Multi-Branch Siamese Networks with Online Selection for Object Tracking

Zhenxi Li[1], Guillaume-Alexandre Bilodeau[1], and Wassim Bouachir[2]

[1]LITIV lab, Polytechnique Montreal
{zhenxi.li, guillaume-alexandre.bilodeau}@polymtl.ca
[2]TELUQ University
wassim.bouachir@teluq.ca

Abstract. In this paper, we propose a robust object tracking algorithm based on a branch selection mechanism to choose the most efficient object representations from multi-branch siamese networks. While most deep learning trackers use a single CNN for target representation, the proposed Multi-Branch Siamese Tracker (MBST) employs multiple branches of CNNs pre-trained for different tasks, and used for various target representations in our tracking method. With our branch selection mechanism, the appropriate CNN branch is selected depending on the target characteristics in an online manner. By using the most adequate target representation with respect to the tracked object, our method achieves real-time tracking, while obtaining improved performance compared to standard Siamese network trackers on object tracking benchmarks.

Keywords: Object tracking · Siamese networks · Online branch selection.

1 Introduction

Model-free visual object tracking is one of the most fundamental problems in computer vision. Given the object of interest marked in the first video frame, the objective is to localize the target in subsequent frames, despite object motion, changes in viewpoint, lighting variation, among other disturbing factors. One of the most challenging difficulties with model-free tracking is the lack of prior knowledge on the target object appearance. Since any arbitrary object may be tracked, it is impossible to train a fully specialized tracker.

Recently, convolutional neural networks (CNNs) have demonstrated strong power in learning feature representations. To fully exploit the representation power of CNNs in visual tracking, it is desirable to train them on large datasets specialized for visual tracking, and covering a wide range of variations in the combination of target and background. However, it is truly challenging to learn a unified representation based on videos that have completely different characteristics. Some trackers [1] train regression networks for tracking in an entirely offline manner. Other works [2,3,6] propose to train deep CNNs to address the general similarity learning problem in an offline phase and evaluate the similarity

online during tracking. However, since these works have no online adaptation, the representations they learned offline are general but not always discriminative.

Rather than applying a single fixed network for feature extraction, we propose to use multiple network branches with an online branch selection mechanism. It is well known that different networks designed and trained for different tasks have diverse feature representations. With the online branch selection mechanism, our tracker dynamically selects the most efficient and robust branch for target representation, even if the target appearance changes. Our goal is to improve the generalization capability with multiple networks.

The main contributions of our work are summarized as follows. First, we propose a multi-branch framework based on a siamese network for object tracking. The proposed architecture is designed to extract appearance representation robust against target variations and changing contrast with background scene elements. Second, to make the full use of the different branches, we propose an effective and generic branch selection mechanism to dynamically select branches according to their discriminative power. Third, on the basis of multiple branches and branch selection mechanism, we present a novel deep learning tracker achieving real-time and improved tracking performance. Our extensive experiments compare the proposed Multi-Branch Siamese Tracker (MBST) with state-of-the-art trackers on OTB benchmarks [4,5].

2   Related Work

Siamese Network Based Trackers. Object tracking can be addressed using similarity learning. By learning a deep embedding function, we can evaluate the similarity between an exemplar image patch and a candidate patch in a search region. These procedures allow to track the target to the location that obtains the highest similarity score. Inspired by this idea, the pioneering work of SiamFC [2] proposed a fully-convolutional Siamese Network in which the similarity learning with deep CNNs is addressed using a Siamese architecture. Since this approach does not need online training, it can easily achieve real-time tracking. Due to the robustness and real-time performance of the SiamFC [2] approach, several subsequent works proceeded along this direction to address the tracking problem. In this context, EAST [7] employs an early-stopping agent to speed up tracking where easy frames are processed with cheap features, while challenging frames are processed with deep features. CFNet [3] incorporates a Correlation Filter into a shallow siamese network, which can speed up tracking without accuracy drop comparing to a deep Siamese network. TRACA [8] applies context-aware feature compression before tracking to achieve high tracking performance. SA-Siam [6] utilizes the combination of semantic features and appearance features to improve generalization capability. In our work, we use the Siamese Network as embedding function to extract feature representations. All branches use the Siamese architecture to apply identical transformation on target patch and search region.

