LEARNING OBJECT SEGMENTATION USING A MULTI NETWORK SEGMENT CLASSIFICATION APPROACH

S. Albertini, I. Gallo, M. Vanetti, A. Nodari
University of Insubria, Dipartimento di Informatica e Comunicazione, Varese Italy
ignazio.gallo@uninsubria.it

Keywords: object segmentation, multi-net system, GrabCut

Abstract: In this study we propose a new strategy to perform an object segmentation using a multi neural network approach. We started extending our previously presented object detection method applying a new segment based classification strategy. The result obtained is a segmentation map post processed by a phase that exploits the GrabCut algorithm to obtain a fairly precise and sharp edges of the object of interest in a full automatic way. We tested the new strategy on a clothing commercial dataset obtaining a substantial improvement on the quality of the segmentation results compared with our previous method. The segment classification approach we propose achieves the same improvement on a subset of the Pascal VOC 2011 dataset which is a recent standard segmentation dataset, obtaining a result which is inline with the state of the art.

1 INTRODUCTION

Object segmentation is an important task in computer vision whereas it is a critical part in many applications such as content based image retrieval, understanding of a scene, automatic annotations, etc. However it is still an open problem due to the heterogeneity of some classes of objects and the complexity of different backgrounds.

Usually an object of interest of an image is detected through the bounding box which surrounds it. The strength of this work consist in the detection of the object in a cognitive manner, locating the object through the use of a segmentation process. A typical segmented object produced by our system is shown in Figure 1. Usually, the images fetched from the web have low quality due to the low resolution, compression artifacts and sometimes they are revised in order to fit some particular need. In this circumstance, object segmentation is not so simple as we would expect; with our work we want to face the problem and find a solution to the object segmentation in the web images environment.

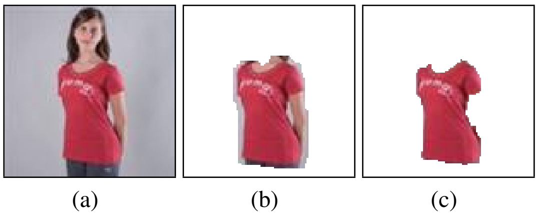

Figure 1: (a) Typical low resolution web image (100×100) of a commercial product; (b) Automatic segmentation of the shirt using our model; (c) Refinment of the segmented object with GrabCut.

The model proposed in this study, like other works which propose biologically inspired systems (Riesenhuber and Poggio, 1999; Serre et al., 2005), is partially inspired by the human visual perception system. In fact, analyzing how the visual system works, a neuron n of the visual cortex receives a bottom-up signal X from the retina (lower-level-input) and a signal M from an object-model-concept m (top-down priming signal). The neuron n is activated if both signals are strong enough. The visual perception uses many levels in the transition from the retina to the object perception. By analogy, we propose a *Multi-net system* (Sharkey, 1999) based on a tree-structure where leaf nodes represent the bottom-up signal extracted from the input image. The intermediate levels nodes represent the knowledge of the previous experience, going in the direction of the root node.

The choice of a tree-based learning architecture is further supported by the recent interest in deep learning models and the conviction that a shallow architecture can't learn very complex problems, such as visual object detection and segmentation (Bengio, 2009). In particular, a function represented compactly by a specific architecture, may require an exponentially greater number of computational elements to be represented by an architecture with a smaller depth of even a few levels. Since the number of training examples required to generalize the problem grows with the number of computational elements constituting the architecture, successfully training a model with an insufficient depth may require too much examples and then becomes very hard in practice.

In this paper we show the improved results of the object segmentation produced by our previous algorithm called MNOD (Gallo and Nodari, 2011).

In particular that model uses the concept of sliding window to segment objects of interest in a given image. In order to improve the response of the algorithm on the boundary of the object of interest, avoiding particularly fuzzy segmentations, we have adopted a new approach based on regions of pixels instead of sliding windows. This new proposed strategy is inspired by a recent work (Li et al., 2010) and in this study we have analyzed its integration in the tree-based learning architecture previously proposed.

Once determined the object segmentation mask, we improved the result setting up a post processing phase based on the GrabCut (Rother et al., 2004) algorithm. There are other works in literature that make use of the GrabCut algorithm as a refinement phase. For example, in (Hernandez et al., 2010) it is employed in video segmentation after a detection phase to separate human shapes from the environment. In order to properly fragment a human target, in (Wang et al., 2010) the GrabCut is initialized with multiple rectangle areas, obtained from a mean shift detection, that enclose different part of the bodies. In contexts where we need to fully exploit the information extracted from a particular object, especially when the images have low resolution, a near perfect segmentation is crucial if we want to extract the visual characteristics of an object. The model we propose exploits the output segmentation mask as initialization for the GrabCut algorithm in order to enhance the quality of the final segmentation.

