# **Cross-Domain Object Detection with YOLO**

|  |  |  |
| --- | --- | --- |
| Michele D’Addetta  Politecnico di Torino  s257801@studenti.polito.it | Lorenzo Montoro  Politecnico di Torino  s266049@studenti.polito.it | Giorgio Giacalone  Politecnico di Torino  s267545@studenti.polito.it |

**Abstract**

*There are some papers about cross-domain classification, but only few of them are about object detection. In this paper, we present the results obtained and the methodologies adopted for cross-domain supervised object detection, basing ourselves on the paper by Inoue, Furuta, Yamasaki, Aizawa. For this paper, we have access to images with instance-level annotations in multiple source domains and images with only image-level annotations in a target domain. The classes to be detected in the target domain are a subset of those in the source domains. We start from a YOLO implementation using a pre-trained Darknet, and then we apply a two-step progressive domain adaptation technique by fine-tuning the Darknet on two types of artificially and automatically generated samples. Finally, we test our YOLO network on a subset of the Comic dataset, achieving an improvement of accuracy in detection of PERC% with reference to the detections made with the network pre-trained with Pascal VOC.*

# **Introduction**

The idea behind this paper is to train a network implementing the YOLO algorithm to detect objects in real-time in a target domain, which has only information about objects in the images, but no information about where they are placed. So, our goal is to use instance-level annotations of other domains and try to generate them on the target one. To make this possible, we have used the Darknet as implemented by [reference to pjreedie] with no pre-trained weights applied. Object detection is the process of autonomously recognizing objects (e.g. people, animals, cars etc.) in an image or video not only classifying those objects correctly deciding what they are, but also recognizing their positions and sizes in the input image. In fact, object detection is about assigning the right label (i.e. the class) and the right bounding box (i.e. a rectangular shaped frame) to an object. So, we define as image-level annotation of an image the set of classes coupled with the objects contained in that image, without knowing where objects are. The knowledge of the class and the position of objects in an image defines the instance-level annotation of the image. The position of an object is defined with a bounding box **b**, defined as *(x, y, width, height)*, where *x* and *y* are the coordinates in pixel of the upper left vertex of the box.

Right now, most object detection networks and researches focus on images from the real world­­­­­­­ obtaining great results, but object detection can be very useful also in other domains than just real-world one. The methods explained in this paper aim for detect objects in domains covered by few datasets with lack of annotations, i.e. painting or comic, and this is very useful to improve performances on them and transfer knowledge from a well-known domain. For example, an automated museum guide will exploit the knowledge obtained from these techniques to recognize objects in form of statues, paintings, etc.

We start from a Darknet pretrained on images with instance-level annotations from a source domain, then we fine-tune it in the target domain. This approach seems the best one, but there are no instance-level annotations available in the target domain. To generate this information, we use the same methods applied in the paper made by Inoue, that are *Domain Transfer* and *Pseudo Labeling*, that will be explained later. To perform *Domain Transfer*, we use a CycleGAN implemented by [AMICO FRIZZ], in order to transform data from source domains in the target one. Once the annotations are automatically created, we fine-tune the Darknet on them. The results obtained by the original paper applying this task achieve an improvement of 5 to 20 percentage points in terms of mean average precision, but the domain adaptation was made starting from a single source domain to a target one. The source was always Pascal VOC, which contains real-world images, and three networks were created to transfer instance-level knowledge to three different domains, that are Clipart, Comic and Watercolor. In our implementation, we want to know how much the results change if we use several sources and transfer all their information to the target one. All the sources have instance-level annotations, meanwhile the target one has only information about the objects in the images, so image-level annotations. The domains we take as source domains are Pascal VOC 2007 and 2012, Watercolor and Clipart. Comic, instead, is used as target domain.

Figure 1: Samples of images from the four used datasets: Pascal VOC, Clipart, Comic, Watercolor (from upper left to lower right)

# **Related Work**

## **Supervised Detection: YOLOv3**

****Many methods can be used to realize object detection in images and video, such as Fast R-CNN, Faster R-CNN, YOLO. The first two use a similar approach: they first define some region of interest in the image, then they try to classify the objects into them. These two approaches are more accurate than YOLO, but they are slower. In fact, YOLO is faster and able to detect objects in real time, without a big latency between the request and the response. The approach of YOLO in the detection is quite different, and this is because YOLO splits the image in a grid and creates a set of bounding boxes in each of them. Then it regresses from each box to a box made by *(x,y,width,height,confidence)*, where the first four values have the same meaning explained before, and the confidence says the probability of having a correct prediction. Once this is done for each box, it predicts scores for the classes in the dataset, including the background class, and outputs them. YOLOv3 is able to learn instance-level annotations from a training dataset and then to detect objects in an image in about 40 milliseconds, which makes it suitable for real-time detection.

