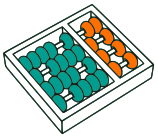


MO433 — Intro. to Unsupervised Machine Learning
INSTITUTO DE COMPUTAÇÃO — UNICAMP

Final Project
Deep Unsupervised Saliency Detection

João Victor da Silva Guerra, Leonardo Alves de Melo, and Marcos Felipe de Menezes Mota *

Abstract

Saliency detection is the problem of identifying visually interesting objects in images that are consistent with human perception. This question is of interest in computer vision applications such as context-aware image editing, image caption generation, and scene segmentation. The machine learning models that achieve the best performance in the saliency detection task are supervised deep learning models. Since these models are supervised there is an extra step of labeling the training sets. Labeling of datasets for the saliency detection problem is pixel-based thus being a labor-intensive process that hinders the possibility of big datasets for training. Such a scenario would greatly benefit from unsupervised methods. Here, we explore unsupervised techniques for saliency detection using deep unsupervised learning and reproduce the paper from [1], which proposes an unsupervised framework that obtained state-of-the-art saliency detection performance. We employed the PyTorch framework to reproduce the proposed architecture with the MSRA-B dataset. The main experiments of the paper are also reproduced and the results obtained discussed.

1 Introduction

Saliency detection targets visually interesting regions in images that can attract human visual attention, which generates a namely saliency map that identifies a pixel's unique features. These saliency maps aim at simplifying and/or changing the image into a more meaningful representation to analyze [1], [2]. In computer vision, salient object detection is commonly a process consist of two stages: (1) detect the most salient object; (2) segment the accurate object region [3]. In recent years, with the powerful learning capabilities of deep neural networks, convolutional models have been employed to address the task of salient object detection.

A saliency map is used to filter the part of interest for a certain data. In the case of computer vision, it can simplify the content prediction, since there is no longer an unused background. It is applicable in various contexts, such as in the medical field, to process Magnetic Resonance Images (MRI) [4], in robotics, to guide robots actions [5], and even in surveillance videos [6], to detect unusual sounds and behaviors in a crowded area [7], [8]. The creation of models that generate reliable saliency maps can help improve all of the areas mentioned areas and more [7], [9], [10].

In this project, we will explore a deep unsupervised state-of-the-art model for saliency detection described in [1], which we attempt to reproduce their experiments and evaluate using quantitative and qualitative comparisons.

2 Related Work

Literature in saliency detection models is vast and it has many different formulations. The first approach of saliency detection is based on cognitive theories and started in 1980 [11]. After this initial attempt, a computational formulation based on heuristics dominated and lastly the machine learning approach [11]. We organize the literature into three categories that ease the understanding of the implemented framework: unsupervised detection, deep supervised detection, and deep unsupervised detection.

2.1 Unsupervised Saliency Detection

Traditional unsupervised methods for saliency detection use heuristic features in the image, called visual priors, for region detection. The main visual prior used in unsupervised methods are contrast-based because the brain

* 117410, 156188 and 211893. j117410@dac.unicamp.br, leonardo.alves.melo.1995@gmail.com, and marcos.mota@ic.unicamp.br

is very sensitive to high contrast objects, but other priors such as location, orientation, and texture are used along contrast [11].

The contrast prior can be divided into local and global contrast. Local contrast uses a fixed region (eg. 9×9 pixel box) and measures contrast against surrounding regions. Global contrast prior compares one image region against all other image regions, high contrast regions look brighter and defines a saliency map. One example of method that used global contrast prior is the HC method [11].

Other examples of visual priors are based on location. These models try to identifies where is the center or the background of the image. Background prior uses border pixels as a seed to detect the background region and define the salient object based on that. Center-prior is a similar approach that tries to estimate the center of the object close to the center of the image [11]. An example of saliency detection methods using background prior are RBD and RC, the RC uses global contrast and background prior together. The FT method is an example of center-prior, but it also uses global contrast to improve its accuracy.

2.2 Deep Supervised Saliency Detection

Deep learning saliency maps were introduced in [2], which applied visualization techniques to compute images, saliency maps being one of them. Recently, deep supervised neural networks have been employed for saliency detection [12]–[16]. Building upon the capabilities of convolutional neural networks (CNN), these deep supervised saliency detection methods achieved state-of-the-art performance with greater performance compared to unsupervised methods [1]. However, such supervised methods depend on large-scale manual supervision as pixel-level human annotations, which is highly time-consuming, labor-intensive, and could hinder the generalization ability of the models [1], [3].

2.3 Deep Unsupervised Saliency Detection with Noisy Labels

Currently, deep learning approaches are the main choices in saliency detection; however, few studies have addressed the field with learning from unreliable and noisy labels [1], [17]. In the literature, "Supervision by Fusion" (SBF; [17]) has been the first successful unsupervised learning framework to train a salient object detector, which employs a two-stream framework to create supervisory signals through an intra-image and inter-image fusion processes. Afterward, Deep Noise Model-based Saliency Detector (DNMSD; [1]) presented an end-to-end deep learning framework that learns saliency maps from multiple noisy unsupervised saliency methods. Instead of removing noise in saliency labeling from unsupervised saliency methods with fusion strategies as in [17], authors explicitly learn an adaptive noise from noisy saliency maps. The latter method outperforms traditional unsupervised methods and also achieves similar performance with current state-of-the-art deep supervised saliency detection methods [1]. With the success and performance enhancement of such deep unspervised methods, the field can overcome the requirement for time-consuming human annotated data and start relying only on unsupervised annotation for saliency detection.

3 Framework

The end-to-end framework for deep unsupervised saliency detection proposed in [1] is composed of two modules: latent saliency prediction module (SPM) and noise modeling module (NMM). The main idea of this approach is to explicitly model the noise in saliency maps in an unsupervised fashion. Such an idea tries to reduce the blurred edges in saliency maps of previous methods. Thus, the framework learns unsupervised saliency from existing unsupervised saliency detection methods, models the noise commonly found in such methods, and optimize a loss function composed by both SPM and NMM losses together.

In Figure 1, we illustrate our deep noise model based saliency detector framework. First, in the SPM, we have a set of images, $\boldsymbol{X} = \{\boldsymbol{x}_i, i = 1...N\}$, and a CNN, which is parametrized by a parameter Θ, that maps the image, $\boldsymbol{x}_i$, to a latent saliency map, $\bar{\boldsymbol{y}}_i = f(x_i;\Theta)$. With this predicted saliency map, we move to NMM that adds a noise, $\boldsymbol{n}_i^j$, sampled from a i.d.d. zero-mean gaussian distribution, $q(\Sigma)$, which is parametrized by a parameter set, $\Sigma = \{\sigma^i_{mn}\}$, where i is the index of the training image and (m,n) are pixel coordinates. With this parametrization, it is simple to sample noise, n_i^j, for any index i and labeller j. Taken both modules, we obtain the saliency map with noise, $\hat{\boldsymbol{y}}_i^j = \bar{\boldsymbol{y}}_i + \boldsymbol{n}_i^j$. Equipped with the unified loss function, composed by SPM and NMM losses, and using

existing unsupervised methods as pseudo ground truth, $\boldsymbol{Y} = \{\boldsymbol{y}_i^j, i = 1...N, j = 1...M\}$, both modules are optimized together, producing better saliency detection maps.

