Flow R-CNN: Flow-enhanced object detection

Athanasios Psaltis*, Anastasios Dimou*†, Federico Alvarez†, Petros Daras*

*Centre for Research and Technology Hellas
Thessaloniki, Greece
Email: {at.psaltis, dimou, daras}@iti.gr
†Universidad Politécnica de Madrid
Madrid, Spain
Email: fag@gatv.ssr.upm.es

Abstract—**This work addresses the problem of multi-task object detection in an efficient, generic but at the same time simple way, following the recent and highly promising studies in the computer vision field, and more specifically the Region-based CNN (R-CNN) approach. A flow-enhanced methodology for object detection is proposed, by adding a new branch to predict an object-level flow field. Following a scheme grounded on neuroscience, a pseudo-temporal motion stream is integrated in parallel to the classification, bounding box regression and segmentation mask prediction branches of Mask R-CNN. Extensive experiments and thorough comparative evaluation provide a detailed analysis of the problem at hand and demonstrate the added value of the involved object-level flow branch. The overall proposed approach achieves improved performance in the six currently broadest and most challenging publicly available semantic urban scene understanding datasets, surpassing the region-based baseline method.**

I. Introduction

Object detection and recognition is a fundamental task for the human visual system. It has been proved that the human brain uses multiple object properties to achieve the required recognition performance. Appearance features such as shape, structure, color, and texture comprise essential information for this purpose. Therefore, most object representation methods have concentrated on single frame cues for recognition.

However, the vast majority of objects are not stationary. The motion characteristics of an object constitute a unique signature that can be used for recognition of the object. Intuitively, exploiting the motion characteristics of an object can improve our object recognition capabilities. The role of motion information in object recognition has been already examined by a number of studies [1]. Both rigid and non-rigid motion, have been studied for their role in different tasks.

As it has been highlighted, object recognition involves a number of heterogeneous modalities, namely appearance, shape and motion. Specialized neural networks have been developed to model each one of them. However, these modalities are strongly interconnected and it has been shown in the literature that employing a multi-target learning technique to address them all in parallel can have important advantages. It drastically reduces the overhead between the networks and it allows the network to generalize better.

Given the lack of motion information in single frames, only appearance-related features have been employed until now in multi-target learning methods. The shape of an object has been shown to be correlated with its motion characteristics. This is further confirmed by recent literature that has shown it is possible to predict the flow of an object *et al.* [2] from a single frame. This pseudo-flow information can be used as a substitute for the actual motion information.

In this paper, a neuroscience-inspired scheme is proposed to improve object detection by introducing to the Mask R-CNN architecture an additional pseudo-temporal stream (branch) for motion prediction from still images. An object-level flow field is incorporated in the object recognition process. In particular, the proposed pseudo-temporal information is effectively incorporated into the proposed detection framework by penalizing the global loss computation with an optical flow loss factor. For this purpose, a dense pseudo-flow estimation branch is added that achieves satisfactory motion prediction accuracy at a relatively low computational cost, since the latter is applied solely at the RoI level. Specifically, the resulting network detects object bounding boxes with instance segmentation masks and estimates the object flow predictions for each candidate object.

The remainder of the paper is organized as follows: Related work is reviewed in Section II. The proposed approach is detailed in Section III. Experimental results are discussed in Section IV and conclusions are drawn in Section V.

II. Related Work

Over the past few years, a broad number of techniques have been proposed, targeting object detection from still images or videos, while combining and integrating different approaches. This section analyses the different methodologies available in the literature for object detection from still images and videos, focusing on DL techniques. DL methods can be roughly divided in a) region-based and b) regression-based ones, depending on the number of processing steps/phases they employ.

A. Region-based methods

Current region-based methods perform detection by carrying out a classification on different regions, sub-windows or patches extracted from the image. This is the most popular category of methods, where the aim is to produce region proposals at first and then classifying each proposal into different object categories [3], [4], [5], [6], [7], [8]. Most of the approaches vary on the type of methodology used for choosing the regions, trying to find the balance between an exhaustive search and a fixed number of region proposals. One of the first attempts to utilize CNNs in object detection was the Region-based CNN (R-CNN) [3] in which a number of class-agnostic candidate regions are proposed and fed to a CNN to extract a fixed-length feature descriptor for each region. Thereafter, a unique linear Support Vector Machine (SVM) for each class classifies these regions based on their extracted descriptors. In [4], a Spatial Pyramid Pooling (SPP) layer is introduced, in order to remove the fixed-size constraint of the network. The latter computes a convolutional feature map from the entire image only once and then pools features in arbitrary regions to generate fixed-length representations for training the detectors.