## **Cross-Domain Object Detection**

The target domains may have few or zero information about the position of the objects in the images. This may lead to an unfeasible detection. The idea is to take instance-level annotations from a source domain and use them in the target one as starting point. So, the network learns from a well-known domain, that has big and several datasets of images coupled with instance-level annotations. But if it tries to predict objects in other domains’ images, the results significantly worsen, because the features in the images are different and the network is not able to understand them. In order to increase performance, the network has to use knowledge of the source domain and to extract patterns from target one, and this will make the prediction more reliable. As we said before, we use the same approach, in terms of methods, of [Inoue]. The [Inoue] one takes a real-world dataset as source and adapts it to a target one, which has only image-level annotations. In our implementation, instead, we use multi-domain source and one target domain. The domains used as source have all the instance-level annotations. We also have instance-level annotations of the target one, but we use them only for testing, in order to get the prediction precision. We want to know if the results get worse with multiple domains or if they can be used to make the network more robust.

## **Domain Adaptation**

There are lots of methods to perform the domain adaptation task. The one proposed here is the CycleGAN, that is an unsupervised method able to translate images from a source to a target domain and vice versa. It uses two GANs, which are unsupervised generative models able to generate images similar to the set used as training. The GAN uses an implicit density distribution to generate data and uses a two-player approach. There are a generator and a discriminator in each of the GAN that tries to fool the generator creating real-like images and to distinguish between real and fake images respectively. The CycleGAN generates a fake image from source to target and tries to reconstruct the image to the source from the target. The cycle-consistency losses are used to update the parameters in the network in order to create images from source to target like the real target ones.

# **Dataset**

In our implementation, four datasets have been used. The source ones are Pascal VOC [2007] [2012], Clipart [] and Watercolor. The target one is Comic. Comic and Watercolor datasets belong to BAM! dataset [], which contains several domains, but some of them are not suitable for the detection, because they contain just one object placed in the center in most of the images, so the detection is not so challenging. The only dataset with instance-level annotations is Pascal VOC, but, as explained in [Inoue paper], some of the images of the other domains have been annotated in order to have some information for the testing and to understand the results. Only the source domains’ ones are used in the training. Examples of images for each dataset are shown in *Fig. 1*.

## **Pascal VOC**

Pascal VOC is the biggest dataset among the used ones, and it is composed of images from real world. The dataset consists of two subparts made in different years, which are 2007 and 2012. The first one contains 9963 images, whereas the second one has 17125 images. Both datasets address 20 classes. This set is used to pre-train the Darknet and to transform the contained images in the target domain for the domain transfer.

## **Clipart**

Clipart is a dataset made by drawings, pictures and cartoons-like images. It contains 1000 images belonging to 20 classes, that are the same of Pascal VOC. The images have been taken from CMPlaces dataset and correctly annotated. This set is used during the domain transfer phase.

## **Watercolor**

Watercolor is a subpart of the BAM! dataset and contains 2000 paintings, which are made by objects of 6 classes. This set is used during the domain transfer.

## **Comic**

Comic is a part of BAM! dataset too, and, as the name says, it has images taken from comics. The dataset is made of 2000 images, some of them are colored and others are black and white. The classes belonging to the dataset images are 6.This set is the target domain, so it is used as aim during the domain transfer and to perform the pseudo labeling.

Remark that both Comic and Watercolor have only 6 classes, specifically *bicycle, bird, car, cat, dog* and *person*. These classes are a subpart of the one of Clipart and Comic.

Another important observation is that BAM! dataset images are not instance-level annotated, but we take a subset of them that have been annotated by the creator of the paper [Inoue paper]. So, comic has instance level annotations, that we will use only to test the results. In this way we will have a quantitative value that shows how good our network performs.