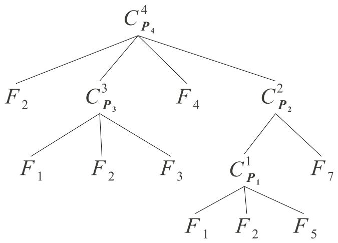

Figure 2: Generic structure of the proposed MNOS model and the existing MNOD. The nodes $C^n_{\mathbf{P}}$ represent the supervised neural models which receive their input directly from leaf nodes F_i and/or other internal nodes $C^m_{\mathbf{P}}$. In the MNOS model a node $C^n_{\mathbf{P}}$ may use either the sliding window or segments as contextual information to be classified.

2 THE EXISTING MODEL

The Multi-Net for Object Detection (MNOD) (Gallo and Nodari, 2011) is a Multi-Net System (Sharkey, 1999) which consists of a tree of neural networks, as shown in Figure 2. Each node n, properly configured with its own parameters $\mathbf{P}$, acts like an independent module $C^n_{\mathbf{P}}$ and it can be replaced by any node of the same type in the tree.

Leaf nodes F_i apply operators and filters on the input images in order to generate feature-images that sharpen the input data peculiarities. The internal nodes aggregates and takes in input the output images produced by their child nodes. Each internal node reads the input images using a sliding window and generates the pattern vectors for the neural network simply relying on the intensity pixel values that fall in the window and gives a map image in output where each pixel has got an intensity value proportional to the probability it belongs to the object. The particular distinction of this model lies in the connection between nodes, which means that the output of a node becomes the input of a parent node. The links between the nodes in the tree structure define the flow of image segmentation process that cross the whole structure from the leaves to the root.

That structure allows to diversify a node $C^n_{\mathbf{P}}$ just adjusting the parameters $\mathbf{P}$, but it was shown that it is sufficient to tune the input image scale and sliding window dimension in order to obtain configurations leading to good results (Gallo and Nodari, 2011). Then, we can refer to $\mathbf{P}_n = \{I_S, W_S\}$ as the configuration for the node n, given I_S and W_S respectively image size and sliding window size. Using different combination of these two parameters, we are able to construct models specialized to recognize specific objects of different class.

The segmentation map produced by the MNOD is the root node output image. This map can be considered as a soft segmentation map and it is used to generate the detection bounding box. The main disadvantage of this kind of soft map relies in a very fuzzy result over the boundaries of the object of interest. The main goal of the present work consists in improving the MNOD algorithm in order to obtain output maps with crisper borders to delimit precisely the object of interests.

3 THE PROPOSED METHOD

In this paper we propose a variant of MNOD, called Multi-Net for Object Segmentation (MNOS). The idea is to change the algorithm's image aggregation method from a sliding window to a segment-based approach.

The new solution generates a partition image composed of segments which are used as primitive elements for the prediction. A segment is defined as a set of pixels that share properties of homogeneity based on the point position and color channels values. Formally, let I be an image, the application of the algorithm to extract segments should produces a set $S = \{S_1, \cdots, S_N\}$ such that $\forall i \neq j$ with $1 \leq i, j \leq N$, then $S_i \cap S_j = \varnothing$ and $\bigcup_{i=1}^{N} S_i = I$.

For each segment that composes the image, the neural network estimates the probability a segment represents a component that belongs to the object of interest. That task is performed by training the neural network with the images and their respective ground truth masks. Given S_i, a segment calculated from the image I and M_I as its ground truth mask, the probability is calculated by $|M_I \cap S_i|/|S_i|$, where $|\cdot|$ is the number of pixels with nonzero value. The input patterns for the neural network are generated extracting features from the region of the image represented by the segments.

The algorithm used to calculate the partition image is a K-means clustering (Hartigan and Wang, 1979) where each image point is represented by a pattern whose dimensionality depends on the MNOS node in question: patterns are placed in a space with a number of dimensions that depends from the number of node's children elements. Specifically, three values are about the three color channels, two values about the (x, y) point position on the image canvas, the rest are the intensity values resulting from each child node segmentation map for that pixel. So, in order to improve the overall segmentation result, we need also to enhance the segmentation quality. By employing a K-means, like the previously described, we exploit the segmentation results from the child nodes obtaining segments that are increasingly similar to the object of interest as you move up in the MNOS structure towards the root node.

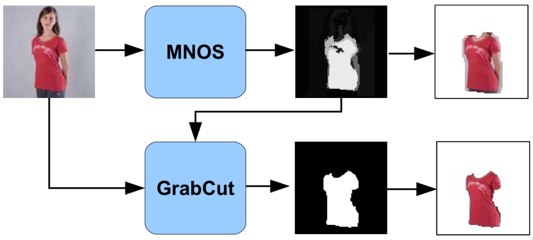

Figure 3: A simple flowchart representing the refinement process which tries to improve the MNOS's output by applying the GrabCut algorithm as a post processing phase. The GrabCut segmentation map is an optimization of the MNOS mask.

