RECOGNITION AND TRACKING OF VEHICLES IN HIGHWAYS USING DEEP LEARNING

Ludwin Lope Cala*, Roseli Aparecida Francelin Romero†

*Department of Computer Science
University of São Paulo
São Carlos-SP, Brazil 13566-590

Emails: ldwnlp@usp.br, rafrance@icmc.usp.br

Abstract— Unmanned Aerial Vehicles (UAVs) are becoming increasingly popular. Researchers are trying to use them in various tasks, such as, surveillance of environments, persecution, collection of images, etc. In this work, we propose a vehicle tracking system to turn UAVs able to recognize a vehicle and monitor it in highways. It is based on a combination of bio-inspired algorithms: VOCUS2, CNN and LSTM. The proposed system was tested with real videos collected by the aerial robot and the results show that in spite of the proposed system is simpler than others, it achieved a good classification performance and overcame other existing approaches in terms of precision.

Keywords— Computer Vision, Deep Learning, Recurrent Neural Network, Tracking, Detection and Classification, Drone.

Resumo— Veículos Aéreos Não Tripulados (VANTs) estão se tornando cada vez mais populares. Pesquisadores estão tentando usá-los em várias tarefas, tais como, vigilância de ambientes, perseguição, coleta de imagens, etc. Neste trabalho, propomos um sistema de rastreamento de veículos para tornar os UAVs capazes de reconhecer um veículo e monitorá-lo em rodovias. O sistema é baseado numa combinação de algoritmos bio-inspirados: VOCUS2, CNN e LSTM. O sistema proposto foi testado com vídeos reais coletadas por um robô aéreo e os resultados mostram que, apesar do sistema proposto ser mais simples do que outros, obteve um bom desempenho de classificação e superou outras abordagens existentes em termos de precisão.

Palavras-chave— Visão Computacional, Aprendizado Profundo, Rede Neural Recorrente, Rastreamento, Detecção e Classificação, Drone.

1 Introduction

UAVs are constantly being improved and going out to the market to perform different tasks in various environments. Nowadays, several research works are focused on autonomous flight of UAVs, so there is no need of a pilot. Previous works in the area like obstacle detection gives way to another stage that is the autonomous flight of UAVs. Also new sophisticated hardwares with high computational power allows us to use Deep Learning (DL) techniques that nowadays are reaching records in the world competitions of recognition and classification of objects such as ImageNet Large Scale Visual Recognition Challenge (ILSVRC) (Jia Deng et al., 2009).

We take the idea of the work of Montanari (2015) that applied traditional artificial vision techniques (described in more detail in the Section 2.1) to detect and track vehicles with **images** taken from an UAV and got 79.82% of accuracy to the classification.

The aim of this work is to develop a system for tracking vehicles by UAV (drone), supposing it is not possible to chase a suspicious vehicle by ground. The system will be used to track suspect cars, which police authorities with access to an UAV can use in cases of persecution.

For this, we are proposing an architecture for classification and tracking of images in **videos**. This architecture is composed by three phases: saliency, recognition and tracking. For saliency, the VOCUS2 method (Frintrop et al., 2015) is used to detect the more salient objects present in the frame (a image of the video). For recognizing the objects detected, we are proposing a deeper network based on Convolutional Neural Network (CNN). Finally, for tracking a specific object (car), Long Sort Term Memory (LSTM) network is added in the output of the deeper network.

This article is organized as follows. In Section 2, are described some related works. In Section 3, a system for recognizing and tracking of vehicles in a highway by using UAV, is proposed. The experiments, results and comparisons are described in Section 4. Finally, in Section 5, conclusions about the results obtained and limitations of the proposed vision system are presented, followed by the future works.

2 RELATED WORKS

In this section, are described some related works of the area. In Section 2.1 are found works about detection and classification of objects and in Section 2.2, works about the tracking of moving object.