# **Proposed Method**

In order to achieve the project goal, we use YOLO as our object detector algorithm. It is implemented by the Darknet network, a supervised model able to predict instance-level annotations, consisting of objects class and their bounding boxes. We pre-train it on Pascal VOC dataset so that it starts learning something about the images features and the information contained in them. However, if we try to detect objects directly on the target domain, the results are very poor, because it hasn’t instance-level information about Comic to be used for training. In fact, source and target domains are from different distributions, and the accuracy of the prediction decreases significantly. To improve the model, we apply *Domain Transfer* and *Pseudo Labeling*. These are two methods also implemented by [Inoue Paper] able to increase the prediction accuracy in the target domain without having instance-level annotations about them. This is done transferring knowledge from the source domains to the target one. Instance-level annotations of Clipart and Watercolor were made by the creators of [INOUE paper], so we consider them reliable. Our purpose is to implement a multi-source approach and see if the model can reach higher level of accuracy. If it works better, then this approach is extendible to all the unknown target domains starting from the knowledge of multiple sources, considered reliable and very detailed. The steps followed during the workflow are shown in *Fig. 2*.

## **Pre-training**

Darknet implementation is taken from [stronzo1]. Once loaded, the object detector needs to gain some knowledge about the source domains, so as starting point it has been pretrained over 30000 iterations with LR of 0.001 and using a step-down policy, that reduces the learning rate after 15000 and 25000 steps, with a γ equal to 0.1.

## **Domain Adaptation**

Feature and output spaces address to the same task, namely to create instance level annotations, but they have very different marginal distributions. If we plot domains in the features space, they appear in a very different way. The main idea is to take images from source domains and transform them in images similar to the target ones. Then their instance-level annotations can be used by the net to understand images from a new domain, learning their features using the original domain annotations. In order to generate new samples, that will have the same appearance of the target domain, we use a CycleGAN []. Each generated image contains the same objects of the original, having a comic fashion.