Built upon R-CNN success, the Fast R-CNN [5] targets the inefficiency of having to pass each of the candidate regions individually through the CNN by forward passing the input image to the network once, generating its feature map and applying Region of Interest (RoI) pooling for each of the candidate regions to extract their feature representations. Based on the previously mentioned methods, Faster R-CNN [7] introduced a trainable mechanism for the purpose of proposing candidate regions called Regional Proposal Network (RPN). Given a number of uniformly generated anchors across the image, the RPN distinguishes them between foreground and background before passing the former to the classifier. Moreover, Mask R-CNN [6] extended the Faster R-CNN by adding an extra head for segmentation and replaced the ROI pooling with RoI align resulting in higher accuracy predictions. In [9], T.-Y. Lin et al. proposed Feature Pyramid Networks (FPN) on the basis of Faster R-CNN. The latter presented a top-down architecture with lateral connections for building high-level semantics at all scales. Later, a variety of improvements have been proposed, including R-FCN [8] and Light-head R-CNN [10].

In contrast to previous Region-based detectors, such as Fast/Faster R-CNN [5], [7], that apply a costly per-region sub-network hundreds of times, the proposed Region-based detector in [8] is fully convolutional with almost all computations shared on the entire image, while also improves speed by reducing the amount of work needed for each RoI. The latter introduces position-sensitive score maps to address a dilemma between translation-invariance in image classification and translation-variance in object detection. Additional classifiers are added in [11] aiming at progressively increasing the IoU's of the proposed regions with the ground truth objects which results in improved predictions.

B. Regression-based methods

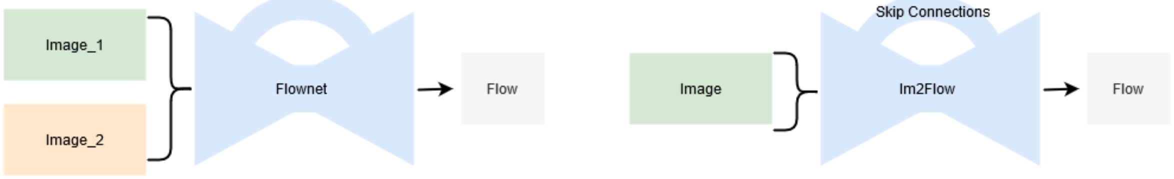
In contrary to the R-CNN family methods where region proposal and region classification are done by discrete modules, in one-stage methods the regions are generated and classified in a single forward pass. Methods belonging at this category try to map directly from image pixels to bounding box coordinates and class probabilities [12], [13], [14], [15], [16], [17]. In particular, Liu et al. [12] describe a method for detecting objects in images, using a single deep neural network. The approach, named Single shot multi-box detector (SSD), discretizes the output space of bounding boxes into a set of default boxes over different aspect ratios and scales per feature map location. At prediction time, the network generates scores for the presence of each object category in each default box and produces adjustments to the box to better match the object shape. Additionally, the network combines predictions from multiple feature maps with different resolutions to naturally handle objects of various sizes. In [13], a CNN-based technique is proposed, which models the problem of object detection as an iterative search in a multi-scale grid-space of all possible bounding boxes.

The YOLO [14] and the SSD [12] algorithms are the most representative one-stage/regression object detection approaches. Later, R. Joseph has made a series of improvements on the basis of YOLO and has proposed its v2 and v3 editions [15], which further improve the detection accuracy while keeps a very high detection speed. Moreover, an approach for introducing addition context into the SSD model is described in [16], where a state-of-the-art feature extractor (Residual-101 [17]) is combined with the aforementioned detection framework [12]. The proposed SSD+Residual-101 architecture is augmented with a set of deconvolution layers in order to introduce additional large-scale context in object detection.

Although this category of methods offers faster performance compared to the RPN based one, they are limited in terms of prediction accuracy due to the high imbalance between positive and negative regions fed to the classifier (the positive and negative terms refer to the presence and the absence of ground truth object, respectively). Lin *et al.* [18] addresses the imbalance by having the ambiguous regions contribute more in the loss calculation, thus valuing the hard examples more than the easily classified ones. To this end, the authors introduced a novel loss function named 'focal loss' by reshaping the standard cross-entropy loss so that the detector will put more focus on hard misclassified examples during training. Focal Loss enables the one-stage detectors to achieve comparable accuracy of two-stage detectors while maintaining a very high detection speed.

C. Flow-based object detection

Optical flow techniques have been applied to video-based object detection tasks over the years, as the incorporation of temporal information in the object detection task can improve the feature quality and recognition accuracy. The majority of them incorporate optical flow vectors obtained after applying an Optical Flow algorithm in the visual analysis loop for marking the detected object in the video frame. In [19], a DL framework, called T-CNN, which incorporates temporal and contextual information from tubelets/boxes obtained in videos is presented, by propagating detection results across adjacent frames according to pre-computed optical flows. Zhu *et al.* [20] propose a flow-guided feature aggregation, an



(a) FlowNet

(b) Im2Flow

Fig. 1: Optical flow estimation architectures: a)FlowNet architecture: including the refinement part, is trained in an end-to-end manner, b) Im2Flow architecture: an encoder-decoder model that infers flow given a single image

accurate and end-to-end learning framework for video object detection, which leverages temporal coherence on feature-level. The later enhances the visual features by employing an optical flow network to estimate the motions between the nearby frames and the reference frame. Recently, a unified approach is introduced, which is based on the principle of multi-frame end-to-end learning of features and cross-frame motion. It belongs to the category of feature-level methods, and introduces a *Spatially-adaptive Partial Feature Updating* to fix the inaccurate feature propagation caused by inaccurate optical flow.