The features extracted from each segment are of two types: based on the input image or based on the segment geometry. Features that fall in the first category are the gray level quantized histogram calculated on the pixel inside the segment, an histogram generated from the pixels that lay in a region around the segment and the seven Hu moments (Hu, 1962). In the second category we have features based on geometrical properties of the segment, like area, perimeter and bounding box center and location.

The algorithm 1 shows how the patterns for the segment-based nodes are generated starting from the set of input images and the ground truth mask. We remark that the input images could be either feature-images or output maps generated from child nodes. The algorithm 2 describes how a sub-tree starting from a node is recursively trained. Assuming $C^n_{\mathbf{P}_n}$ as the root node, that algorithm corresponds to the training of a MNOS model. Algorithm 3 shows the segmentation task carried out by a generic node from a MNOS model.

A MNOS node can be used together with standard sliding window nodes because they expose the same functionalities: they take a set of images as input and returns the predicted segmentation map that can become the input for a node in the next layer. The output image of a node is generated from the neural network prediction, assigning an activation value to each pixel of the segments that is the one predicted by the network.

We used a model based on the idea that the MNOS performs an implicit aggregation process while the information flows though the structure, from the leaf nodes to the root node, in a bottom-up process.

Using the sliding window nodes on the first levels, followed by nodes that aggregate their results using segments, makes the proposed approach biologically plausible. In fact the first levels perform a prediction at a pixel level, while the next layers aggregate the image points in regions and then fulfill their prediction considering not raw intensity values but features extracted from the segments, assigning a probability value to a whole region and finally producing crisp and homogeneous boundaries along estimated object masks.

Algorithm 1 Creation of the neural network patterns for MNOS nodes.

```
Require: Set of input images $I = \{I_1, \dots, I_N\}$,
    ground truth output image $O$, set of segments from
    partition image $S = \{S_1, \dots, S_N\}$;
    $F_I = \{F_{I1}, \dots, F_{IN}\}$ set of features on images;
    $F_S = \{F_{S1}, \dots, F_{SM}\}$ set of features on segment geometry.
Ensure: A set of patters $P$, where $|P.in| = |S|$ are the
    input patterns, $|P.out| = |S|$ are the truth output patterns.

  $P \leftarrow \varnothing$
  for all $S_i \in S$ do
    $p \leftarrow \varnothing$
    for all $I_j \in I$ do
      for all $F_t \in F_I$ do
        $f \leftarrow F_t(I_j \cap S_i)$
        Concatenate $f$ to $p$
      end for
    end for
    for all $F_r \in F_S$ do
      $f \leftarrow F_r(S_i)$
      Concatenate $f$ to $p$
    end for
    $P.in \leftarrow P.in \cup p$
    if (training the node) then
      $h \leftarrow S_i \cap O$
      $o \leftarrow |h| / |S_i|$
      $P.out \leftarrow P.out \cup o$
    end if
  end for
  return $P$
```

Algorithm 2 Training of a MNOS node $C^n_{\mathbf{P}_n}$.

```
Require: $D = \{< I_1^{in}, I_1^{out} >, \dots, < I_N^{in}, I_N^{out} >\}$ the
    set of images with their gound truth segmentation mask.

  for $i = 1$ to $N$ do
    for all node $C \in C^n.children$ do
      $C.Train(D)$
    end for
    inList $\leftarrow \varnothing$
    outList $\leftarrow \varnothing$
    maskList $\leftarrow \varnothing$
    for all $< I_i^{in}, I_i^{out} > \in D$ do
      childSeg $\leftarrow \varnothing$
      for all node $C \in C^n.children$ do
        $s \leftarrow$ C prediction on $I_i^{in}$
        Resized $s$ to $I_S^n \in \mathbf{P}_n$
        childSeg $\leftarrow$ childSeg $\cup s$
      end for
      $o \leftarrow$ Resize $I_i^{out}$ to $I_S^n \in \mathbf{P}_n$
      $m \leftarrow$ generate partition mask with K-means
      from $I_1^{in}$
      inList $\leftarrow$ inList $\cup$ childSeg
      outList $\leftarrow$ outList $\cup o$
      maskList $\leftarrow$ maskList $\cup m$
    end for
  end for
  Train the MLP network with <inList, outList,
  maskList> generating the patterns as described by
  algorithm 1
```

3.1 Post processing with GrabCut

In order to improve the quality of the mask produced by the proposed method, we analyzed a solution which takes advantage from the great detection ability of the MNOS. In this section we explain the integration of the the GrabCut algorithm described in (Rother et al., 2004) as a post processing phase for the MNOS result. The GrabCut is the state of the art in the interactive segmentation algorithms: a supervisor must specify a bounding box (or a lasso) on the image which encloses the desired object to segment. Then the algorithm calculates the parameters for background and foreground initialization, starting from a Gaussian Mixture Models parameterized with the color distribution of the two mutual exclusive regions defined. In the final step an iterative graph cut of the final segmentation is performed.