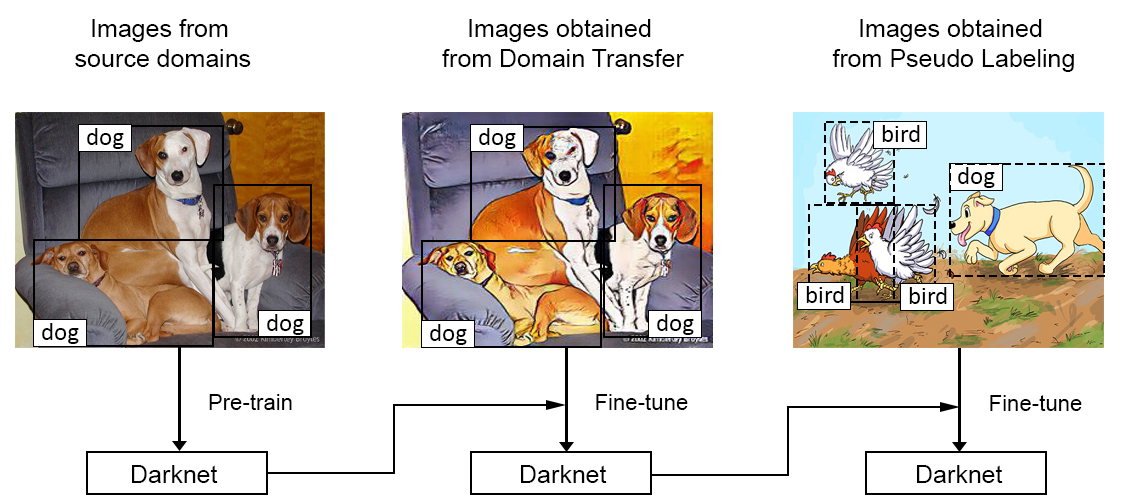
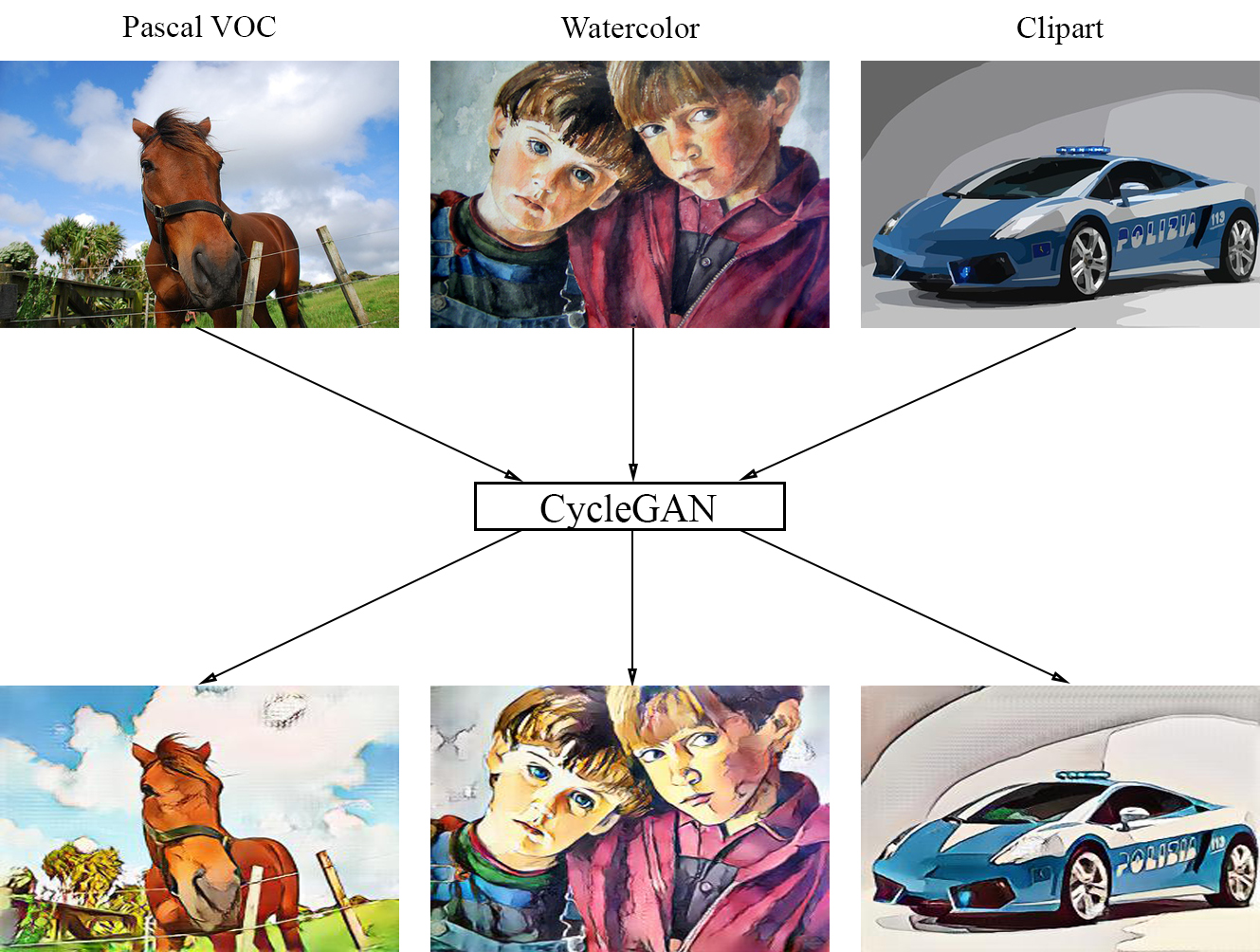
The implementation is taken from [stronzo2] and it is trained for 20 epochs over the three source domains creating new samples. In the first ten epochs, the CycleGAN uses a learning rate of 2x10-4, then, in the last ten epochs, the learning rate is decreased linearly after each epoch, until it becomes 0. After the training, we obtain the parameters of the net, that will be used to transform all the source images. They have been stored and used for training again the Darknet for 10000 iterations with a constant LR value of 10-5. The training configuration used by the creators of [INOUE] is different, but when we tried it the results were sub-optimal. In fact, the net didn’t gain enough information about those generated images. Using our training configuration, the detector will try to extract as more instance-level annotations from the generated images as it can. The problem is that the quality of images generated by the CycleGAN is lower than the real target domains pictures, but after this phase, the Darknet has a deep knowledge of instance-level annotations. These can be used to classify the target domain images. Samples of generated images for each source dataset are shown in *Fig. 3*.

Figure 3: Examples of images transformed by Domain Transfer from the sources (top) to the target (bottom).