From the above analysis, it can be deduced that despite the fact that optical flow features have been extensively applied in video object detection task, the current study is the first study, to the best of authors knowledge, that addresses the problem of object detection from still images, by incorporating object flow predictions of each detected object. In addition, object detection-related literature has in principle concentrated on appearance and contextual information analysis, while the respective pseudo-temporal information has not been examined yet, *i.e.* leaving great potential for further performance improvement unexplored.

III. Flow R-CNN

Objects inherently have motion characteristics, the capturing and encoding of which could be of paramount importance for achieving robust detection performance. According to recent neuroscience reports [21], [21], [22], the cerebral cortex can predict the path of a moving object (visual motion), even in cases where the object is traveling faster than the brains' visual processing rate, in order to adapt human behavior to surrounding objects moving in real-time. Neuro-scientists conclude that there is a specific part, called the Medial Superior Temporal (MST) area, in the cerebral cortex, which lies in the dorsal/parietal stream of the visual area of the primate brain where the whole visual processing takes place. The MST cooperates with the Middle Temporal (MT) area, in order to estimate the motion field of each moving object in a scene. In other words, that specific part is responsible for estimating the final or close to the final location of a moving object. Therefore, it is evident that the human brain can reveal the implied motion using a single still image. Given a single static image, the brain's ventral stream interprets the instantaneous semantic content, and at the same time the dorsal stream predicts what is going to happen based on scene spatial configuration, *e.g.* the ventral stream detects a car, while the dorsal stream anticipates that the car is moving forward. In this section, Flow R-CNN is thoroughly presented, including a detailed analysis of the new object-based motion branch.

A. *Object-based motion analysis*

CNNs have been extensively employed for optical flow estimation, achieving a huge improvement in prediction quality. In the current study, the literature approach of [23] is selected (Fig. 1a), where the information included in a pair of successive images is first spatially compressed in a contractive part of the CNN and then refined in an expanding part. However, for small displacement, *FlowNet* is not reliable. Thus, the authors proposed an extension of their previous model, called *FlowNetSD* [24], where they replaced several network parameters including kernel size and window stride of selected layers. Despite the very good results of these methods, they pose an impermeable constraint, as they require a pair of images as input to obtain satisfactory results. On the contrary, inspired by the aforementioned neuro-scientific notion of visual dynamics, Gao *et al.* have introduced an encoder-decoder CNN (refer to Fig. 1b) equipped with a novel optical flow encoding scheme that is able to translate a single static image into an accurate flow field. Their main idea is to learn a motion prior over short-term dynamics from a large set of videos and transfer the learned motion from videos to static images to infer their motion. The current study adopts the findings of Gao *et al.* [2] for object-level flow estimation.

B. *Mask R-CNN*

The baseline of this work is Mask R-CNN that belongs, as briefly stated in section II, to the Region-based/two-stage approaches. The latter is equipped with an RPN mechanism in the first stage in order to propose candidate RoIs. In the second stage, another part of the network takes the proposed RoIs and locates the relevant areas of the feature map by utilizing a RoIAlign layer. The extracted features are further processed in parallel to perform classification, bounding box regression and instance-level semantic segmentation. Both stages are connected to the backbone. Backbone could be any Convolution Network, but usually, ResNet or VGG are used to extract raw images.

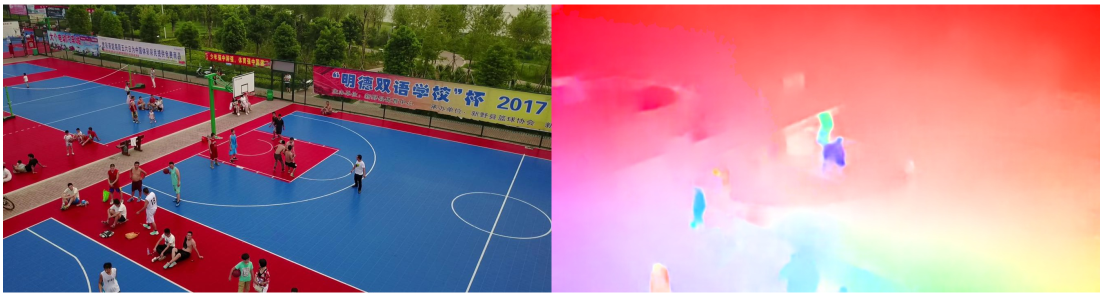


Fig. 2: An example of a computed flow field given a static image

C. Proposed architecture

In the current work, the proposed approach mimics the visual perception procedures that take place in the human brain, following an appropriate deep neuro-physiologically grounded architecture. The primary visual cortex is emulated by the backbone of the network, generating high-level feature representations, while the dorsal (‘where’) and ventral (‘what’) stream are incarnated by the flow-estimation and object classification branches respectively, predicting object categories with each respective motion in a collaborative way. The above notion is presented in Fig. 3.