In the early experiments we used the detection bounding box calculated from the MNOS's output to automatically initialize the GrabCut. The main problem using the bounding box is its inefficiency in specifying the samples used by the algorithm to generate the initialization parameters. The area of a bounding box is always greater than the area of an object of interest. There are borderline cases in which long and narrow objects lead to the generation of very large bounding box.

We can remark that the GrabCut interactive phase can be viewed as a labelling process. We want to assign to each pixel a priori information whether it more probably belongs to background or foreground, considering four labels:

Algorithm 3 MNOS prediction from a node $C^n_{\mathbf{P}_n}$.

```
Require: Image $I$.
Ensure: Segmentation image map $S_I$.
  childSeg $\leftarrow \varnothing$
  for all node $C \in C^n.children$ do
    $s \leftarrow$ C prediction on $I$
    Resized $s$ to $I^n_S \in \mathbf{P}_n$
    childSeg $\leftarrow$ childSeg $\cup s$
  end for
  $m \leftarrow$ generate partition mask with K-means from $I$
  Use the neural model to predict the activation level for each segment in $m$
  Compose the output image $S_I$ assigning to the pixel of each segment the respective activation value
  return $S_I$
```

- Definitely background
- Probably background
- Probably foreground
- Definitely background

In order to overcome the limitations resulting from the use of the bounding box in the GrabCut initialization process, we propose a method of initialization based on the MNOS output, whose area of initialization is more related to the boundary of the object of interest. The whole process, including the post processing phase with GrabCut, is graphically represented by the flowchart shown in Figure 3. We can observe how the GrabCut, to segment the object of interest contained in the input image is initialized with the segmentation map produced by MNOS. That solution produces good experimental results in controlled domais where the MNOS model yield very good output maps as proven in section 4.1.

In order to apply the algorithm to the MNOS segmentation mask, we define three regions on the images and assign the labels to fulfill the GrabCut interactive phase. We binarize the MNOS mask and erode the mask first, then we dilate it, so cleaning it from the eventual noise generated by the MNOS. The resulting activated pixels are labeled as "probably foreground". Next, we calculate a region surrounding the "probably foreground" region and assign to these pixels the label "probably background". Finally, the remaining region is labeled as "definitely background", so it will never be included in the GrabCut object segmentation mask.

The GrabCut version we employ doesn't make use of a border matting phase described in (Rother et al., 2004), so we generate hard masks.

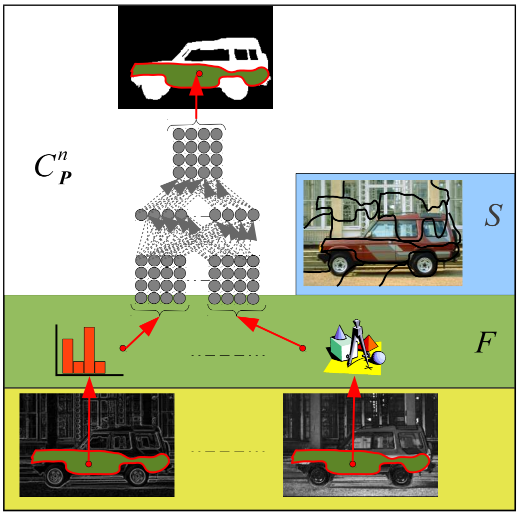

Figure 4: Generic structure of a MNOS node $C^n_{\mathbf{P}}$. It generate the set of segments $S = \{S_1, \dots, S_N\}$ on the original scaled image. Then, for each input image from child nodes and for each segment it calculate the features $F = F_I \cup F_S$ and compose the patterns for the neural network. Finally, the neural model prediction values are used to shape the output image of the node $C^n_{\mathbf{P}}$.

4 EXPERIMENTS

To test the performance of the algorithm proposed in this study and to analyze the results in a real application, we created a dataset from the fashion domain called Drezzy Dataset. It consists of 2068 images of 200×200 and 100×100 pixels in the VOC2011 format (Everingham et al., 2011) whose cardinality is described in Table 1. In order to make our results available to performance comparisons, we have uploaded it at this URL[1].

Table 1: Description of the Drezzy dataset cardinality for each class.