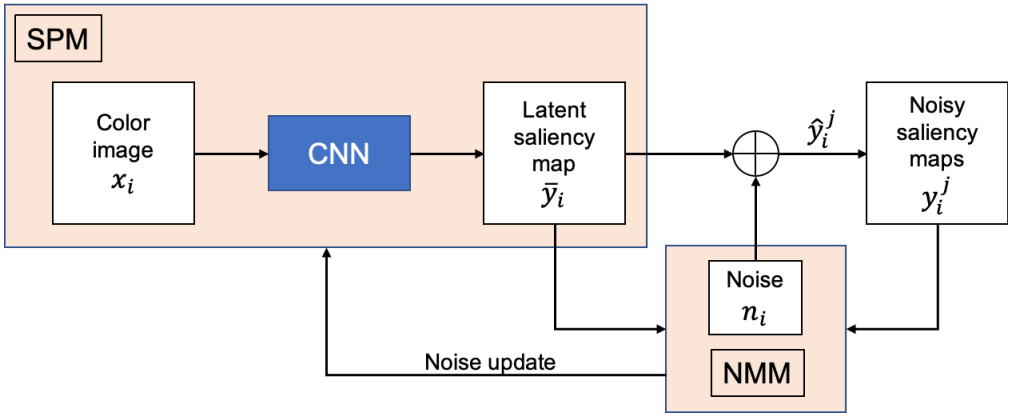


Figure 1. Schematic representation of our deep unsupervised saliency detection framework, which consists of a latent saliency prediction module and a noise modeling module.

3.1 Loss Function

As we mentioned before, one of the central aspects of the work proposed in [1] is the loss function of their framework, which is one of the main contributions of the work and essential to the end-to-end training feature. Given the variance Σ of the zero-mean Gaussian distribution, where the noise $\boldsymbol{n}_i^j \sim \boldsymbol{q}_i(\Sigma)$ is sampled from, and the parameter Θ of the CNN, that predicts the saliency map, $\bar{\boldsymbol{y}}_i = f(\boldsymbol{x}_i, \Theta)$, the loss function for our framework is:

$$\mathscr{L}(\Theta, \Sigma) = \mathscr{L}_{\text{pred}}(\Theta, \Sigma) + \lambda \cdot \mathscr{L}_{\text{noise}}(\Theta, \Sigma), \tag{1}$$

where λ is the regularization term, which balances both losses terms. With an increase in the noise, modeling variance makes prediction loss large and decrease noise loss. Meanwhile, keeping the variance lower will decrease the prediction loss but increase the noise loss [1]. Therefore, optimizing the loss function is to achieve a balance between the NMM, which increases generalization, and the SPM with good accuracy. We define in detail the losses for both SPM and NMM below.

3.1.1 Saliency Prediction

The main component of the SPM is a CNN, that applies a traditional cross-entropy loss, which is computed element-wisely at a pixel level across the images. Given an $m \times n$ image $\boldsymbol{x}_i$, its predicted label $\hat{y}$, and its ground truth label y, the cross-entropy loss, L_{CE}, is:

$$L_{CE} = -\big(y \cdot \log(\hat{y}) + (1 - y) \cdot \log(1 - \hat{y})\big) \tag{2}$$

Further, the SPM loss, $\mathscr{L}_{pred}(\Theta, \Sigma)$, is:

$$\mathscr{L}_{\text{pred}}(\Theta, \Sigma) = \sum_{i=1}^{N} \sum_{j=1}^{M} \sum_{m,n} L_{CE}\left(\mathbf{y}_{i,mn}^i, \hat{\mathbf{y}}_{i,mn}^j\right), \tag{3}$$

where $\mathbf{y}_{i,mn}^i$ and $\hat{\mathbf{y}}_{i,mn}^i$ is our predicted noisy saliency map and pseudo ground truth saliency maps (i.e. unsupervised methods), respectively. The first is computed by $\hat{\boldsymbol{y}}_i^j = \bar{\boldsymbol{y}}_i + \boldsymbol{n}_i^j$ and truncated to the interval $[0, 1]$.

3.1.2 Noise Modeling

For each image $\boldsymbol{x}_i$, an empirical noise can be computed from the saliency map outputed from the SPM, $\hat{\boldsymbol{n}}_i^j = \boldsymbol{y}_i^j - \bar{\boldsymbol{y}}_i$. Assuming the empirical noise follows a zero-mean Gaussian probability distribution, $p(\hat{\Sigma}_i)$, which the complete set of parameters being $\hat{\Sigma} = \{\hat{\sigma_{i,mn}}\}$, with the variance of each pixel denoted by $\sigma_{i,mn}$.

Since it is intractable to estimate the true posterior distribution $q(\Sigma)$, the original work proposed to approximate it by sequentially optimizing the parameters of the zero-mean Gaussian prior distribution [1]. In addition, the noise is randomly produced based on an parameters set Σ. Since given an image $\boldsymbol{x}_i$ and the SPM parameters Θ, the NMM works as a probabilistic encoder that generates a distribution over possible values of noise n. The procedure has two stages: (1) a noise map n_i is produced from some prior $q(\Sigma^*)$, and (2) a noise map $\hat{n}_i^j$ is generated and the corresponding pixel variances $\hat{\sigma}_i$ are estimated.

Hence, we want to maximize the agreement between the true distribution $q(\Sigma_i)$ of the NMM and the approximated empirical distribution of noise $p(\hat{\Sigma}_i)$ by minimizing the forward Kullback-Leibler (KL) divergence. Thus, our NMM loss, $\mathscr{L}_{\text{noise}}$, is:

$$\mathscr{L}_{\text{noise}}(\Theta,\Sigma) = \sum_{i=1}^{N} \text{KL}\left(q(\Sigma_i)||p(\hat{\Sigma}_i)\right) \tag{4}$$

Since a zero-mean Gaussin distribution is our noise model prior, the KL divergence has a closed-form as follows:

$$\text{KL}\left(q(\sigma)||p(\hat{\sigma})\right) = \log\left(\frac{\hat{\sigma}}{\sigma}\right) + \frac{\sigma^2 + (\mu-\hat{\mu})^2}{2\cdot\hat{\sigma}^2} - \frac{1}{2} \tag{5}$$

Based on the previous equation, we derive an update rule for the variances of the noise at pixel level of each $m \times n$ image, as described in [1], as follows:

$$(\sigma_i^{t+1})^2 = (\sigma_i^t)^2 + \alpha\left((\hat{\sigma}_i^t)^2 - (\sigma_i^t)^2\right), \tag{6}$$

where α is the step size and was set to 0.01 as in [1].