Multi-Branch Tracking Frameworks. The diversity of target representation from a single fixed network is limited. The learned features may not be

discriminative in all tracking situations. There are many works using diverse features with context-aware or domain-aware scheme.

TRACA [8] is a multi-branch tracker, which utilizes multiple expert auto-encoders to robustly compress raw deep convolutional features. Since each of expert auto-encoders is trained according to a different context, it performs context-dependent compression. MDNet [9] is composed of shared layers and multiple branches of domain-specific layers. BranchOut [18] employs a CNN for target representation, with a common convolutional layers and multiple branches of fully connected layers. It allows different number of layers in each branch to maintain variable abstraction levels of target appearances.

A common insight of these multi-branch trackers is the possibility to make a robust tracker by utilizing different feature representations. Our method shares some insights and design principles with other multi-branch trackers. Our network architecture is composed of multiple branches separately trained offline and focusing on different types of CNN features. In addition, we use an AlexNet [11] branch in our framework that is designed and pretrained for image classification. In our multi-branch frameworks, the combination of branches trained in different scenarios ensures a better use of diverse feature representations.

Online Branch Selection. Different models produce various feature maps on different tracked targets in different scales, rotations, illumination and other factors. Using all features available for a single object tracking is neither efficient nor effective. BranchOut [18] selects a subset of branches randomly for model update to diversify learned target appearance models. MDNet [9] learns domain-independent representations from pretraining, and identifies branches through online learning.

In our online branch selection mechanism, we analyse the feature representation of each branch to select the most robust branch at every T frames. This allows us to use diverse feature representations and to handle various challenges in the object tracking problem more efficiently.

3   Multi-Branch Siamese Tracker

We propose a multi-branch siamese network for tracking. Given that different neural network models produce diverse feature representations, we use many of them as branches in our tracker to produce diverse feature representations and select the most robust branch with our online branch selection mechanism.

3.1   Network Architecture

Using multiple target representations is shown to be beneficial for object tracking [6,10], as different CNNs can provide various feature representations. In our work, we ensemble N_e siamese networks including N_s context-dependent branches and one AlexNet branch as $N_e = N_s + 1$. The context-dependent branches have the same structure as SiamFC [2] and the AlexNet branch has
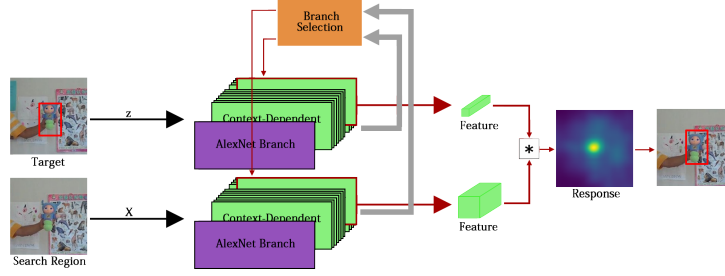
Fig. 1. The architecture of our MBST tracker. Context-dependent branches are indicated by green blocks and AlexNet branch is indicated by purple blocks.

the same structure as AlexNet [11]. Each branch of the tracker is a siamese network applying identical transformation φ_i to both inputs and combining their representation by a cross-correlation layer. The architecture of the proposed tracker is illustrated in Fig. 1.

The input consists of a target patch cropped from the first video frame and another patch containing the search region in the current frame. The target patch z has a size of $W_z \times H_z \times 3$, corresponding to the width, height and color channels of the image patch. The search region X has a size of $W_X \times H_X \times 3$ ($W_z < W_X$ and $H_z < H_X$), representing also the width, height and color channels of the search region. X can be considered as a collection of candidate patches x in the search region with the same dimension as z.

From what we observed, there are two strategies to improve the discriminative ability of the tracking networks. The first one is training the network in different contexts, while the second one is to use multiple networks designed and trained for different tasks. In our approach, we utilize context-dependent branches pretrained in different contexts in addition to another branch pretrained for image classification task to improve our tracking performance. We note that more branches could be added with other pre-trained networks at the cost of slower performances.