2.1 Detection and Classification of Objects

Object detection and classification is an active area of research. Nowadays, there are com-

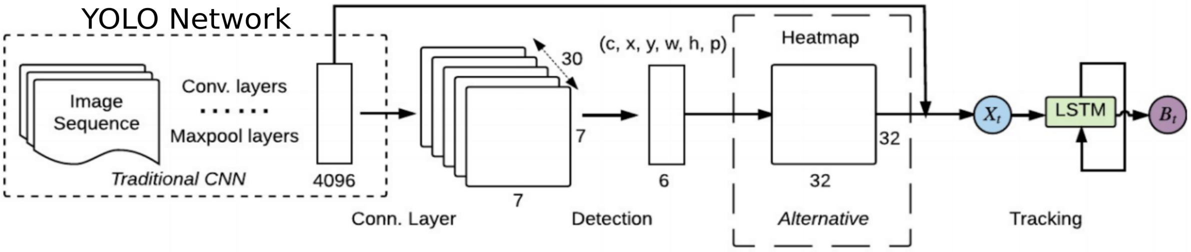


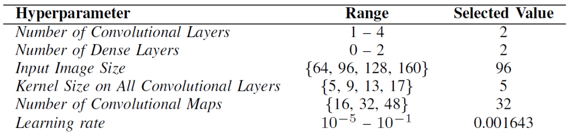
Figure 1: ROLO architecture proposed by Ning et al. (2016).

petitions to classify various kinds of objects. One of the most important competitions is ImageNet Large Scale Visual Recognition Challenge (ILSVRC). The database was presented for the first time at the 2009 Conference on Vision and Pattern Recognition (CVPR), in Florida, by researchers from the Department of Computer Science at Princeton University (Jia Deng et al., 2009). This competition contain 1000 categories and 1.2 million images each year. Deep learning techniques were introduced in the competition in 2012. In 2015, it was possible to beat the human level of accuracy at classifying of objects.

The baseline of our work is the work developed by Montanari (2015). In this work, the VOCUS2 technique, a visual attention technique, is used to obtain the saliency object segmentation. For the classification of the images, the bag-of-features (Salton and McGill, 1986) technique was applied, and to track the objects Camshift and Kalman filters were used obtaining an accuracy of 79.82%. More especifically, to classify the images, it was used an algorithm to extract the characteristics of the image, then to construct a matrix for representing the characteristics and finally to obtain the corresponding classification.

In Huttunen et al. (2016), it was proposed a system for recognizing 4 types of vehicles: bus, truck, van and small car, obtaining 97% of accuracy. In Fig. 2, are shown the network architectures tested, chosen among the architectures that had the better accuracy. The network receives an image input of 96 x 96 with 3 channels (RGB) and has a first convolutional layer with 32 feature maps, followed by a max-pooling (Scherer et al., 2010), reducing the image dimension to 48 x 48. The second layer is also a convolutional layer with 32 feature maps, followed by a max-pooling, reducing the image dimension to 24 x 24. Finally, there are 2 fully connected layers of 100 neurons each one, to produce an output layer constituted by 4 output neurons with the SoftMax function (Duan et al., 2003).

In Riveros and Caceres (2016), it was proposed an automobile classifier based on convolutional neural networks, which obtained 95.6% of accuracy for a dataset collected by a security camera. The network architecture is based on LeNet-5 and it was tested with different activation functions (RELU, sigmoid, PreRELU) and different functions for the pooling layer (AVG, MAX, STO). Different ways to initialize weights (Xavier, Uniforme, Gaussian) were applied for getting the best results with RELU, MAX and XAVIER, respectively. All these functions are described in details on (Riveros and Caceres, 2016).



Hyperparameter	Range	Selected Value
Number of Convolutional Layers	1 – 4	2
Number of Dense Layers	0 – 2	2
Input Image Size	{64, 96, 128, 160}	96
Kernel Size on All Convolutional Layers	{5, 9, 13, 17}	5
Number of Convolutional Maps	{16, 32, 48}	32
Learning rate	10^{-5} – 10^{-1}	0.001643

Figure 2: Neural network architectures tested by Huttunen et al. (2016).

2.2 Moving Object Tracking

Object tracking is an important task in the field of computer vision. In its simplest form, tracking can be defined as the problem of estimating the trajectory of an object in an environment or around a scene. Almost all tracking algorithms require the detection of objects (Yilmaz et al., 2006).