The introduced Flow R-CNN exhibits the following advantageous characteristics: a) it enhances the two-stage detector by introducing an additional pseudo-temporal stream, and b) it incorporates the aforementioned stream in a multi-task learning process. In particular, the current study adopts the findings of Gao *et al.* [2] while moving their concept one step further, utilizing the pseudo-temporal object-level motion patterns combined with the appearance/contextual information to distinguish objects in still images. To this end, an object detection architecture is designed that takes into account the implied movement estimation.

The proposed Flow R-CNN model is built upon the Mask R-CNN model, estimating a per RoI flow field given a single static in an Im2Flow inspired branch. As a feature extractor (backbone) the ResNet variant is selected. The three existing branches of the baseline model (object classification, bounding box prediction, and mask prediction) remain intact and a 4th sub-network is integrated to the RoI head, in an end-to-end manner, to estimate a flow field for each predicted region proposal. The flow branch is inspired by the encoder-decoder logic of the Im2Flow model, where a motion prior-learned from videos in the form of a two-dimensional vector, taken from several urban scene understanding datasets, is transferred to images, bridging the still-image detection with video object motion understanding.

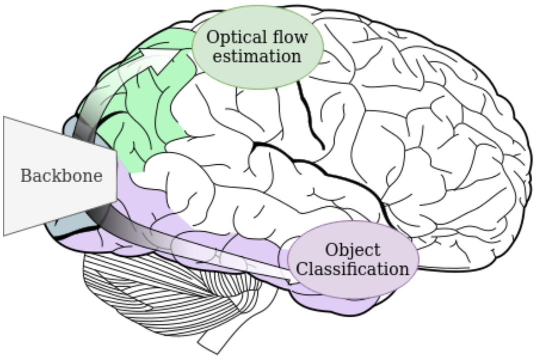


Fig. 3: The dorsal stream (green) and ventral stream (purple) are shown. They originate from primary visual cortex

Prior to the application of the proposed Flow R-CNN, optical flow estimation is realized for each dataset described in section IV. More specifically, videos from several datasets were used to model the motion patterns of objects and people in the scene, then embed the resulting knowledge into a representation for individual images (*i.e.* a two dimensional vector as can be seen in Fig:2). Finally, the proposed Flow R-CNN combines, in a way, the appearance information from the static image and the predicted motion dynamics from the introduced 4th branch in order to improve the detection accuracy. A graphical representation of the developed Flow R-CNN model is illustrated in Fig. 4. It needs to be highlighted that in the current implementation a modified version of the Im2Flow was used, where the encoder part was replaced by a region-based model backbone (ResNet) and the decoder part with the object-level flow estimation branch. The developed CNN branch consists of one convolutional layer, which models the correlations among the RoI features, an average pooling layer, and two fully connected layers, for computing the respective flow field. Experimentation with additional configurations, regarding the number of convolutional and fully connected layers (as well as their parameterization), did not lead to improved recognition performance.

In the training phase, a multi-task loss L_{total} is defined on each sampled proposal, as shown in (1). The classification loss L_{class}, the bounding-box loss L_{bbox} and the instance segmentation loss L_{seg} are identical to the ones define in Mask R-CNN model.

$$L_{total} = L_{class} + L_{bbox} + L_{seg} + L_{of} \tag{1}$$

For each RoI, an additional object-level flow loss L_{of} is computed to supervise the per-object motion by penalizing the predicted m x m x 2 optical flow output. This requires optical flow data for every image in the database, and in some cases, the authors of this study have extracted such info using state-of-the-art optical flow estimators [24]. The predicted object-level flow fields were compared with the cropped and resized region from the ground truth motion fields, penalized with an $l1$ loss function. It is assumed that the loss of the optical flow estimation branch enhances the learning process of the composite model while retaining key parts of the baseline model unaffected.

IV. Experimental results

A. Object detection datasets

Major research efforts have been made in the field of computer vision to understand the complex urban scenarios. The respective progress is bonded with the availability of vast amounts of annotated training data (*e.g.* cars, bicycles, pedestrians *etc.*) under varying conditions. In this section, experimental results, as well as comparative evaluation from the application of the proposed object detection method, are presented. For the evaluation, the 'KITTI'[25], 'V-KITTI'[26], 'Visdrone'[27], 'Cityscapes'[28], 'Berkeley Deep Drive'[29] as well as the 'UDacity'[30] datasets were used.