Class Name	# Images
Bags	285
Shoes	400
Hats	158
Ties	203
Man clothing	150
Man underwear	278
Woman clothing	355
Woman underwear	239

For each experiment we used the segmentation accuracy showed in (1) proposed by the contest "The

[1] http://www.dicom.uninsubria.it/arteLab

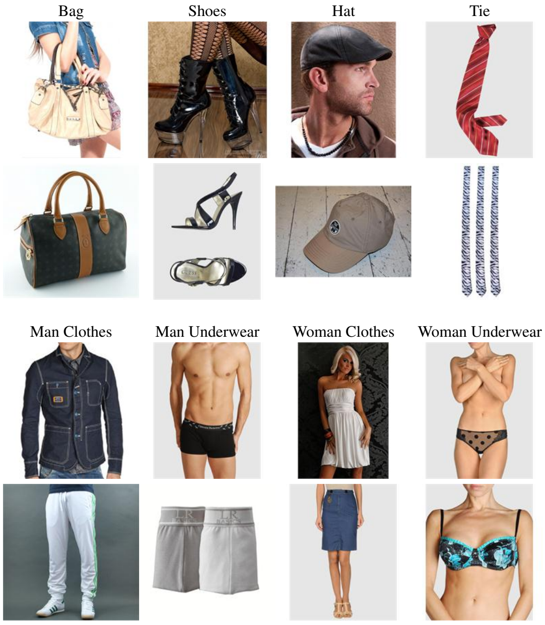

Figure 5: Some examples from the Drezzy dataset grouped by class.

PASCAL Visual Object Classes Challenge” (Everingham et al., 2011) and usually called *Jaccard index* (Jaccard, 1912).

$$\text{Acc} = \frac{\text{TP}}{\text{TP} + \text{FP} + \text{FN}} \quad (1)$$

The values under consideration are calculated pixel-by-pixel: TP are the True Positives, FP the False Positives and FN the False Negatives. We consider the problem of segmentation as a classification problem formalized as a function that takes a pixel and returns 1 if the pixel belongs to the foreground or 0 if it belongs to the background.

Initially we set up the MNOS with all nodes based on segments, but the results of such configuration did not lead to any improvement compared to the MNOD model. We created a hybrid model, so in the following experiment we tested the introduction of the nodes with sliding windows in the MNOS model. In particular, we introduced the nodes with the sliding window in the first levels of MNOS. Table 2 shows how the use of the above mentioned model brings a substantial improvement comparing the object of interest segmentation accuracy.

We suppose that the information carried by the nodes in the first levels that benefits from the sliding window, gives an overview of the image to the next levels by passing a revised image information while reducing the problem of segmentation complexity. We can see it as an implicit aggregation process: low levels, which use the standard MNOD nodes based on sliding window, read the image at a pixel level and perform their predictions based on raw intensity values of image pixels. The higher levels, with nodes based on segments, perform their predictions from the output images of the sliding window layers: they calculate segments and generate features from that region of pixel. We remark that segments used by a node $C^n_{\mathbf{P}}$ are generated with a clustering in a space that is also based on the predictions from its child nodes. Then, the prediction result is referred to a set of points rather than a single point. Anyway, each point was previously evaluated by sliding window nodes, so they are just aggregating them and estimating a general and homogeneous probability value for that set of pixels. So, one of the main advantage about using the nodes based on the segments in the last levels of the MNOS tree is to produce masks with sharp edges around the activated areas of the map, not blurred like what happened with the original MNOD algorithm, as shown in Figure 6. This type of operation reminds the same mechanism of human visual cortex, where different level types process information and pass the results to the next levels for further enhancements.

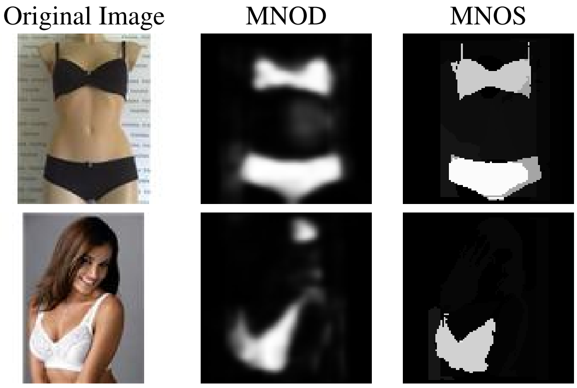

Figure 6: Some examples from the Drezzy dataset of the MNOS’s segmentation results using only Sliding Windows (MNOD) and using Sliding Windows and Segments (MNOS).

4.1 Results applying GrabCut

We described in section 3.1 how we conceived the use of the GrabCut algorithm in order to further increase the segmentation mask quality. The most natural way to integrate the MNOS with the GrabCut is to calculate the detection bounding box, in order to fulfill the interactive initialization needed by the GrabCut. Anyway, that solution is intuitively inefficient, because the bounding box is a coarse solu-

Table 2: Comparison between accuracy values (%) of the MNOD with sliding window (*SL*) and the proposed MNOS method, which uses the segments (*S*) in the last layers and sliding windows in the first layers.