There are several works in the literature to track a vehicle. In the work proposed by Montanari (2015), it was utilized Camshift technique (Bradski, 1998) with Kalman filter (Welch and Bishop, 1995). Riveros and Caceres (2016) used Camshift with correlation filter tracker (Danelljan et al., 2014). In Alper Yilmaz (2006), it can also be found a survey with various descriptions of tracking algorithms.

But this kind of tracking is not robust for object tracking, since a robust object tracking requires the knowledge and the understanding of the object being tracked (Gordon et al., 2017). For this reason, in this work we will focus on ROLO (Ning et al., 2016), a tracking framework that uses a highly efficient image detector called YOLO (Redmon et al., 2015), which is a fast detector CNN (45 fps). The ROLO framework receives the outputs of the last fully connected layer of the YOLO as it can be seen in Fig. 1. This layer

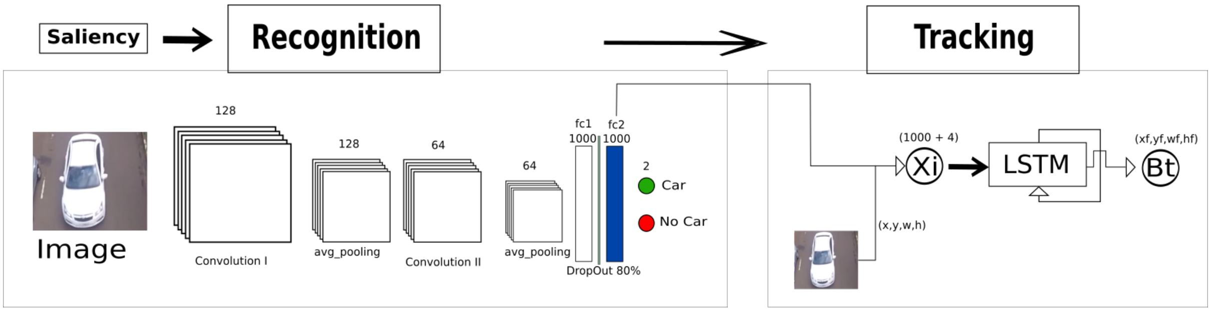


Figure 3: Architecture proposed for classification and tracking

has 4096 neurons. Its output signals are sent into the recurrent network LSTM and for a layer that will extract the position coordinates, represented by the vector: (c,x,y,w,h,p) = (0, x, y, w, h, 0), in this case, of each object being tracked. The LSTM network, described in subsection 3.3, is added and by its turn has a layer of 4102 recurring memory cells (corresponding to 4096 plus 6 components) and therefore it receives 4102 inputs coming from YOLO.

The framework ROLO has two phases. The first one, on the contrary of the second phase, does not utilize the Heatmap, that is a transformation of the output of YOLO for the size 32x32. The second phase includes the Heatmap for avoiding occlusion. Then, the framework ROLO predicts the new position of the object in the first phase. In our case, we have implemented only the first phase, then our proposal is not free of occlusions.

3 THE PROPOSED SYSTEM

In this section, the system proposed for recognition and tracking of vehicles in a highway, is described in details. It is composed by three main steps: Saliency, Recognition and Tracking, as it is shown in Fig. 3. Each constituting step will be described on the next subsections.

3.1 Saliency

This step consists in to highlight that part of the image in which the vehicle is. The main point is to decrease the searching of the object in the image to be analyzed, avoiding to look for it in the whole image. For this, we use an algorithm to detect the object more salient in the image and then to be able to analyze only the saliency object. We found an algorithm of saliency called VOCUS2, which is an improvement of the VOCUS algorithm. This algorithm calculates the feature channels of the image in parallel and the center-neighbor contrast is computed by Difference-of-Gaussians. It is based on the concepts from human perception, which is useful to obtain an object of greater saliency in an image. The salient segment will be the input for the next stage that will be described in the next section.

3.2 Recognition

In this subsection is described how the objects are recognized in the image. For this purpose, it is necessary to explain how a CNN (Le Cun et al., 1998) can recognize or classify the saliency object. A CNN is biologically inspired by the visual cortex of animals. After to receive as input an image, it processes the image in the convolutional layers and finally in the fully layers the image is classified. A typical CNN processes the image and produces an output. In our case, this output is binary, representing car or no-car. The components of CNN are inputs, weights, activation functions and outputs. The main component in CNN are the weights, because they are updated during the training process. Once the detection of the object is completed by using CNN, the next stage is to track the object and this process will be described in the next Section 3.3.