Figure 2: The steps in the workflow.

## **Pseudo-Labeling**

So far, our object detector could predict images of target dataset, but the results are not so accurate. In fact, even if the Darknet is trained over images that looks like Comic ones, the real target images are quite different.

In order to achieve better performance, Darknet needs to understand as much as possible the real data from the target domain, in order to classify images correctly.

Pseudo-Labeling technique, also implemented by [], is the choice made to help the model to gain more accuracy on target domain, by creating pseudo instance-level annotations for each image from target domain. The objective is to classify each image of Comic with the parameters obtained from the Domain Transfer and pick, for each class from its image-level annotation, the top-1 confident detection. In our implementation, we select the best prediction for each class present in a certain image, according to its image-level annotation. If an image contains more than one instance of a certain class *c*, we take the *k*-most confident predictions on *c*, where *k* is the minimum between the obtained predictions for *c* and the number of instances of *c* in the image. Remark that if the Darknet predicts an object belonging to a class that hasn’t instances in a certain image, this prediction is discarded. Finally, we save them as instance-level annotations and discard the least confident ones. The created pseudo-annotations are used to fine-tune the Darknet for 10000 iterations, using a learning rate of 10-5.

# **Experiments and results**

## **Evaluation metrics**

In order to evaluate the accuracy of the detection on the target domain, we use Average Precision (*AP*) and its mean value (*mAP*).

We calculate the Average Precision (*AP*) for each class

of the target dataset, and then we average these results to obtain *mAP*. As explained before the classes of the target domain, that is Comic, are six, and all of them are a subset of the source domains classes.

We evaluate three different cases:

* *Case 1*: we pre-train Darknet on Pascal VOC and we use the same parameters and network architecture to fine-tune during the *Domain Transfer*. Then, in order to execute the *Pseudo Labeling*, we define the pseudo-labels on the training images, discard the predictions of objects not in the set of Comic classes and we modify the last layer of the Darknet. In this way, the net will be able to classify only objects belonging to the six classes of Comic.
* *Case 2*: we pre-train Darknet on Pascal VOC, but, in this case, we change the last layer of the Darknet, and we use the pretrained weights architecture to fine-tune during the *Domain Transfer*. So, now during the *Pseudo Labeling* we use a network that only knows the classes of the target domain. The pseudo-annotations are generated using the technique explained before in the *Pseudo-Labeling* paragraph, taking the top-k predictions for each class.
* *Case 3*: we pre-train Darknet on Pascal VOC, and also in this case, we change the last layer of the Darknet, and we use the pretrained weights architecture to fine-tune during the *Domain Transfer*. The pseudo-annotations that will be used during the fine-tuning are generated using the *Pseudo-Labeling*, but in this case, we take only the top-1 confident detection for each class for all the images. This is the approach used by the creators of [INOUE].

Another parameter we keep in consideration, for the calculation of results, is the *IoU* (Intersection over Unit), which is the percentage of overlap between the bounding boxes of ground truth and the bounding boxes predicted by YOLO. This parameter is used to determine a threshold to define if a predicted bounding box can be considered correct or not. In our case, we use a threshold for *IoU* of 50%, which means that if a detected bounding box overlaps more than the half of the ground truth bounding box and the detected class corresponds to the real one, then the predicted bounding box is considered correct.

In order to calculate *AP*, the detector predicts classes and bounding boxes on the test images, then it ranks all the predictions in descending order depending on the value of confidence of the prediction. Then, if the *IoU* is greater than 50% and the class is correct, the detection is considered as *True Positive (TP)*, otherwise there will be two possible classifications of the detection:

* *False Positive (FP)*, if the *IoU* is lower than 50% or the bounding box around the object is duplicated
* *False Negative (FN)*, if the *IoU* is greater than 50%, but the predicted class is not the real class

Once the predictions are ranked and classified in terms of *TP, FP* and *FN*, the precision is calculated for each class as:

|  |  |  |
| --- | --- | --- |
| (1) |  | (1) |

where *n* is the number of objects predicted for the class *c*, *nTP* is the number of objects correctly detected until the *i-th* analysed detection.

The *mAP* value is computed as follows:

|  |  |  |
| --- | --- | --- |
| (1) |  | (2) |

where *c* is a class of the dataset, *C* is the total number of classes in the target dataset and *AP(c)* is the Average Precision of class *c*.