	Acc Train		Acc Test	
Dataset	**SL**	**S**	**SL**	**S**
Bags	90,54	92,20	79,68	79,00
Shoes	92,40	91,14	83,61	88,39
Hats	75,25	87,32	61,32	62,55
Ties	98,88	96,81	77,57	81,52
Man clothing	83,11	84,61	69,00	73,40
Man underwear	64,40	77,20	54,77	65,25
W. clothing	59,07	65,33	59,11	62,64
W. underwear	57,90	69,47	54,94	66,68

tion as it should discard much information from the MNOS result. As previously discussed, a clever idea is to initialize the GrabCut with a region mask that would preserve the segmentation information given by the MNOS. Figure 8 shows the differences using the two solutions on examples taken from each dataset. The first column shows what happens when we initialize the GrabCut with a bounding box calculated from the MNOS mask, while the second one shows the region mask approach: the deep blue pixels (in the proposed method strategy column) are labeled as “probably foreground”, while the cyan pixels are labeled as “probably foreground”. All the others are “definitely background”. These approaches were both tested on the MNOS segmentation maps and the results are showed in table 3. We had better accuracy results, as expected, using the region mask approach.

There are basically two reasons why the GrabCut may worsen the segmentation performed by the MNOS:

- Flawed MNOD mask, because it could lead to a wrong GrabCut initialization
- Inability of the GrabCut, when the contrast and the color distribution is not well distributed, for example with camouflage, between the background and foreground samples

Comparing the MNOS segmentation accuracies without and with the GrabCut post processing phase, employing the region mask initalization showed in table 4, we see a general improvement but for the woman underwear dataset. In general, the GrabCut works well if the initalization information is clean, so it makes sense to work on MNOS improvements because the GrabCut don’t flatten the final accuracy results independently from the quality of the MNOS segmentation map. Otherwise, it could lead to unpleasant results that can also worsen the MNOS segmentation mask.

Table 3: Results comparison between the GrabCut initialized with bounding box (BB) and region mask (RM) applyed to the MNOS segmentation maps.

Dataset	**BB**	**RM**	**Diff**
Bags	89,33	89,29	-0,04
Shoes	93,23	93,49	+0,26
Hats	80,45	82,19	+1,74
Ties	92,43	92,39	-0,04
Man clothing	78,51	81,50	+2,99
Man underwear	69,25	80,10	+10,85
Woman clothing	65,54	68,08	+2,54
Woman underwear	54,76	61,22	+6,46

Table 4: Results comparison between the MNOS segmentation accuracy and the accuracies obtained after the application of the GrubCut as a post processing phase. The last column *Diff* resume the performance gain between the two methods.

Dataset	**MNOS**	**GrabCut**	**Diff**
Bags	79,00	89,29	+10,29
Shoes	88,39	93,49	+5,10
Hats	62,55	82,19	+19,64
Ties	81,52	92,39	+10,87
Man cloathing	73,40	81,50	+8,10
Man underwear	65,25	80,10	+14,85
Woman clothing	62,64	68,08	+5,44
Woman underwear	66,68	61,22	-5,46

4.2 Experiment with a standard dataset

In order to evaluate the proposed method and have the opportunity to compare it with other objects segmentation methods, we made a simple experiment with some classes of the standard datasets VOC2011 (Everingham et al., 2011). This dataset consists of 20 classes of objects and 5.034 segmentations divided into train and validation sets. In this experiment we chose to work only on a subset of eight classes of objects for simplicity and in order to use a simplified configuration compatible with the chosen classes.

The main goal of this experiment is to demonstrate the MNOS gives better accuracy results compared with the existing MNOD, using a setting of simple configurations. In other words, in this experiment our objective isn’t to push the specific configurations to achieve the best results.

Table 5: Parameters range fixed in order to compare the two models MNOD and MNOS when trained with the VOC2011 dataset. For each layer L the maxmum number of node was fixed to N. For each node one of the fixed W_s, I_s sizes and eventually a set of leaf nodes were used.

L	N	W_s	I_s	*Leaves*
1	4	3x3	50, 90	brightness,rgb
2	4	3x3,5x5	10, 50, 90	brightness,rgb
3	4	3x3,7x7	10, 50, 90	brightness,rgb
4	1	5x5,7x7	10, 50, 90	brightness,rgb

For the configuration of the two models we fixed the parameters and type of leaves for each level and the number of levels, as summarized in Table 5. We set the number of levels to 4 for both MNOD and MNOS models. For the MNOS model we have configured the first two layers with the sliding window and the last two with segments. For the first three layers we chose a maximum of 4 nodes looking for the best configuration with parameters W_s, I_s and *Leaves* selected from the collections described in Table 5. For the last layer we only need to configure the best root node, using the constraints showed in table 5

Table 6 summarizes the segmentation accuracies when the new MNOS model is compared with the existing MNOD. The first columns list the accuracy results obtained in eight classes with MNOS and MNOD model. We can notice an overall increase comparing the results without the GrabCut post processing. In fact, the GrabCut algorithm don't always improve the MNOS results. In the last column of table 6 we highlight the only three classes where the post process actually brings an improvement in accuracy results. It is possible to conclude the reason why the GrabCut cannot give a good contribute lies in the fact that the MNOS masks lack of accuracy, so the region mask we use to initialize the GrabCut is inaccurate and then the post processing could lead to worsen the MNOS mask accuracy, amplifying errors.