Since each image has a corresponding noise map, it is difficult to converge simultaneously the parameters of SPM (Θ) and NMM (Σ). As a solution, we update them in rounds. At the first round, noise variance is initialized to be zero, we train the CNN for one epoch, using Eqs. (2) and (3), and we update the NMM using Eq. (6). Based on the variance of the saliency prediction and noisy labels, then the noise variance is updated for each image and retrain the network. Using the updated noise variance sampled from NMM as initialization, the next round can proceed. The next i rounds, we train SPM on unsupervised labels for 20 epochs, using Eq. (1), and we update the NMM using Eq. (6).

3.2 Architecture and Implementation

We build our SPM upon the DeepLab network [18], where a pre-trained ResNet-101 [19] for image classification is re-purposed with modifications in its last classifier layer to convolutional layers.

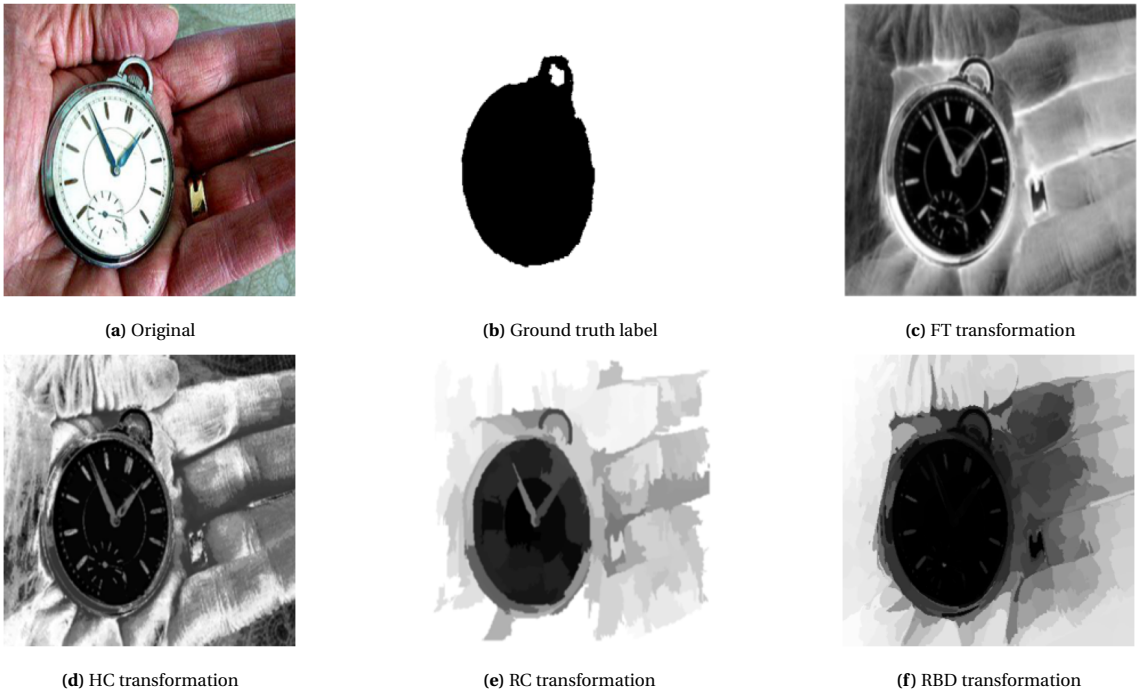
The input of our model are rescaled images (x_i) and transformations (y_i^j) of 256×256. For training, the noise model is used to update the predicted saliency map $\hat{y}_i^j$, and it is not considered in the validation and testing stages, where our SPM outputs the latent salient map ($\ddot{y}_i$).

4 Experiments

We try to minimize the loss function of the framework in the training process using different types of optimizers (i.e. Adam and SGD) and learning rate schedulers (i.e. StepLR and ReduceLROnPlateau) with the MSRA-B dataset and the four image transformations. The dataset, unsupervised image transformation, the evaluation metrics, and the baseline experiments are described below.

4.1 Dataset

In our project, we propose to reproduce the work of [1] with MSRA-B dataset [20], 5000 images with pixel-accurate object labeling, which most images only have one salient object. One example of the dataset with its label and unsupervised transformations are shown in Figure 2. Further, we split the MSRA-B dataset in 2500, 500, and 2000 images as training, validation, and test sets respectively.



(a) Original **(b)** Ground truth label **(c)** FT transformation

(d) HC transformation **(e)** RC transformation **(f)** RBD transformation

Figure 2. An example from the dataset and its ground truth saliency map and unsupervised transformations.

4.2 Images Transformations

In this project, we applied four unsupervised transformations in our images from dataset MSRA-B (Figures 2c, 2d, 2e and 2f). The first one is the Frequency-Tuned (FT), described by [21], which exploits features of color and luminance, is simple to implement, computationally efficient, and in the paper is used to output full resolution saliency maps with well-defined boundaries of salient objects. The second and third one is Histogram Based Contrast (HC) and Region-Based Contrast (RC), both described in [22], which simultaneously evaluates global contrast differences and spatially weighted coherence scores, being simple, efficient, naturally multi-scalable, and produces full-resolution, high-quality saliency maps. The last one is Robust Background Detection (RBD), which characterizes the spatial layout of image regions with respect to image boundaries and has an intuitive geometrical interpretation, as is described in [23].

4.3 Evaluation Metrics

We evaluate our model with a set of evaluation metrics, including mean absolute error (MAE; Eq. (7)), Precision (P; Eq. (8)), Recall (R; Eq. (9)) and F-measure (F_β; Eq. (10)). The MAE represents the dissimilarity between the estimated binary and the ground truth saliency map.

$$MAE = \frac{1}{N}\sum_{j=1}^{N}|y_j - \hat{y}_j|, \tag{7}$$

where N is the number of images in the set, y_j is the ground truth saliency maps and $\hat{y}_j$ is the predicted binary saliency map.

$$P = \frac{TP}{TP+FP} \tag{8}$$

$$R = \frac{TP}{TP+FN}, \tag{9}$$

where TP is the number of true positives, FP is the number of false positives and FN is the number of false negatives.

$$F_\beta = (1+\beta^2)\frac{P \cdot R}{\beta^2 \cdot P + R}, \tag{10}$$

where P is the precision that corresponds to the correctly detected salient pixels, R is the recall that corresponds to the fraction of detected salient pixels w.r.t. the ground truth salient pixels and $\beta = 0.3$.

4.4 Baseline Experiments

We explored three different baseline experiments based on the utilization of the multiple saliency maps of our model. The description of each baseline experiment is provided below.

- **Baseline Experiment 1 (Real):** The SPM is trained directly with the clean ground truth labels provided by the MSRA-B dataset;
- **Baseline Experiment 2 (Noise):** The SPM is trained on our four noisy unsupervised labels (FT, HC, RC, and RBD);
- **Baseline Experiment 3 (Average):** Instead of training the SPM with our noisy unsupervised labels, we train the average saliency map of those unsupervised labels as pseudo ground truth.

In the complete model (denoted by **Complete**), the SPM and NMM are trained together with the procedure previously described in Section 3.1.2. However, we just trained their experiments with one round.