3.3 Tracking

To track a vehicle, it is necessary a method for memorizing the final positions of the vehicle for a short period and then predict a future position of the vehicle. In the literature, it can be found several techniques, but we have chosen the recurrent neural network LSTM (Hochreiter and Schmidhuber, 1997). It was created in 1997 by Hochreiter and Schimdhuber, but its popularity has grown in recent years for different applications, obtaining good accuracy. LSTM is composed by memory cells, each memory cell has three doors. The first one is the input gate, the second is the output gate or inference and the last, in which its information or the result of the memory cell serves as input for the next cell. It is similar to a flip flop or a bistable multi-vibrator that serves as a memory to keep a bit. However, this output is not connected to itself as the flip flop, instead it is connected to the memory cell.

It is necessary to emphasize that ROLO framework has been adapted by us. For this, in-

Table 1: Characteristics of the architectures

	Arch. 1	Arch. 2
Number of training epoch.	5000	5000
Number of convolutional layers	2	2
Number of layers fully connected	**2(100, 100)**	**2(1000, 1000)**
Input image size	96x96	96x96
Number of feature maps	**{32, 32}**	**{128, 64}**
Kernel size of the convolutional layers	5x5	5x5
Pooling kernel size	2x2 (avg)	2x2 (avg)
Cost function	Mean Squared Sum	Mean Squared Sum
Optimizer	AdagradOptimizer	AdagradOptimizer
Learning Rate	0.001643	0.001643
Dropouts	80%	80%

stead of YOLO, we are proposing a new network for recognition based on CNN, called by us Architecture 2, which characteristics are shown in Table 1. Furthermore, we insert LSTM after the last fully-connected layer, as it is shown in Fig. 3. Furthermore, the data (x,y,w,h), that is the position of car in the image is obtained by other algorithm called Correlation Filter Tracker (CFT) (Danelljan et al., 2014).

All these algorithms mentioned above (VOCUS2, CNN and LSTM) are useful to detect, recognize and track a vehicle and they will serve to be part of the aerial robot of our laboratory.

4 EXPERIMENTAL RESULTS

In this section, it will be describe how the data has been collected and how the training for vehicle recognition has been performed.

Table 2: Table of errors in each Cross-Validation Folds

Fold	Error %
I	14.75
II	12.08
III	9.27
IV	1.73
Avg. Error	**9.45**

4.1 Data Collection

For collecting videos, a drone took images in our university. The images collected were processed with the help of the saliency algorithm VOCUS2. Next, each images of the video was manually classified between car and not-car. This dataset was increased up 13705 images taking images from other sceneries.

Table 3: Average of accuracy, precision and classify time in seconds of architectures 1, 2 and Inception-v2

	Arc. 1	Arc. 2	Incept.-v2
Avg. accuracy	0.4158	0.4192	**0.5775**
Avg. precision	0.9313	**0.9572**	0.8546
Avg. class. time	**0.3464**	0.5946	5.0373

4.2 Training for Vehicle Recognition

The dataset was trained with different architectures of networks. Some modifications in CNN proposed in Huttunen et al. (2016) are being adopted here. More convolutional layers, more neurons in the fully connected layers and dropout have been added. Two different architectures for tracking the vehicle are being investigated. Their characteristics are shown in Table 1.