Context-dependent branches: We use N_c context-dependent branches and one general branch as $N_s = N_c + 1$. All these branches have the same architecture as the SiamFC network [2]. Context-dependent branches are trained in three steps. Firstly, we train the basic siamese network on the ILSVRC-2015 [12] video dataset (henceforth ImageNet), including 4,000 video sequences and around 1.3 million frames containing about 2 million tracked objects. We keep the basic siamese network as the general branch. Then, we perform contextual clustering on the low level feature map from the ImageNet Video dataset to find N_c ($N_c = 10$) context-dependent clusters. Finally, we use the N_c clusters to train N_c context-dependent branches initialized by the basic siamese network. These branches take (z, X) as input and extract their feature maps. Then, using a cross correlation layer we combine their feature maps to get a response map. The response map of context-dependent branches is calculated as:
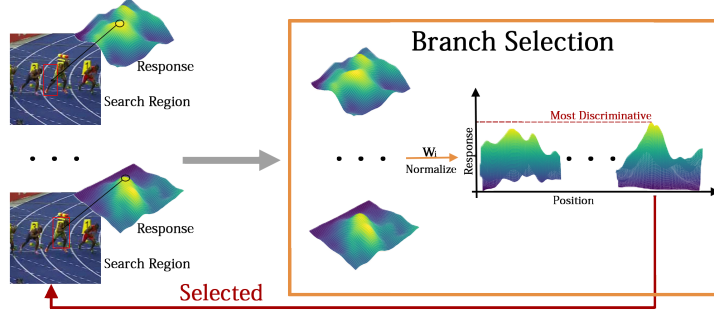
Fig. 2. Online branch Selection mechanism and response map example.

$$h_{s_i}(z, X) = corr(f_{s_i}(z), f_{s_i}(X)), \tag{1}$$

where s_i indicates the contextual index including the general branch $(i = 0)$, $f(\cdot)$ denotes features generated by the network.

The AlexNet branch: We use AlexNet [11] pretrained on the image classification task as a branch with a network trained for a different task. Small modifications are made on the stride to ensure that the output response map has the same dimension as other branches. Since AlexNet is trained for image classification and the deeper layers encode more semantic information of targets, target representations from this branch are more robust to significant appearance variations. The network output corresponds to (z, X) as input, while the generated features are denoted as $f_a(\cdot)$. The response map is expressed as:

$$h_a(z, X) = corr(f_a(z), f_a(X)). \tag{2}$$

In our implementation, MBST is composed of context-dependent branches and AlexNet branch. The output of each branch is a response map indicating the similarity between target z and candidate patch x within the search region X. The branch selection mechanism compares the maps from each branch to select the most discriminative one. The corresponding branch is then used for $T - 1$ frames.

3.2   Online Branch Selection Mechanism

Different branches trained in different scenarios can be used to diversify the target representation. To ensure the optimal exploitation of the diverse representations from our branches, we designed a branch selection mechanism to monitor the tracking output and automatically select the most discriminative branch as illustrated in Fig. 2.

Given the input image pair, each branch applies identical transformation to both inputs and calculates the response map h using a cross-correlation layer. Since the ranges of feature values from different branches are different, we apply

response weights w_i on response map of each branches to normalize their range difference. The discriminative power is then measured based on the weighted response maps from all branches. The heuristic approach we used to measure the discriminative power of branches is formulated as:

$$R(w_i h_{B_i}) = w_i(P(h_{B_i}) - M(h_{B_i})),\tag{3}$$

where h_{B_i} is the response map for each branch B_i, P_{B_i} is the peak value of the response map h_{B_i}, and $M_{h_{B_i}}$ is the minimum value of the response map h_{B_i}.

The objective function of our branch selection mechanism can be written as:

$$B^* = \underset{B_i}{\operatorname{argmax}}\, R(w_i h_{B_i}),\tag{4}$$

where B^* is the selected branch to transform inputs.

4 Experiments

The first aim of our experiments is to investigate the effect of incorporating multiple feature representations with an online branch selection mechanism. For this purpose, we performed ablation analysis on our framework. We then compare our method with state-of-the-art trackers. The experimental results demonstrate that our method achieves improved performance with respect to the basic SiamFC tracker [2].

4.1 Implementation Details

Network structure: The context-dependent branches have exactly the same structure as the SiamFC network [2]. For the AlexNet branch, we use AlexNet [11] pretrained on ImageNet dataset [12] with a small modification to ensure that the output response map has the same dimension as other branches, which is 17×17. Other branches could also be used based on other network architectures.