Finally, our baseline experiments and complete model are trained for 20 epochs. As mentioned above, we explored different optimizers and learning rate schedulers and their parameters are as follows. The Adam optimizer with betas of 0.9 and 0.99 and base learning rate of $1e-3$ and the SGD optimizer with momentum 0.9 and base learning rate of $1e-3$. The StepLR decays the learning rate of each parameter group by 0.9 every 10 epochs and the ReduceLROnPlateau decays the learning rate by 0.9 when the validation loss has stopped improved for 10 epochs and waits for one epoch before resuming normal operation after the learning rate has been reduced.

5 Results

5.1 Loss function

The loss functions, Eq. (1), per epoch at the training stage, for our baseline experiments and complete model, are shown in Figure 3. We noticed that the loss function decreases and smoothly stabilizes in all scenarios, which is the expected behavior. This minimization of the loss function indicates that the baseline experiments and our framework are learning better saliency detection maps. In addition, our framework is also learning better parameters for the NMM, from which the noise is sampled.

5.2 Evaluation Metrics

Here, we compare the evaluation metrics (MAE and F-measure) between our baseline experiments and complete models, which are shown in Table 1. The combination of SGD optimizer and ReduceLROnPlateau scheduler showed the best evaluation metrics for all baseline experiments and our framework except for Average that Adam optimizer had slightly better metrics, agreeing with the implementation in [1]. Our framework outperforms both Noise and Average experiments, showing that our framework works better with unsupervised labels as pseudo ground truth. Further, the Average also outperforms the Noise experiment, because the first has 3000 averaged training unsupervised label, which probably better identifies the salient object than 12000 training unsupervised labels; and we suggest that the chosen unsupervised saliency methods are complementary to some extent, since averaging than yield better supervision than each one of them individually. Unexpectedly, our framework presented a worse performance than the Real experiment, which is trained with a ground truth saliency map. This indicates that our selection of unsupervised saliency methods could not have been the best option, which leads to a worse performance when compared to [1].

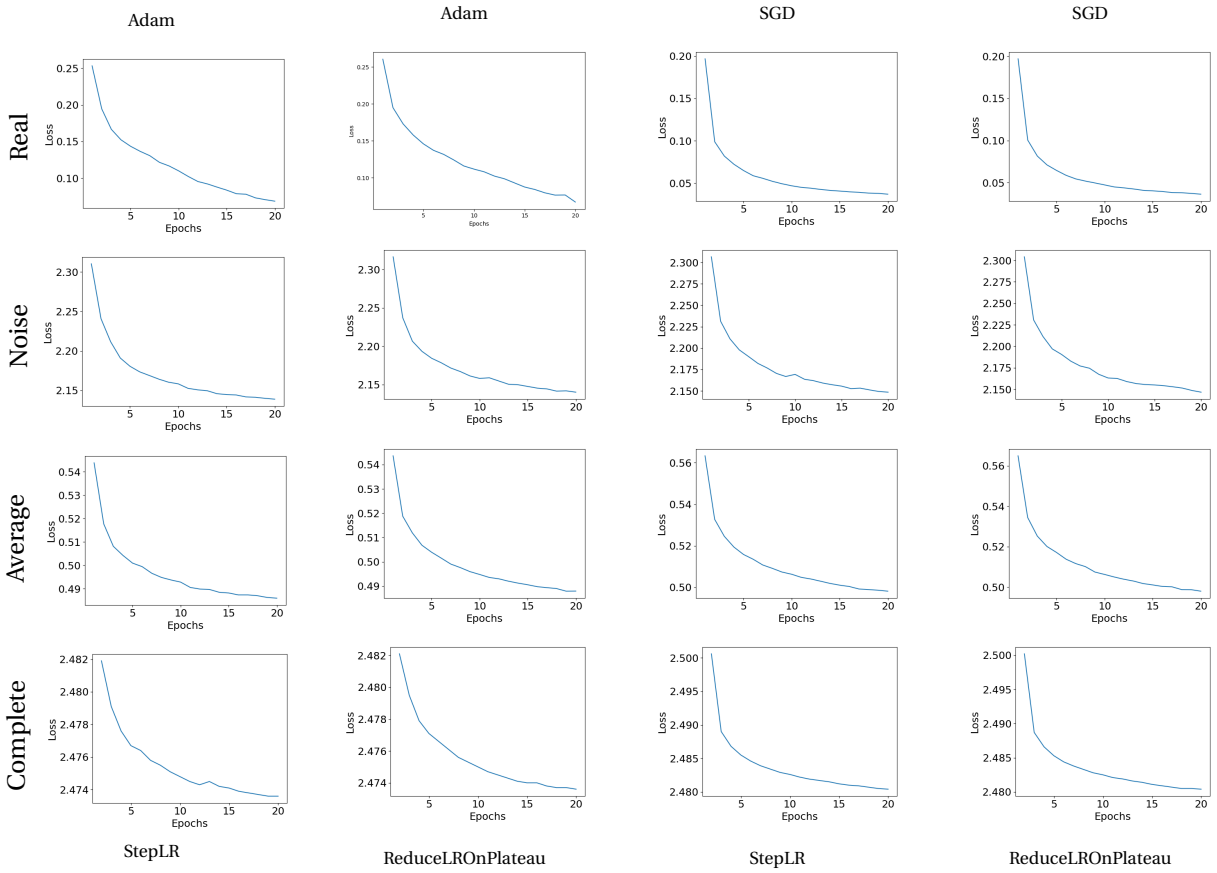


Figure 3. Training loss function for each baseline experiment and our framework with different optimizers and learning rate schedulers on MSRA-B dataset. The experiment, the optimizer and the learning rate scheduler are shown in the right, top and bottom of the figure, respectively.

5.3 Saliency Maps

A predicted saliency map for the baseline experiments and our framework with the SGD optimizer and the ReduceLROnPlateau scheduler (best combination), which achieved the best evaluation metrics (MAE and F-measure), are shown in Figure 4 with their original image and their ground truth saliency map.

Figure 4. Comparison of a predicted saliency between the baseline experiments and our framework.

Additionally, some examples from the validation set are shown for each baseline experiment and our framework, with the SGD optimizer and the ReduceLROnPlateau scheduler, below.

5.3.1 Real

The saliency maps with the SPM trained directly with the clean ground truth labels are shown in Figure 5.

Optimizer	**Adam**				**SGD**			
Scheduler	**StepLR**		**ReduceLROnPlateau**		**StepLR**		**ReduceLROnPlateau**	
Experiment	F_β	***MAE***	F_β	***MAE***	F_β	***MAE***	F_β	***MAE***
Real	0.8934	0.0518	0.8925	0.0520	0.9248	0.037	0.9256	0.036
Noise	0.6514	0.1400	0.5937	0.1567	0.6640	0.1389	0.6682	0.1376
Average	0.6897	0.1359	0.7316	0.1227	0.7061	0.1317	0.7287	0.1267
Complete	0.6904	0.1432	0.7013	0.1384	0.7810	0.1043	0.7888	0.1033

Table 1. Perfomance of each baseline experiment and our framework with different optimizers and learning rate schedulers on MSRA-B dataset.

