the accuracy is greater using Vgg instead of AlexNet. The accuracy using Adaptation is a bit greater and the effect of the Guided Zoom does not increase the accuracy.

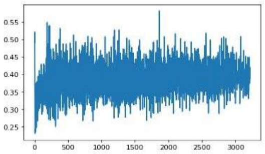
The graphs before shows the trends of the loss in function of the number of steps. The loss tends to zero except for the discriminator and the evidence, maybe with better values of LR we can obtain better performances. We report the graphs obtained by running the experiment with Vgg with 2 epochs on the left and with 5 epochs on the right.



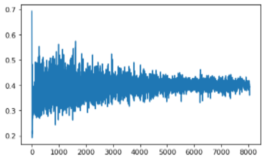
Source Loss before Adaptation



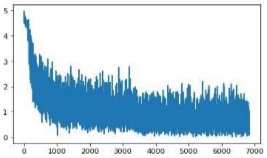
Source Loss before Adaptation



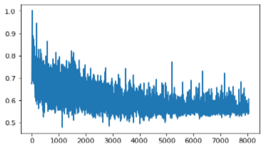
Discriminator Loss



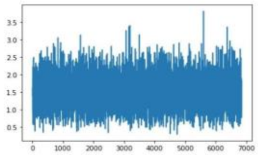
Discriminator Loss



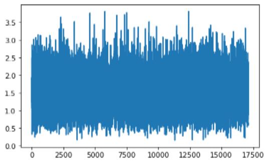
Target Loss after Adaptation



Target Loss after Adaptation



Evidence Loss



Evidence Loss

## 5. Conclusions

After all the experiments we are arrived at the conclusion that the technique of the Guided Zoom can improve the accuracy if implemented correctly with the rights instruments on domain adaptation tasks. For example taking into account more classes instead of the top-3 that we have chosen for our implementation or maybe including in the training of the Evidence CNN only the correctly classified images of the Conventional CNN and using other patches performing adversarial erasing. Other improvements could be for example the tuning of the hyperparameters or training on more epochs. Another improvement could be the introduction of different strategy for the Decision Refinement like assign at each top-k probability a weight. Other improvements could be reached using a better implementation of the ADDA.

## References

[1] Sarah Adel Bargal, Andrea Zunino, Vitali Petsiuk, Jianming Zhang, Kate Saenko, Vittorio Murino, Stan Sclaroff, Guided Zoom: Questioning NetworkEvidence for Fine-Grained Classification. BMVC, 2019.

[2] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, “Adversarialdiscriminative domain adaptation,”CoRR, vol. abs/1702.05464, 2017.

[3] R.R. Selvaraju, A. Das, R. Vedantam, M. Cogswell, D. Parikh, and D. Batra. “Grad-CAM: Visual Explanationsfrom Deep Networks via Gradient-based Localization”, 2016.

[4] Link to the repository with the code :

https://github.com/RiccardoRasicci/ML-Project