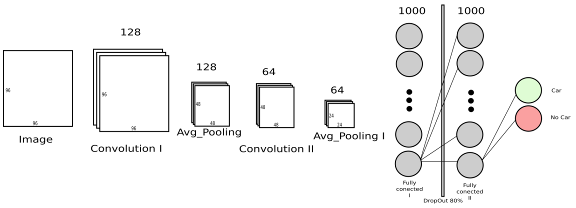


Figure 4: CNN Architecture 2

For the validation of the models proposed, we apply the cross-validation technique on Architecture 2, Fig. 4. From a total of 13705 images collected (classified by hands between it is or no-car); The data was fragmented in 4 folds, considering approximately 25% of data for testing and 75% for training, of both groups (car and no-car). The training was performed with 5000 epochs for each fold. The CNN inputs are images of size 96x96. Each image have a whole car or parts of it. The results obtained by the Architecture 2 is shown in Table 2. It can be seen that an average accuracy of 90.55% and an average error of classification of 9.45%. This shows a good performance of the proposed approach. We have trained considering more one CNN architecture: Architecture 1, shown in Table 1. In this case, it was considered for training only Fold I, because it was the fold that presented the higher error rate (14.75%), when using Architecture 1 the error of classification was 16.11%. Another test was done with Fold I, was trained again with another type of architecture, more complex, as shown in Szegedy et al. (2015). This architecture is of type inception, which we will call here by Inception-v2, with 1000 epochs of training, giving an approximate error of 0.12%.

In Table 3, are shown the mean values of accuracy and precision, tested with different videos collected by the drone of LAR laboratory, considering the architectures: 1, 2 and Inception-v2. It can be seen a difference of approximately 17% of accuracy between the proposed architectures (1 and 2) and Inception-v2, what implies that Inception-v2 was better than our architectures (1 and 2), in terms of accuracy. It is necessary to note that Inception-v2 has already trained weights and we used the transfer learning technique to train Fold I. Furthermore, Inception-v2 takes more time to classify as it can be seen in Table 3. On the other hand, it is needed to highlight the good precision obtained by our architectures (1 and 2). They presented almost 10% better than Inception-v2. This means that the true positives are almost perfect in our architectures (1 and 2). These results indicate that it is necessary to increase the dataset to improve the accuracy for any video. Comparing the precision of Architectures 1 and 2, they are not different statistically, according Test-T since p-value = 0.5363. However, as the average precision obtained for Architecture 2 was better than Architecture 1, according Table 3, Architecture 2 will be considered for the task of tracking of vehicles presented in next section.

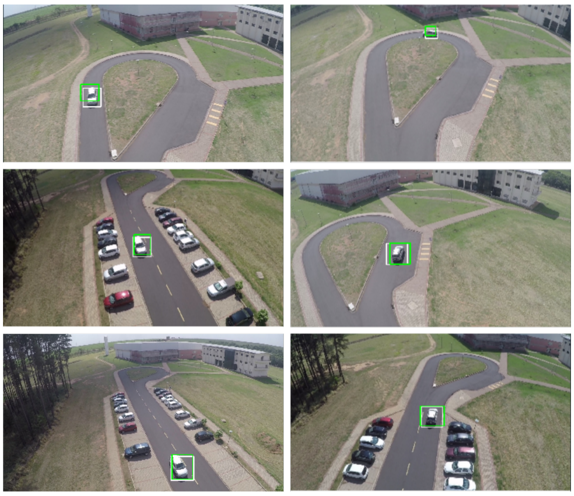


Figure 5: Test Vehicle Tracking: Box green see our LSTM track and the white box is from ground truth (manually labeled) (Video#1).

Table 4: Training for Vehicle Tracking

No. Epochs	IoU
Video#1	
200	0.37077
1000	0.70266
Video#2	
200	0.75823
500	0.79602

Table 5: Comparison between CFT and LSTM tracker

	CFT	Our Tracker
IoU of Video#1	0.3154	0.7027
IoU of Video#2	0.9163	0.7960
Avg. IoU	**0.6159**	**0.7494**

4.3 Tracking of Vehicles

In Fig. 5, we can see an example of vehicle tracking from the perspective of the drone. In Table 4, are shown the number of epochs used to train the LSTM recurrent network and the average of Intersection-over-Union (IoU). It can noted also the variation of number of epochs from one video to another to have a good IoU. Video #1 has 800 frames whereas Video #2 has 500 frames. For Video #1, one observe that are necessary 1000 epochs for getting IoU approximately equals to 70%, whereas for Video #2, only 200 epochs.

In Table 5, it is shown a comparison between CFT and our LSTM tracker. It can be seen that CFT in video#2 got a better IoU but in the video#1, it got the worst. Certainly, we need to consider more videos to get a conclusion, but, considering an average of IoU obtained in these two videos, our proposal was better than CFT for tracking of vehicles.