- **KITTI dataset [25]:** The KITTI object detection benchmark consists of 7481 training images and 7518 test images, comprising a total of 80.256 labeled objects. All images are color and the goal of the challenge is to detect objects from three common urban categories, namely *Car*, *Pedestrian*, and *Cyclist*. For evaluation, an Average Precision (AP) is computed.
- **V-KITTI dataset [26]:** The Virtual KITTI dataset contains 50 photo-realistic high-resolution synthetic videos for a total of approximately 21.000 frames, generated from 5 different virtual words in urban settings under different imaging and weather condition. These worlds were created using the Unity game engine and a novel real-to-virtual cloning method. These photo-realistic synthetic videos are automatically, exactly, and fully annotated for 2D and 3D multi-object tracking and at the pixel level with category, instance, flow, and depth labels. For the particular task of object detection the V-KITTI contains detailed class annotation for the objects of interest (*Car*, *Van*).
- **Visdrone dataset [27]:** The Visdrone benchmark dataset consists of 288 video clips formed by 261.908 frames and 10.209 static images, captured by various drone-mounted cameras, covering a wide range of aspects including location, environment (urban and country), objects, and density (sparse and crowded scenes). The dataset was collected using various drone platforms, in different scenarios, and under various weather and lighting conditions. From those only 8.559 images are used for the object detection task, with more than 540k bounding boxes in ten predefined categories, such as such as *Pedestrians*, *Cars*, *Bicycles*, and *Tricycles*. The dataset is further divided into training, validation and testing sets, having 6.471, 548 and 1580 images, respectively.
- **Cityscapes dataset [28]:** The Cityscapes dataset contains a diverse set of stereo video sequences recorded in street scenes from 50 different cities, with high-quality pixel-level annotations of 5.000 frames in addition to a larger set of 20.000 weakly annotated frames. A number of 30 visual classes for annotation were defined, which are further grouped into eight categories: flat, construction, nature, vehicle, sky, object, human, and void. However, instance-level labeling is available only for humans and vehicles (*Person*, *Rider*, *Car*, *Truck*, *Bus*, *Train*, *Motorcycle*, and *Bicycle*). Around 3000 images are used for the training, 500 for the validation, as well as 1500 images with annotation being held for benchmarking purposes.
- **Berkeley Deep Drive (BDD) dataset [29]:** The BDD dataset is a new driving dataset comprised of over 100K videos with diverse kinds of annotations including image-level tagging, object bounding boxes, drivable areas, lane markings, and full-frame instance segmentation. The dataset possesses geographic, environmental, and weather diversity, which is useful for training models so that they are less likely to be surprised by new conditions. The latter contains 10 object categories (*bus, traffic light, traffic sign, person, bike, truck, motor, car, train*, and *rider*) spread over 100.000 images with over 1.8M object instance labeled bounding boxes, making it suitable for robust object detection and semantic instance segmentation. The dataset is divided further into 3 domains, namely 'clear weather', 'city street' and 'daytime'. The current study selects only the 'city street' as a training domain which has a number of around 36.000 images in the training set.
- **Udacity dataset [30]:** The UDacity dataset contains over 600K urban objects in a variety of outdoor urban videos involving *Pedestrians, Cars, Bicycles* and other objects moving in the scene. Part of the data was collected using an HD camera mounted in a vehicle. Around 375.000 annotated objects for 100k images are used for training. The train/validation and test splits are 40%, 40% and 20%, respectively.

B. Experimental environment

In order to define the experimental protocol a set of parameters should be initialized as follows:

- The R-CNN part of the model was pre-trained using the COCO [31] dataset, while for the fine-tuning, the training and validation sets from each dataset were used.
- Images were resized such that their scale (longer edge) is 512 pixels.
- The RPN anchors span 5 scales and 3 aspect ratios, and the IoU threshold of positive and negative anchors was 0.7 and 0.3 respectively.
- As in Mask R-CNN, a RoI was considered positive if it has Intersection over union (IoU) with a ground-truth bounding box of at least 0.5, otherwise it was discarded as negative.
- The optical flow loss L_{of} was defined only on positive RoIs. During training, a set of 64 samples was selected for each input image, while at test time the proposal number