5.3.2 Noise

The saliency maps of the SPM, trained with our four unsupervised saliency methods (FT, HC, RC and RBD) as ground truth labels, are shown in Figure 6.

5.3.3 Average

The saliency maps of the SPM, trained with the average saliency map of our four unsupervised saliency methods (FT, HC, RC and RBD) as ground truth labels, are illustrated in Figure 7.

5.3.4 Complete

The saliency maps of our framework, SPM and NMM trained together, with our four unsupervised saliency methods (FT, HC, RC and RBD) as pseudo ground truth, are shown in Figure 8.

5.3.5 Comparison

First of all, we observed that the quality of the generated saliency maps are improving over the epochs, producing better saliency maps for the images, in all scenarios (Figures 4, 6, 6, 7 and 8). Also, over epochs, the predicted saliency maps adjust their edges to better fit the maps; however, in some cases it still far from what it should be (e.g. images 3 and 4 of Noise experiment).

In the Real experiment, the saliency maps fit well to the ground truth labels provided by the MSRA-B dataset, agreeing with the evaluation metrics; however, with extra some pixels and some degree of smoothness in the edges of the saliency. In Figure 5, the predicted saliency map of image 3 does not identify pixels in a region, probably due to the low difference in contrast (color and/or luminance) in this region.

In the Noise experiment, the saliency maps had the worst quality compared to the other scenarios, agreeing with the evaluation metrics. Figure 6 suggests that the saliency maps are only identifying regions with high contrast difference with the neighborhood, such as the red clothes in image 1, the light spots and the yellow leaf in image 2, the red cloth in the image 3 and the white feathers and yellow beak in image 4.

In the Average experiment, the saliency maps had an improvement in quality compared to the Noise experiment, also agreeing with the evaluation metrics. Based on Figure 7, we argue that the saliency maps identified regions that are not present in the ground truth labels, but had some visual appeal in the image, such as the forest in the background of image 1.

Putting together Noise and Average experiments, their generated saliency maps indicate that our selection of unsupervised saliency methods could not have been the best possible combination, corroborating what we discussed in Section § 5.2. Probably, our four unsupervised methods (FT, HC, RC, and RBD) identifies some non-complementary features and, when we train with the four maps separated, we learn only to identify regions of high agreement between them. On the other hand, when we train on the average map of the methods, the maps possibly complement each other to learn higher quality saliency map when compared to the Noise experiment.

Finally, our framework produced interesting saliency maps, clearly improving the performance compared to the Noise and Average experiments, which indicates the importance of integrating the NMM into the framework. However, there is still some gap to close w.r.t. the Real experiment, that uses ground truth labels in training, which is possible to achieve as shown in [1]. Based on Figure 8, we recognize that it is the method with the best improvement from epoch one to the last when compared to the baseline experiments. With more noise updates, probably

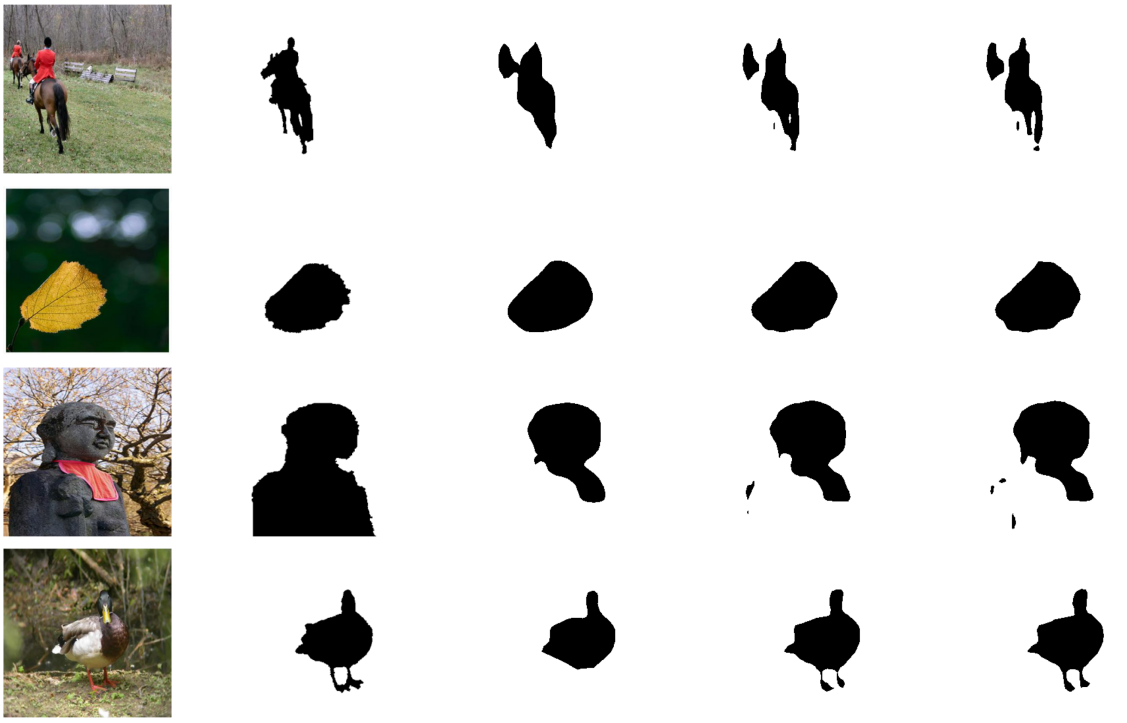


Figure 5. Comparison between the predicted saliency maps of the Real baseline experiment.