We test the accuracy of the detection after Domain Transfer (DT) and then also after Pseudo Labeling (PL). As test set, we split the whole Comic dataset and use a part as test and the other during the fine-tuning of the Pseudo Labeling. As training set, we use the whole Pascal VOC and half of Clipart and Watercolor datasets. We leave the other two splits because we use them for testing in the intermediate results of the entire workflow.

## **Results and discussion**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | AP for each class | | | | | |  |
| Method | bicycle | bird | car | cat | dog | person | *mAP* |
| Pre-train | 25.7 | 6.6 | 9.5 | 8.4 | 7.4 | 21.0 | 13.1 |
| *Case 1* |  |  |  |  |  |  |  |
| DT | 43.8 | 20.4 | 22.0 | 15.7 | 19.6 | 53.2 | 29.1 |
| DT+PL | 0.3 | 0.5 | 8.1 | 1.3 | 0.2 | 44.2 | 9.1 |
| *Case 2* |  |  |  |  |  |  |  |
| DT | 34.2 | 13.5 | 19.6 | 13.5 | 13.9 | 43.3 | 23.0 |
| DT+PL | 6.7 | 10.4 | 21.5 | 11.1 | 17.9 | 40.4 | 18.0 |
| *Case 3* |  |  |  |  |  |  |  |
| DT | 34.2 | 13.5 | 19.6 | 13.5 | 13.9 | 43.3 | 23.0 |
| DT+PL | 28.5 | 11.3 | 20.3 | 11.1 | 20.3 | 42.0 | 22.3 |
| DT+PL1000iters | 35.0 | 14.2 | 19.8 | 11.7 | 19.2 | 42.8 | 23.8 |
| Table 1: *AP* results for each class and *mAP* in percentage. |  |  |  |  |  |  |  |

The *Pre-train* row in the *Table 1* shows the *mAP* of the pretrained network using Pascal VOC and tested on the target dataset. As expected, the results are very poor, because the Darknet has no knowledge on the target domain.

As the *Table 1* shows, the results obtained using the first approach are very good for the *Domain Transfer*, meanwhile the *Pseudo Labeling* phase worsen. This is because the Domain Transfer is done using all the source classes and only in the Pseudo Labeling the last layer is changed. In fact, the pseudo labels cannot improve the detection as expected.

About the last case, the results given both in *Domain Transfer* and *Pseudo Labeling* are very similar, with a slight decrease after the pseudo labeling. We expect higher *mAP* for the latter one, but it seems that the Darknet is overfit or our implementation of Pseudo Labeling is not completely correct. Another possible cause of the deterioration may be due to the multiple source domains used during the Domain Transfer. In fact, the pseudo labels are generated after the fine-tuning of the *DT* phase, and these may not be correct because the annotations are not supervised, but automatically created. Executing the *PL* fine-tuning, the loss value decreases during the iterations, but also the *mAP* decreases. Testing the net with weights of the Darknet after 1000 iterations of *PL* fine-tuning, the *mAP* value obtained is 23.8%. This is the highest computed during the whole project, so we think that the best way to do *PL* with our implementation is to run the tuning only for few iterations. This is a big difference between our implementation and the one made by [INOUE], which uses 10000 iterations.

Moreover, comparing our results and the [INOUE ] ones, the latter obtain a better *mAP*, and we attribute this to the multiple source domains and to a possible bad implementation. The *AP* for each class are shown in the last row of *Table 1*.

# **Conclusion**

Detection of objects in a domain that doesn’t have instance-level annotations for training improve quite significantly using the proposed methods, so this approach can be considered a good workflow that can be adapted to any domain needed. In fact, we consider these techniques a very powerful tool, because of the flexibility of its usage among every possible domain. However, if results obtained with *Domain Adaptation* are similar to the ones obtained by Inoue[], we can’t say the same thing about *Pseudo-labeling*. So, in a future work, or having more time to spend on this project, we would like to inspect with more attention the implementation of *PL* in order to find possible problems and make it work as expected. In fact, in the original paper by Inoue[], the highest accuracy in detection is obtained with DT + PL, but we haven’t been able to reproduce the same results.

## Blind review

Many authors misunderstand the concept of anonymizing for blind review. Blind review does not mean that one must remove citations to one’s own work—in fact it is often impossible to review a paper unless the previous citations are known and available.