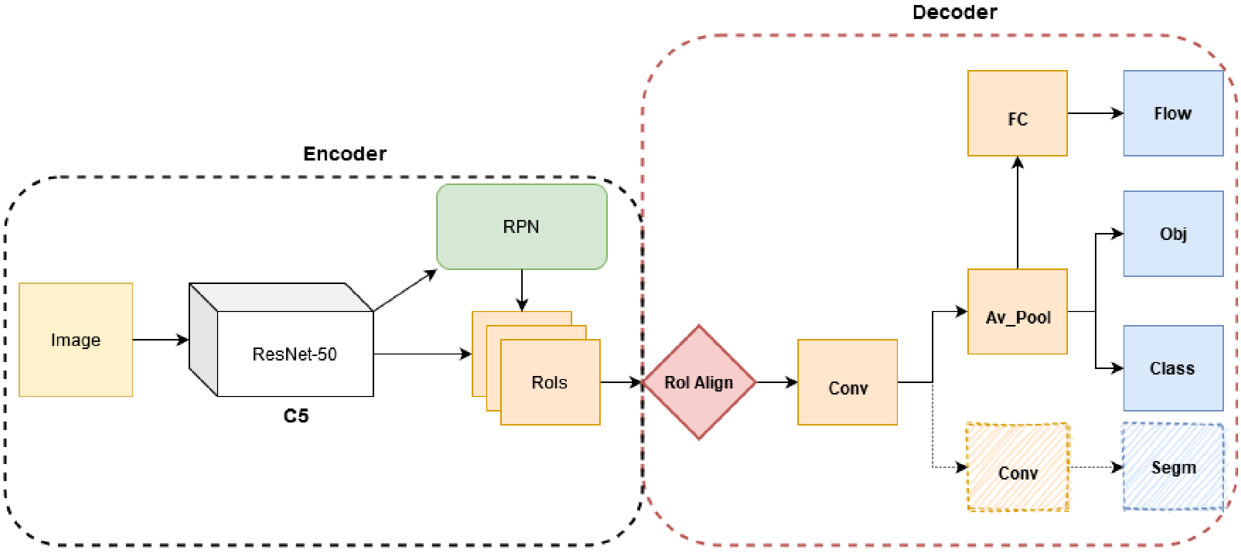


Fig. 4: Overall Flow R-CNN architecture: a composite region-based object detection model, the backbone of the network is used for the image decoding, while the object-level flow estimation branch is used to infer the optical flow field. Sketched part of the network, i.e. the segmentation branch, remains intact during training.

was set 300 followed by an NMS mechanism. The NMS process was performed twice, at the RPN results as well as at the predicted classes (class-specific NMS).

- The training phase is divided into two stages: a) only the flow branch being trained, b) all layers from ResNet stage 4 and up being fine-tuned.
- The ‘Keras 2’ deep learning framework with ‘Tensorflow’ backend was used for experimentation on two Nvidia GeForce GTX TITAN X GPUs. Ubuntu 18.04.
- The model was trained using SGD, utilizing batches of 2 images with learning rate (lr), initially set equal to $1e^{-3}$.
- Momentum was set to 0.9 and weight decay to 0.0001.

TABLE I: Comparative results on KITTI dataset

	Easy		Moderate		Hard	
	Mask	Flow	Mask	Flow	Mask	Flow
Car	0.893	0.905	0.843	0.849	0.733	0.736
Pedestrian	0.804	0.812	0.672	0.677	0.619	0.622
Cyclist	0.739	0.746	0.635	0.638	0.554	0.556
mAP	0.812	0.821	0.717	0.721	0.635	0.638

TABLE II: Comparative results on V-KITTI dataset

Class	Mask R-CNN	Flow R-CNN
Car	0.932	0.958
Van	0.917	0.940
mAP	0.924	0.949

C. Comparative evaluation

In Tables I, II, III, IV, V and VI, quantitative object detection results are given in the form of the mean Average Precision (mAP), *i.e.* computes the average precision value for recall value over 0 to 1. The current study follows the evaluation protocol defined by COCO challenge and adopts the primary challenge metric mAP that computes mAP over all classes and over 10 IoU thresholds. Averaging over the 10 IoU thresholds rather than only considering one general threshold of $mAP^{IoU=.5}$ tends to reward models that are better at precise localization. For providing a better insight, indicative object detection results obtained by the application of the proposed approach against the baseline one are presented in Fig.5. It can be observed that the proposed scheme exhibits improved recognition performance (especially in the case of moving cars) over the baseline in varying urban scenarios (night-view, top-view, car-view).

From the first group (Table I) of the provided results (*i.e.* KITTI dataset), it can be seen that the introduced Flow R-CNN model slightly improved the results of the respective Mask R-CNN model in all categories (Car, Pedestrian, Cyclist) as well as in every application scenario (Easy, moderate, hard). However, there was a significant improvement for the ‘Car’ category, about 1.2%, that supports the initial claim. The latter demonstrates the increased discrimination capabilities of the flow information stream. The same applies to the V-KITTI experiments (Table II), where the proposed architecture surpasses the baseline by a large margin (over 2% improvement).

TABLE III: Comparative results on Visdrone dataset

Class	Mask R-CNN	Flow R-CNN
Pedestrian	0.205	0.223
People	0.071	0.064
Bicycle	0.029	0.033
Car	0.406	0.428
Van	0.208	0.232
Truck	0.148	0.181
Tricycle	0.132	0.148
Awn	0.091	0.085
Bus	0.216	0.253
Motor	0.153	0.151
mAP	0.166	0.180