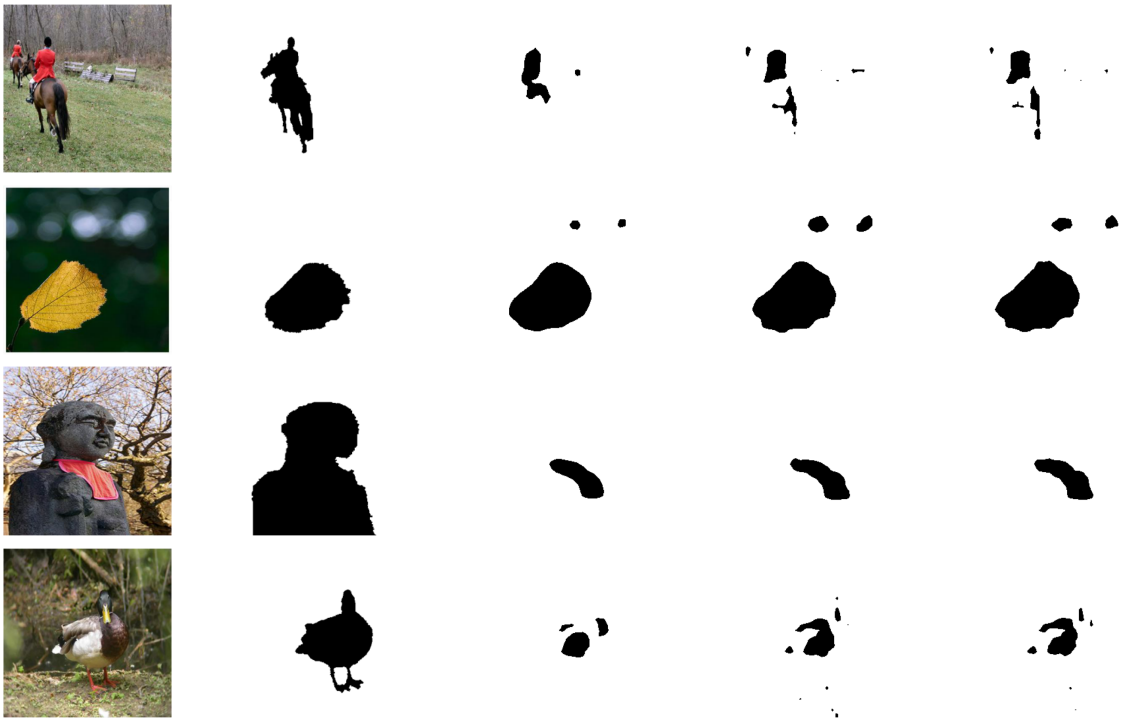


Figure 6. Comparison between the predicted saliency maps of the Noise baseline experiment.

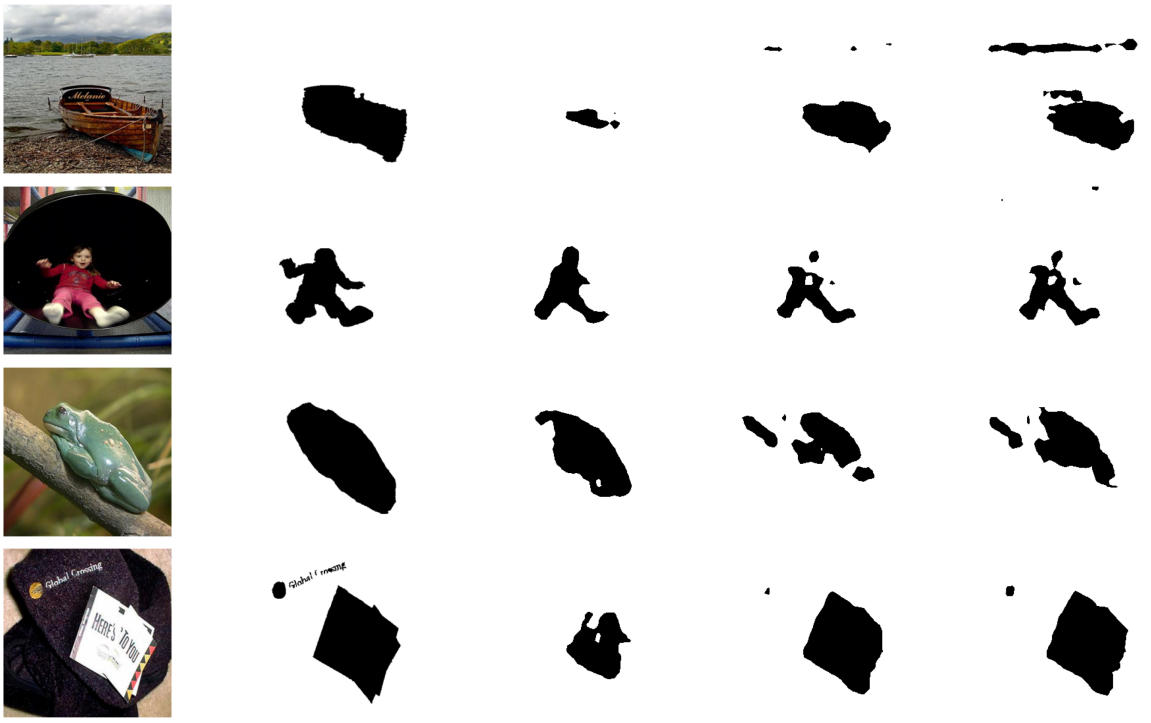


Figure 7. Comparison between the predicted saliency maps of the Average baseline experiment.

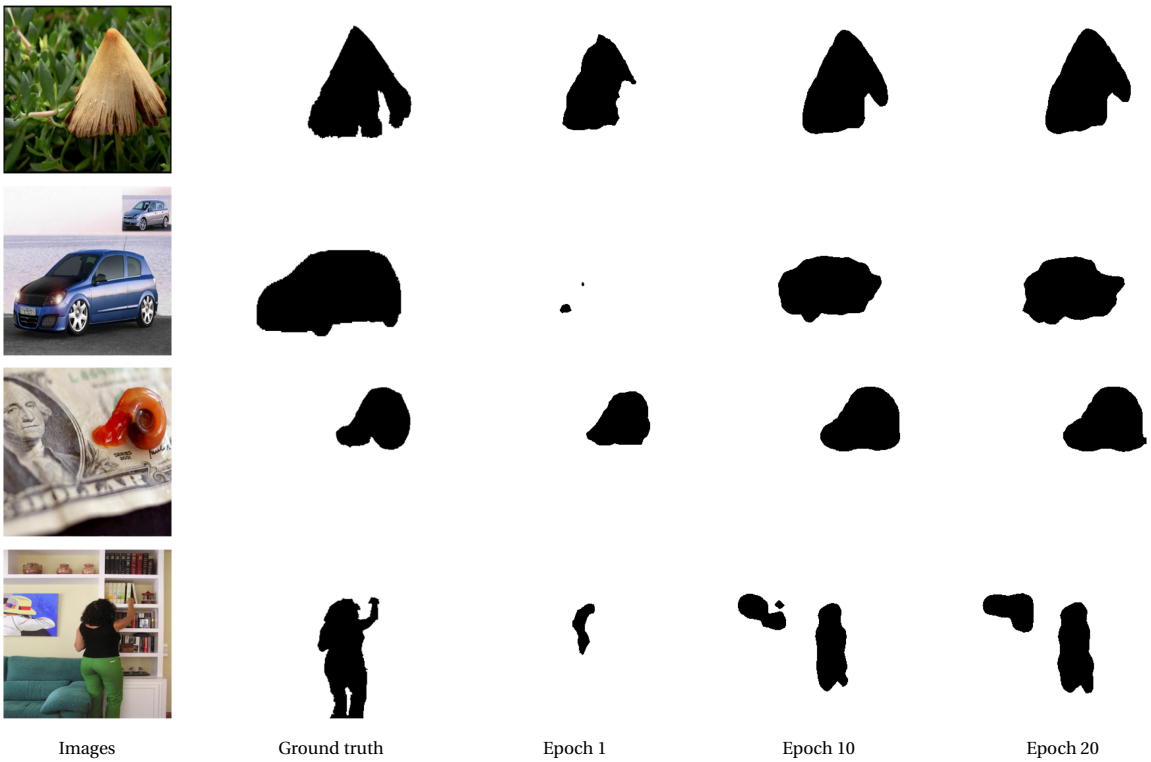


Figure 8. Comparison between the predicted saliency maps of our framework.

we would achieve even higher quality saliency maps with our framework. In addition, a different selection of unsupervised saliency methods as pseudo ground truth could led also to higher quality saliency maps. Further, we could have explored other optimizers and learning rate schedulers and the optimization of some hyperparameters in our framework, in order to achieve even better performance.