On the other hand, when the MNOS mask is good, the GrabCut actually leads to an improvement in the final segmentation. Let's look at the image in figure 7(a), taken from the VOC2011 dataset for the class "train". We calculate the MNOS mask, which produces the segmentation in figure 7(b). It is a fairly good result, because it is a simple image. It almost segmented the object, except for some details. So, we can generate a good initialization map for the post processing, and the GrabCut is able to perfect the result, as we can see in 7(c). Obviously, that's an optimal situation.

Table 6: Results comparison between the existing MNOS segmentation accuracy (%) and the new MNOS algorithm on the VOC2011 dataset. The column *Diff* resumes the performance gain between the two methods. The last column *GC* shows the post processing accuracy results when applied to the MNOS ouput maps.

Class	**MNOD**	**MNOS**	**Diff**	**GC**
boat	23,51	28,87	+5,36	26,43
dog	23,68	29,23	+5,55	24,20
horse	29,39	35,14	+5,75	32,41
motorbike	45,37	47,13	+1,76	**50,02**
pottedplant	12,45	14,73	+2,28	**21,03**
sheep	28,26	30,22	+1,96	**31,75**
train	38,73	46,80	+8,07	41,37
tvmonitor	16,62	19,52	+2,9	15,86

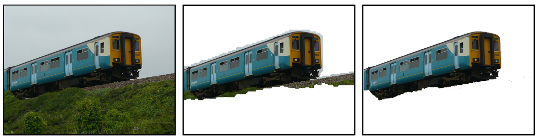
(a) (b) (c)

Figure 7: (a) Typical image of the VOC2011 dataset with an object that belongs to the "train" class; (b) Automatic segmentation of the train using our MNOS model; (c) Refinment of the segmented object with GrabCut.

5 CONCLUSIONS

In this paper we described an object segmentation algorithm based on a multi network system and inspired from a previously presented object detection algorithm, the MNOD. It is composed by a set of neural networks combined together to provide a single output result. The model results highlight the benefits of our solution. The proposed algorithm can be configured for different classes of objects and its nodes may be of different types using sliding windows or segments to read their input. We presented a model that use the sliding window in the first layers of the tree, and segments in the subsequent layers. Then, we studied a post processing phase using the GrabCut algorithm. We fulfilled its interactive initialization by exploiting the MNOS output segmentation map.

We tested the proposed model on different datasets composed by images representing commercial products from the web. The MNOS algorithm was pushed in order to achieve better accuracy results than the MNOD model. We also obtained good results on some classes of the VOC2011 dataset. Moreover, the results show that our algorithm is robust to the change of perspective for the same object and at