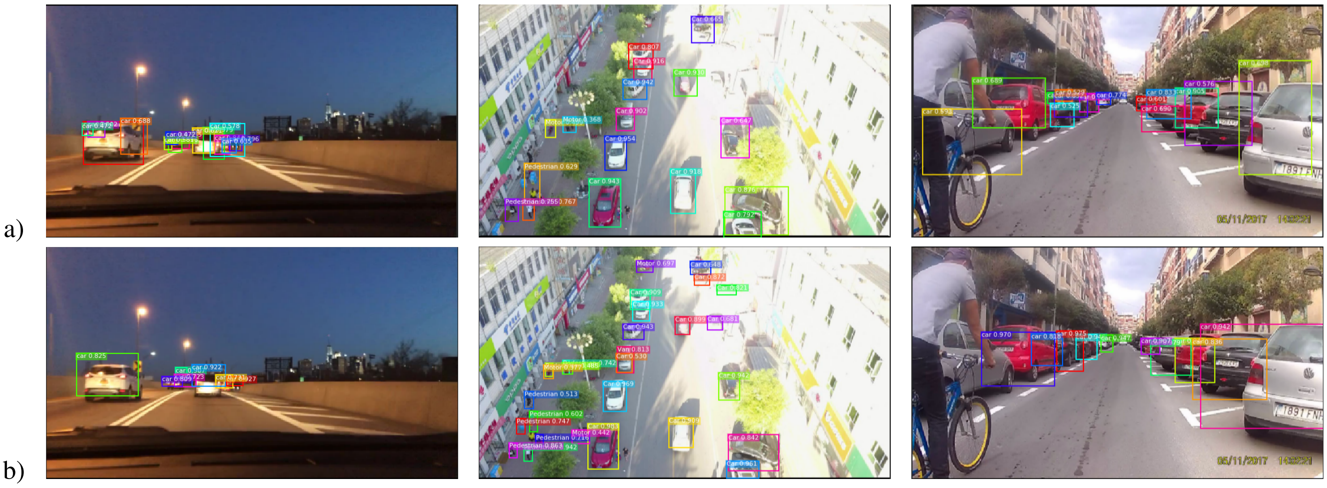


Fig. 5: Object detection results obtained from the application of the Mask R-CNN (a) and Flow R-CNN (b) models to the supported datasets[29], [27], [30]

TABLE IV: Comparative results on Cityscapes dataset

Class	Mask R-CNN	Flow R-CNN
Person	0.345	0.364
Rider	0.271	0.307
Car	0.488	0.505
Truck	0.296	0.306
Bus	0.401	0.387
Train	0.302	0.252
Motorcycle	0.237	0.256
Bicycle	0.182	0.204
mAP	0.315	0.323

TABLE V: Comparative results on BDD dataset

Class	Mask R-CNN	Flow R-CNN
Bike	0.383	0.391
Bus	0.481	0.489
Car	0.732	0.746
Motor	0.194	0.198
Person	0.531	0.537
Rider	0.349	0.352
Traffic-light	0.479	0.473
Traffic-sign	0.558	0.547
Truck	0.506	0.514
mAP	0.421	0.424

Concerning the Visdrone experiments (Table III), it can be observed that the introduced scheme perform reasonably well in categories where the motion is evident (‘Car’, ‘Van’, ‘Track’, *etc*.), while fails to recognize those that have complex structure or cover small portion of the image due to camera positioning (*e.g*. based on drone footage).

The exhibited results of the Cityscapes dataset (Table IV) suggest that incorporating the flow stream into the learning process of an R-CNN architecture may have a positive impact in the detection and recognition of moving objects, such as ‘Cars’, ‘Motorcycles’ and ‘Trucks’, by 1.7%, 1.9% and 1%, respectively. Moreover, regarding the BDD experiments (Table V) the influence of the motion branch to the R-CNN scheme is evident in the presented results, as most classes have superior recognition performance, whereas a slight increase is reported for the overall mAP (0.3%), as static objects (‘traffic-light’ and ‘traffic-sign’) over-shade the performance. Udacity dataset (Table VI) has quite similar content and category types to the previous one, and due to the lack of optical flow training data for this group, as Udacity is composed of still images, it was decided to transfer the acquired knowledge from the previous set (*i.e*. the BDD). This limitation has led the model to fail in most cases, except in the case of cars, that hold a significant portion of the dataset, demonstrating the need for data but also highlighting the cumulative capabilities that the introduced model offers to the moving objects. An evaluation of the proposed Flow R-CNN in six different datasets using various backbones is shown in Table VII. It can be observed that the introduced model achieves improved performance in all datasets using deeper ResNet architectures, while also benefiting from advanced schemes such as the FPN-variant; highlighting the generalizability of the proposed design.