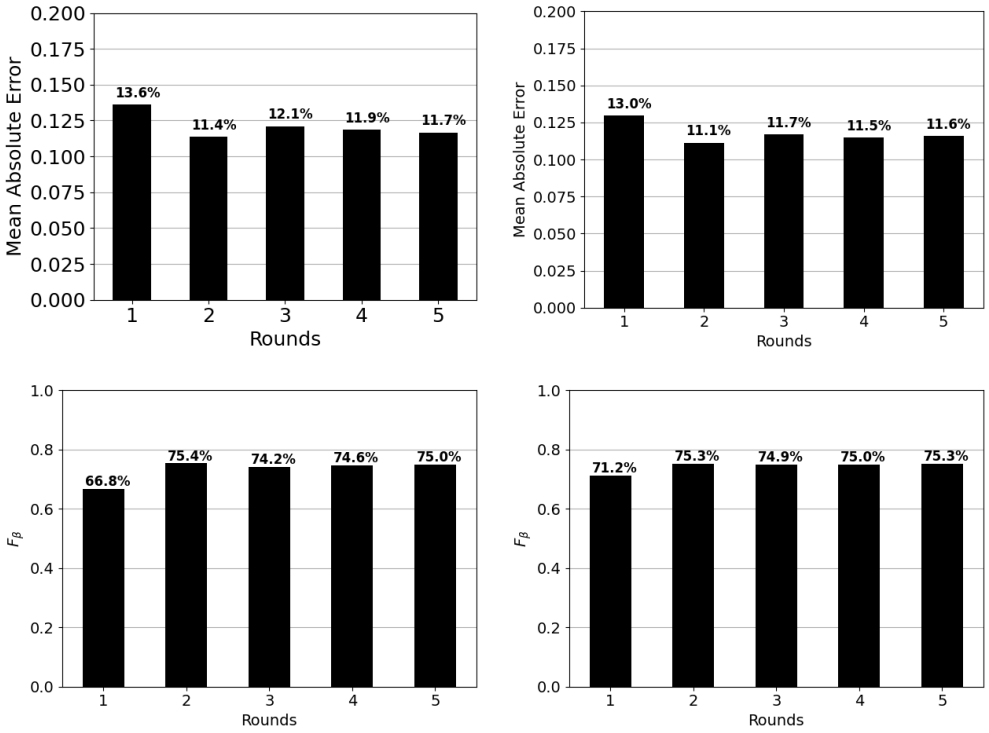
5.3.6 Ablation Studies

Here, we trained the SPM for 20 epochs, with zero noise initialization, before updating the NMM for the first time and start training its parameters (Round 1). After that, we trained four i rounds of our framework, each round lasting for 20 epochs until a new noise update.

In this scenario, we explored two training regimes with our framework: training with the unsupevised labels individually (Complete experiment) and training with the average saliency map of them as pseudo ground truth. This latter regime is due to the Average experiment achieved better results when compared to the Noise experiment (see Table 1).

The MAE and F-measure for both training regimes for each round are shown in Figure 9. Based on it, we observed that the MAE reduces and the F-measure increase over the rounds as expected; however, there is an unexpected behavior for both training regimes, from 1st round to 2nd round, in which the MAE increased and the F-measure decreased. In addition, examples of a saliency map at the end of each round with the original image and the ground truth saliency map for the training regime with individual unsupervised labels and with average labels are shown in Figures 10 and 11, respectively. Our framework starts with the zero noise initialization and consistently improves the SPM through the epochs and also improves performance with each noise update in NMM.

Finally, we argue that both training regimes achieved similar results w.r.t. to the evaluation metrics and qualitative results (i.e. quality of saliency maps), except in the first round in which the second regime (average unsupervised labels) achieved slighty better performance (i.e. evaluation metrics and qualitative results). In general, the NMM was a significant addition to our framework, as it improved its performance at each noise update; however, presenting similar results in both training regimes. Therefore, we restate that our framework would have achieved better results with a better selection of unsupervised saliency methods as pseudo ground truth.



(a) 1st regime (Individual labels)

(b) 2nd regime (Average labels)

Figure 9. Evaluation metrics for both training regimes for each round on MSRA-B dataset.

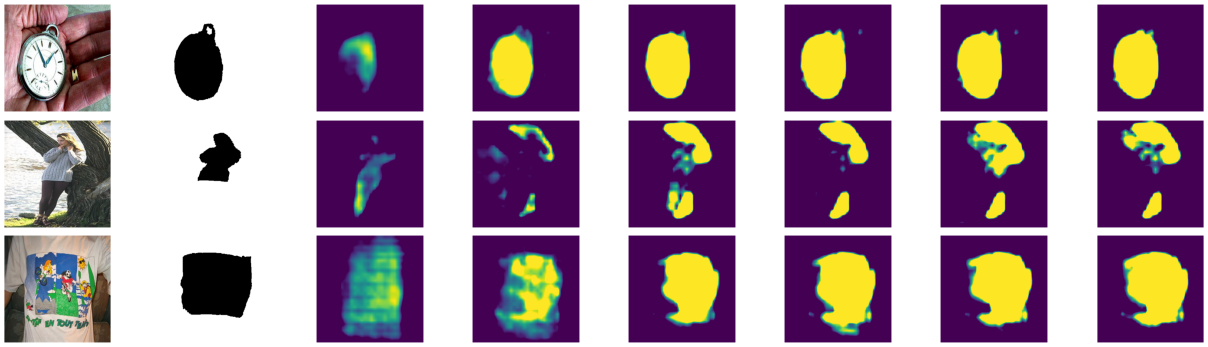


Figure 10. Examples of image, ground truth label and predicted saliency maps generated by each round for the training regime with individual unsupervised labels.

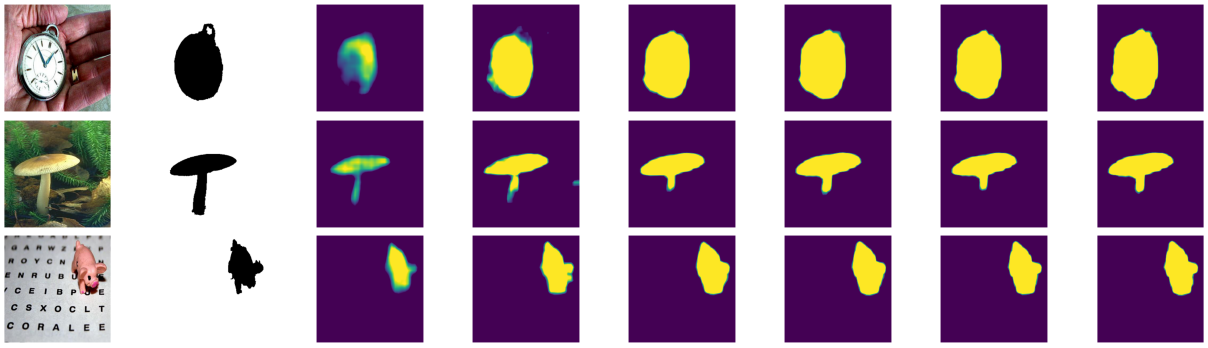


Figure 11. Examples of image, ground truth label and predicted saliency maps generated by each round for the training regime with average unsupervised labels.