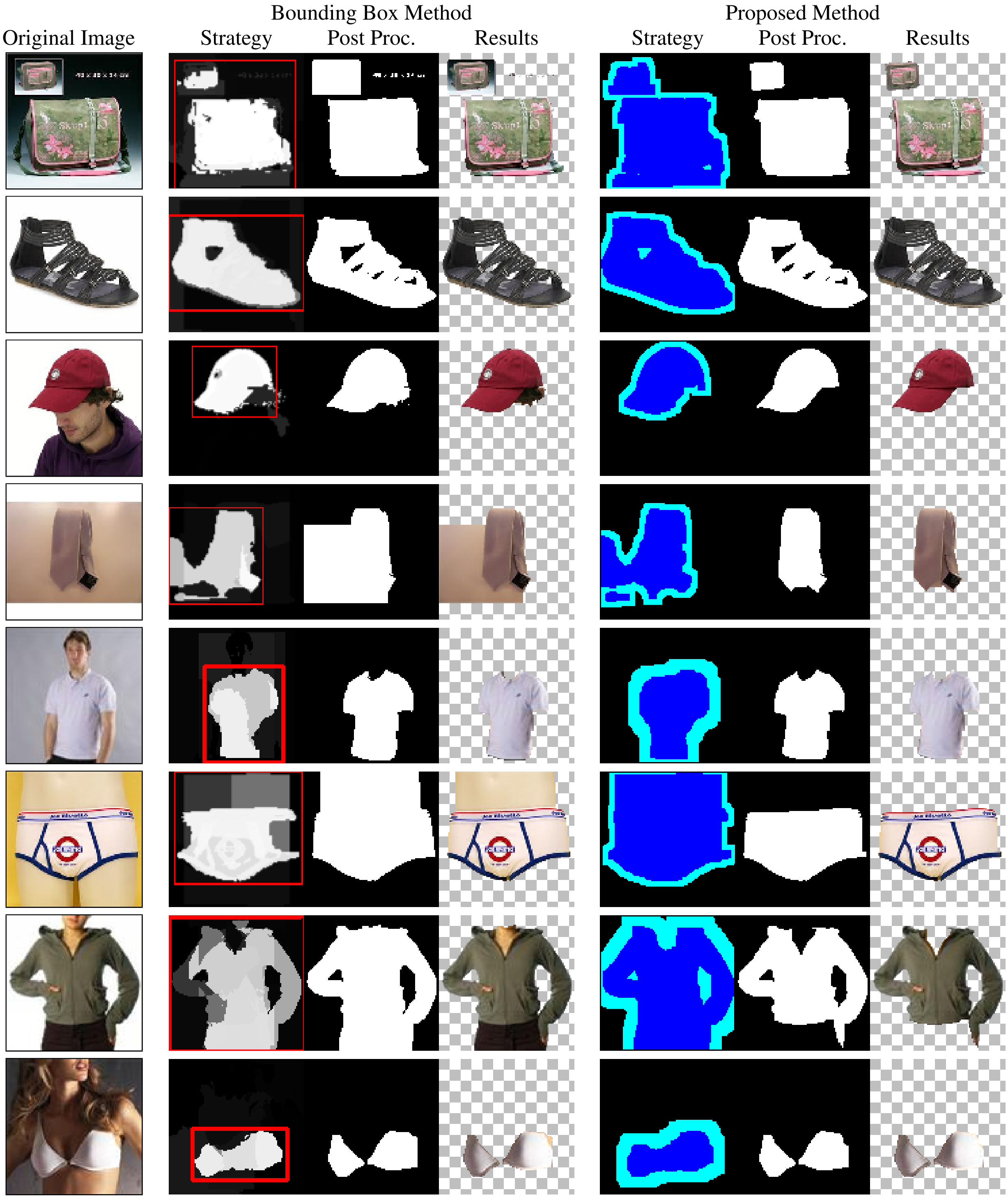

Figure 8: An example for each class of the Drezzy Dataset using the GrabCut as a post processing phase with a MNOS configured by sliding window and segment layers. The original image is reported in the first column. For each GrabCut configuration method (bounding Box and Proposed Method) we divided it in three columns: Strategy which summarize the configuration input; Post Processing which reports the Grab cut result on the MNOS segmentation mask and the result calculated as the AND between the original image and the segmentation mask after the post processing phase.

the same time, it is robust for objects of the same type but different shapes in different poses or even articulated and slightly occluded.

The GrabCut post processing phase led to very good results when the segmentation map is accurate and clean. Anyway, with very difficult images, like the ones in the VOC2011 dataset, the MNOS algorithm often produces segmentation masks that aren't accurate enough to provide a good initialization for the GrabCut, so it often worsens the MNOS result.

The most important extension we plan to realize is to make our model works with multiple classes instead as a single class segmentation algorithm. Moreover, it is possible to use different configuration regarding the set of features and the segment extraction technique.

REFERENCES

Bengio, Y. (2009). Learning deep architectures for AI. *Foundations and Trends in Machine Learning*, 2(1):1–127.

Gallo, I. and Nodari, A. (2011). Learning object detection using multiple neural netwoks. In *VISAP 2011*. INSTICC Press.

Hartigan, J. and Wang, M. (1979). A k-means clustering algorithm. *Applied Statistics*, 28:100–108.

Hernandez, A., Reyes, M., Escalera, S., and Radeva, P. (2010). Spatio-temporal grabcut human segmentation for face and pose recovery. pages 33–40.

Hu, M.-K. (1962). Visual pattern recognition by moment invariants. *Information Theory, IRE Transactions on*, 8(2):179–187.

Jaccard, P. (1912). The distribution of the flora in the alpine zone. *New Phytologist*, 11(2):37–50.

Li, F., Carreira, J., and Sminchisescu, C. (2010). Object recognition as ranking holistic figure-ground hypotheses. In *CVPR*, pages 1712–1719. IEEE.

Riesenhuber, M. and Poggio, T. (1999). Hierarchical models of object recognition in cortex. *Nature Neuroscience*, 2(11):1019–1025.

Rother, C., Kolmogorov, V., and Blake, A. (2004). Grabcut: Interactive foreground extraction using iterated graph cuts. volume 23, pages 309–314.

Serre, T., Wolf, L., and Poggio, T. (2005). Object recognition with features inspired by visual cortex. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, CVPR '05, pages 994–1000, Washington, DC, USA. IEEE Computer Society.

Sharkey, A. J. (1999). *Combining Artificial Neural Nets: Ensemble and Modular Multi-Net Systems*, chapter Multi-Net Systems. Springer.

Wang, F., Yu, S., and Yang, J. (2010). Robust and efficient fragments-based tracking using mean shift. *AEU - International Journal of Electronics and Communications*, 64(7):614 – 623.