TABLE VI: Comparative results on Udacity dataset

Class	Mask R-CNN	Flow R-CNN
Bike	0.625	0.629
Bus	0.949	0.951
Car	0.724	0.736
Motorbike	0.738	0.736
Person	0.747	0.752
Traffic-light	0.502	0.498
Traffic-sign	0.701	0.696
mAP	0.712	0.714

TABLE VII: Comparative results on six datasets using different backbone architectures

Backbone	KITTI	V-KITTI	Visdrone	Cityscapes	BDD	Udacity
ResNet-50	0.724	0.949	0.180	0.323	0.424	0.714
ResNet-101	0.731	0.956	0.185	0.329	0.430	0.720
ResNet-50-FPN	0.735	0.961	0.194	0.334	0.432	0.725
ResNet-101-FPN	0.742	0.967	0.207	0.340	0.438	0.731

V. Conclusions

In this paper, the problem of multi-task object detection using DL techniques was investigated following the recent and highly promising studies in the computer vision field, and more specifically the R-CNN approach. A methodology for incorporation of pseudo-temporal information in Region-based CNN object detection schemes was presented, in contrast to the vast majority of literature methods that rely only on the use of appearance information and semantic knowledge. Additionally, following a neuro-scientifically grounded scheme, the pseudo-temporal stream was integrated parallel to the classification, bounding box regression and segmentation mask prediction branches of Mask R-CNN, and it was effectively incorporated into the learning process by penalizing the global loss computation with an optical flow loss factor. Extensive experiments and thorough comparative evaluation were reported, which provide a detailed analysis of the problem at hand and demonstrate the added value of the involved instance-level motion branch. The overall proposed approach achieved improved performance in the six currently broadest and most challenging publicly available semantic urban scene understanding datasets, surpassing the baseline method. Future work includes the investigation of re-adjusting the proposed pseudo-temporal branch utilizing a more sophisticated optical flow estimation methodology.

Acknowledgment

The work presented in this paper was supported by the European Commission under contract H2020-787061 ANITA.

References

[1] H. Hill and A. Johnston, "Categorizing sex and identity from the biological motion of faces," *Current biology*, vol. 11, no. 11, pp. 880–885, 2001.

[2] R. Gao, B. Xiong, and K. Grauman, "Im2flow: Motion hallucination from static images for action recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5937–5947.

[3] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

[5] R. Girshick, "Fast r-cnn object detection with caffe," *Microsoft Research*, 2015.

[6] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask r-cnn," in *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[7] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[8] J. Dai, Y. Li, K. He, and J. Sun, "R-fcn: Object detection via region-based fully convolutional networks," in *Advances in neural information processing systems*, 2016, pp. 379–387.

[9] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.

[10] Z. Li, C. Peng, G. Yu, X. Zhang, Y. Deng, and J. Sun, "Light-head r-cnn: In defense of two-stage object detector," *arXiv preprint arXiv:1711.07264*, 2017.

[12] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.

[11] Z. Cai and N. Vasconcelos, "Cascade r-cnn: High quality object detection and instance segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[13] M. Najibi, M. Rastegari, and L. S. Davis, "G-cnn: An iterative grid based object detector," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[14] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[15] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7263–7271.

[16] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg, "Dssd: Deconvolutional single shot detector," *arXiv preprint arXiv:1701.06659*, 2017.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[18] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

[19] K. Kang, H. Li, J. Yan, X. Zeng, B. Yang, T. Xiao, C. Zhang, Z. Wang, R. Wang, X. Wang *et al.*, "T-cnn: Tubelets with convolutional neural networks for object detection from videos," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 10, pp. 2896–2907, 2017.

[20] X. Zhu, Y. Wang, J. Dai, L. Yuan, and Y. Wei, "Flow-guided feature aggregation for video object detection," in *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[21] Z. Kourtzi and N. Kanwisher, "Activation in human mt/mst by static images with implied motion," *Journal of cognitive neuroscience*, vol. 12, no. 1, pp. 48–55, 2000.

[22] G. W. Maus, J. Ward, R. Nijhawan, and D. Whitney, "The perceived position of moving objects: transcranial magnetic stimulation of area mt+ reduces the flash-lag effect," *Cerebral cortex*, vol. 23, no. 1, pp. 241–247, 2013.

[23] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox, "Flownet: Learning optical flow with convolutional networks," in *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[24] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "Flownet 2.0: Evolution of optical flow estimation with deep networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2462–2470.

[25] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *International Journal of Robotics Research (IJRR)*, 2013.

[26] A. Gaidon, Q. Wang, Y. Cabon, and E. Vig, "Virtual worlds as proxy for multi-object tracking analysis," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4340–4349.

[27] P. Zhu, L. Wen, D. Du, X. Bian, H. Ling, Q. Hu, H. Wu, Q. Nie, H. Cheng, C. Liu *et al.*, "Visdrone-vdt2018: The vision meets drone video detection and tracking challenge results," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 0–0.

[28] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3213–3223.

[29] F. Yu, W. Xian, Y. Chen, F. Liu, M. Liao, V. Madhavan, and T. Darrell, "Bdd100k: A diverse driving video database with scalable annotation tooling," *arXiv preprint arXiv:1805.04687*, 2018.

[30] A. Dominguez-Sanchez, M. Cazorla, and S. Orts-Escolano, "A new dataset and performance evaluation of a region-based cnn for urban object detection," *Electronics*, vol. 7, no. 11, p. 301, 2018.

[31] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.