Blind review means that you do not use the words “my” or “our” when citing previous work. That is all. (But see below for techreports)

Saying “this builds on the work of Lucy Smith [1]” does not say that you are Lucy Smith, it says that you are building on her work. If you are Smith and Jones, do not say “as we show in [7]”, say “as Smith and Jones show in [7]” and at the end of the paper, include reference 7 as you would any other cited work.

An example of a bad paper:

An analysis of the frobnicatable foo filter.

In this paper we present a performance analysis

of our previous paper [1], and show it to be inferior

to all previously known methods. Why the

previous paper was accepted without this analysis

is beyond me.

[1] Removed for blind review

An example of an excellent paper:

An analysis of the frobnicatable foo filter.

In this paper we present a performance analysis of

the paper of Smith et al. [1], and show it to be

inferior to all previously known methods. Why the

previous paper was accepted without this analysis

is beyond me.

[1] Smith, L and Jones, C. “The frobnicatable

foo filter, a fundamental contribution to human

knowledge”. Nature 381(12), 1-213.

If you are making a submission to another conference at the same time, which covers similar or overlapping material, you may need to refer to that submission in order to explain the differences, just as you would if you had previously published related work. In such cases, include the anonymized parallel submission [4] as additional material and cite it as

[1] Authors. “The frobnicatable foo filter”, Face and Gesture 2014 submission ID 324, Supplied as additional material efg324.pdf.

Finally, you may feel you need to tell the reader that more details can be found elsewhere, and refer them to a technical report. For conference submissions, the paper must stand on its own, and not require the reviewer to go to a techreport for further details. Thus, you may say in the body of the paper “further details may be found in [5]”. Then submit the techreport as additional material. Again, you may not assume the reviewers will read this material.

Sometimes your paper is about a problem which you tested using a tool which is widely known to be restricted to a single institution. For example, let’s say it’s 1969, you have solved a key problem on the Apollo lander, and you believe that the CVPR70 audience would like to hear about your solution. The work is a development of your celebrated 1968 paper entitled ”Zero-g frobnication: How being the only people in the world with access to the Apollo lander source code makes us a wow at parties”, by Zeus *et al.*

You can handle this paper like any other. Don’t write “We show how to improve our previous work [Anonymous, 1968]. This time we tested the algorithm on a lunar lander [name of lander removed for blind review]”. That would be silly, and would immediately identify the authors. Instead write the following:

We describe a system for zero-g frobnication. This system is new because it handles the following cases: A, B. Previous systems [Zeus et al. 1968] didn’t handle case B properly. Ours handles it by including a foo term in the bar integral.

...

The proposed system was integrated with the Apollo lunar lander, and went all the way to the moon, don’t you know. It displayed the following behaviours which show how well we solved cases A and B: ...

As you can see, the above text follows standard scientific convention, reads better than the first version, and does not explicitly name you as the authors. A reviewer might think it likely that the new paper was written by Zeus et al, but cannot make any decision based on that guess. He or she would have to be sure that no other authors could have been contracted to solve problem B.

## Miscellaneous

When citing a multi-author paper, you may save space by using “*et alia*”, shortened to “*et al*.” (not “*et. al.*” as “*et*” is a complete word.) However, use it only when there are three or more authors. Thus, the following is correct:

“Frobnication has been trendy lately. It was introduced

by Alpher [3], and subsequently developed by Alpher and Fotheringham-Smythe [1], and Alpher *et al.* [2].”

This is incorrect: “... subsequently developed by Alpher et al. [1] ...” because reference [1] has just two authors. If you use the \etal macro provided, then you need not worry about double periods when used at the end of a sentence as in Alpher et al.

For this citation style, keep multiple citations in numerical (not chronological) order, so prefer [1, 3, 4] to [3, 1, 4].

# Formatting your paper

All text must be in a two-column format. The total allowable width of the text area is inches (17.5 cm) wide byinches (22.54 cm) high. Columns are to be 31/4 inches (8.25 cm) wide, with a 5/16 inch (0.8 cm) space between them. The main title (on the first page) should begin 1.0 inch (2.54 cm) from the top edge of the page. The second and following pages should begin 1.0 inch (2.54 cm) from the top edge. On all pages, the bottom margin should be inches (2.86 cm) from the bottom edge of the page for 8.5 × 11-inch paper; for A4 paper, approximatelyinches (4.13 cm) from the bottom edge of the page.