Data Dimensions: In our experiment, the target image patch z has a dimension of $127 \times 127 \times 3$, and the search region X has a dimension of $255 \times 255 \times 3$. But since all branches are fully convolution layers, they can also be adapted to any other dimension easily. The embedding output for z and X has a dimension of $6 \times 6 \times 256$ and $22 \times 22 \times 256$ respectively.

Training: We use the ImageNet dataset [12] for training and only consider color images. For simplicity, we randomly pick a pair of images, we crop z in the center and X in the center of another image. Images are scaled such that the bounding box, plus an added margin for context, has a fixed area. The basic siamese branch is trained for 50 epochs with an initial learning rate of 0.01. The learning rate decays after every epoch with a decay factor δ of 0.869. The context-dependent branches are fine-tuned based on the parameters of the general branch with a learning rate 0.00001 for 10 epochs. For the AlexNet branch, we directly use AlexNet [11] pretrained on ImageNet dataset [12].

Table 1. Ablation study of MBST on OTB benchmarks. Various combinations of general siamese branch, context-dependent branches and AlexNet branch are evaluated.

General	Context	AlexNet	OTB-2013 AUC	OTB-2013 Prec.	OTB-50 AUC	OTB-50 Prec.	OTB-100 AUC	OTB-100 Prec.	FPS
✓			0.600	0.791	0.519	0.698	0.585	0.766	**65.0**
	✓		0.601	0.798	0.523	0.707	0.584	0.768	18.6
		✓	0.581	0.761	0.501	0.678	0.560	0.741	63.6
✓	✓		0.594	0.784	0.535	0.721	0.587	0.770	16.9
✓		✓	0.605	0.796	0.536	0.718	0.599	0.783	42.9
	✓	✓	0.616	0.811	0.570	0.767	0.614	0.806	16.9
✓	✓	✓	**0.620**	**0.816**	**0.573**	**0.773**	**0.617**	**0.811**	16.9

Our experiments are performed on a PC with a Intel i7-3770 3.40 GHz CPU and a Nvidia Titan X GPU. We evaluated our results using the Python implementation of the OTB toolkit. The average testing speed of MBST is 17 fps.

Hyperparameters: The weights w_i for context-dependent branches have the same value of 1.0. For AlexNet branch, we perform a grid search from 8.0 to 12.0 with step 0.5. Evaluation suggests that the best performance is achieved when w_i is 10.5. This value is thus used for all the test sequences. In order to handle scale variations, we rescale the inputs into three different resolutions.

4.2   Dataset and Evaluation Metrics

OTB: We evaluate the proposed tracker on the OTB benchmarks [4,5] with eleven interference attributes for the video sequences. The OTB benchmark uses the precision and success rate for quantitative analysis. For the precision plot, we calculate the average Euclidean distance between the center locations of the tracked targets and the manually labeled ground truth. Then the average center location error over all the frames of one sequences is used to summarize the overall performance. As the representative precision score for each tracker, we use the score for the threshold of 20 pixels. For the success plot, we compute the IoU (intersection over union) between the tracked and ground truth bounding boxes. A success plot is obtained by evaluating the success rate at different IoU thresholds. The area-under-curve (AUC) of the success plot is reported.

4.3   Ablation Analysis

To verify the contribution of each branch and the online branch selection mechanism of our algorithm, we implemented several variations of our approach and evaluated them on the OTB benchmarks.

Multiple branches improve the tracking result. We compared our full branches algorithm with various combination of branches as illustrated in Table 1. We evaluate the performances of the original branch, context-dependent branches and AlexNet branch alone. Note that branch selection is applied only when we evaluate the context-dependent branches, since many branches are
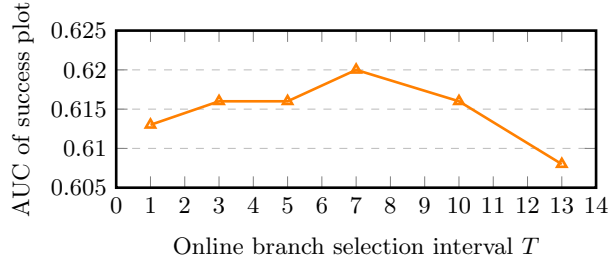
Fig. 3. Curve for the branch selection interval T on OTB2013 benchmark [4].

available. For the other experiments in Table 1, we combine these branches with online branch selection for testing. Results clearly demonstrate that the proposed multiple branches architecture allows a better use of diverse feature representations. The best FPS is achieved by the general siamese branch, which is expected since it needs less computations with only one branch.