5 CONCLUSION

In this paper, it was presented an architecture based on deeper networks to turn a vehicle able of recognizing and tracking specific car in a highway. The proposed system can receive and analyze images captured by an UAV (drone), aiming to track a vehicle. It is constituted by bio-inspired algorithms: VOCUS2, CNN and LSTM. The results obtained showed 90.55 % of average accuracy in the dataset tested, what demonstrates a good performance. From the comparative results, in terms of accuracy, Inception-v2 network was better than the architecture proposed, but it holds to note that for the classification Inception-v2 network took longer time. On the other hand, in terms of precision, the proposed architecture had a better performance compared to Inception-v2 network. In the tracking of vehicle task, the proposed approach was compared with the correlation filter tracker technique. The obtained result showed that the proposed system got an IoU mean of approximately 75% compared to 62% obtained for the correlation filter tracker. One limitation is that it is useful for images took to daylight, it does not at night light. As future works, we intend to increase the dataset and to use transfer learning with the proposed architecture, for training in real time for a specific car that is being tracked.

Acknowledgements

This work was supported by the CNPq (Brazilian National Research Council), by the FAPESP (São Paulo Research Foundation) 2017/01687-0 and by the INCT (National Institute of Science and Technology) under the grant CNPq 465755/2014-3 and FAPESP 2014/50851-0.

References

Alper Yilmaz, Omar Javed, M. S. (2006). Object tracking: A survey, *ACM Comput. Surv. 38* .

Bradski, G. R. (1998). Computer vision face tracking for use in a perceptual user interface.

Danelljan, M., Hager, G., Shahbaz Khan, F. and Felsberg, M. (2014). Accurate scale estimation for robust visual tracking, *Proceedings of the British Machine Vision Conference*, BMVA Press.

Duan, K., Keerthi, S. S., Chu, W., Shevade, S. K. and Poo, A. N. (2003). Multi-category classification by soft-max combination of binary classifiers, *In 4th International Workshop on Multiple Classifier Systems*.

Frintrop, S., Werner, T. and Martin Garcia, G. (2015). Traditional saliency reloaded: A good old model in new shape, *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Gordon, D., Farhadi, A. and Fox, D. (2017). Re3 : Real-time recurrent regression networks for object tracking, *CoRR* **abs/1705.06368**.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory, *Neural computation* **9**(8): 1735–1780.

Huttunen, H., Yancheshmeh, F. S. and Chen, K. (2016). Car type recognition with deep neural networks, *IEEE Intelligent Vehicles Symposium (IV)* .

Jia Deng, W. D., Richard Socher, L.-J. L. and Kai Li, L. F.-F. (2009). Imagenet: A large-scale hierarchical image database, *conference on Computer Vision and Pattern Recognition* .

Le Cun, Y., Bottou, L., Bengio, Y. and Haffner, P. (1998). Gradient based learning applied to document recognition, *Proceedings of IEEE* **86**(11): 2278–2324.

Montanari, R. (2015). *Detecção e classificação de objetos em imagens para rastreamento de veículos*, Master's thesis.

Ning, G., Zhang, Z., Huang, C., He, Z., Ren, X. and Wang, H. (2016). Spatially supervised recurrent convolutional neural networks for visual object tracking, *CoRR* **abs/1607.05781**.

Redmon, J., Divvala, S. K., Girshick, R. B. and Farhadi, A. (2015). You only look once: Unified, real-time object detection, *CoRR* **abs/1506.02640**.

Riveros, E. R. L. and Caceres, J. C. G. (2016). Detection and monitoring of vehicles with surveillance camera using a cost allocation algorithm.

Salton, G. and McGill, M. J. (1986). *Introduction to Modern Information Retrieval*, McGraw-Hill, Inc., New York, NY, USA.

Scherer, D., Muller, A. and Behnke, S. (2010). Evaluation of pooling operations in convolutional architectures for object recognition, *Proceedings of the 20th International Conference on Artificial Neural Networks: Part III*, ICANN'10, pp. 92–101.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. and Wojna, Z. (2015). Rethinking the inception architecture for computer vision, *CoRR* **abs/1512.00567**.

Welch, G. and Bishop, G. (1995). An introduction to the kalman filter.

Yilmaz, A., Javed, O. and Shah, M. (2006). Object tracking:a survey, *ACM Computing Surveys* .