## Margins and page numbering

All printed material, including text, illustrations, and charts, must be kept within a print area  inches (17.5 cm) wide by inches (22.54 cm) high. Page numbers should be in footer with page numbers, centered and .75 inches from the bottom of the page and make it start at the correct page number rather than the 4321 in the example (how to do that depends on your version of word or open office. Failure to use the correct page number, or place it properly, could result in the paper not being included in Xplore, (even if it passes PDF express (which does not check page number)

## Type-style and fonts

Wherever Times is specified, Times Roman may also be used. If neither is available on your word processor, please use the font closest in appearance to Times to which you have access.

MAIN TITLE. Center the title 1-3/8 inches (3.49 cm) from the top edge of the first page. The title should be in Times 14-point, boldface type. Capitalize the first letter of nouns, pronouns, verbs, adjectives, and adverbs; do not capitalize articles, coordinate conjunctions, or prepositions (unless the title begins with such a word). Leave two blank lines after the title.

AUTHOR NAME(s) and AFFILIATION(s) are to be centered beneath the title and printed in Times 12-point, non-boldface type. This information is to be followed by two blank lines.

The ABSTRACT and MAIN TEXT are to be in a twocolumn format.

MAIN TEXT. Type main text in 10-point Times, singlespaced. Do NOT use double-spacing. All paragraphs should be indented 1 pica (approx. 1/6 inch or 0.422 cm). Make sure your text is fully justified—that is, flush left and flush right. Please do not place any additional blank lines between paragraphs.

Figure and table captions should be 9-point Roman type as in Figures 1 and 2. Short captions should be centred. Callouts should be 9-point Helvetica, non-boldface type. Initially capitalize only the first word of section titles and first-, second-, and third-order headings.

FIRST-ORDER HEADINGS. (For example, 1. Introduction) should be Times 12-point boldface, initially capitalized, flush left, with one blank line before, and one blank line after.

SECOND-ORDER HEADINGS. Should be Times 11-point boldface, initially capitalized, flush left, with one blank line before, and one after. If you require a third-order heading (we discourage it), use 10-point Times, boldface, initially capitalized, flush left, preceded by one blank line, followed by a period and your text on the same line.

## Footnotes

Please use footnotes[[1]](#footnote-1) sparingly. Indeed, try to avoid footnotes altogether and include necessary peripheral observations in the text (within parentheses, if you prefer, as in this sentence). If you wish to use a footnote, place it at the bottom of the column on the page on which it is referenced. Use Times 8-point type, single-spaced.

## References

List and number all bibliographical references in 9-point Times, single-spaced, at the end of your paper. When referenced in the text, enclose the citation number in square brackets, for example [4]. Where appropriate, include the name(s) of editors of referenced books.

## Illustrations, graphs, and photographs

All graphics should be centered. Please ensure that any point you wish to make is resolvable in a printed copy of the paper. Resize fonts in figures to match the font in the body text, and choose line widths which render effectively in print. Many readers (and reviewers), even of an electronic copy, will choose to print your paper in order to read it.

You cannot insist that they do otherwise, and therefore must not assume that they can zoom in to see tiny details on a graphic. When placing figures in LATEX, it’s almost always best to use \includegraphics, and to specify the figure width as a multiple of the line width as in the example below

\usepackage[dvips]{graphicx} ...

\includegraphics[width=0.8\linewidth]

{myfile.eps}

# References

1. FirstName Alpher, , and J. P. N. Fotheringham-Smythe. Frobnication revisited. Journal of Foo, 13(1):234–778, 2003.
2. FirstName Alpher, , FirstName Fotheringham-Smythe, and FirstName Gamow. Can a machine frobnicate? Journal of Foo, 14(1):234–778, 2004.
3. FirstName Alpher. Frobnication. Journal of Foo, 12(1):234–778, 2002.
4. Actual Author Name. The frobnicatable foo filter, 2014. Face and Gesture (to appear ID 324).
5. Actual Author Name. Frobnication tutorial, 2014. Some URL al tr.pdf.

1. This is what a footnote looks like. It often distracts the reader from the main flow of the argument. [↑](#footnote-ref-1)