Online branch selection for every frame is not necessary. As shown in Fig. 3, we conduct experiments on the branch selection interval T by changing the value: $T = 1, 3, 5, 7, 10, 13$. When the value of branch selection interval is less than 7 frames, the tracking performance is reduced. This can be explained by the fact that a frequent execution of the selection mechanism increases the possibility of selecting an inappropriate branch. When the value of branch selection interval is more than 7 frames, the tracking performance is also decreased because we keep for a too long period a branch that is not discriminative anymore. In our experiments, the optimal value of branch selection interval T was 7 frames.

4.4   Comparison with State-of-the Art Trackers

We compare MBST with CFNet [3], SiamFC [2], Staple [13], LCT [14], Struck [15], MEEM [16], SCM [17], LMCF [19], MUSTER [20], TLD [21] on OTB benchmarks. The precision plots and success plots of one path evaluation (OPE) are shown in Fig. 4. Based on precision and success plots, the overall comparison suggests that the proposed MBST achieved the best performance among these state-of-the-art trackers on OTB benchmarks. Notably, it outperforms SiamFC [2] as well as its variation CFNet [3] on all datasets. This demonstrates that diverse feature representations are important to improve tracking, as feature maps from various CNNs can be quite different. Fig. 5 demonstrates that our tracker effectively handles all kinds of challenging situations that often require high-level semantic understanding. For example, our tracker significantly outperforms SiamFC in the case of deformation, occlusion and out-of-plane rotations because the contrast between the object and the background changes and switching to another feature map may give a better discriminativity. Therefore, our approach is beneficial each time the appearance of the object changes significantly during its tracking.
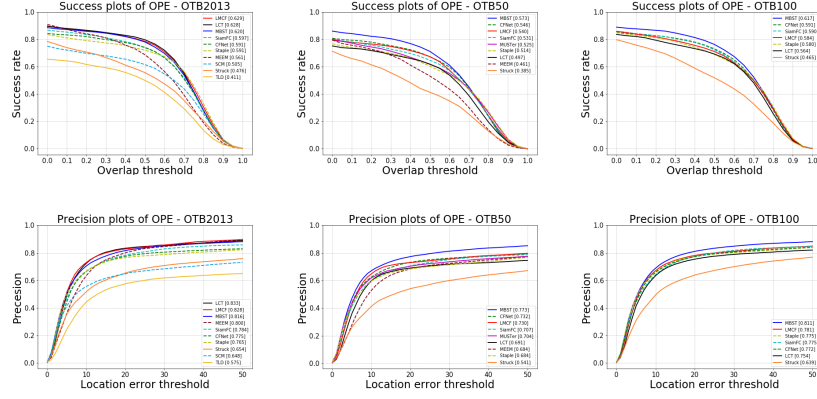
Fig. 4. The success plots and precision plots on OTB benchmarks. Curves and numbers are generated with Python implemented OTB toolkit.

5   Conclusion

In this paper, we propose a Multi-Branch Siamese Network with Online Selection. We ensemble multiple siamese networks to diversify target feature representations. Using our online branch selection mechanism, the most discriminative branch is selected against target appearance variations. Our tracker benefits from the diverse target representation, and can handle all kinds of challenging situations in visual object tracking. Our experiment results show improved performances compared to standard Siamese network trackers, while outperform several recent state-of-the-art trackers.
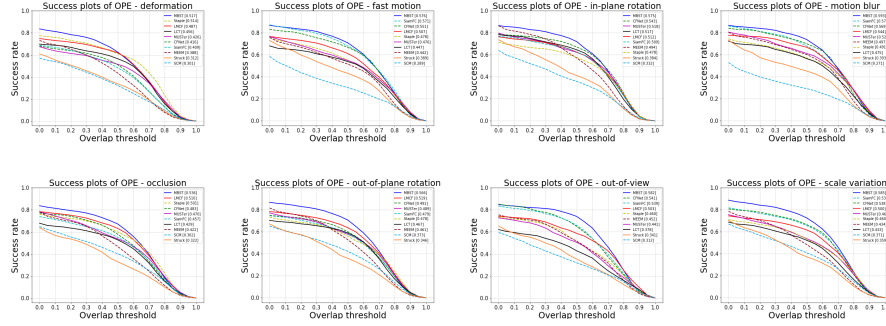

Fig. 5. The Success plot on OTB50 for eight challenge attributes: deformation, fast motion, in-plane rotation, motion blur, occlusion, out-of-plane rotation, out-of-view, scale variation.