6 Conclusion

In this project, we learned about saliency detection in images and reproduced a state-of-the-art unsupervised framework to learn and predict saliency maps in the MSRA-B dataset. The proposal and reproduction of this deep unsupervised state-of-the-art framework was an essential step in our learning and professional development processes in the field of unsupervised machine learning. The DNMSD framework [1] is the second successful deep unsupervised saliency detector, preceded by the SBF framework [17], which was able to close the gap between the deep unsupervised methods and the deep supervised ones. Further, the performance achieved in the saliency maps was without human-annotated saliency maps in network training, which is less labor-intensive and time-consuming, showing their importance to the field of unsupervised machine learning. However, we identified that it highly relies on the set of unsupervised saliency methods chosen to be the pseudo ground truth. Since we achieved worse metrics when compared to the original paper [1] by using a different set of unsupervised methods. Finally, we argue that if we selected a better composition of unsupervised methods and trained our framework for more rounds, i.e. more noise updates, we could have achieved even higher quality saliency maps. As an interesting application, we could apply this framework to create masks to other deep unsupervised computer vision tasks, e.g. classification.

References

[1] J. Zhang, T. Zhang, Y. Dai, M. Harandi, and R. I. Hartley, "Deep unsupervised saliency detection: A multiple noisy labeling perspective," *CoRR*, vol. abs/1803.10910, 2018. arXiv: 1803.10910. [Online]. Available: http://arxiv.org/abs/1803.10910.

[2] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *CoRR*, vol. abs/1312.6034, 2014. arXiv: 1312.6034. [Online]. Available: https://arxiv.org/pdf/1312.6034.pdf.

[3] A. Borji, M.-M. Cheng, H. Jiang, and J. Li, "Salient object detection: A survey," *CoRR*, vol. abs/1411.5878, 2014. arXiv: 1411.5878. [Online]. Available: http://arxiv.org/abs/1411.5878.

[4] G. Amit, O. Hadad, S. Alpert, T. Tlusty, Y. Gur, R. Ben-Ari, and S. Hashoul, "Hybrid mass detection in breast MRI combining unsupervised saliency analysis and deep learning," in *Medical Image Computing and Computer Assisted Intervention - MICCAI 2017*, Springer International Publishing, 2017, pp. 594–602. DOI: 10.1007/978-3-319-66179-7_68. [Online]. Available: https://doi.org/10.1007/978-3-319-66179-7_68.

[5] Y. Zhang, J. Shen, M. Rotea, and N. Gans, "Robots looking for interesting things: Extremum seeking control on saliency maps," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011, pp. 1180–1186.

[6] F. F. E. Guraya, F. A. Cheikh, A. Tremeau, Y. Tong, and H. Konik, "Predictive saliency maps for surveillance videos," in *2010 Ninth International Symposium on Distributed Computing and Applications to Business, Engineering and Science*, 2010, pp. 508–513.

[7] B. Dunin-Keplicz, A. Jankowski, A. Skowron, and M. Szczuka, *Monitoring, Security, and Rescue Techniques in Multiagent Systems (Advances in Soft Computing)*. Berlin, Heidelberg: Springer-Verlag, 2005, ISBN: 3540232451.

[8] M. T. Nguyen, P. Siritanawan, and K. Kotani, "Saliency map extraction in human crowd rgb data," in *2019 58th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, 2019, pp. 941–946.

[9] Z. Ren, S. Gao, L. Chia, and I. W. Tsang, "Region-based saliency detection and its application in object recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 5, pp. 769–779, 2014.

[10] A. De Abreu, C. Ozcinar, and A. Smolic, "Look around you: Saliency maps for omnidirectional images in vr applications," in *2017 Ninth International Conference on Quality of Multimedia Experience (QoMEX)*, 2017, pp. 1–6.

[11] I. Ullah, M. Jian, S. Hussain, J. Guo, H. Yu, X. Wang, and Y. Yin, "A brief survey of visual saliency detection," *Multimedia Tools and Applications*, pp. 1–41, 2020.

[12] G. Li and Y. Yu, "Visual saliency based on multiscale deep features," *CoRR*, vol. abs/1503.08663, 2015. arXiv: 1503.08663. [Online]. Available: http://arxiv.org/abs/1503.08663.

[13] Q. Hou, M.-M. Cheng, X. Hu, A. Borji, Z. Tu, and P. H. S. Torr, "Deeply supervised salient object detection with short connections," *CoRR*, vol. abs/1611.04849, 2016. arXiv: 1611.04849. [Online]. Available: http://arxiv.org/abs/1611.04849.

[14] P. Zhang, D. Wang, H. Lu, H. Wang, and X. Ruan, "Amulet: Aggregating multi-level convolutional features for salient object detection," *CoRR*, vol. abs/1708.02001, 2017. arXiv: 1708.02001. [Online]. Available: http://arxiv.org/abs/1708.02001.

[15] Z. Luo, A. Mishra, A. Achkar, J. Eichel, S. Li, and P. Jodoin, "Non-local deep features for salient object detection," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 6593–6601.

[16] L. Zhou and X. Gu, "Deep supervised visual saliency model addressing low-level features," *Journal of Ambient Intelligence and Humanized Computing*, Sep. 2019. DOI: 10.1007/s12652-019-01441-9. [Online]. Available: https://doi.org/10.1007/s12652-019-01441-9.

[17] D. Zhang, J. Han, and Y. Zhang, "Supervision by fusion: Towards unsupervised learning of deep salient object detector," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 4068–4076.

[18] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *CoRR*, vol. abs/1606.00915, 2016. arXiv: 1606.00915. [Online]. Available: http://arxiv.org/abs/1606.00915.

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015. arXiv: 1512.03385. [Online]. Available: http://arxiv.org/abs/1512.03385.

[20] J. Wang, H. Jiang, Z. Yuan, M.-M. Cheng, X. Hu, and N. Zheng, "Salient object detection: A discriminative regional feature integration approach," *International Journal of Computer Vision*, vol. 123, no. 2, pp. 251–268, 2017, ISSN: 1573-1405. DOI: 10.1007/s11263-016-0977-3.

[21] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk, "Frequency-tuned salient region detection," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 1597–1604.

[22] M.-M. Cheng, N. Mitra, X. Huang, P. Torr, and S.-M. Hu, "Salient object detection and segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, Oct. 2011. DOI: 10.1109/TPAMI.2014.2345401.

[23] W. Zhu, S. Liang, Y. Wei, and J. Sun, "Saliency optimization from robust background detection," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2814–2821.