References

1. Held, D., Thrun, S and Savarese, S.: Learning to Track at 100 FPS with Deep Regression Networks. In: Leibe, B., Matas, J., Sebe, N. and Welling, M., ECCV 2016, pp. 749–765. Springer
2. Bertinetto, L., Valmadre, J., Henriques, J.F., Vedaldi, A. and Torr, P.H.: Fully-convolutional siamese networks for object tracking. In: ECCV 2016, pp. 850–865. Springer
3. Valmadre, J., Bertinetto, L., Henriques, J.F., Vedaldi, A. and Torr, P.H: End-to-end representation learning for correlation filter based tracking. In: CVPR 2017, pp. 5000–5008. IEEE
4. Wu, Y., Lim, J. and Yang, M.H.: Online object tracking: A benchmark. In: CVPR 2013, pp. 2411–2418
5. Wu, Y., Lim, J. and Yang, M.H.: Object tracking benchmark. TPAMI**37**(9), 1834–1848(2015)
6. He, A., Luo, C., Tian, X. and Zeng, W.: A twofold siamese network for real-time object tracking. In: CVPR 2018, pp. 4834–4843
7. Huang, C., Lucey, S. and Ramanan, D.: Learning policies for adaptive tracking with deep feature cascades. In: ICCV 2017, pp: 105–114
8. Choi, J., Chang, H.J., Fischer, T., Yun, S., Lee, K., Jeong, J., Demiris, Y. and Choi, J.Y.: Context-aware Deep Feature Compression for High-speed Visual Tracking. In: CVPR 2018, pp: 479–488
9. Nam, H. and Han, B: Learning multi-domain convolutional neural networks for visual tracking. In: CVPR 2016, pp: 4293–4302
10. Nam, H., Baek, M. and Han, B.: Modeling and propagating cnns in a tree structure for visual tracking. arXiv preprint arXiv:1608.07242(2016)
11. Krizhevsky, A., Sutskever, I. and Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NIPS 2012, pp: 1097–1105.
12. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. Imagenet large scale visual recognition challenge. IJCV**115**(3), 211–252(2015)
13. Bertinetto, L., Valmadre, J., Golodetz, S., Miksik, O. and Torr, P.H.: Staple: Complementary learners for real-time tracking. In: CVPR 2016, pp: 1401–1409
14. Ma, C., Yang, X., Zhang, C. and Yang, M.H.: Long-term correlation tracking. In: CVPR 2015, pp: 5388–5396
15. Hare, S., Saffari, A. and Torr, P.H.: Struck: Structured output tracking with kernels. In: ICCV 2011, pp: 263–270
16. Zhang, J., Ma, S. and Sclaroff, S.: MEEM: robust tracking via multiple experts using entropy minimization. In: ECCV 2014, pp: 188–203
17. Zhong, W., Lu, H., and Yang, M.H.: Robust object tracking via sparsity-based collaborative model. In: CVPR 2012, pp: 1838–1845
18. Han, B., Sim, J. and Adam, H.: BranchOut: Regularization for online ensemble tracking with convolutional neural networks. In: ICCV 2017, pp: 2217–2224
19. Wang, M., Liu, Y. and Huang, Z.: Large margin object tracking with circulant feature maps. In: CVPR 2017, pp: 21–26
20. Hong, Z., Chen, Z., Wang, C., Mei, X., Prokhorov, D. and Tao, D.: Multi-store tracker (muster): A cognitive psychology inspired approach to object tracking. In: CVPR 2015, pp: 749–758
21. Kalal, Z., Mikolajczyk, K., Matas, J., et al: Tracking-learning-detection. TPAMI**34**(7), 1409(2012)